Prenexová normální forma (pnf)

- cíl: převést libovolnou (uzavřenou) formuli do tvaru, v němž jsou všechny kvantifikátory na začátku a následuje otevřené (= bez kvantifikátorů) jádro v nkf (ndf)

$$Qx_1 \dots Qx_n((A_{1_1} \vee \dots \vee A_{1_{l_1}}) \wedge (A_{2_1} \vee \dots \vee A_{2_{l_2}}) \wedge \dots \\ \wedge (A_{m_1} \vee \dots \vee A_{m_{l_m}}))$$

- příklad: $\forall x \forall y \exists z \forall w((P(x,y) \vee \neg Q(z)) \wedge (R(x,w) \vee R(y,w)))$
- **Věta:** pro každou formuli existuje ekvivalentní formule v konjunktivní (disjunktivní) prenexové normální formě.

Pnf: algoritmus převodu

1. eliminovat zbytečné kvantifikátory
2. přejmenovat korektně proměnné tak, aby u každého kvantifikátoru byla jiná proměnná
3. eliminovat všechny spojky různé od $\neg, \wedge$ a $\vee$
4. přesunout negaci dovnitř, je-li potřeba:
   $\neg\forall x A$ nahradit $\exists x \neg A$
   $\neg(A \wedge B)$ nahradit $\neg A \vee \neg B$ apod.
5. přesunout kvantifikátory doleva ($\circ \in \{\wedge, \vee\}, Q \in \{\forall, \exists\}$):
   $A \circ QxB$ nahradit $Qx(A \circ B)$
   $QxA \circ B$ nahradit $Qx(A \circ B)$
6. použít distributivní zákony k převodu jádra do nkf (ndf):
   $A \vee (B \wedge C)$ nahradit $(A \vee B) \wedge (A \vee C)$
   $(A \wedge B) \vee C$ nahradit $(A \vee C) \wedge (B \vee C)$

Převod do pnf – příklad

Příklad: převeďte do konjunktivní pnf formuli
$\forall x \exists y \neg (P(x,y) \Rightarrow \forall z R(y)) \vee \neg \exists x Q(x)$.

1. $\forall x \exists y \neg (P(x,y) \Rightarrow R(y)) \vee \neg \exists x Q(x)$ (zbytečné $\forall z$)
2. $\forall x_1 \exists y \neg (P(x_1,y) \Rightarrow R(y)) \vee \neg \exists x_2 Q(x_2)$ (přejmenování x)
3. $\forall x_1 \exists y \neg (\neg P(x_1,y) \vee R(y)) \vee \neg \exists x_2 Q(x_2)$ (eliminace $\Rightarrow$)
4. $\forall x_1 \exists y (P(x_1,y) \wedge \neg R(y)) \vee \forall x_2 \neg Q(x_2)$ (přesun negace 2x)
5. $\forall x_1 (\exists y (P(x_1,y) \wedge \neg R(y)) \vee \forall x_2 \neg Q(x_2))$ (posun $\forall x_1$ doleva)
   $\forall x_1 \exists y ((P(x_1,y) \wedge \neg R(y)) \vee \forall x_2 \neg Q(x_2))$ (posun $\exists y$ doleva)
   $\forall x_1 \exists y \forall x_2 ((P(x_1,y) \wedge \neg R(y)) \vee \neg Q(x_2))$ (posun $\forall x_2$ doleva)
6. $\forall x_1 \exists y \forall x_2 ((P(x_1,y) \vee \neg Q(x_2)) \wedge (\neg R(y) \vee \neg Q(x_2)))$ (jádro do nkf)

Skolemizace

- převod formulí na formule bez existenčních kvantifikátorů v jazyce, který je rozšířen o tzv. *Skolemovy funkce*; zachovává splnitelnost
- idea převodu: formuli $\forall x_1 \ldots \forall x_n \exists y P(x_1, \ldots, x_n, y)$ transformujeme na $\forall x_1 \ldots \forall x_n P(x_1, \ldots, x_n, f(x_1, \ldots, x_n))$
- příklad: mějme celá čísla s $+$. Formuli $\forall x \exists y(x + y = 0)$ převedeme na $\forall x(x + f(x) = 0)$. Interpretace f – unární funkce, která pro daný argument vrátí opačné číslo.
- *Skolemova normální forma* je prenexová normální forma pouze s univerzálními kvantifikátory.
- **Věta:** každou formuli A lze převést na takovou formuli A' ve Skolemově normální formě, že A je splnitelná právě když A' je splnitelná.

