

Výroková a predikátová logika - XI

Petr Gregor

KTIML MFF UK

ZS 2020/21

Unifikace

Nechť $S = \{E_1, \dots, E_n\}$ je (konečná) množina výrazů.

- **Unifikace** pro S je substituce σ taková, že $E_1\sigma = E_2\sigma = \dots = E_n\sigma$, tj. $S\sigma$ je singleton.
- S je **unifikovatelná**, pokud má unifikaci.
- Unifikace σ pro S je **nejobecnější unifikace (mgu)**, pokud pro každou unifikaci τ pro S existuje substituce λ taková, že $\tau = \sigma\lambda$.

Např. $S = \{P(f(x), y), P(f(a), w)\}$ je unifikovatelná pomocí nejobecnější unifikace $\sigma = \{x/a, y/w\}$. Unifikaci $\tau = \{x/a, y/b, w/b\}$ dostaneme jako $\sigma\lambda$ pro $\lambda = \{w/b\}$. τ není mgu, nelze z ní získat unifikaci $\varrho = \{x/a, y/c, w/c\}$.

Pozorování Jsou-li σ, τ různé nejobecnější unifikace pro S , liší se pouze přejmenováním proměnných.

Unifikační algoritmus

Nechť S je (konečná) neprázdná množina výrazů a p je **nejlevější** pozice, na které se nějaké dva výrazy z S liší. Pak **neshoda** v S je množina $D(S)$ podvýrazů začínajících na pozici p ze **všech** výrazů v S .

Např. pro $S = \{P(x, y), P(f(x), z), P(z, f(x))\}$ je $D(S) = \{x, f(x), z\}$.

Vstup Neprázdná (konečná) množina výrazů S .

Výstup Nejobecnější unifikace σ pro S nebo “ S není unifikovatelná”.

- (0) Nechť $S_0 := S$, $\sigma_0 := \emptyset$, $k := 0$. (inicializace)
- (1) Je-li S_k singleton, vydej substituci $\sigma = \sigma_0\sigma_1 \cdots \sigma_k$. (mgu pro S)
- (2) Zjistí, zda v $D(S_k)$ existuje proměnná x a term t **neobsahující** x .
- (3) Pokud ne, vydej “ S není unifikovatelná”.
- (4) Jinak $\sigma_{k+1} := \{x/t\}$, $S_{k+1} := S_k\sigma_{k+1}$, $k := k + 1$ a jdi na (1).

Poznámka Test výskytu proměnné x v termu t v kroku (2) může být “drahý”.

Unifikační algoritmus - příklad

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

- 1) $S_0 = S$ není singleton a $D(S_0) = \{y, h(w), h(b)\}$ obsahuje term $h(w)$ a proměnnou y nevyskytující se v $h(w)$. Pak $\sigma_1 = \{y/h(w)\}$, $S_1 = S_0\sigma_1$, tj.

$$S_1 = \{P(f(h(w), g(z)), h(b)), P(f(h(w), g(a)), t), P(f(h(b), g(z)), h(w))\}.$$
- 2) $D(S_1) = \{w, b\}$, $\sigma_2 = \{w/b\}$, $S_2 = S_1\sigma_2$, tj.

$$S_2 = \{P(f(h(b), g(z)), h(b)), P(f(h(b), g(a)), t)\}.$$
- 3) $D(S_2) = \{z, a\}$, $\sigma_3 = \{z/a\}$, $S_3 = S_2\sigma_3$, tj.

$$S_3 = \{P(f(h(b), g(a)), h(b)), P(f(h(b), g(a)), t)\}.$$
- 4) $D(S_3) = \{h(b), t\}$, $\sigma_4 = \{t/h(b)\}$, $S_4 = S_3\sigma_4$, tj.

$$S_4 = \{P(f(h(b), g(a)), h(b))\}.$$
- 5) S_4 je singleton a nejobecnější unifikace pro S je

$$\sigma = \{y/h(w)\}\{w/b\}\{z/a\}\{t/h(b)\} = \{y/h(b), w/b, z/a, t/h(b)\}.$$

Unifikační algoritmus - korektnost

Tvrzení Pro každé S unifikační algoritmus vydá po konečně mnoha krocích korektní výsledek, tj. nejobecnější unifikaci σ pro S nebo pozná, že S není unifikovatelná. (*) Navíc, pro každou unifikaci τ pro S platí, že $\tau = \sigma\tau$.

