

Desátá přednáška

NAIL062 Výroková a predikátová logika

Jakub Bulín (KTIML MFF UK)

Zimní semestr 2024

Program

- unifikace, unifikační algoritmus
- rezoluční pravidlo, rezoluční důkaz
- korektnost rezoluce
- lifting lemma a úplnost rezoluce

Materiály

Zápisky z přednášky, Sekce 8.4–8.6 z Kapitoly 8

8.4 Unifikace

Příklady substitucí

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

Příklady substitucí

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

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

Příklady substitucí

Místo **všech základních** použijeme 'vhodné' substitute (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 resolucí $\{P(b), \neg P(a)\}$

Příklady substitucí

Místo **všech základních** použijeme 'vhodné' substitute (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 resolucí $\{P(b), \neg P(a)\}$
- nebo $\{x/y\}$ a resolucí přes $P(y)$ máme $\{Q(y, a), \neg Q(b, y)\}$

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 resolucí $\{P(b), \neg P(a)\}$
- nebo $\{x/y\}$ a resolucí 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**

Příklady substitucí

Místo **všech základních** použijeme 'vhodné' substitute (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 $\{\neg P(y), \neg Q(f(y), y)\}$

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 $\{P(f(a)), \neg P(a)\}$

Příklady substitucí

Místo **všech základních** použijeme '**vhodné**' substitute (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 resolucí $\{P(b), \neg P(a)\}$
- nebo $\{x/y\}$ a resolucí 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 $\{\neg P(a), \neg Q(f(a), a)\}$, resolucí $\{P(f(a)), \neg P(a)\}$
- 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 $\{P(f(z)), \neg P(z)\}$

Příklady substitucí

Místo **všech základních** použijeme '**vhodné**' substitute (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 resolucí $\{P(b), \neg P(a)\}$
- nebo $\{x/y\}$ a resolucí 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 $\{\neg P(a), \neg Q(f(a), a)\}$, resolucí $\{P(f(a)), \neg P(a)\}$
- **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

Příklady substitucí

Místo **všech základních** použijeme '**vhodné**' substitute (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 resolucí $\{P(b), \neg P(a)\}$
- nebo $\{x/y\}$ a resolucí 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 $\{\neg P(a), \neg Q(f(a), a)\}$, resolucí $\{P(f(a)), \neg P(a)\}$
- **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

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

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

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)

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

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 $S\sigma = \{E\sigma \mid E \in S\}$

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

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

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

- substituce lze skládat, $\sigma\tau$ znamená nejprve σ a potom τ

Skládání substitucí

- substituce lze skládat, $\sigma\tau$ znamená nejprve σ a potom τ
- chceme, aby platilo $E(\sigma\tau) = (E\sigma)\tau$, pro libovolný výraz E

Skládání substitucí

- substituce 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 substituce

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

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 substitucí

- substituce 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 substituce

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

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

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

Skládání substitucí

- substituce 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 substituce

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

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 $\sigma_1\sigma_2\cdots\sigma_n$)

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

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

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

Vlastnosti skládání

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

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

Vlastnosti skládání

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

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

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):

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

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

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

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:

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:

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

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.

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** 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

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

- **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 $\tau = \sigma\lambda$

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

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

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

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 $\varrho = \{x/a, y/c, w/c\}$

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

Unifikační algoritmus

- postupně od začátku výrazů aplikuje substituce
- buď p nejlevější pozice, na které se nějaké dva výrazy z S liší

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

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
- $S = \{P(x, y), P(f(x), z), P(z, f(x))\}, p = 3, D(S) = \{x, f(x), z\}$

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
- $S = \{P(x, y), P(f(x), z), P(z, f(x))\}, p = 3, D(S) = \{x, f(x), z\}$

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

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
- $S = \{P(x, y), P(f(x), z), P(z, f(x))\}, p = 3, D(S) = \{x, f(x), z\}$

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

výstup: nejobecnější unifikace σ nebo info, že není unifikovatelná

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
- $S = \{P(x, y), P(f(x), z), P(z, f(x))\}, p = 3, D(S) = \{x, f(x), z\}$

