

Математическая логика и логическое программирование

mk.cs.msu.ru → Лекционные курсы
→ Математическая логика и логическое программирование (3-й поток)

Блок 13

Теорема Лёвенгейма-Сколема
Теорема компактности Мальцева
Автоматизация доказательства теорем

Лектор:
Подымов Владислав Васильевич
E-mail:
valdus@yandex.ru

ВМК МГУ, 2024/2025, осенний семестр

Вступление

Корректность и **полнота** метода семантических таблиц в ЛП:
 $\models \varphi \Leftrightarrow$ для таблицы $\langle \mid \varphi \rangle$ существует успешный табличный вывод

Полнота метода семантических таблиц в ЛП (доказательство):
если $\models \varphi$, то успешный табличный вывод можно построить, придерживаясь особой стратегии

Свойства и «приёмы», обсуждавшиеся
для метода семантических таблиц, позволяют

- ▶ обосновать несколько нетривиальных утверждений, не имеющих прямого отношения к методу, и
- ▶ естественно поставить важные алгоритмические вопросы, касающиеся проблемы общезначимости формул логики предикатов

Теорема Лёвенгейма-Сколема

Вспомним **определение выполнимости формул логики предикатов**:
Формула выполнима \Leftrightarrow она выполняется хотя бы в одной интерпретации

Представим себе, что для проверки выполнимости формулы разрешено перебрать **сколько угодно** интерпретаций и в каждой проверить выполнимость формулы

Следует ли перебрать все существующие в природе интерпретации, или же достаточно ограничиться только какими-нибудь «простыми»?

В таком переборе намного важнее природы предметов оказывается их **количество** (как было отмечено, например, в **блоке 7**)

Хотелось бы в полноценных рассуждениях о выполнимости формулы использовать только интерпретации с хотя и бесконечной, но всё же как можно меньшей предметной областью

Теорема Лёвенгейма-Сколема

Для любого предложения φ справедлива равносильность:

φ выполнимо $\Leftrightarrow \varphi$ имеет модель с не более чем счётной
предметной областью

Доказательство.

Совместим знания о корректности и полноте вывода
и определение выполнимости таблицы:

$\models \varphi \Leftrightarrow$ для $T_0 = \langle \varphi \mid \rangle$ не существует успешного табличного вывода

Построив вывод согласно стратегии
из доказательства теоремы о полноте, получим

- ▶ бесконечную ветвь $T_0 \rightarrow T_1 \rightarrow \dots$ вывода,
состоящую только из незакрытых таблиц
- ▶ интерпретацию \mathcal{I} с **не более чем счётной** предметной областью,
в которой выполняются все таблицы этой ветви —
в том числе и таблица T_0 ▼

Теорема компактности Мальцева

Вспомним **теорему о логическом следствии**:

методы проверки общезначимости формул можно применить и для проверки логического следования одних формул из других

$$\psi_1, \dots, \psi_k \models \varphi \Leftrightarrow \models \psi_1 \& \dots \& \psi_k \rightarrow \varphi$$

Здесь $\{\psi_1, \dots, \psi_k\}$ — это **конечная** база знаний, относительно которой требуется проверить достоверность извлечённого следствия

А можно ли предложить что-нибудь аналогичное для **бесконечных** баз знаний?

Оказывается, что, независимо от размера набора знаний, в достоверности логического следствия можно убедиться, выбрав для рассмотрения только некоторый **конечный** поднабор

Теорема компактности Мальцева

Для любого предложения φ и любого множества предложений Γ справедлива равносильность:

$\Gamma \models \varphi \Leftrightarrow$ существует конечное подмножество Γ' множества Γ ,
такое что $\Gamma' \models \varphi$

Доказательство.

$\Gamma \models \varphi \Leftrightarrow$ таблица $T = \langle \Gamma \mid \varphi \rangle$ невыполнима (почему?)
 \Leftrightarrow существует успешный табличный вывод \mathfrak{D} для T

Подмножество Γ_1 всех формул множества Γ , к которым применяются правила вывода в \mathfrak{D} , конечно, как и минимальное подмножество Γ_2 , такое что для каждого листа \mathfrak{D} существует формула из Γ_2 , содержащаяся в обеих частях таблицы этого листа (почему?)

Тогда для таблицы $\langle \Gamma_1 \cup \Gamma_2 \mid \varphi \rangle$,
также существует успешный табличный вывод (почему?)

