Математическая логика и логическое программирование

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

Блок 14

Теорема Лёвенгейма-Сколема
Теорема компактности Мальцева
Автоматизация доказательства теорем

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

ВМК МГУ, 2022/2023, 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»?

При переборе интерпретаций неважна природа предметов, но (как было отмечено, например, в блоке 7) важно **количество** предметов

Оказывается, что можно рассуждать о выполнимости формулы, имея в виду только «достаточно небольшие» по размеру интерпретации

Теорема Лёвенгейма-Сколема

Для любого предложения φ справедлива равносильность:

$\Vdash \varphi \quad \Leftrightarrow \quad \varphi$ **имеет модель с <u>не более чем счётной</u> предметной областью**

Доказательство.

Вспомним о корректности и полноте вывода
и определение выполнимости таблицы:
$\Vdash \varphi \Leftrightarrow$ для $T_0 = \langle \varphi \mid \, \rangle$ не существует успешного табличного вывода

Построив вывод согласно стратегии из доказательства теоремы о полноте, получим

- бесконечную ветвь $T_0 \to T_1 \to \dots$ 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

$$\psi_1, \ldots, \psi_k \models \varphi \Leftrightarrow \models \psi_1 \,\&\, \ldots \,\&\, \psi_k \to \varphi$$

Здесь $\{\psi_1, \ldots, \psi_k\}$ — 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:

$\Gamma \models \varphi \quad \Leftrightarrow \quad$ существует <u>конечное</u> 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 формул множества Γ, к которым применяются правила вывода в $\mathfrak{D}$, <u>конечно</u>, как и подмножество Γ_2 формул, содержащихся в обеих частях каких-либо листьев вывода (почему?)

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

Значит, $\Gamma' \models \varphi$ ▼

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 вывода[1], то в результате получится прувер: средство доказательства теорем логики предикатов

First-order theorem prover

К пруверу 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 вывода[1], то в результате получится прувер: средство доказательства теорем логики предикатов

First-order theorem prover

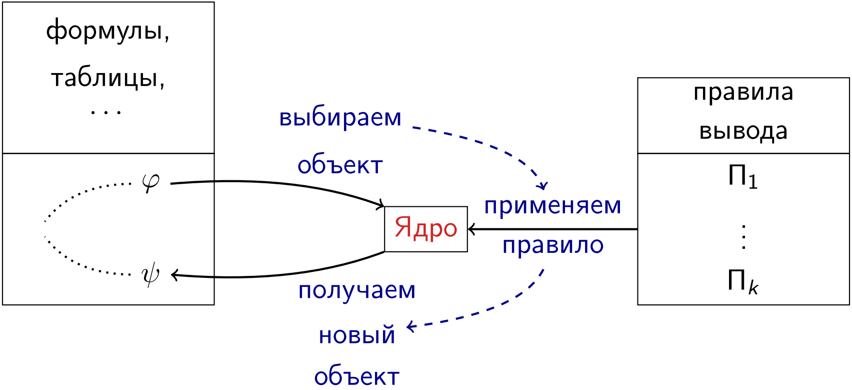
Один из **очень многих** примеров того, чего позволило добиться использование пруверов:[2]
строго доказана корректность *nix-микроядра L4,
и в процессе доказательства найдены и исправлены сотни ошибок в коде

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 пруверы:



Автоматизация доказательства теорем

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

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

Насколько эффективен может быть такой прувер?

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 прувером осуществляется полный перебор всех формул, возникающих в таблицах, или перебор слишком большого числа термов, то такой прувер, вероятно, окажется неэффективным

Автоматизация доказательства теорем

Избежать полного перебора формул — непростая задача:

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

Автоматизация доказательства теорем

Избежать полного перебора формул — непростая задача:

База знаний	Запрос
В огороде бузина $\text{Растёт}(\textbf{бузина}, \textbf{огород})$ Всё в огороде посадил дядька $\forall x\ (\text{Растёт}(x, \textbf{огород}) \to$ $\quad \exists y\ (\text{Посадил}(y, x)\ \&\ \text{Дядька}(y)))$	В Киеве дядька $\exists y\ (\text{Дядька}(y)$ $\quad \&\ \text{Живёт}(y, \textbf{Киев}))$

Автоматизация доказательства теорем

Избежать полного перебора формул — непростая задача:

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

Автоматизация доказательства теорем

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

Избежать перебора большого числа термов — непростая задача

Для примера представим себе, что при построении вывода придётся перебрать все термы, составленные из **одного** функционального символа $\mathsf{f}^{(2)}$, 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 нолей меньше) имеет особое название — гугол: это бессмысленно большое число, превосходящее число атомов в наблюдаемой вселенной

Автоматизация доказательства теорем

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

Избежать перебора большого числа термов — непростая задача

Существуют способы повышения эффективности перебора термов при построении логического вывода для доказательства общезначимости формул,[1]

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 математическую основу для логического программирования

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