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

výstup: nejobecnější unifikace σ nebo info, že není unifikovatelná

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

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
- $S = \{P(x, y), P(f(x), z), P(z, f(x))\}, p = 3, D(S) = \{x, f(x), z\}$

vstup: 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 $\sigma = \sigma_0 \sigma_1 \cdots \sigma_k$

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
- $S = \{P(x, y), P(f(x), z), P(z, f(x))\}, p = 3, D(S) = \{x, f(x), z\}$

vstup: 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

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
- $S = \{P(x, y), P(f(x), z), P(z, f(x))\}, p = 3, D(S) = \{x, f(x), z\}$

vstup: 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)

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
- $S = \{P(x, y), P(f(x), z), P(z, f(x))\}, p = 3, D(S) = \{x, f(x), z\}$

vstup: 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

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
- $S = \{P(x, y), P(f(x), z), P(z, f(x))\}, p = 3, D(S) = \{x, f(x), z\}$

vstup: 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

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

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 $S_1 = S_0\sigma_1$

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 $S_1 = S_0\sigma_1$

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

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 $S_1 = S_0\sigma_1$

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

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

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 $S_1 = S_0\sigma_1$

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

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

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

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 $S_1 = S_0\sigma_1$

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

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

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

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

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 $S_1 = S_0\sigma_1$

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

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

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

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

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

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 $S_1 = S_0\sigma_1$

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

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

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

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

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

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

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 $S_1 = S_0\sigma_1$

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

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

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

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

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

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

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

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 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 .

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':

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': Buď τ lib. unifikace pro S . Indukcí pro $0 \leq i \leq k$ ukážeme $\tau = \sigma_0\sigma_1 \cdots \sigma_i\tau$

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': Buď 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.

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: $u\sigma_{i+1}\tau = u\tau$

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 $\tau = \sigma_0\sigma_1 \cdots \sigma_i\sigma_{i+1}\tau$.

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 .

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**.

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:

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:

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

Příklad rezolučního kroku

Chceme-li ukázat $T \models \varphi$, skolemizací najdeme CNF formuli S ekvisplnitelnou 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.

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 $\sigma = \{x/f(f(z)), y/f(z), u/f(z)\}$

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 $\sigma = \{x/f(f(z)), y/f(z), u/f(z)\}$

- $C_1\sigma = \{P(f(f(z))), Q(f(f(z)), f(z))\}$

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 $\sigma = \{x/f(f(z)), y/f(z), u/f(z)\}$

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

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 $\sigma = \{x/f(f(z)), y/f(z), u/f(z)\}$

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

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

Rezoluční pravidlo

Mějme klauzule C_1 a C_2 s disjunktími množinami proměnných tvaru

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

Rezoluční pravidlo

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.

Rezoluční pravidlo

Mějme klauzule C_1 a C_2 s disjunktími 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 .

Rezoluční pravidlo

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

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

Rezoluční pravidlo

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

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

- Disjunktní množ. proměnných získáme přejmenováním. Proč? Z $\{\{P(x)\}, \{\neg P(f(x))\}\}$ odvodíme \square , nahradíme-li $\{P(x)\}$ klauzulí $\{P(y)\}$. Ale $S = \{P(x), P(f(x))\}$ není unifikovatelná.

Rezoluční pravidlo

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

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

- Disjunkt ní množ. proměnných získáme přejmenováním. Proč? Z $\{\{P(x)\}, \{\neg P(f(x))\}\}$ odvodíme \square , nahradíme-li $\{P(x)\}$ klauzulí $\{P(y)\}$. Ale $S = \{P(x), P(f(x))\}$ není unifikovatelná.
- Proč potřebujeme z klauzule odstranit více literálů najednou? $S = \{\{P(x), P(y)\}, \{\neg P(x), \neg P(y)\}\}$ je zamítnutelná, ale nemá zamítnutí, které by v každém kroku odstranilo jen jeden.

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

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

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 $k < i$.