Значит, $\Gamma_1 \cup \Gamma_2 \models \varphi$ ▼

Теорема компактности Мальцева

Множество предложений Γ называется **выполнимым** ($\models \Gamma^1$), если у этого множества есть хотя бы одна модель

Следствие. Для любого множества предложений Γ верно:

$$\not\models \Gamma$$

$$\Leftrightarrow$$

существует конечное подмножество Γ' множества Γ , такое что $\not\models \Gamma'$

Доказательство.

$$\not\models \Gamma$$

\Leftrightarrow (определения **логического следования** и **невыполнимости**)

$\Gamma \models \varphi$ для произвольно взятого невыполнимого предложения φ

\Leftrightarrow (**теорема компактности Мальцева**)

существует подмножество Γ' множества Γ , такое что $\Gamma' \models \varphi$

(для произвольно взятого невыполнимого предложения φ)

\Leftrightarrow (определения **логического следования** и **невыполнимости**)

существует подмножество Γ' множества Γ , такое что $\not\models \Gamma' \blacktriangledown$

1 Снова то же необщепотребимое обозначение

Автоматизация доказательства теорем

В свете того, что есть **стратегия** построения успешного табличного вывода для произвольной невыполнимой таблицы, естественно возникает вопрос:

А можно ли поручить проверку общезначимости формул ЛП компьютеру, чтобы он делал всю работу за нас?

Если формула общезначима, то это можно обосновать, придерживаясь упомянутой стратегии

А если необщезначима ... (?)

... то на этот счёт пока есть только **теорема о корректности:**
успешного вывода для соответствующей таблицы не существует

Познакомившись получше с логикой предикатов и логическими программами, обсудим и то, что целиком переложить такую работу на компьютер **невозможно**¹

Но если всё же попытаться, то ...

¹ То есть о том, что проблема общезначимости формул
алгоритмически неразрешима

Автоматизация доказательства теорем

Если программно реализовать стратегию построения логического вывода¹, то в результате получится **прувер**: средство доказательства теорем логики предикатов

First-order theorem prover

К прuverу разумно было бы предъявить такие требования:

- ▶ **корректность**: выдаются только правильные ответы — обязательно
- ▶ **полнота**: ответы выдаются всегда — желательно как можно лучше к этому приблизиться
- ▶ **эффективность**: ответы выдаются за разумное время — очень желательно

¹ Не обязательно стратегию из доказательства теоремы полноты.
Не обязательно полную стратегию. Не обязательно табличного вывода

Автоматизация доказательства теорем

Если программно реализовать стратегию построения логического вывода¹, то в результате получится **прувер**: средство доказательства теорем логики предикатов

First-order theorem prover

Один из **очень многих** примеров того, чего позволило добиться использование пруверов:² **строго доказана** корректность *nix-микроядра L4, и в процессе доказательства найдены и исправлены сотни ошибок в коде

1 Не обязательно стратегию из доказательства теоремы полноты.

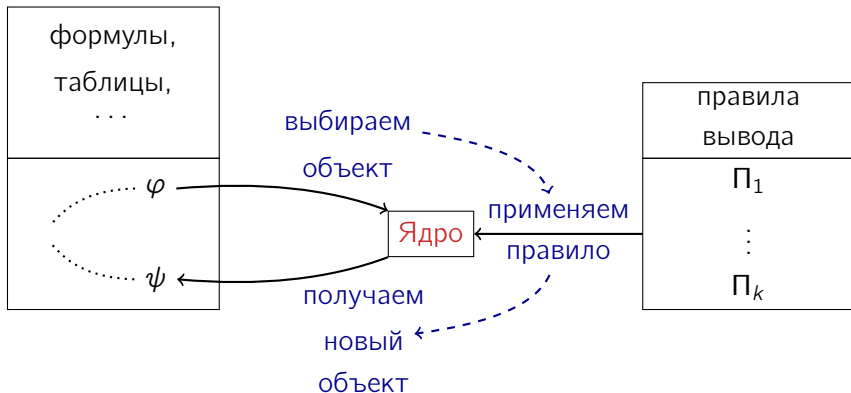
Не обязательно полную стратегию. Не обязательно табличного вывода

2 Klein et al. seL4: formal verification of an OS kernel. 2009.

Конкретно этот пример выбран только из-за наглядности, понятности и при этом «неоспоримой полезности» формулировки результата

Автоматизация доказательства теорем

Как устроены пружеры:



Автоматизация доказательства теорем

Представим себе прuver, способный проверять общезначимость формул логики предикатов **методом семантических таблиц**:

- ▶ корректный
- ▶ выдающий ответ «да» для **всех** общезначимых формул

Насколько эффективен может быть такой прuver?

