

Desátá přednáška

NAIL062 Výroková a predikátová logika

Jakub Bulín (KTIML MFF UK)

Zimní semestr 2023

Program

- grounding, Herbrandova věta
- unifikace, unifikační algoritmus
- rezoluční pravidlo, rezoluční důkaz

Materiály

Zápisky z přednášky, Sekce 8.3-8.5 z Kapitoly 8

8.3 Grounding

- **základní (ground) instance** otevřené φ ve volných proměnných x_1, \dots, x_n je $\varphi(x_1/t_1, \dots, x_n/t_n)$, kde vš. t_i jsou konstantní

Herbrandova věta říká, že je-li **otevřená** teorie **nesplnitelná**, lze to doložit “na konkrétních prvcích”: existuje konečně mnoho **základních instancí** axiomů, jejichž konjunkce je nesplnitelná

- např. pro $T = \{P(x, y) \vee R(x, y), \neg P(c, y), \neg R(x, f(x))\}$ substituujeme **konstantní** termy $\{x/c, y/f(c)\}$:

$$(P(c, f(c)) \vee R(c, f(c))) \wedge \neg P(c, f(c)) \wedge \neg R(c, f(c))$$

- základní atomické sentence chápeme jako prvovýroky:

$$(p_1 \vee p_2) \wedge \neg p_1 \wedge \neg p_2$$

- to už snadno zamítneme výrokovou rezolucí
- p_1 znamená “platí $P(c, f(c))$ ”, p_2 znamená “platí $R(c, f(c))$ ”

Přímá redukce do výrokové logiky

Herbrandova věta + korektnost a úplnost výrokové rezoluce dává následující, neefektivní postup (S' je moc velká, i nekonečná):

1. $S \rightsquigarrow S' =$ množina všech základních instancí klauzulí z S
2. atomické sentence v S' chápeme jako prvovýroky
3. S nespílitelná $\Leftrightarrow S'$ zamítnutelná 'na úrovni výrokové logiky'

Např. pro $S = \{\{P(x, y), R(x, y)\}, \{\neg P(c, y)\}, \{\neg R(x, f(x))\}\}$

$$S' = \{\{P(c, c), R(c, c)\}, \{P(c, f(c)), R(c, f(c))\}, \{P(f(c), c), R(f(c), c)\}, \dots, \\ \{\neg P(c, c)\}, \{\neg P(c, f(c))\}, \{\neg P(c, f(f(c)))\}, \{\neg P(c, f(f(f(c))))\}, \dots, \\ \{\neg R(c, f(c))\}, \{\neg R(f(c), f(f(c)))\}, \{\neg R(f(f(c)), f(f(f(c))))\}, \dots\}$$

S' je nespílitelná obsahuje konečnou nespílitelnou podmnožinu:

$$\{\{P(c, f(c)), R(c, f(c))\}, \{\neg P(c, f(c))\}, \{\neg R(c, f(c))\}\} \vdash_R \square$$

Efektivnější je hledat vhodné základní instance **unifikací** [za chvíli]

Herbrandův model

Mějme jazyk $L = \langle \mathcal{R}, \mathcal{F} \rangle$ s alespoň jedním konstantním symbolem. L -struktura $\mathcal{A} = \langle A, \mathcal{R}^{\mathcal{A}}, \mathcal{F}^{\mathcal{A}} \rangle$ je **Herbrandův model**, jestliže:

- A je množina všech konst. L -termů (**Herbrandovo univerzum**)
- pro každý n -ární $f \in \mathcal{F}$ a (konstantní) $"t_1", \dots, "t_n" \in A$:
$$f^{\mathcal{A}}("t_1", \dots, "t_n") = "f(t_1, \dots, t_n)"$$
- speciálně, pro konstantní symbol $c \in \mathcal{F}$ je $c^{\mathcal{A}} = "c"$
- na relační symboly neklademe podmínky

Např. $L = \langle P, f, c \rangle$ (P unární rel., f binární funkční, c konstantní) **Herbrandův model** je každá struktura $\mathcal{A} = \langle A, P^{\mathcal{A}}, f^{\mathcal{A}}, c^{\mathcal{A}} \rangle$, kde

- $A = \{ "c", "f(c, c)", "f(c, f(c, c))", "f(f(c, c), c)" \dots \}$
- $c^{\mathcal{A}} = "c"$
- $f^{\mathcal{A}}("c", "c") = "f(c, c)", f^{\mathcal{A}}("c", "f(c, c)") = "f(c, f(c, c))",$
 $f^{\mathcal{A}}("f(c, c)", "c") = "f(f(c, c), c)",$ atd.
- $P^{\mathcal{A}} \subseteq A$ může být libovolná

Herbrandova věta

Věta (Herbrandova): Je-li T otevřená, v jazyce bez rovnosti a s alespoň jedním konstantním symbolem, potom:

- buď má T Herbrandův model, nebo
- existuje konečně mnoho základních instancí axiomů T , jejichž konjunkce je nesplnitelná.