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.

Příklad

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

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:

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:

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

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:

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

rezoluční strom:

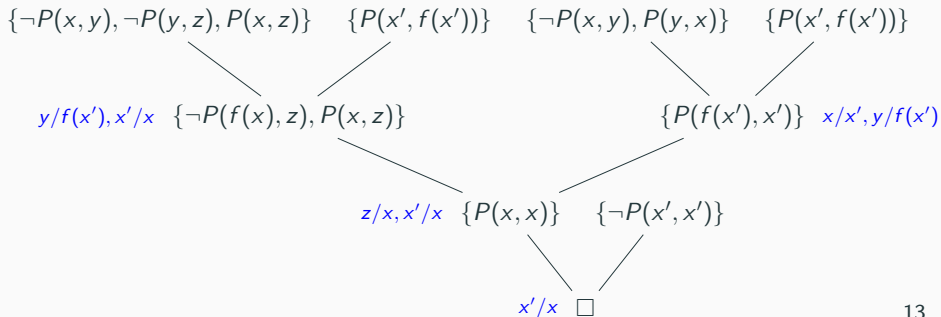
Příklad

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

rezoluční zamítnutí:

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

rezoluční strom:



8.6 Korektnost a úplnost

Korektnost rezolučního kroku

Tvrzení: Mějme klauzule C_1 , C_2 a jejich rezolventu C . Platí-li v nějaké struktuře \mathcal{A} klauzule C_1 a C_2 , potom v ní platí i C .

Korektnost rezolučního kroku

Tvrzení: Mějme klauzule C_1 , C_2 a jejich rezolventu C . Platí-li v nějaké struktuře \mathcal{A} klauzule C_1 a C_2 , potom v ní platí i C .

Důkaz: Bud' $C_1 = C'_1 \sqcup \{A_1, \dots, A_n\}$, $C_2 = C'_2 \sqcup \{\neg B_1, \dots, \neg B_m\}$, a $C = C'_1\sigma \cup C'_2\sigma$, kde $S\sigma = \{A_1\sigma\}$ (a σ je nejobecnější).

Korektnost rezolučního kroku

Tvrzení: Mějme klauzule C_1 , C_2 a jejich rezolventu C . Platí-li v nějaké struktuře \mathcal{A} klauzule C_1 a C_2 , potom v ní platí i C .

Důkaz: Buď $C_1 = C'_1 \sqcup \{A_1, \dots, A_n\}$, $C_2 = C'_2 \sqcup \{\neg B_1, \dots, \neg B_m\}$, a $C = C'_1\sigma \cup C'_2\sigma$, kde $S\sigma = \{A_1\sigma\}$ (a σ je nejobecnější). Klauzule jsou otevřené formule, proto platí i jejich instance:

$$\mathcal{A} \models C_1\sigma \quad \text{a} \quad \mathcal{A} \models C_2\sigma$$

Korektnost rezolučního kroku

Tvrzení: Mějme klauzule C_1 , C_2 a jejich rezolventu C . Platí-li v nějaké struktuře \mathcal{A} klauzule C_1 a C_2 , potom v ní platí i C .

Důkaz: Buď $C_1 = C'_1 \sqcup \{A_1, \dots, A_n\}$, $C_2 = C'_2 \sqcup \{\neg B_1, \dots, \neg B_m\}$, a $C = C'_1\sigma \cup C'_2\sigma$, kde $S\sigma = \{A_1\sigma\}$ (a σ je nejobecnější). Klauzule jsou otevřené formule, proto platí i jejich instance:

$$\mathcal{A} \models C_1\sigma \quad \text{a} \quad \mathcal{A} \models C_2\sigma$$

Po aplikaci unifikace máme:

Korektnost rezolučního kroku

Tvrzení: Mějme klauzule C_1 , C_2 a jejich rezolventu C . Platí-li v nějaké struktuře \mathcal{A} klauzule C_1 a C_2 , potom v ní platí i C .