Důkaz V každém kroku eliminuje jednu proměnnou, někdy tedy skončí.

- Skončí-li neúspěchem po k krocích, nelze unifikovat $D(S_k)$, tedy ani S .
- Vydá-li $\sigma = \sigma_0\sigma_1 \cdots \sigma_k$, je σ evidentně **unifikace** pro S .
- Dokážeme-li, že σ má vlastnost (*), je σ **nejobecnější** unifikace pro S .

- (1) Nechť τ je unifikace pro S . Ukážeme, že $\tau = \sigma_0\sigma_1 \cdots \sigma_i\tau$ pro každé $i \leq k$.
- (2) Pro $i = 0$ platí (1). Nechť $\sigma_{i+1} = \{x/t\}$, předpokládejme $\tau = \sigma_0\sigma_1 \cdots \sigma_i\tau$.
- (3) Stačí dokázat, že $v\sigma_{i+1}\tau = v\tau$ pro každou proměnnou v .
- (4) Pro $v \neq x$ je $v\sigma_{i+1} = v$, tedy platí (3). Nyní $v = x$ a $v\sigma_{i+1} = x\sigma_{i+1} = t$.
- (5) Jelikož τ unifikuje $S_i = S\sigma_0\sigma_1 \cdots \sigma_i$ a proměnná x i term t jsou v $D(S_i)$, musí τ unifikovat x a t , tj. $t\tau = x\tau$, jak bylo požadováno pro (3). □

Obecné rezoluční pravidlo

Nechť klauzule C_1, C_2 neobsahují stejnou proměnnou a jsou ve 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 $S = \{A_1, \dots, A_n, B_1, \dots, B_m\}$ lze unifikovat a $n, m \geq 1$. Pak klauzule

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

kde σ je **nejobecnější unifikace** pro S , je **rezolventa** klauzulí C_1 a C_2 .

Např. v klauzulích $\{P(x), Q(x, z)\}$ a $\{\neg P(y), \neg Q(f(y), y)\}$ lze unifikovat $S = \{Q(x, z), Q(f(y), y)\}$ pomocí nejobecnější unifikace $\sigma = \{x/f(y), z/y\}$ a získat z nich rezolventu $\{P(f(y)), \neg P(y)\}$.

***Poznámka** Podmínce o různých proměnných lze vyhovět přejmenováním proměnných v rámci klauzule. Je to nutné, např. z $\{\{P(x)\}, \{\neg P(f(x))\}\}$ lze po přejmenování získat \square , ale $\{P(x), P(f(x))\}$ nelze unifikovat.*

Rezoluční důkaz

Pojmy zavedeme jako ve VL, jen navíc dovolíme přejmenování proměnných.

- **Rezoluční důkaz (odvození)** klauzule C z formule S je **konečná** posloupnost $C_0, \dots, C_n = C$ taková, že pro každé $i \leq n$ je $C_i = C'_i \sigma$, kde $C'_i \in S$ a σ je přejmenování proměnných, nebo je C_i rezolventou nějakých dvou předchozích klauzulí (i stejných).
- Klauzule C je (rezolucí) **dokazatelná** z S , psáno $S \vdash_R C$, pokud má rezoluční důkaz z S .
- **Zamítnutí** formule S je rezoluční důkaz \square z S .
- S je (rezolucí) **zamítnutelná**, pokud $S \vdash_R \square$.

Poznámka *Eliminace více literálů najednou je někdy nezbytná, např.*

$S = \{\{P(x), P(y)\}, \{\neg P(x), \neg P(y)\}\}$ je rezolucí zamítnutelná, ale nemá zamítnutí, při kterém by se v každém kroku eliminoval pouze jeden literál.

Příklad rezoluce

Mějme teorii $T = \{\neg P(x, x), P(x, y) \rightarrow P(y, x), P(x, y) \wedge P(y, z) \rightarrow P(x, z)\}$.

Je $T \models (\exists x)\neg P(x, f(x))$? Tedy, je následující formule T' nespelnitelná?

$T' = \{\{\neg P(x, x)\}, \{\neg P(x, y), P(y, x)\}, \{\neg P(x, y), \neg P(y, z), P(x, z)\}, \{P(x, f(x))\}\}$

$T' \vdash_R \square$



x'/x

$\{P(x, x)\}$

$\{\neg P(x', x')\}$

$z/x, x'/x$

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

$\{P(f(x'), x')\}$

$y/f(x), x'/x$

$x/x', y/f(x')$

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

$\{P(x', f(x'))\}$

$\{\neg P(x, y), P(y, x)\}$

$\{P(x', f(x'))\}$

Korektnost rezoluce

Nejprve ukážeme, že obecné rezoluční pravidlo je korektní.