Důkaz: T_{ground} = množina všech základních instancí axiomů T

Zkonstruujeme “systematické tablo” τ z T_{ground} s $F \perp$ v kořeni, ale z jazyka L , bez rozšíření o pomocné konstantní symboly na L_C . (Nepotřebujeme je, protože v T_{ground} nejsou kvantifikátory.)

Pokud má τ bezespornou větev, je “kanonický model” (opět bez pomocných symbolů) Herbrandovým modelem T .

Jinak je τ důkaz sporu, T_{ground} (a tedy i T) je nesplnitelná. Tablo τ je konečné, používá jen konečně mnoho $\alpha_{\text{ground}} \in T_{\text{ground}}$, jejich konjunkce už je nesplnitelná. □

- konstatní symbol potřebujeme, aby existovaly vůbec nějaké konstantní termy (ale není-li v L žádný, můžeme ho přidat)
- Herbrandův model je podobný kanonickému, ale nepřidáváme pomocné symboly, a neříkáme nic o relacích
- je-li jazyk s rovností, najdeme Herbrandův model pro T^* (přidané axiomy rovnosti) a faktorizujeme podle $=^A$

Důsledky Herbrandovy věty

Důsledek: Je-li T otevřená v jazyce s konstantním symbolem, potom T má model, právě když má model teorie T_{ground} .

Důkaz: \Rightarrow V modelu T platí i všechny základní instance axiomů. Je tedy i modelem T_{ground} .

\Leftarrow Pokud T nemá model, podle Herbrandovy věty je nějaká konečná podmnožina teorie T_{ground} nesplnitelná. \square

Důsledek: Mějme otevřenou $\varphi(x_1, \dots, x_n)$ v L s konst. symbolem. Potom existuje $m \in \mathbb{N}$ a konstantní L -termy t_{ij} ($i \in [m], j \in [n]$), že sentence $(\exists x_1) \dots (\exists x_n) \varphi(x_1, \dots, x_n)$ je pravdivá, právě když je následující formule (výroková) tautologie:

$$\varphi(x_1/t_{11}, \dots, x_n/t_{1n}) \vee \dots \vee \varphi(x_1/t_{m1}, \dots, x_n/t_{mn})$$

Důkaz: Je **pravdivá**, právě když $(\forall x_1) \dots (\forall x_n) \neg \varphi$ neboli $\neg \varphi$ je **nesplnitelná**. Stačí aplikovat Herbrandovu větu na $T = \{\neg \varphi\}$. \square

8.4 Unifikace

Příklady substitucí

Místo **všech základních** použijeme '**vhodné**' substituce (**unifikace**):

1. $\{P(x), Q(x, a)\}$ a $\{\neg P(y), \neg Q(b, y)\}$

- substitucí $\{x/b, y/a\}$ získáme $\{P(b), Q(b, a)\}$ a $\{\neg P(a), \neg Q(b, a)\}$, z nich rezolucí $\{P(b), \neg P(a)\}$
- nebo $\{x/y\}$ a rezolucí přes $P(y)$ máme $\{Q(y, a), \neg Q(b, y)\}$
- šlo by např. $\{x/a\}$, získat $\{Q(a, a), \neg Q(b, a)\}$, ale to je **horší**

2. $\{P(x), Q(x, z)\}$ a $\{\neg P(y), \neg Q(f(y), y)\}$

- lze použít $\{x/f(a), y/a, z/a\}$, získat $\{P(f(a)), Q(f(a), a)\}$ a $\{\neg P(a), \neg Q(f(a), a)\}$, rezolucí $\{P(f(a)), \neg P(a)\}$
- **lepší** je $\{x/f(z), y/z\}$, dává $\{P(f(z)), Q(f(z), z)\}$ a $\{\neg P(z), \neg Q(f(z), z)\}$, rezolventu $\{P(f(z)), \neg P(z)\}$
- proč lepší? **obecnější**, rezolventa 'říká více': $\{P(f(a)), \neg P(a)\}$ je důsledkem $\{P(f(z)), \neg P(z)\}$, ale nejsou ekvivalentní
- $\{x/f(a), y/a, z/a\}$ získáme **složením** $\{x/f(z), y/z\}$ a $\{z/a\}$

