Метод резолюции

Как найти доказательство для формулы исчисления предикатов

- Задачи проверки истинности и доказуемости формул исчисления предикатов неразрешимы.
- Однако, эти задачи неплохо решаются людьми в практических ситуациях.
- Налицо типичная постановка задачи искусственного интеллекта — можно попробовать что-то придумать.

Что можно сделать для разрешимости исчисления предикатов?

- По теореме о полноте можем рассматривать $(\models)$ вместо $(\vdash)$. Напомним: $\models \alpha$, если для всех $M = \langle D, F, P, E \rangle$ выполнено $M \models \alpha$.

Что можно сделать для разрешимости исчисления предикатов?

- По теореме о полноте можем рассматривать ($\models$) вместо ($\vdash$). Напомним: $\models \alpha$, если для всех $M = \langle D, F, P, E \rangle$ выполнено $M \models \alpha$.
- Что мешает проверке истинности:
  1. слишком сложные формулы — кванторы по бесконечным множествам;
  2. слишком большое разнообразие D, включая несчётные;
  3. даже $D = \mathbb{N}$ в формальной арифметике представляет проблему.

Что можно сделать для разрешимости исчисления предикатов?

- По теореме о полноте можем рассматривать $(\models)$ вместо $(\vdash)$. Напомним: $\models \alpha$, если для всех $M = \langle D, F, P, E \rangle$ выполнено $M \models \alpha$.
- Что мешает проверке истинности:
  1. слишком сложные формулы — кванторы по бесконечным множествам;
  2. слишком большое разнообразие D, включая несчётные;
  3. даже $D = \mathbb{N}$ в формальной арифметике представляет проблему.
- Будем последовательно бороться:
  1. упростим формулу (борьба с кванторами);
  2. заменим произвольное D на какое-то рекурсивно-перечислимое множество, устроенное некоторым фиксированным образом (борьба с разнообразием D);
  3. устроим правильный перебор, позволяющий быстро находить решения, если они есть (борьба с бесконечностью D).

Шаги рассуждения

1. Упростим формулу — избавимся от кванторов.
2. Заменяем модель (D и значения функциональных и предикатных символов).
3. Правильный перебор

Упрощаем формулу α, сколемизация

1. Для любой α найдётся β с поверхностными кванторами, что $\vdash \alpha \leftrightarrow \beta$. *В качестве примера* пусть в β оказались чередующиеся кванторы:

$$\beta := \forall x_1.\exists x_2.\forall x_3.\exists x_4 \dots \forall x_{n-1}.\exists x_n.\varphi$$

2. Исходная задача: проверка $\vdash \alpha$. Это эквивалентно $\vdash \beta$. Эквивалентно $\models \beta$. То есть, при любом D:
   - при любом x_1 найдётся такой x_2, что ...
   - при любом x_3 найдётся x_4, что ... (и т.д.) ...
   - что найдётся x_n, что φ истинен.
3. Заменим x_{2k} *функциями Сколема* $e_{2k}(x_1, x_3, \dots, x_{2k-1})$. Получим:

$$\eta := \forall x_1.\forall x_3 \dots \forall x_{n-1}.\varphi[x_2 := e_2(x_1), x_4 := e_4(x_1, x_3), \dots, x_n := e_n(x_1, x_3, \dots, x_{n-1})]$$

4. Сколемизация сохраняет выполнимость: $(\Rightarrow)$ если β истинна, то рассмотрим все $x_1, x_2, \dots,$ что φ истинна — и положим $e_k(x_1, x_3, \dots, x_{k-1}) := x_k$; $(\Leftarrow)$ если η истинна, то β истинна в той же оценке.