Algoritmus převodu do Skolemovy nf

1. převést formuli do (konjunktivní) pnf
2. provést Skolemizaci: odstranit všechny existenční kvantifikátory a nahradit jimi vázané proměnné pomocnými Skolemovými funkcemi

- příklad 1: převeďte do Skolemovy nf formuli
  $\forall x \exists y \neg (P(x,y) \Rightarrow \forall z R(y)) \vee \neg \exists x Q(x)$
  1. $\forall x_1 \exists y \forall x_2 ((P(x_1,y) \vee \neg Q(x_2)) \wedge (\neg R(y) \vee \neg Q(x_2)))$
  2. $\forall x_1 \forall x_2 ((P(x_1,f(x_1)) \vee \neg Q(x_2)) \wedge (\neg R(f(x_1)) \vee \neg Q(x_2)))$

- příklad 2: převeďte do Skolemovy nf následující formuli v pnf
  $\forall x \exists y \forall z \exists w (P(x,y) \vee \neg Q(z,w))$
  $\forall x \forall z (P(x,f_1(x)) \vee \neg Q(z,f_2(x,z)))$

Herbrandova věta I

- motivace: hledáme snazší prostředky k určení, zda daná množina formulí je splnitelná
- pracujeme s množinou S formulí ve Skolemově nf (univerzální kvantifikátory se často při zápisu vynechávají), jejími konstantami (alespoň jedna, příp. přidaná mimo S), funkčními a predikátovými symboly
- *Herbrandovo univerzum* $U(S)$ je množina všech uzavřených termů, které lze utvořit z konstant a funkčních symbolů z S (tzv. *základní termy*)
  př.: pro $S = \{P(f(0))\}$ je $U(S) = \{0, f(0), f(f(0)), f(f(f(0))), \ldots\}$
- *Herbrandova báze* $B(S)$ je množina všech atomických formulí, které lze vytvořit nad prvky $U(S)$;
  $B(S) = \{P(t_1, \ldots, t_n) | t_i \in U(S), P$ je predik. symbol figurující v $S\}$
  př.: pro $S = \{P(f(0))\}$ je $B(S) = \{P(0), P(f(0)), P(f(f(0))), \ldots\}$

Herbrandova věta II

- *Herbrandova interpretace* je libovolná podmnožina báze $B(S)$ zahrnující ty aplikace predikátů na prvky univerza, které jsou pravdivé
  Poznámka: s funkcemi a konstantami lze pracovat i nadále pouze na symbolické úrovni
- *Herbrandův model* $M(S)$ množiny S je taková Herbrandova interpretace, ve které jsou všechny formule z S pravdivé
- **Herbrandova věta:** buď existuje Herbrandův model S nebo existuje konečně mnoho uzavřených instancí prvků S, jejichž konjunkce neplatí
- slabší tvrzení: S je splnitelná právě tehdy, když existuje její Herbrandův model
- závěr: k rozhodnutí o splnitelnosti množiny již nepotřebujeme brát v úvahu všechny možné interpretace, stačí pracovat pouze se ‚symbolickými' Herbrandovými interpretacemi

Herbrandovy modely – příklady

Příklad 1: $S = \{P(0), P(s(x)) \vee \neg P(x)\}$ (předp. $\forall$ kvantifikovány)

- $U(S) = \{0, s(0), s(s(0)), s(s(s(0))), \ldots\}$
  $B(S) = \{P(0), P(s(0)), P(s(s(0))), P(s(s(s(0)))), \ldots\}$
  $M(S) = B(S)$ (minimální Herbrandův model je celá báze)
  poznámka: P vyjadřuje vlastnost ‚být korektní přirozené číslo' (pomocí následníků nuly)

Příklad 2: $S' = \{P(0), P(s(x)) \vee \neg P(x), R(x, s(x)) \vee \neg P(x)\}$

- $U(S') = \{0, s(0), s(s(0)), s(s(s(0))), \ldots\}$ (stejné jako pro S)
  $$B(S') = \{P(0), P(s(0)), P(s(s(0))), P(s(s(s(0)))), \ldots, \\ R(0,0), R(0, s(0)), R(s(0), 0), R(s(0), s(0)), \ldots\}$$
  $$M(S') = \{P(0), P(s(0)), P(s(s(0))), P(s(s(s(0)))), \ldots, \\ R(0, s(0)), R(s(0), s(s(0))), R(s(s(0)), s(s(s(0)))) \ldots\}$$
  poznámka: P je stejné jako pro S, $R(x, y)$ reprezentuje binární vlastnost ‚y je následníkem x' (resp. ‚x je předchůdcem y')