Substituce formálně

- **substituce** je konečná množina $\sigma = \{x_1/t_1, \dots, x_n/t_n\}$, kde x_i jsou navzájem různé proměnné, t_i jsou termy, t_i není x_i
 - **základní**: všechny termy t_i jsou konstantní
 - **přejmenování proměnných**: vš. t_i navzájem různé proměnné
 - **výraz** je term nebo literál (atomická formule nebo její negace)
 - **instance** výrazu E **při substituci** $\sigma = \{x_1/t_1, \dots, x_n/t_n\}$, $E\sigma$: simultánně nahradíme všechny výskyty x_i za termy t_i
 - pro množinu výrazů S je $S\sigma = \{E\sigma \mid E \in S\}$
-
- simultánně proto, aby výskyt x_i v termu t_j nevedl ke zřetězení
 - např. $S = \{P(x), R(y, z)\}$, $\sigma = \{x/f(y, z), y/x, z/c\}$

$$S\sigma = \{P(f(y, z)), R(x, c)\}$$

Skládání substitucí

- substitute lze skládat, $\sigma\tau$ znamená nejprve σ a potom τ
- chceme, aby platilo $E(\sigma\tau) = (E\sigma)\tau$, pro libovolný výraz E
- např. pro výraz $E = P(x, w, u)$ a substitute

$$\sigma = \{x/f(y), w/v\} \quad \tau = \{x/a, y/g(x), v/w, u/c\}$$

máme $E\sigma = P(f(y), v, u)$ a $(E\sigma)\tau = P(f(g(x)), w, c)$, takže:

$$\sigma\tau = \{x/f(g(x)), y/g(x), v/w, u/c\}$$

- skládání není komutativní, $\sigma\tau$ je (typicky) jiná než $\tau\sigma$, zde

$$\tau\sigma = \{x/a, y/g(f(y)), u/c, w/v\}$$

- ale je asociativní (takže nemusíme psát závorky v $\sigma_1\sigma_2\cdots\sigma_n$)

Bud' $\sigma = \{x_1/t_1, \dots, x_n/t_n\}$ a $\tau = \{y_1/s_1, \dots, y_m/s_m\}$, označme $X = \{x_1, \dots, x_n\}$ a $Y = \{y_1, \dots, y_m\}$. Složení σ a τ je substitute

$$\sigma\tau = \{x_i/t_i\tau \mid x_i \in X, x_i \neq t_i\tau\} \cup \{y_j/s_j \mid y_j \in Y \setminus X\}$$

Vlastnosti skládání

Tvrzení: Pro libovolné substituce σ , τ , ϱ a výraz E platí:

$$(i) (E\sigma)\tau = E(\sigma\tau) \quad (ii) (\sigma\tau)\varrho = \sigma(\tau\varrho)$$

Důkaz: (i) Buď $\sigma = \{x_1/t_1, \dots, x_n/t_n\}$ a $\tau = \{y_1/s_1, \dots, y_m/s_m\}$.

Stačí pro E proměnnou (substituce nemění ostatní symboly):

- pro $E = x_i$ je $E\sigma = t_i$ a $(E\sigma)\tau = t_i\tau = E(\sigma\tau)$
- pro $E = y_j \notin X$ je $E\sigma = E$ a $(E\sigma)\tau = E\tau = s_j = E(\sigma\tau)$
- je-li E jiná proměnná, potom $(E\sigma)\tau = E = E(\sigma\tau)$.

(i) opakovaným užitím (i) máme pro lib. výraz, tedy i proměnnou:

$$E((\sigma\tau)\varrho) = (E(\sigma\tau))\varrho = ((E\sigma)\tau)\varrho = (E\sigma)(\tau\varrho) = E(\sigma(\tau\varrho))$$

Z toho plyne, že $(\sigma\tau)\varrho$ a $\sigma(\tau\varrho)$ jsou touž substitucí.

(Podrobněji, zřejmě platí: $\pi = \{z_1/v_1, \dots, z_k/v_k\}$ právě když $z_i\pi = v_i$ a $E\pi = E$ je-li E proměnná různá od všech z_i .)