Důkaz: Buď $C_1 = C'_1 \sqcup \{A_1, \dots, A_n\}$, $C_2 = C'_2 \sqcup \{\neg B_1, \dots, \neg B_m\}$, a $C = C'_1\sigma \cup C'_2\sigma$, kde $S\sigma = \{A_1\sigma\}$ (a σ je nejobecnější). Klauzule jsou otevřené formule, proto platí i jejich instance:

$$\mathcal{A} \models C_1\sigma \quad \text{a} \quad \mathcal{A} \models C_2\sigma$$

Po aplikaci unifikace máme:

$$C_1\sigma = C'_1\sigma \cup \{A_1\sigma\}$$

$$C_2\sigma = C'_2\sigma \cup \{\neg A_1\sigma\}$$

Korektnost rezolučního kroku

Tvrzení: Mějme klauzule C_1 , C_2 a jejich rezolventu C . Platí-li v nějaké struktuře \mathcal{A} klauzule C_1 a C_2 , potom v ní platí i C .

Důkaz: Buď $C_1 = C'_1 \sqcup \{A_1, \dots, A_n\}$, $C_2 = C'_2 \sqcup \{\neg B_1, \dots, \neg B_m\}$, a $C = C'_1\sigma \cup C'_2\sigma$, kde $S\sigma = \{A_1\sigma\}$ (a σ je nejobecnější). Klauzule jsou otevřené formule, proto platí i jejich instance:

$$\mathcal{A} \models C_1\sigma \quad \text{a} \quad \mathcal{A} \models C_2\sigma$$

Po aplikaci unifikace máme:

$$C_1\sigma = C'_1\sigma \cup \{A_1\sigma\}$$

$$C_2\sigma = C'_2\sigma \cup \{\neg A_1\sigma\}$$

Chceme ukázat, že $\mathcal{A} \models C[e]$ pro lib. ohodnocení e .

Korektnost rezolučního kroku

Tvrzení: Mějme klauzule C_1 , C_2 a jejich rezolventu C . Platí-li v nějaké struktuře \mathcal{A} klauzule C_1 a C_2 , potom v ní platí i C .

Důkaz: Buď $C_1 = C'_1 \sqcup \{A_1, \dots, A_n\}$, $C_2 = C'_2 \sqcup \{\neg B_1, \dots, \neg B_m\}$, a $C = C'_1\sigma \cup C'_2\sigma$, kde $S\sigma = \{A_1\sigma\}$ (a σ je nejobecnější). Klauzule jsou otevřené formule, proto platí i jejich instance:

$$\mathcal{A} \models C_1\sigma \quad \text{a} \quad \mathcal{A} \models C_2\sigma$$

Po aplikaci unifikace máme:

$$C_1\sigma = C'_1\sigma \cup \{A_1\sigma\}$$

$$C_2\sigma = C'_2\sigma \cup \{\neg A_1\sigma\}$$

Chceme ukázat, že $\mathcal{A} \models C[e]$ pro lib. ohodnocení e .

- Je-li $\mathcal{A} \models A_1\sigma[e]$, potom $\mathcal{A} \not\models \neg A_1\sigma[e]$ a musí $\mathcal{A} \models C'_2\sigma[e]$.
Tedy i $\mathcal{A} \models C[e]$.

Korektnost rezolučního kroku

Tvrzení: Mějme klauzule C_1 , C_2 a jejich rezolventu C . Platí-li v nějaké struktuře \mathcal{A} klauzule C_1 a C_2 , potom v ní platí i C .

Důkaz: Buď $C_1 = C'_1 \sqcup \{A_1, \dots, A_n\}$, $C_2 = C'_2 \sqcup \{\neg B_1, \dots, \neg B_m\}$, a $C = C'_1\sigma \cup C'_2\sigma$, kde $S\sigma = \{A_1\sigma\}$ (a σ je nejobecnější). Klauzule jsou otevřené formule, proto platí i jejich instance:

$$\mathcal{A} \models C_1\sigma \quad \text{a} \quad \mathcal{A} \models C_2\sigma$$

Po aplikaci unifikace máme:

$$C_1\sigma = C'_1\sigma \cup \{A_1\sigma\}$$

$$C_2\sigma = C'_2\sigma \cup \{\neg A_1\sigma\}$$

Chceme ukázat, že $\mathcal{A} \models C[e]$ pro lib. ohodnocení e .