Unifikace – motivace

- směřujeme k rezoluci v predikátové logice:
  - formule umíme reprezentovat v konjunktivní pnf odpovídající klauzulární formě ($\forall$ nepíšeme, ale předpokládáme univerzální kvantifikaci všech proměnných)
  - literály nyní představují atomické formule a jejich negace
  - zůstává jediný problém: jak instanciovat proměnné, aby bylo možné použít rezoluční pravidlo
- příklad: mějme klauzule $C_1 = \{P(f(x)), \neg Q(a,x)\}$ a $C_2 = \{\neg P(f(g(a)))\}$; nahradíme-li x termem $g(a)$, získáme rezolventu $\{\neg Q(a, g(a))\}$
  poznámka: C_1 odpovídá formuli $\forall x(P(f(x)) \vee \neg Q(a,x))$, takže můžeme použít libovolnou instanci
- obecně řeší uvedený problém se substitucemi proměnných *unifikace*

Substituce

- *konečná substituce* ϕ je konečná množina $\{x_1/t_1, x_2/t_2, \ldots, x_n/t_n\}$, kde všechna x_i jsou vzájemně různé proměnné a každé t_i je term různý od x_i. Jsou-li všechna t_i uzavřené termy, jedná se o *uzavřenou substituci*. Pokud jsou t_i proměnné, označujeme ϕ jako *přejmenování proměnných*.
- označme libovolný term nebo literál jako *výraz* E; pak $E\phi$ je výsledek nahrazení všech výskytů všech x_i odpovídajícími termy t_i (obdobně pro množiny výrazů)
- poznámka: substituce proměnných probíhají paralelně, ne postupně
- příklad:
  $S = \{P(f(x, g(y))), \neg P(y, x), Q(y, z, a)\}$
  $\phi = \{x/h(y), y/g(z), z/c\}$
  $S\phi = \{P(f(h(y), g(g(z)))), \neg P(g(z), h(y)), Q(g(z), c, a)\}$

Kompozice substitucí

- *kompozice substitucí*
  $\phi = \{x_1/t_1, \ldots, x_n/t_n\}$ a $\psi = \{y_1/s_1, \ldots, y_m/s_m\}$ je množina
  $\phi\psi = \{x_1/t_1\psi, \ldots, x_n/t_n\psi, y_1/s_1, \ldots, y_m/s_m\}$ beze všech $x_i/t_i\psi$, pro která $x_i = t_i\psi$, a všech $y_j/s_j, y_j \in \{x_1, \ldots, x_n\}$
- pro *prázdnou substituci* ϵ a libovolnou substituci ϕ platí $\phi\epsilon = \epsilon\phi = \phi$
- pro libovolný výraz E a substituce ϕ, ψ, σ platí $(E\phi)\psi = E(\phi\psi)$ a $(\phi\psi)\sigma = \phi(\psi\sigma)$
- příklad:
  $S = \{f(x, g(y)), \neg P(y, x), Q(y, z, a)\}$
  $\phi = \{x/h(y), y/w, z/g(w, y)\}$, $\psi = \{x/a, y/f(b), w/y\}$
  $\phi\psi = \{x/h(f(b)), z/g(y, f(b)), w/y\}$
  $S\phi = \{f(h(y), g(w)), \neg P(w, h(y)), Q(w, g(w, y), a)\}$
  $S(\phi\psi) = (S\phi)\psi =$
  $\{f(h(f(b)), g(y)), \neg P(y, h(f(b))), Q(y, g(y, f(b)), a)\}$