□

Unifikace

- **unifikace** pro $S = \{E_1, \dots, E_n\}$ je substituce σ taková, že $E_1\sigma = E_2\sigma = \dots = E_n\sigma$, tj. $S\sigma$ obsahuje jediný výraz
- pokud má S unifikaci, je **unifikovatelná**
- unifikace pro S je **nejobecnější**, pokud pro každou unifikaci τ pro S existuje substituce λ taková, že $\tau = \sigma\lambda$

NB: různé nejobecnějších unifikace pro S se liší jen přejmenováním proměnných

Např. pro $S = \{P(f(x), y), P(f(a), w)\}$

- $\sigma = \{x/a, y/w\}$ je nejobecnější unifikace
- $\tau = \{x/a, y/b, w/b\}$ je unifikace, ale není nejobecnější, nelze z ní získat např. unifikaci $\varrho = \{x/a, y/c, w/c\}$
- z nejobecnější unifikace σ získáme $\tau = \sigma\lambda$ pro $\lambda = \{w/b\}$

Unifikační algoritmus

- postupně od začátku výrazů aplikuje substituce
- buď p nejlevější pozice, na které se nějaké dva výrazy z S liší
- $D(S)$ je množina všech podvýrazů začínajících na pozici p
- $S = \{P(x, y), P(f(x), z), P(z, f(x))\}, p = 3, D(S) = \{x, f(x), z\}$

vtup: konečná množina výrazů $S \neq \emptyset$

výstup: nejobecnější unifikace σ nebo info, že není unifikovatelná

(0) nastav $S_0 := S, \sigma_0 := \emptyset, k := 0$

(1) pokud $|S_k| = 1$, vrať $\sigma = \sigma_0 \sigma_1 \cdots \sigma_k$

(2) zjisti, zda je v $D(S_k)$ proměnná x a term t **neobsahující** x

(3) pokud ano, nastav $\sigma_{k+1} := \{x/t\}, S_{k+1} := S_k \sigma_{k+1},$
 $k := k + 1$, a jdi na (1)

(4) pokud ne, odpověz, že S není unifikovatelná

NB: hledání x a t v kroku (2) je relativně výpočetně náročné

Ukázkový běh

$$S = S_0 = \{P(f(y, g(z)), h(b)), P(f(h(w), g(a)), t), P(f(h(b), g(z)), y)\}$$

($k = 0$) $|S_0| > 1$, $D(S_0) = \{y, h(w), h(b)\}$, proměnná y není v $h(w)$, nastavíme $\sigma_1 := \{y/h(w)\}$ a $S_1 = S_0\sigma_1$

$$S_1 = \{P(f(h(w), g(z)), h(b)), P(f(h(w), g(a)), t), P(f(h(b), g(z)), h(w))\}$$

($k = 1$) $D(S_1) = \{w, b\}$, $\sigma_2 = \{w/b\}$, $S_2 = S_1\sigma_2$

$$S_2 = \{P(f(h(b), g(z)), h(b)), P(f(h(b), g(a)), t)\}$$

($k = 2$) $D(S_2) = \{z, a\}$, $\sigma_3 = \{z/a\}$, $S_3 = S_2\sigma_3$

$$S_3 = \{P(f(h(b), g(a)), h(b)), P(f(h(b), g(a)), t)\}$$

($k = 3$) $D(S_3) = \{h(b), t\}$, $\sigma_4 = \{t/h(b)\}$, $S_4 = S_3\sigma_4$

$$S_4 = \{P(f(h(b), g(a)), h(b))\}$$

($k = 4$) $|S_4| = 1$, nejobecnější unifikace pro S je $\sigma = \sigma_1\sigma_2\sigma_3\sigma_4 = \{y/h(w)\}\{w/b\}\{z/a\}\{t/h(b)\} = \{y/h(b), w/b, z/a, t/h(b)\}$

Důkaz korektnosti

Tvrzení: Unifikační algoritmus je korektní. Pro sestrojenou σ navíc platí, že je-li τ libovolná unifikace, potom $\tau = \sigma\tau$.

Důkaz: Algoritmus vždy skončí, neboť v každém kroku eliminuje proměnnou. Skončí-li neúspěchem, nelze unifikovat S_k , tedy ani S .

Odpoví-li $\sigma = \sigma_0\sigma_1 \cdots \sigma_k$, zjevně jde o unifikaci. Zbývá dokázat, že je nejobecnější, k tomu stačí dokázat vlastnost 'navíc': Bud' τ lib. unifikace pro S . Indukcí pro $0 \leq i \leq k$ ukážeme $\tau = \sigma_0\sigma_1 \cdots \sigma_i\tau$

(báze indukce) Pro $i = 0$ je $\sigma_0 = \emptyset$, $\tau = \sigma_0\tau$ tedy platí triviálně.

(indukční krok) Bud' $\sigma_{i+1} = \{x/t\}$. Ukažme, že pro lib. proměnnou platí: $u\sigma_{i+1}\tau = u\tau$ Z toho okamžitě plyne i $\tau = \sigma_0\sigma_1 \cdots \sigma_i\sigma_{i+1}\tau$.

