KATEDRA
INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

JAN LAŠTOVIČKA
https://www.inf.upol.cz/lide/jan-lastovicka

LOGICKÉ PROGRAMOVÁNÍ

5. Unifikace

poznámky k přednášce, verze z 30. října 2026

Písmena θ, λ, σ označují substituce. Písmena s, t s případnými indexy označují termy.

Substituci, která je identitou na množině všech proměnných, nazýváme **prázdná substituce** a značíme ϵ.

Složením substitucí θ, λ rozumíme substituci $\theta\lambda$ takovou, že $\theta\lambda(X) = (X\theta)\lambda$ pro každou proměnnou X. Jestliže $\theta = \{X_1/s_1, \ldots, X_n/s_n\}$ a $\lambda = \{Y_1/t_1, \ldots, Y_m/t_m\}$, pak $\theta\lambda = \{Z_1/(Z_1\theta)\lambda, \ldots, Z_k/(Z_k\theta)\lambda\}$, kde $\{Z_1, \ldots, Z_k\}$ je množina všech proměnných takových, že $Z_i \in \{X_1, \ldots, X_n, Y_1, \ldots, Y_m\}$ a $(Z_i\theta)\lambda \neq Z_i$. Například složením substitucí $\{X/succ(Y)\}$ a $\{Y/zero\}$ je substituce $\{X/succ(zero), Y/zero\}$.

Tvrzení 1. *Platí* $(t\theta)\lambda = t(\theta\lambda)$.

Důkaz. Tvrzení dokážeme strukturální indukcí. Nechť X je proměnná. Poté $(X\theta)\lambda = X(\theta\lambda)$. Předpokládejme, že tvrzení platí pro $t_1, \ldots, t_n$. Nechť $f(t_1, \ldots, t_n)$ je term. Poté $(f(t_1, \ldots, t_n)\theta)\lambda = f((t_1\theta)\lambda, \ldots, (t_n\theta)\lambda) = f(t_1(\theta\lambda), \ldots, t_n(\theta\lambda)) = f(t_1, \ldots, t_n)(\theta\lambda)$. □

Tvrzení 2. *Skládání substitucí je na množině všech substitucí asociativní operace.*

Důkaz. Dokážeme $(\theta\lambda)\sigma = \theta(\lambda\sigma)$. Podle Tvrzení 1 máme $X((\theta\lambda)\sigma) = (X(\theta\lambda))\sigma = ((X\theta)\lambda)\sigma = (X\theta)(\lambda)\sigma) = X(\theta(\lambda\sigma))$. □

Unifikátorem dvou termů rozumíme substituci, jejíž aplikace na oba termy je týž term. Například $\{X/succ(Y)\}$ a $\{X/succ(zero), Y/zero\}$ jsou unifikátory termů $succ(X)$ a $succ(succ(Y))$, termy $succ(X)$ a $zero$ nemají žádný unifikátor.

Rovnicí rozumíme výraz $t_1 \approx t_2$, kde t_1, t_2 jsou termy. Například $succ(X) \approx succ(succ(Y))$ je rovnice. Substituci θ nezveme **řešením rovnice** $t_1 \approx t_2$, jestliže θ je unifikátor termů t_1, t_2. Množinu proměnných rovnice definujeme rovností $\mathrm{var}(t_1 \approx t_2) = \mathrm{var}(t_1) \cup \mathrm{var}(t_2)$. Aplikaci substituce na rovnici definujeme rovností $(t_1 \approx t_2)\theta = t_1\theta \approx t_2\theta$.

Soustavou rovnic rozumíme konečnou množinu rovnic. Substituce θ je **řešením soustavy rovnic**, jestliže je řešením každé rovnice v soustavě. Dvě soustavy rovnic E_1, E_2 nazveme **ekvivalentní**, jestliže substituce θ je řešením soustavy E_1 právě tehdy, když θ je řešením soustavy E_2. Množinu proměnných soustavy rovnic E definujeme rovností $\mathrm{var}(E) = \bigcup\{\mathrm{var}(t_1 \approx t_2) \mid t_1 \approx t_2 \in E\}$. Aplikaci substituce θ na soustavu rovnic E definujeme rovností $E\theta = \{(t_1 \approx t_2)\theta \mid t_1 \approx t_2 \in R\}$. O soustavě rovnic $\{X_1 \approx t_1, \ldots, X_n \approx t_n\}$ řekneme, že je **vyřešená**, jestliže proměnné $X_1, \ldots, X_n$ jsou po dvou různé a žádná z nich se nevyskytuje v žádném termu $t_1, \ldots t_n$. Soustavu rovnic **nelze vyřešit**, jestliže neexistuje s ní ekvivalentní vyřešená soustava.

