Метод резолюции

Метод резолюции (что мы умеем, повторение)

Дана формула α.

1. Упростим формулу — поверхностные кванторы всеобщности, сколемизация. Умеем строить формулу β:

   $$\beta := \forall x_1.\forall x_2.\forall x_k.\delta_1(x_1, \dots, x_k) \,\&\, \cdots \,\&\, \delta_n(x_1, \dots, x_k)$$

   α доказуема тогда и только тогда, когда при всех оценках предикатных и функциональных символов найдётся значение сколемовских функций e_k, при которых β всегда истинна (слоёный пирог из кванторов).
2. Упрощаем предметное множество — заменили произвольный D на эрбранов универсум H. Выполнимость формулы эквивалентна выполнимости на эрбрановом универсуме.
3. Осталось избавиться от кванторов всеобщности и организовать правильный перебор (эрбранов универсум может быть бесконечным).

Оценка формулы на эрбрановом универсуме

Определение

Эрбранов универсум H_φ — всевозможные комбинации функциональных символов из формулы φ. Если в формуле нет нульместных функциональных символов, к множеству символов формулы добавляется свежий нульместный функциональный символ a и все комбинации с его участием.

Например, для $P(0) \vee (P(x) \rightarrow P(x'))$ эрбрановым универсумом будет $\{0, 0', 0'', 0''', \dots\}$, для $P(x')$ это будет $\{a, a', a'', a''', \dots\}$.

Определение

Если φ — бескванторная формула, то её эрбранова оценка задаётся как $\langle H_\varphi, F, P, E\rangle$, функции F определяются как текстовые подстановки $[\![f(\theta)]\!] = “f(“ + +[\![\theta]\!] + +“)“$, предикаты P задаются перечислением истинных.

Например, для $P(0) \vee (P(x) \rightarrow P(x'))$ эрбранова оценка при истинных предикатах $\{P(0'), P(0''), P(0''''')\}$ такова: $[\![\varphi]\!]^{x:=0} =$ И и $[\![\varphi]\!]^{x:=0''} =$ Л

Противоречивые системы дизъюнктов

Теорема (о выполнимости)

Формула выполнима тогда и только тогда, когда она выполнима в какой-то эрбрановой оценке.

Доказательство.

Доказано на предыдущей лекции. $\square$

Определение

Система дизъюнктов $S = \{\delta_1, \ldots, \delta_n\}$ *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 оценках.

Основные примеры.

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

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

Определение

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

Пример

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

Определение

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

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

Противоречивые множества основных примеров

Определение

Система основных примеров E противоречива в эрбрановой оценке (интерпретации), если для любой эрбрановой оценки M найдётся такой $\varepsilon \in E$, что $[\![\varepsilon]\!]_M =$ Л.

Теорема

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

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 оценки 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.

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: $\beta = \forall x_1. \dots .\forall x_k.\delta_1 \,\&\, \cdots \,\&\, \delta_n$.
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 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 оценке $\mathcal{M}$.

Подмножество $\mathcal{E}$	выполнено в оценке	количество оценок
$\{P(e)\}$	$[\![P(e)]\!] = И$	2 варианта
$\{P(e), P(e')\}$	$[\![P(e)]\!] = [\![P(e')]\!] = И$	4 варианта
$\dots$		
$\{P(e), \dots, P(e''''), \neg P(e'''')\}$	невыполнимо	64 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:

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

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 $\pi = \mathcal{U}[\alpha, \beta]$, что $\pi(\alpha) = \pi(\beta)$.
Также, η 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 f — одноместная функция).*
- *Проверим формулу на соответствие 11 схеме аксиом:*

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

Пусть $\pi = \mathcal{U}[P(x), P(f(t, g(t), y))]$, 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:

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

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

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)
```