- Je-li $\mathcal{A} \models A_1\sigma[e]$, potom $\mathcal{A} \not\models \neg A_1\sigma[e]$ a musí $\mathcal{A} \models C'_2\sigma[e]$.
Tedy i $\mathcal{A} \models C[e]$.
- Je-li $\mathcal{A} \not\models A_1\sigma[e]$, musí být $\mathcal{A} \models C'_1\sigma[e]$ a opět $\mathcal{A} \models C[e]$. \square

Věta (O korektnosti rezoluce): Pokud je CNF formule S rezolucí zamítnutelná, potom je nesplnitelná.

Věta (O korektnosti rezoluce): Pokud je CNF formule S rezolucí zamítnutelná, potom je nespílitelná.

Důkaz: Víme, že $S \vdash_R \square$, vezměme tedy nějaký rezoluční důkaz \square z S .

Věta (O korektnosti rezoluce): Pokud je CNF formule S rezolucí zamítnutelná, potom je nespílitelná.

Důkaz: Víme, že $S \vdash_R \square$, vezměme tedy nějaký rezoluční důkaz \square z S . Kdyby existoval model $\mathcal{A} \models S$, díky korektnosti rezolučního pravidla bychom dokázali (indukcí podle délky důkazu) i $\mathcal{A} \models \square$,

Věta (O korektnosti rezoluce): Pokud je CNF formule S rezolucí zamítnutelná, potom je nespílitelná.

Důkaz: Víme, že $S \vdash_R \square$, vezměme tedy nějaký rezoluční důkaz \square z S . Kdyby existoval model $\mathcal{A} \models S$, díky korektnosti rezolučního pravidla bychom dokázali (indukcí podle délky důkazu) i $\mathcal{A} \models \square$, což ale není možné. \square

úplnost rezoluce dokážeme převedením na případ výrokové logiky:
rezoluční důkaz 'na úrovni VL' je možné 'zvednout' na úroveň PL

úplnost rezoluce dokážeme převedením na případ výrokové logiky:
rezoluční důkaz 'na úrovni VL' je možné 'zvednout' na úroveň PL

Lifting lemma: Bud'te C_1 a C_2 klauzule s disj. množ. proměnných, C_1^* a C_2^* jejich základní instance, C^* rezolventa C_1^* a C_2^* . Potom C_1 a C_2 mají rezolventu C takovou, že C^* je základní instance C .

úplnost rezoluce dokážeme převedením na případ výrokové logiky:
rezoluční důkaz 'na úrovni VL' je možné 'zvednout' na úroveň PL

Lifting lemma: Bud'te C_1 a C_2 klauzule s disj. množ. proměnných, C_1^* a C_2^* jejich základní instance, C^* rezolventa C_1^* a C_2^* . Potom C_1 a C_2 mají rezolventu C takovou, že C^* je základní instance C .

(důkaz na příštím slidu)

úplnost rezoluce dokážeme převedením na případ výrokové logiky: rezoluční důkaz 'na úrovni VL' je možné 'zvednout' na úroveň PL

Lifting lemma: Bud' C_1 a C_2 klauzule s disj. množ. proměnných, C_1^* a C_2^* jejich základní instance, C^* rezolventa C_1^* a C_2^* . Potom C_1 a C_2 mají rezolventu C takovou, že C^* je základní instance C .

(důkaz na příštím slidu)

Důsledek: Bud' S CNF formule a označme S^* množinu všech jejích základních instancí. Pokud $S^* \vdash_R C^*$ pro nějakou základní klauzuli C^* ('na úrovni VL'), potom existuje klauzule C a základní substituce σ taková, že $C^* = C\sigma$ a $S \vdash_R C$ ('na úrovni PL').