Substituce θ **je obecnější než** substituce λ, jestliže pro nějakou substituci σ platí, že $\lambda = \theta\sigma$. Například substituce $\theta = \{X/succ(Y)\}$ je obecnější než substituce $\lambda = \{X/succ(zero), Y/zero\}$, protože pro $\sigma = \{Y/zero\}$ je $\lambda = \theta\sigma$.

Pokud T je množina substitucí a $\sigma \in T$, pak řekneme, že σ je **nejobecnější** substituce z T, jestliže σ je obecnější než každá substituce z T. Například můžeme říci, že θ je nejobecnější unifikátor termů t_1, t_2.

Tvrzení 3. *Jestliže soustava rovnic* $\{X_1 \approx t_1, \ldots, X_n \approx t_n\}$ *je vyřešená, pak* $\{X_1/t_1, \ldots, X_n/t_n\}$ *je její nejobecnější řešení.*

Důkaz. Označme $E = \{X_1 \approx t_1, \ldots, X_n \approx t_n\}$ a $\theta = \{X_1/t_1, \ldots, X_n/t_n\}$. Jistě je substituce θ řešením soustavy rovnic E. Dokážeme, že je nejobecnější. Nechť σ je řešením soustavy rovnic E. Máme $X_i\theta = t_i$ a $X_i\sigma = t_i\sigma$ pro $1 \leq i \leq n$. Dále platí, že $X_i\theta\sigma = t_i\sigma = X_i\sigma$ pro $1 \leq i \leq n$ a $X\theta\sigma = X\sigma$ pro $X \notin \{X_1, \ldots, X_n\}$. Tedy $\theta\sigma = \sigma$ a proto je θ obecnější než σ. □

Pravidla pro řešení soustavy rovnic:

1. $$\frac{E \cup \{f(s_1, \ldots, s_n) \approx f(t_1, \ldots, t_n)\}}{E \cup \{s_1 \approx t_1, \ldots, s_n \approx t_n\}}$$

2. $$\frac{E \cup \{f(s_1, \ldots, s_n) \approx g(t_1, \ldots, t_m)\}}{\textbf{failure}} \quad f \neq g$$

3. $$\frac{E \cup \{X \approx X\}}{E}$$

4. $$\frac{E \cup \{t \approx X\}}{E \cup \{X \approx t\}} \quad t \text{ není proměnná}$$

5. $$\frac{E \cup \{X \approx t\}}{\textbf{failure}} \quad t \neq X \text{ a } t \text{ obsahuje } X$$

6. $$\frac{E \cup \{X \approx t\}}{E\{X/t\} \cup \{X \approx t\}} \quad t \text{ neobsahuje obsahuje } X \text{ a } E \text{ obsahuje } X$$

Algoritmus řešení soustavy rovnic:

- Vstup: Soustava rovnic E.
- Výstup: Vyřešená soustava rovnic, nebo **failure**.
- Průběh: Dokud je to možné, aplikujeme na E pravidla pro řešení soustavy rovnic.

První příklad průběhu algoritmu:

1. $\{\underline{f(g(Y), a) \approx f(X, X)}\}$ (1. pravidlo)
2. $\{g(Y) \approx X, \underline{a \approx X}\}$ (4. pravidlo)
3. $\{g(Y) \approx X, \underline{X \approx a}\}$ (6. pravidlo)
4. $\{\underline{g(Y) \approx a}, X \approx a\}$ (2. pravidlo)
5. **failure**

Druhý příklad průběhu algoritmu:

1. $\{\underline{f(g(Y,b),X) \approx f(X,g(c,Z))}\}$ (1. pravidlo)
2. $\{g(Y,b) \approx X, \underline{X \approx g(c,Z)}\}$ (6. pravidlo)
3. $\{\underline{g(Y,b) \approx g(c,Z)}, X \approx g(c,Z)\}$ (1. pravidlo)
4. $\{Y \approx c, \underline{b \approx Z}, X \approx g(c,Z)\}$ (4. pravidlo)
5. $\{Y \approx c, \underline{Z \approx b}, X \approx g(c,Z)\}$ (6. pravidlo)
6. $\{Y \approx c, Z \approx b, X \approx g(c,b)\}$

Třetí příklad průběhu algoritmu:

1. $\{\underline{f(g(X),X,Z) \approx f(Y,Y,Z)}\}$ (1. pravidlo)
2. $\{g(X) \approx Y, X \approx Y, \underline{Z \approx Z}\}$ (3. pravidlo)
3. $\{g(X) \approx Y, \underline{X \approx Y}\}$ (6. pravidlo)
4. $\{\underline{g(Y) \approx Y}, X \approx Y\}$ (4. pravidlo)
5. $\{\underline{Y \approx g(Y)}, X \approx Y\}$ (5. pravidlo)
6. **failure**

O uspořádané množině řekneme, že je **dobře uspořádána**, jestliže každá její neprázdná podmnožina má nejmenší prvek. Množina všech přirozených čísel s přirozeným uspořádáním je dobře uspořádána.