Сколемизация, избавляемся от чередований кванторов

1. Было: $\beta := \forall x_1.\exists x_2.\forall x_3.\exists x_4 \dots \forall x_{n-1}.\exists x_n.\varphi$ (n чередований кванторов).
2. Сколемизация не сохраняет общезначимость: $\vdash \forall x.\exists y.y > x$, но $\forall x.e(y) > x$ ложно при $e(y) := x$.
3. Поэтому мы проверяем иное свойство: при любом D найдутся e_i, что
$$\eta := \forall x_1.\forall x_3 \dots \forall x_{n-1}.\varphi[x_2 := e_2(x_1), \dots, x_n := e_n(x_1, x_3, \dots, x_{n-1})]$$
Два чередования: при любом D — найдутся e_{2k} — что $\forall x_1.\forall x_3.\dots.\forall.x_{n-1}.\varphi$
4. Как бы убрать и это чередование? С помощью отрицания — рассмотрим $\alpha' := \neg\alpha$ и сколемизированное представление для неё:
$$\eta' := \forall x_1.\forall x_3 \dots \forall x_{m-1}.\varphi'[x_2 := e'_2(x_1), \dots, x_m := e'_n(x_1, x_3, \dots, x_{m-1})]$$
Тогда $\models \alpha$ соответствует невыполнимости α', то есть невыполнимости η'.
5. То есть, $\models \alpha$ тогда и только тогда, когда при любом D — при любых e' — найдутся $x_1, x_3, \dots x_{m-1}$, что ложно
$$\varphi'[x_2 := e'_2(x_1), x_4 := e'_4(x_1, x_3), \dots, x_m := e'_n(x_1, x_3, \dots, x_{m-1})]$$

Преобразуем формулу в КНФ

Определение

КНФ (c конъюнктов, в каждом $d(c)$ дизъюнктов, каждый — предикатный символ с возможным отрицанием):

$$\zeta := \forall x_1.\forall x_3 \dots \forall x_{n-1}. \bigwedge_{k=\overline{1,c}} \left(\bigvee_{i=\overline{1,d(c)}} \delta_i^k \right)$$

при этом

$$\delta_i^k := P_i^k(\theta_{i,1}^k, \dots, \theta_{i,a(k,i)}^k) \text{ или } \delta_i^k := \neg P_i^k(\theta_{i,1}^k, \dots, \theta_{i,a(k,i)}^k)$$

Теорема

Для любой φ найдётся эквивалентная ей формула в КНФ.

Шаги рассуждения

1. Упростим формулу — поверхностные кванторы всеобщности, сколемизация, КНФ.
2. Заменяем модель (D и значения функциональных и предикатных символов).
3. Правильный перебор

Эрбранов универсум, основные термы

Определение

Пусть φ *— формула и* $\mathcal{F}_k$ *— все* k*-местные функциональные символы из* φ*. Тогда:*
$H^0_\varphi := \mathcal{F}_0$ *(либо* $\{a\}$*, если* $\mathcal{F}_0 = \varnothing$*);*
$H^{k+1}_\varphi := H^k_\varphi \cup \{\text{“}f(\text{“} + +x_1 + +\text{“}, \text{“} \dots \text{“}, \text{“} + +x_n + +\text{“})\text{“} \mid x_i \in H^k_\varphi, f \in \mathcal{F}_0\}$
Тогда $H_\varphi = \bigcup_n H^n_\varphi$ *— эрбранов универсум, его элементы — основные термы.*

Пример ($\varphi := P(a) \vee Q(f(b))$)

$$\begin{aligned}
H^0_\varphi &= \{a, b\} \\
H^1_\varphi &= \{a, b, f(a), f(b)\} \\
H^2_\varphi &= \{a, b, f(a), f(b), f(f(a)), f(f(b))\} \\
&\dots \\
H_\varphi &= \{f^{(n)}(x) \mid n \in \mathbb{N}_0, x \in \{a, b\}\}
\end{aligned}$$

Пример

$\varphi := P(0) \vee (P(x) \to P(x'))$ $\quad H_\varphi = \{0, 0', 0'', 0''', \dots\}$
$\varphi := P(x')$ $\quad H_\varphi = \{a, a', a'', a''', \dots\}$

Эрбранова интерпретация

Определение

Для бескванторной φ *рассмотрим* H_φ*, зададим оценку функциональных символов* f *из* φ*:*

$$\mathcal{F}_f(\llbracket\overline{\theta}\rrbracket) := \text{“}f(\text{“} + +\llbracket\overline{\theta}\rrbracket + +\text{“})\text{“}$$