Tvrzení Nechť C je rezolventa klauzulí C_1, C_2 . Pro každou L -strukturu \mathcal{A} ,

$$\mathcal{A} \models C_1 \text{ a } \mathcal{A} \models C_2 \Rightarrow \mathcal{A} \models C.$$

Důkaz Nechť $C_1 = C'_1 \sqcup \{A_1, \dots, A_n\}$, $C_2 = C'_2 \sqcup \{\neg B_1, \dots, \neg B_m\}$, σ je nejobecnější unifikace pro $S = \{A_1, \dots, A_n, B_1, \dots, B_m\}$ a $C = C'_1\sigma \cup C'_2\sigma$.

- Jelikož C_1, C_2 jsou otevřené, platí i $\mathcal{A} \models C_1\sigma$ a $\mathcal{A} \models C_2\sigma$.
- Máme $C_1\sigma = C'_1\sigma \cup \{S\sigma\}$ a $C_2\sigma = C'_2\sigma \cup \{\neg(S\sigma)\}$.
- Ukážeme, že $\mathcal{A} \models C[e]$ pro každé e . Je-li $\mathcal{A} \models S\sigma[e]$, pak $\mathcal{A} \models C'_1\sigma[e]$ a tedy $\mathcal{A} \models C[e]$. Jinak $\mathcal{A} \not\models S\sigma[e]$, pak $\mathcal{A} \models C'_2\sigma[e]$ a tedy $\mathcal{A} \models C[e]$. \square

Věta (korektnost) *Je-li formule S rezolucí zamítnutelná, je S nesplnitelná.*

Důkaz Nechť $S \vdash_R \square$. Kdyby $\mathcal{A} \models S$ pro nějakou strukturu \mathcal{A} , z korektnosti rezolučního pravidla by platilo i $\mathcal{A} \models \square$, což není možné. \blacksquare

Lifting lemma

Rezoluční důkaz na úrovni VL lze “zdvihnout” na úroveň PL.

Lemma Necht' $C_1^* = C_1\tau_1$, $C_2^* = C_2\tau_2$ jsou *základní instance* klauzulí C_1 , C_2 *neobsahující stejnou proměnnou* a C^* je rezolventa C_1^* a C_2^* . Pak existuje rezolventa C klauzulí C_1 a C_2 taková, že $C^* = C\tau_1\tau_2$ je základní instance C .

Důkaz Předpokládejme, že C^* je rezolventa C_1^* , C_2^* přes *literál* $P(t_1, \dots, t_k)$.

- Pak lze psát $C_1 = C'_1 \sqcup \{A_1, \dots, A_n\}$ a $C_2 = C'_2 \sqcup \{\neg B_1, \dots, \neg B_m\}$, kde $\{A_1, \dots, A_n\}\tau_1 = \{P(t_1, \dots, t_k)\}$ a $\{\neg B_1, \dots, \neg B_m\}\tau_2 = \{\neg P(t_1, \dots, t_k)\}$.
- Tedy $(\tau_1\tau_2)$ unifikuje $S = \{A_1, \dots, A_n, B_1, \dots, B_m\}$ a je-li σ mgu pro S z unifikačního algoritmu, pak $C = C'_1\sigma \cup C'_2\sigma$ je rezolventa C_1 a C_2 .
- Navíc $(\tau_1\tau_2) = \sigma(\tau_1\tau_2)$ z vlastnosti (*) pro σ a tedy

$$\begin{aligned} C\tau_1\tau_2 &= (C'_1\sigma \cup C'_2\sigma)\tau_1\tau_2 = C'_1\sigma\tau_1\tau_2 \cup C'_2\sigma\tau_1\tau_2 = C'_1\tau_1 \cup C'_2\tau_2 \\ &= (C_1 \setminus \{A_1, \dots, A_n\})\tau_1 \cup (C_2 \setminus \{\neg B_1, \dots, \neg B_m\})\tau_2 \\ &= (C_1^* \setminus \{P(t_1, \dots, t_k)\}) \cup (C_2^* \setminus \{\neg P(t_1, \dots, t_k)\}) = C^*. \quad \square \end{aligned}$$

Úplnost

Důsledek Necht' S' je množina všech základních instancí klauzulí formule S . Je-li $S' \vdash_R C'$ (na úrovni VL), kde C' je základní klauzule, pak existuje klauzule C a základní substituce σ t.ž. $C' = C\sigma$ a $S \vdash_R C$ (na úrovni PL).