Эффективность построения вывода определяется тем,

- ▶ как на каждом шаге выбираются формулы для применения к ним правил и
- ▶ какие термы подставляются при применении правил $L\forall$ и $R\exists$

Если прuverом осуществляется полный перебор всех формул, возникающих в таблицах, или перебор слишком большого числа термов, то такой прuver, вероятно, окажется неэффективным

Автоматизация доказательства теорем

Избежать полного перебора формул — непростая задача:

База знаний	Запрос
В огороде бузина Растёт(бузина , огород)	В Киеве дядька $\exists y$ (Дядька(y) & Живёт(y , Киев))

Автоматизация доказательства теорем

Избежать полного перебора формул — непростая задача:

База знаний	Запрос
<p>В огороде бузина Растёт(бузина, огород)</p> <p>Всё в огороде посадил дядька $\forall x (\text{Растёт}(x, \text{огород}) \rightarrow$ $\exists y (\text{Посадил}(y, x) \& \text{Дядька}(y)))$</p>	<p>В Киеве дядька $\exists y (\text{Дядька}(y)$ $\& \text{Живёт}(y, \text{Киев}))$</p>

Автоматизация доказательства теорем

Избежать полного перебора формул — непростая задача:

База знаний	Запрос
<p>В огороде бузина Растёт(бузина, огород)</p> <p>Всё в огороде посадил дядька $\forall x (\text{Растёт}(x, \text{огород}) \rightarrow$ $\exists y (\text{Посадил}(y, x) \& \text{Дядька}(y)))$</p> <p>Бузину сажают только Киевляне $\forall x (\text{Посадил}(x, \text{бузина})$ $\rightarrow \text{Живёт}(x, \text{Киев}))$</p>	<p>В Киеве дядька $\exists y (\text{Дядька}(y)$ $\& \text{Живёт}(y, \text{Киев}))$</p>

Автоматизация доказательства теорем

$$L\forall \frac{\langle \Gamma, \forall x \varphi(x) \mid \Delta \rangle}{\langle \Gamma, \forall x \varphi(x), \varphi(x)\{x/t\} \mid \Delta \rangle}$$

$$R\exists \frac{\langle \Gamma \mid \Delta, \exists x \varphi(x) \rangle}{\langle \Gamma \mid \Delta, \exists x \varphi(x), \varphi(x)\{x/t\} \rangle}$$

Избежать перебора большого числа термов — непростая задача

Для примера представим себе, что при построении вывода придётся перебрать все термы, составленные из **одного** функционального символа $f^{(2)}$, используемого не более 10 раз, и **двух** различных констант (*вроде бы это не очень большие термы?*)

Можно легко посчитать, что существует **более** 10^{300} различных термов такого вида

Число 10^{100} (на 200 нулей меньше) имеет особое название — **гугол**: это бессмысленно большое число, превосходящее число атомов в наблюдаемой вселенной

Автоматизация доказательства теорем

$$L\forall \frac{\langle \Gamma, \forall x \varphi(x) \mid \Delta \rangle}{\langle \Gamma, \forall x \varphi(x), \varphi(x)\{x/t\} \mid \Delta \rangle}$$

$$R\exists \frac{\langle \Gamma \mid \Delta, \exists x \varphi(x) \rangle}{\langle \Gamma \mid \Delta, \exists x \varphi(x), \varphi(x)\{x/t\} \rangle}$$

Избежать перебора большого числа термов — непростая задача

Существуют способы повышения эффективности перебора термов при построении логического вывода для доказательства общезначимости формул¹

Далее обсудим один из таких способов и основанный на нём метод:
метод резолюций

Заодно в процессе обсуждения познакомимся и с другими важными понятиями и задачами и методами, касающимися логики предикатов, и подготовим основу для обсуждения логического программирования

¹ J.A. Robinson: метод резолюций. С.Ю. Маслов: обратный вывод