Оценку для P *(*k*-местного предикатного символа из* φ*) зададим набором истинных значений* $S_P \subseteq (H_\varphi)^k$*:*

$$P(\theta_1, \dots, \theta_{a(i)}) \text{ истинно тогда и только тогда, когда } \langle\llbracket\theta_1\rrbracket, \dots, \llbracket\theta_k\rrbracket\rangle \in S_P$$

Также пусть $E : \mathcal{V} \to H_\varphi$*, тогда* $\langle H_\varphi, \mathcal{F}, \mathcal{P}, E\rangle$ *задаёт эрбранову интерперетацию.*

Пример

Пусть $\varphi := P(0) \vee (P(x) \to P(x'))$ *и* $S_P := \{0', 0'', 0'''''\}$*, тогда*

$$\llbracket\varphi\rrbracket^{x:=0} = \llbracket P(0) \vee (P(0) \to P(0'))\rrbracket = \text{И}$$
$$\llbracket\varphi\rrbracket^{x:=0''} = \llbracket P(0'') \vee (P(0'') \to P(0'''))\rrbracket = \text{Л}$$

Выполнимость не теряется. Заменяем D на H

Теорема

Формула выполнима тогда и только тогда, когда она выполнима на Эрбрановом универсуме.

Доказательство.

($\Rightarrow$) Пусть $M \models \forall \overline{x}.\varphi$. Тогда построим отображение eval : $H \to M$ (смысл названия вдохновлён языками программирования: eval(“$f(f(b))$“) перейдёт в $f(f(b))$, где f и b — из M).
Предикатам дадим согласованную оценку:
$P_H(t_1, \dots, t_n) = P_M(eval(t_1), \dots, eval(t_n))$. Очевидно, любая формула сохранит своё значение, кванторы всеобщности по меньшему множеству также останутся истинными.
($\Leftarrow$) Очевидно. $\square$

Шаги рассуждения

1. Упростим формулу — поверхностные кванторы всеобщности, сколемизация, КНФ.
2. Заменяем модель.
3. Правильный перебор.

Противоречивые системы дизъюнктов

Определение

Система дизъюнктов $S = \{\delta_1, \dots, \delta_n\}$ *противоречива, если для каждой оценки* $M = \langle D, P, F, E \rangle$ *найдётся* δ_t *и такой набор* $\overline{d} \in D$*, что* $[\![\delta_t]\!]^{\overline{x} := \overline{d}} = Л$.

Теорема

Система дизъюнктов противоречива, если она невыполнима в эрбрановых интерпретациях.

Основные примеры.

Рассмотрим сколемизированную формулу β в КНФ. Заметим, что если $\beta = \forall x_1. \dots . \forall x_k . \delta_1 \mathbin{\&} \delta_2 \mathbin{\&} \cdots \mathbin{\&} \delta_n$, то

$$\vdash \beta \leftrightarrow (\forall x_1. \dots . \forall x_k . \delta_1) \mathbin{\&} \cdots \mathbin{\&} (\forall x_1. \dots . \forall x_k . \delta_n)$$

Определение

Дизъюнкт с подставленными значениями из эрбранового универсума H_β (как строками) вместо переменных называется основным примером формулы β.

Пример

Пусть $\beta := \forall x. P(0) \mathbin{\&} (P(x) \vee P(x'))$, тогда $P(0''') \vee P(0'''')$ — основной пример, а $P(0''''')$ — нет.

Противоречивые множества основных примеров

Определение

Система основных примеров — все основные примеры, опровергаемые хоть при какой-то эрбрановой интерпертации $\mathcal{M}$:

$$\mathcal{E}_S = \{\delta_t[\overline{x} := \overline{d}] \mid \text{существует } \mathcal{M}, \text{ что } [\![\delta_t]\!]_{\mathcal{M}}^{\overline{x}:=\overline{d}} = Л; \quad d_i \in H_\beta\}$$

Определение

Система основных примеров E противоречива в эрбрановых интерпретациях, если для любой эрбрановой интерпретации $\mathcal{M}$ найдётся такой $\varepsilon \in E$, что $[\![\varepsilon]\!]_{\mathcal{M}} = Л$.