Důkaz Indukcí dle délky rezolučního odvození pomocí lifting lemmatu. \square

Věta (úplnost) Je-li formule S nespíitelná, je $S \vdash_R \square$.

Důkaz Je-li S nespíitelná, dle (důsledku) Herbrandovy věty je nespíitelná i množina S' všech základních instancí klauzulí z S .

- Dle úplnosti rezoluční metody ve VL je $S' \vdash_R \square$ (na úrovni VL).
- Dle předchozího důsledku existuje klauzule C a substituce σ taková, že $\square = C\sigma$ a $S \vdash_R C$ (na úrovni PL).
- Jediná klauzule, jejíž instance je \square , je klauzule $C = \square$. \blacksquare

Lineární rezoluce

Stejně jako ve VL, rezoluční metodu lze značně omezit (bez ztráty úplnosti).

- **Lineární důkaz** klauzule C z formule S je konečná posloupnost dvojic $(C_0, B_0), \dots, (C_n, B_n)$ t.ž. C_0 je **varianta** klauzule v S a pro každé $i \leq n$
 - B_i je varianta klauzule v S nebo $B_i = C_j$ pro nějaké $j < i$, a
 - C_{i+1} je rezolventa C_i a B_i , kde $C_{n+1} = C$.
- C je **lineárně dokazatelná** z S , psáno $S \vdash_L C$, má-li lineární důkaz z S .
- **Lineární zamítnutí** S je lineární důkaz \square z S .
- S je **lineárně zamítnutelná**, pokud $S \vdash_L \square$.

Věta S je lineárně zamítnutelná, právě když S je nesplnitelná.

Důkaz (\Rightarrow) Každý lineární důkaz lze transformovat na rezoluční důkaz.

(\Leftarrow) Plyne z úplnosti lineární rezoluce ve VL (nedokazováno), neboť lifting lemma zachovává **linearitu** odvození. \square

LI-rezoluce

Stejně jako ve VL, pro Hornovy formule můžeme lineární rezoluci dál omezit.

- **LI-rezoluce** (“linear input”) z formule S je lineární rezoluce z S , ve které je každá boční klauzule B_i variantou klauzule ze (vstupní) formule S .
- Je-li klauzule C dokazatelná LI-rezolucí z S , píšeme $S \vdash_{LI} C$.
- **Hornova formule** je množina (i nekonečná) Hornových klauzulí.
- **Hornova klauzule** je klauzule obsahující nejvýše jeden pozitivní literál.
- **Fakt** je (Hornova) klauzule $\{p\}$, kde p je pozitivní literál.
- **Pravidlo** je (Hornova) klauzule s právě jedním pozitivním a aspoň jedním negativním literálem. Pravidla a fakta jsou **programové klauzule**.
- **Cíl** je neprázdná (Hornova) klauzule bez pozitivního literálu.

Věta Je-li Hornova T splnitelná a $T \cup \{G\}$ nespjitelná pro cíl G , lze \square odvodit LI-rezolucí z $T \cup \{G\}$ začínající G .

Důkaz Plyne z Herbrandovy věty, stejné věty ve VL a lifting lemmatu. \square

Program v Prologu

Program (v Prologu) je Hornova formule obsahující pouze **programové klauzule**, tj. **fakta** nebo **pravidla**.

$syn(X, Y) :- otec(Y, X), muz(X).$	$\{syn(X, Y), \neg otec(Y, X), \neg muz(X)\}$
$syn(X, Y) :- matka(Y, X), muz(X).$	$\{syn(X, Y), \neg matka(Y, X), \neg muz(X)\}$
$muz(jan).$	$\{muz(jan)\}$
$otec(jiri, jan).$	$\{otec(jiri, jan)\}$
$matka(julie, jan).$	$\{matka(julie, jan)\}$
<hr/>	
$?- syn(jan, X)$	$P \models (\exists X) syn(jan, X) ? \quad \{\neg syn(jan, X)\}$

Zajímá nás, zda daný **existenční dotaz** vyplývá z daného programu.