Pro $u \neq x$ je $u\sigma_{i+1} = u$, tedy i $u\sigma_{i+1}\tau = u\tau$. Je-li $u = x$, máme $u\sigma_{i+1} = x\sigma_{i+1} = t$. Protože τ unifikuje $S_i = S\sigma_0\sigma_1 \cdots \sigma_i$ a $x, t \in D(S_i)$, τ unifikuje i x a t , tzn. $t\tau = x\tau$, tj. $u\sigma_{i+1}\tau = u\tau$. \square

8.5 Rezoluční metoda

Příklad rezolučního kroku

Chceme-li ukázat $T \models \varphi$, skolemizací najdeme CNF formuli S ekvivalentní s $T \cup \{\neg\varphi\}$. Stačí najít rezoluční zamítnutí S .

Jediným podstatným rozdílem bude **rezoluční pravidlo**.

Rezolventou dvojice klauzulí bude klauzule, kterou lze odvodit aplikací (**nejobecnější**) **unifikace**. Nejprve příklad:

$$C_1 = \{P(x), Q(x, y), Q(x, f(z))\}, C_2 = \{\neg P(u), \neg Q(f(u), u)\}$$

Vyberme z C_1 **oba** pozitivní literály začínající Q , z C_2 negativní.

$S = \{Q(x, y), Q(x, f(z)), Q(f(u), u)\}$ lze unifikovat pomocí nejobecnější unifikace $\sigma = \{x/f(f(v)), y/f(v), z/v, u/f(v)\}$

- $C_1\sigma = \{P(f(f(v))), Q(f(f(v)), f(v))\}$
- $C_2\sigma = \{\neg P(f(v)), \neg Q(f(f(v)), f(v))\}$

z nich odvodíme rezolventu $C = \{P(f(f(v))), \neg P(f(v))\}$

Mějme klauzule C_1 a C_2 s disjunktními množinami proměnných tvaru

$$C_1 = C'_1 \sqcup \{A_1, \dots, A_n\}, \quad C_2 = C'_2 \sqcup \{\neg B_1, \dots, \neg B_m\}$$

kde $n, m \geq 1$ a $S = \{A_1, \dots, A_n, B_1, \dots, B_m\}$ lze unifikovat. Buď σ nejobecnější unifikace S . **Rezolventa** C_1 a C_2 je potom klauzule

$$C = C'_1\sigma \cup C'_2\sigma$$

Disjunktční množiny proměnných lze získat přejmenováním. Proč?

Např. z $\{\{P(x)\}, \{\neg P(f(x))\}\}$ lze získat \square , nahradíme-li $\{P(x)\}$ klauzulí $\{P(y)\}$. Ale $S = \{P(x), P(f(x))\}$ není unifikovatelná.

Rezoluční důkaz (odvození) klauzule C z formule S je konečná posloupnost klauzulí $C_0, C_1, \dots, C_n = C$ taková, že pro každé i je buď

- $C_i = C'_i \sigma$ pro nějakou $C'_i \in S$ a přejmenování proměnných σ
- nebo C_i je rezolventou nějakých C_j, C_k kde $j < i$ a $k < i$.

Existuje-li, je C **rezolucí dokazatelná** z S , $S \vdash_R C$. **(Rezoluční zamítnutí** S je rez. důkaz \square z S , potom je S **(rezolucí) zamítnutelná**.

Proč potřebujeme odstranit více literálů z jedné klauzule najednou?
Např. $S = \{\{P(x), P(y)\}, \{\neg P(x), \neg P(y)\}\}$ je zamítnutelná, ale neexistuje zamítnutí, které by v každém kroku odstranilo jen jeden.

Příklad

$$S = \{\{\neg P(x, y), \neg P(y, z), P(x, z)\}, \{\neg P(x, x)\}, \\ \{\neg P(x, y), P(y, x)\}, \{P(x, f(x))\}\}$$

Rezoluční zamítnutí:

$$\begin{aligned} &\{\neg P(x, y), \neg P(y, z), P(x, z)\}, \\ &\{P(x', f(x'))\}, \\ &\{\neg P(f(x), z), P(x, z)\}, \\ &\{\neg P(x, y), P(y, x)\}, \{P(x', f(x'))\}, \\ &\{P(f(x'), x')\}, \\ &\{P(x, x)\}, \\ &\{P(x', x')\}, \end{aligned}$$



Příklad rezolučního stromu

$$S = \{\{\neg P(x, y), \neg P(y, z), P(x, z)\}, \{\neg P(x, x)\}, \\ \{\neg P(x, y), P(y, x)\}, \{P(x, f(x))\}\}$$