Теорема

Система дизъюнктов S противоречива тогда и только тогда, когда система её основных примеров $\mathcal{E}_S$ противоречива в эрбрановых интерпретациях.

Теорема Эрбрана

Теорема (Гёделя о компактности)

Если Γ — некоторое семейство бескванторных формул, то Γ имеет модель тогда и только тогда, когда любое его конечное подмножество имеет модель.

Теорема (Эрбрана)

Система дизъюнктов S противоречива тогда и только тогда, когда у $\mathcal{E}_S$ существует конечное противоречивое в эрбрановой интерпретации подмножество.

Доказательство.

($\Leftarrow$) Пусть $\{\varepsilon_1, \ldots, \varepsilon_t\} \subseteq \mathcal{E}_S$ противоречиво, $\varepsilon_i = \delta_{m_i}[\overline{x} := \overline{d_i}]$, где $\overline{d_i}$ — набор значений из H. То есть, для любой эрбрановой интерпретации M существует ε_p, что $[\![\varepsilon_p]\!]_M =$ Л. Отсюда, по теореме о выполнимости S тоже противоречива.
($\Rightarrow$) Если S противоречива, то $\mathcal{E}_S$ противоречива. Тогда у неё нет модели. Тогда у неё найдётся конечное противоречивое подмножество (компактность). $\square$

Возможно убедиться в невыполнимости за конечное время.

Главное — не запутаться в определениях

- Показываем невыполнимость формулы $\varphi = \bigvee \bigwedge \delta_i$ (в КНФ).
- По φ строим H_φ (эрбранов универсум, состоит из основных термов)
- Доопределяем функциональные символы как конкатенацию строк (эрбранова интерпретация). Выполнимость формулы эквивалентна её выполнимости в эрбрановой интерпретации.
- Заменяем формулу φ на множество $\{\delta_1, \ldots, \delta_n\}$ (система дизъюнктов S, убираем кванторы)
- Рассматриваем систему дизъюнктов с подставленными значениями из H_φ (основные примеры, убираем переменные).
- Оставляем только полезные — те, что опровергаются хотя бы в какой-то эрбрановой интерпретации ($\mathcal{E}_S$, система основных примеров).
- Невыполнимость формулы эквивалентна невыполнимости (противоречивости) системы дизъюнктов и эквивалентна противоречивости системы основных примеров.

Общая схема алгоритма

Цель алгоритма: убедиться, что α доказуемо.

1. По формуле α строим её отрицание $\neg\alpha$.
2. Приводим к виду с поверхностными кванторами, проводим сколемизацию, находим КНФ: $\beta = \forall x_1. \dots .\forall x_k.\delta_1 \mathbin{\&} \dots \mathbin{\&} \delta_n$.
3. Убедимся, что при любом D и значениях функциональных и предикатных символов и сколемовских функций e_k найдутся $d_i \in D$, что один из дизъюнктов δ_t при подстановке $\overline{x} := \overline{d}$ ложный.
4. Для этого строим универсум Эрбрана H, и систему основных примеров $\mathcal{E}_S$, её противоречивость эквивалентна невыполнимости β.
5. Конечное противоречивое подмножество по теореме Эрбрана обязательно находится в каком-то начальном отрезке $\{\varepsilon_1, \dots, \varepsilon_t\} \subseteq \mathcal{E}_S$ (если оно есть).

Пример: как проверяем выполнимость формулы?