Unifikace

- substituci ϕ nazveme *unifikátorem* množiny $S = \{E_1, \ldots, E_2\}$, pokud $E_1\phi = E_2\phi = \ldots = E_n\phi$, tedy v případě, že $S\phi$ má jediný prvek. Existuje-li unifikátor množiny, označíme ji jako *unifikovatelnou*.
- příklady:
  1. unifikátorem $\{P(x, c), P(b, c)\}$ je $\phi = \{x/b\}$; žádný jiný neexistuje
  2. unifikátorem $\{P(f(x), y), P(f(a), w)\}$ může být $\phi = \{x/a, y/w\}$, ale také $\psi = \{x/a, y/a, w/a\}$, $\sigma = \{x/a, y/b, w/b\}$ atd.
  3. $\{P(x, a), P(b, c)\}, \{P(f(x), z), P(a, w)\}, \{P(x, w), \neg P(a, w)\}, \{P(x, y, z), P(a, b)\}, \{R(x), P(x)\}$ nejsou unifikovatelné
- unifikátor ϕ množiny S je *nejobecnějším unifikátorem (mgu)* S, pokud pro libovolný unifikátor ψ množiny S existuje substituce λ taková, že $\phi\lambda = \psi$
- příklad: v bodu 2. předchozího příkladu je mgu ϕ, odpovídající substituce λ pro unifikátory ψ resp. σ je $\{w/a\}$ resp. $\{w/b\}$

Rozdíl mezi výrazy

- poznámka: pro unifikovatelné množiny existuje jediný mgu (až na přejmenování proměnných)
- mějme množinu S výrazů. Najdeme první (nejlevější) pozici, na které nemají všechny prvky S stejný symbol. *Rozdíl* $D(S)$ mezi výrazy pak definujeme jako množinu podvýrazů všech $E \in S$ začínajících na této pozici.
  Poznámka: každý unifikátor S nutně unifikuje i $D(S)$.
- příklady:
  - $S_1 = \{f(\mathbf{x}, g(x)), f(\mathbf{h}(y), g(h(z))\}$
    $D(S_1) = \{x, h(y)\}$
    $T_1 = S_1\{x/h(y)\} = \{f(h(y), g(h(\mathbf{y}))), f(h(y), g(h(\mathbf{z}))\}$
    $D(T_1) = \{y, z\}$
  - $S_2 = \{f(\mathbf{h}(x), g(x)), f(\mathbf{g}(x), f(x))\}$
    $D(S_2) = \{h(x), g(x)\}$

Unifikační algoritmus

unifikační algoritmus pro množinu výrazů S

- krok 0: $S_0 := S, \phi_0 := \epsilon$
- krok $k+1$:
  - je-li S_k jednoprvková, ukonči algoritmus s výsledkem $mgu(S) = \phi_0\phi_1\phi_2 \dots \phi_k$
  - jinak, pokud v $D(S_k)$ není proměnná v a term t, který ji neobsahuje, ukonči algoritmus s výsledkem neexistuje $mgu(S)$
  - jinak vyber takovou proměnnou v a term t, který neobsahuje v, $\phi_{k+1} := \{v/t\}$, $S_{k+1} := S_k\phi_{k+1}$ a pokračuj krokem $k+2$

Algoritmus skončí korektně pro libovolnou množinu výrazů S.

Unifikace – příklad

Najděte nejobecnější unifikátor množiny

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

1. $S_0 = S$ není jednoprvková; $D(S_0) = \{y, h(w), h(b)\}$, vyberme ze dvou možností $\phi_1 = \{y/h(w)\}$, $S_1 = S_0\phi_1 =$ $\{P(f(h(w), g(z)), h(b)), P(f(h(w), g(a)), x), P(f(h(b), g(z)), h(w))\}$
2. $D(S_1) = \{w, b\}, \phi_2 = \{w/b\}, S_2 = S_1\phi_2 =$ $\{P(f(h(b), g(z)), h(b)), P(f(h(b), g(a)), x), P(f(h(b), g(z)), h(b))\}$
3. $D(S_2) = \{z, a\}, \phi_3 = \{z/a\}, S_3 = S_2\phi_3 =$ $\{P(f(h(b), g(a)), h(b)), P(f(h(b), g(a)), x), P(f(h(b), g(a)), h(b))\}$
4. $D(S_3) = \{h(b), x\}, \phi_4 = \{x/h(b)\}, S_4 = S_3\phi_4 =$ $\{P(f(h(b), g(a)), h(b)), P(f(h(b), g(a)), h(b)), P(f(h(b), g(a)), h(b))\}$
5. $mgu(S) = \{y/h(w)\}\{w/b\}\{z/a\}\{x/h(b)\} =$ $\{y/h(b), w/b, z/a, x/h(b)\}$