Důsledek Pro program P a cíl $G = \{\neg A_1, \dots, \neg A_n\}$ v proměnných X_1, \dots, X_m

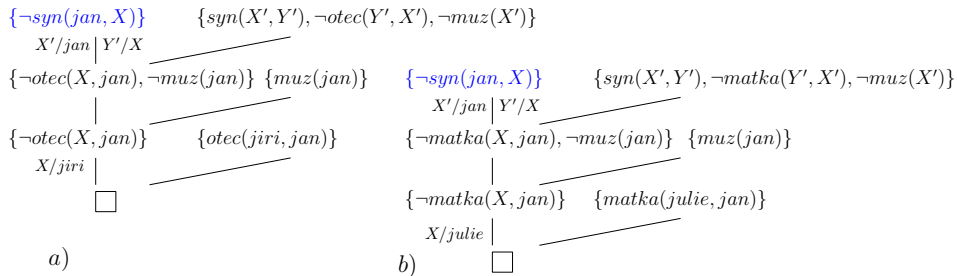
- $P \models (\exists X_1) \dots (\exists X_m) (A_1 \wedge \dots \wedge A_n)$, právě když
- \square lze odvodit LI-rezolucí z $P \cup \{G\}$ začínající (variantou) cíle G .

LI-rezoluce nad programem

Je-li odpověď na dotaz kladná, chceme navíc znát výstupní substituci.

Výstupní substitute σ LI-rezoluce \square z $P \cup \{G\}$ začínající $G = \{\neg A_1, \dots, \neg A_n\}$ je složení mgu v jednotlivých krocích (jen na proměnné v G). Platí,

$$P \models (A_1 \wedge \dots \wedge A_n)\sigma.$$



Výstupní substitute a) $X = \text{jiri}$, b) $X = \text{julie}$.

Hilbertovský kalkul

- základní logické spojky a kvantifikátory: \neg , \rightarrow , $(\forall x)$ (ostatní odvozené)
- dokazují se libovolné formule (nejen sentence)
- logické axiomy** (schémata logických axiomů)

$$(i) \quad \varphi \rightarrow (\psi \rightarrow \varphi)$$

$$(ii) \quad (\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$$

$$(iii) \quad (\neg\varphi \rightarrow \neg\psi) \rightarrow (\psi \rightarrow \varphi)$$

$$(iv) \quad (\forall x)\varphi \rightarrow \varphi(x/t) \quad \text{je-li } t \text{ substituovatelný za } x \text{ do } \varphi$$

$$(v) \quad (\forall x)(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow (\forall x)\psi) \quad \text{není-li } x \text{ volná proměnná ve } \varphi$$

kde φ , ψ , χ jsou libovolné formule (daného jazyka), t je libovolný term a x je libovolná proměnná.

- je-li jazyk s rovností, mezi logické axiomy patří navíc **axiomy rovnosti**
- odvozovací (deduktivní) pravidla**

$$\frac{\varphi, \varphi \rightarrow \psi}{\psi} \quad (\text{modus ponens}), \quad \frac{\varphi}{(\forall x)\varphi} \quad (\text{generalizace})$$

Pojem důkazu

Důkaz (Hilbertova stylu) formule φ z teorie T je **konečná** posloupnost $\varphi_0, \dots, \varphi_n = \varphi$ formulí taková, že pro každé $i \leq n$

- φ_i je logický axiom nebo $\varphi_i \in T$ (axiom teorie), nebo
- φ_i lze odvodit z předchozích formulí pomocí odvozovacích pravidel.

Formule φ je **dokazatelná** v T , má-li důkaz z T , značíme $T \vdash_H \varphi$.

Věta Pro každou teorii T a formuli φ , $T \vdash_H \varphi \Rightarrow T \models \varphi$.

Důkaz

- Je-li $\varphi \in T$ nebo logický axiom, je $T \models \varphi$ (logické axiomy jsou tautologie),
- jestliže $T \models \varphi$ a $T \models \varphi \rightarrow \psi$, pak $T \models \psi$, tj. *modus ponens* je korektní,
- jestliže $T \models \varphi$, pak $T \models (\forall x)\varphi$, tj. *pravidlo generalizace* je korektní,
- tedy každá formule vyskytující se v důkazu z T platí v T . \square

Poznámka Platí i *úplnost*, tj. $T \models \varphi \Rightarrow T \vdash_H \varphi$ pro každou teorii T a formuli φ .