Допустим, формула: $(\forall x.P(x) \mathbin{\&} P(x')) \mathbin{\&} \exists x.\neg P(x'''')$

1. Поверхностные кванторы, сколемизация, КНФ:
   $(\forall x.P(x)) \mathbin{\&} (\forall x.P(x')) \mathbin{\&} (\neg P(e''''))$
2. Строим эрбранов универсум: $H = \{e, e', e'', e''', \dots\}$
3. Если есть противоречие, то среди основных примеров:

$$\mathcal{E} = \{P(e), P(e'), P(e''), P(e'''), P(e''''), \neg P(e''''), \dots\}$$

Либо есть $\mathcal{M}$, что $[\![\mathop{\&}\mathcal{E}]\!]_\mathcal{M} =$ И, либо есть $\{\varepsilon_1, \dots, \varepsilon_n\} \subseteq \mathcal{E}$, что $[\![\varepsilon_t]\!]_\mathcal{M} =$ Л для какого-то t при каждой эрбрановой интерпретации $\mathcal{M}$.

Подмножество $\mathcal{E}$	выполнено в интерпретации	количество интерпретаций
$\{P(e)\}$	$[\![P(e)]\!] =$ И	2 варианта
$\{P(e), P(e')\}$	$[\![P(e)]\!] = [\![P(e')]\!] =$ И	4 варианта
...		
$\{P(e), \dots, P(e''''), \neg P(e'''')\}$	невыполнимо	32 варианта

Правило резолюции (исчисление высказываний)

Пусть даны два дизъюнкта, $\alpha_1 \vee \beta$ и $\alpha_2 \vee \neg\beta$. Тогда следующее правило вывода называется правилом резолюции:

$$\frac{\alpha_1 \vee \beta \qquad \alpha_2 \vee \neg\beta}{\alpha_1 \vee \alpha_2}$$

Теорема

Система дизъюнктов противоречива, если в процессе всевозможного применения правила резолюции будет построено явное противоречие, т.е. найдено два противоречивых дизъюнкта: β *и* $\neg\beta$.

Расширение правила резолюции на исчисление предикатов

Заметим, что правило резолюции для исчисления высказываний не подойдёт для исчисления предикатов.

$$S = \{P(x), \neg P(0)\}$$

Здесь $P(x)$ противоречит $\neg P(0)$, но правило резолюции для исчисления высказываний здесь неприменимо, потому что x можно заменять, это не константа:

$$\frac{P(x) \qquad \neg P(0)}{???}$$

Нужно заменять $P(x)$ на основные примеры, и искать среди них. Модифицируем правило резолюции для этого.

Алгебраические термы

Определение

Алгебраический терм

$$\theta := x|(f(\theta_1, \ldots, \theta_n))$$

где x—переменная, $f(\theta_1, \ldots, \theta_n)$—применение функции. Напомним, что константы — нульместные функциональные символы, собственно переменные будем обозначать последними буквами латинского алфавита.

Определение

Система уравнений в алгебраических термах $\begin{cases} \theta_1 = \sigma_1 \\ \vdots \\ \theta_n = \sigma_n \end{cases}$

где θ_i и σ_i — термы

Уравнение в алгебраических термах

Определение

$\{x_i\} = X$ — множество переменных, $\{\theta_i\} = T$ — множество термов.

Определение

Подстановка — отображение вида: $\pi_0 : X \to T$, тождественное почти везде (за исключением конечного числа переменных).
$\pi_0(x)$ может быть либо $\pi_0(x) = \theta_i$, либо $\pi_0(x) = x$.

Доопределим $\pi : T \to T$, где

1. $\pi(x) = \pi_0(x)$
2. $\pi(f(\theta_1, \ldots, \theta_k)) = f(\pi(\theta_1), \ldots, \pi(\theta_k))$

Определение

Решить уравнение в алгебраических термах — найти такую наиболее общую подстановку π, что $\pi(\theta_1) = \pi(\theta_2)$. Наиболее общая подстановка — такая, для которой другие подстановки являются её частными случаями.

Задача унификации

Определение

Пусть даны формулы α и β. Тогда решением задачи унификации будет такая наиболее общая подстановка $\pi = \mathcal{U}\left[\alpha, \beta\right]$, что $\pi(\alpha) = \pi(\beta)$.
Также, π назовём наиболее общим унификатором.