Lifting lemma

úplnost rezoluce dokážeme převedením na případ výrokové logiky: rezoluční důkaz 'na úrovni VL' je možné 'zvednout' na úroveň PL

Lifting lemma: Bud' C_1 a C_2 klauzule s disj. množ. proměnných, C_1^* a C_2^* jejich základní instance, C^* rezolventa C_1^* a C_2^* . Potom C_1 a C_2 mají rezolventu C takovou, že C^* je základní instance C .

(důkaz na příštím slidu)

Důsledek: Bud' S CNF formule a označme S^* množinu všech jejích základních instancí. Pokud $S^* \vdash_R C^*$ pro nějakou základní klauzuli C^* ('na úrovni VL'), potom existuje klauzule C a základní substituce σ taková, že $C^* = C\sigma$ a $S \vdash_R C$ ('na úrovni PL').

Důkaz: Snadno z Lifting lemmatu indukcí dle délky důkazu. \square

Důkaz Lifting lemmatu

Nechť $C_1^* = C_1\tau_1$ a $C_2^* = C_2\tau_2$, τ_1 a τ_2 zákl. substituce nesdílející žádnou proměnnou. Najdeme rezolventu C , že $C^* = C\tau_1\tau_2$.

Důkaz Lifting lemmatu

Nechť $C_1^* = C_1\tau_1$ a $C_2^* = C_2\tau_2$, τ_1 a τ_2 zákl. substituce nesdílející žádnou proměnnou. Najdeme rezolventu C , že $C^* = C\tau_1\tau_2$.

Bud' C^* rezolventa C_1^* a C_2^* přes literál $P(t_1, \dots, t_k)$. Víme, že:

Důkaz Lifting lemmatu

Nechť $C_1^* = C_1\tau_1$ a $C_2^* = C_2\tau_2$, τ_1 a τ_2 zákl. substituce nesdílející žádnou proměnnou. Najdeme rezolventu C , že $C^* = C\tau_1\tau_2$.

Bud' C^* rezolventa C_1^* a C_2^* přes literál $P(t_1, \dots, t_k)$. Víme, že:

$$C_1 = C_1' \sqcup \{A_1, \dots, A_n\}, \text{ kde } \{A_1, \dots, A_n\}\tau_1 = \{P(t_1, \dots, t_k)\}$$

$$C_2 = C_2' \sqcup \{\neg B_1, \dots, \neg B_m\}, \{\neg B_1, \dots, \neg B_m\}\tau_2 = \{\neg P(t_1, \dots, t_k)\}$$

Důkaz Lifting lemmatu

Nechť $C_1^* = C_1\tau_1$ a $C_2^* = C_2\tau_2$, τ_1 a τ_2 zákl. substituce nesdílející žádnou proměnnou. Najdeme rezolventu C , že $C^* = C\tau_1\tau_2$.

Bud' C^* rezolventa C_1^* a C_2^* přes literál $P(t_1, \dots, t_k)$. Víme, že:

$$C_1 = C_1' \sqcup \{A_1, \dots, A_n\}, \text{ kde } \{A_1, \dots, A_n\}\tau_1 = \{P(t_1, \dots, t_k)\}$$

$$C_2 = C_2' \sqcup \{\neg B_1, \dots, \neg B_m\}, \{\neg B_1, \dots, \neg B_m\}\tau_2 = \{\neg P(t_1, \dots, t_k)\}$$

Tedy $(\tau_1\tau_2)$ unifikuje $S = \{A_1, \dots, A_n, B_1, \dots, B_m\}$. Bud' σ nejob. unifikace pro S z Unifikačního algoritmu. Zvolme $C = C_1'\sigma \cup C_2'\sigma$.

Důkaz Lifting lemmatu

Nechť $C_1^* = C_1\tau_1$ a $C_2^* = C_2\tau_2$, τ_1 a τ_2 zákl. substituce nesdílející žádnou proměnnou. Najdeme rezolventu C , že $C^* = C\tau_1\tau_2$.