Nechť A, B jsou uspořádané množiny, pak na $A \times B$ definujeme **lexikografické uspořádání** tak, že položíme $(a_1, b_1) \leq (a_2, b_2)$, jestliže $a_1 < a_2$ nebo ($a_1 = a_2$ a $b_1 \leq b_2$).

Tvrzení 4. *Jestliže A, B jsou dobře uspořádány, pak množina $A \times B$ je lexikografickým uspořádáním dobře uspořádána.*

Důkaz. Nechť $C \subseteq A \times B$ je neprázdná. Pak množina $\{a \mid (a,b) \in C\}$ je neprázdná a má tedy nejmenší prvek a_0. Množina $\{b \mid (a_0,b) \in C\}$ je také neprázdná a má nejmenší prvek b_0. Uspořádaná dvojice (a_0, b_0) je nejmenším prvkem množiny C. □

Podle předchozího tvrzení je množina $(\mathbb{N}_0)^k$, kde $k > 1$, uspořádaná lexikograficky dobře uspořádána.

Tvrzení 5. *Nechť A je uspořádaná množina, která je dobře uspořádána. Pak neexistuje nekonečná klesající posloupnost prvků z A.*

Důkaz. Pro důkaz sporem předpokládejme, že $(a_n)_{n=0}^{\infty}$ je nekonečná klesající posloupnost prvků z A. Pak množina $B = \{a_n \mid n \in \mathbb{N}_0\}$ je neprázdná a má nejmenší prvek b. Protože $b \in B$ musí existovat přirozené číslo n takové, že $a_n = b$, ale $a_{n+1} < a_n$ a $a_{n+1} \in B$, což je spor. □

Tvrzení 6. *Algoritmus řešení soustavy rovnic vrátí ekvivalentní vyřešenou soustavu, nebo* **failure** *v případě, že soustavu vyřešit nelze.*

Důkaz. Dokážeme, že algoritmus vždy skončí. Uvažujme nekonečnou posloupnost soustav $E_1, E_2, \ldots$ odpovídající krokům algoritmu. Každé soustavě E_i přiřaďme uspořádanou trojici (a_i, b_i, c_i) přirozených čísel, kde a_i je počet proměnných, které se vyskytují ve více než jedné rovnici, b_i počet výskytů funkčních symbolů na levých stranách rovnic a c_i je počet rovnic v soustavě. Kontrolou pravidel ověříme, že v lexikografickém uspořádání je $(a_{i+1}, b_{i+1}, c_{i+1})$ je ostře menší než (a_i, b_i, c_i) pro každé přirozené číslo i. Obdržíme tak nekonečnou klesající posloupnost a dostáváme rozpor s Tvrzením 5, neboť množina $(\mathbb{N}_0)^3$ je lexikografickým uspořádáním dobře uspořádána.

Je jasné, že pokud algoritmus skončí, tak výsledkem je vyřešená soustava rovnic nebo **failure**, když poslední soustava nemá řešení.

Zbývá dokázat, že aplikováním pravidla obdržíme ekvivalentní soustavu. U všech nezamítacích pravidel je to zřejmé až na šesté pravidlo. Nechť $E_1 = \{s_1 \approx t_1, \ldots, s_n \approx t_n\} \cup \{X \approx t\}$ je soustava rovnic splňující předpoklad šestého pravidla. Označme $E_2 = \{s_1\sigma \approx t_1\sigma, \ldots, s_n\sigma \approx t_n\sigma\} \cup \{X \approx t\}$ soustavu vzniklou aplikací šestého pravidla na E_1, kde $\sigma = \{X/t\}$. Pokud θ splňuje $X\theta = t\theta$, pak $\sigma\theta = \theta$, protože $X\sigma\theta = t\theta = X\theta$.

Nechť θ je řešením soustavy E_1. Poté $s_i\theta = t_i\theta$ pro každé $1 \leq i \leq n$ a $X\theta = t\theta$ a tedy $\sigma\theta = \theta$. Proto $s_i\sigma\theta = t_i\sigma\theta$ pro každé $1 \leq i \leq n$. Dostáváme, že θ je řešením soustavy E_2.

Nechť naopak θ je řešení soustavy E_2. Pak $X\theta = t\theta$ a opět $\sigma\theta = \theta$ a proto je θ řešením také soustavy E_1.

□

Algoritmus unifikace:

- Vstup: Termy t_1, t_2.
- Výstup: Nejobecnější unifikátor termů t_1, t_2, nebo **failure**, v případě, že termy nemají unifikátor.
- Průběh: Aplikuj algoritmus řešení soustavy rovnic na $\{t_1 \approx t_2\}$. V případě úspěchu získej z vyřešené soustavy rovnic její nejobecnější řešení.