Пример

- *Формулы $P(a, g(b))$ и $P(c, d)$ не имеют унификатора (мы считаем, что a, b, c, d — нульместные функции, а g — одноместная функция).*
- *Проверим формулу на соответствие 11 схеме аксиом $(\forall x.\varphi) \rightarrow \varphi[x := \theta]$:*

$$(\forall x.P(x)) \rightarrow P(f(t, g(t), y))$$

Для этого решим задачу унификации: $\pi = \mathcal{U}\left[P(x), P(f(t, g(t), y))\right]$, тогда $\pi(x) = f(t, g(t), y)$.

Правило резолюции для исчисления предикатов

Определение

Пусть σ_1 и σ_2 — подстановки, заменяющие переменные в формуле на свежие. Тогда правило резолюции выглядит так:

$$\frac{\alpha_1 \vee \beta_1 \qquad \alpha_2 \vee \neg\beta_2}{\pi(\sigma_1(\alpha_1) \vee \sigma_2(\alpha_2))} \ \pi = \mathcal{U}[\sigma_1(\beta_1), \sigma_2(\beta_2)]$$

σ_1 и σ_2 разделяют переменные у дизъюнктов, чтобы π не осуществила лишние замены, ведь $\vdash (\forall x.P(x) \,\&\, Q(x)) \leftrightarrow (\forall x.P(x)) \,\&\, (\forall x.Q(x))$, но $\nvdash (\forall x.P(x) \vee Q(x)) \to (\forall x.P(x)) \vee (\forall x.Q(x))$.

Пример

$$\frac{Q(x) \vee P(x) \qquad \neg P(a) \vee T(x)}{Q(a) \vee T(x'')} \quad \textit{подстановки: } \sigma_1(x) = x', \sigma_2(x) = x'', \pi(x') = a$$

Метод резолюции

Ищем $\vdash \alpha$.

1. будем искать опровержение $\neg\alpha$.
2. перестроим $\neg\alpha$ в КНФ.
3. будем применять правило резолюции, пока получаем новые дизъюнкты и пока не найдём явное противоречие (дизъюнкты вида β и $\neg\beta$).

Если противоречие нашлось, значит, $\vdash \neg\neg\alpha$. Если нет — значит, $\vdash \neg\alpha$. Процесс может не закончиться.

SMT-решатели

Обычно требуется не логическое исчисление само по себе, а теория первого порядка. То есть, «Satisfability Modulo Theory», «выполнимость в теории» — вместо SAT, выполнимости.

- Иногда можно вложить теорию в логическое исчисление, даже в исчисление высказываний: $\overline{S_2 S_1 S_0} = \overline{A_1 A_0} + \overline{B_1 B_0}$

$$\begin{array}{ll} S_0 = A_0 \oplus B_0 & C_0 = A_0 \mathbin{\&} B_0 \\ S_1 = A_1 \oplus B_1 \oplus C_0 & C_1 = (A_1 \mathbin{\&} B_1) \vee (A_1 \mathbin{\&} C_0) \vee (B_1 \mathbin{\&} C_0) \\ S_2 = C_1 & \end{array}$$

- А можно что-то добавить прямо на уровень унификации / резолюции: Например, можем зафиксировать арифметические функции — и производить вычисления в правиле резолюции вместе с унификацией.
  Тогда противоречие в $\{x = 1 + 3 + 1, \neg x = 5\}$ можно найти за один шаг.

Уточнённые типы (Refinement types), LiquidHaskell

Определение

(Неформальное) Уточнённый тип — тип вида $\{\tau(x) \mid P(x)\}$, где P — некоторый предикат.

Пример на LiquidHaskell:

```
data [a] <p :: a -> a -> Prop> where
    | []  :: [a] <p>
    | (:) :: h:a -> [a<p h>]<p> -> [a]<p>
```

- `h:a` — голова (h) имеет тип a
- `[a<p h>]<p>` — хвост состоит из значений типа a, уточнённых p — $\{t : a \mid p\ h\ t\}$ (карринг: `a <p h>`).

```
{-@ type IncrList a = [a] <{\xi xj -> xi <= xj}> @-}
{-@ insertSort    :: (Ord a) => xs:[a] -> (IncrList a) @-}
insertSort []     = []
insertSort (x:xs) = insert x (insertSort xs)
```