Bud' C^* rezolventa C_1^* a C_2^* přes literál $P(t_1, \dots, t_k)$. Víme, že:

$$C_1 = C_1' \sqcup \{A_1, \dots, A_n\}, \text{ kde } \{A_1, \dots, A_n\}\tau_1 = \{P(t_1, \dots, t_k)\}$$

$$C_2 = C_2' \sqcup \{\neg B_1, \dots, \neg B_m\}, \{\neg B_1, \dots, \neg B_m\}\tau_2 = \{\neg P(t_1, \dots, t_k)\}$$

Tedy $(\tau_1\tau_2)$ unifikuje $S = \{A_1, \dots, A_n, B_1, \dots, B_m\}$. Bud' σ nejob. unifikace pro S z Unifikačního algoritmu. Zvolme $C = C_1'\sigma \cup C_2'\sigma$.

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

Důkaz Lifting lemmatu

Nechť $C_1^* = C_1\tau_1$ a $C_2^* = C_2\tau_2$, τ_1 a τ_2 zákl. substituce nesdílející žádnou proměnnou. Najdeme rezolventu C , že $C^* = C\tau_1\tau_2$.

Bud' C^* rezolventa C_1^* a C_2^* přes literál $P(t_1, \dots, t_k)$. Víme, že:

$$C_1 = C_1' \sqcup \{A_1, \dots, A_n\}, \text{ kde } \{A_1, \dots, A_n\}\tau_1 = \{P(t_1, \dots, t_k)\}$$

$$C_2 = C_2' \sqcup \{\neg B_1, \dots, \neg B_m\}, \{\neg B_1, \dots, \neg B_m\}\tau_2 = \{\neg P(t_1, \dots, t_k)\}$$

Tedy $(\tau_1\tau_2)$ unifikuje $S = \{A_1, \dots, A_n, B_1, \dots, B_m\}$. Bud' σ nejob. unifikace pro S z Unifikačního algoritmu. Zvolme $C = C_1'\sigma \cup C_2'\sigma$.

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

Zde $=$ plyne z vlastnosti 'navíc' Unif. algoritmu $(\tau_1\tau_2) = \sigma(\tau_1\tau_2)$,

a $=$ z toho, že jde o základní substituce nesdílející proměnnou. \square

Věta (O úplnosti rezoluce): Je-li CNF formule S nesplnitelná, potom je zamítnutelná rezolucí.

Věta (O úplnosti rezoluce): Je-li CNF formule S nespílitelná, potom je zamítnutelná rezolucí.

Důkaz:

Věta (O úplnosti rezoluce): Je-li CNF formule S nespílitelná, potom je zamítnutelná rezolucí.

Důkaz: Množina S^* všech základních instancí klauzulí z S je také nespílitelná (důsledek Herbrandovy věty).

Věta (O úplnosti rezoluce): Je-li CNF formule S nespínitelná, potom je zamítnutelná rezolucí.

Důkaz: Množina S^* všech základních instancí klauzulí z S je také nespínitelná (důsledek Herbrandovy věty). Úplnost **výrokové** rezoluce dává $S^* \vdash_R \square$ ('na úrovni VL').

Věta (O úplnosti rezoluce): Je-li CNF formule S nespílitelná, potom je zamítnutelná rezolucí.

Důkaz: Množina S^* všech základních instancí klauzulí z S je také nespílitelná (důsledek Herbrandovy věty). Úplnost **výrokové** rezoluce dává $S^* \vdash_R \square$ ('na úrovni VL').

Z důsledku Lifting lemmatu dostáváme klauzuli C a základní substituci σ takové, že $C\sigma = \square$ a $S \vdash_R C$ ('na úrovni PL').

Věta (O úplnosti rezoluce): Je-li CNF formule S nespílitelná, potom je zamítnutelná rezolucí.

Důkaz: Množina S^* všech základních instancí klauzulí z S je také nespílitelná (důsledek Herbrandovy věty). Úplnost **výrokové** rezoluce dává $S^* \vdash_R \square$ ('na úrovni VL').

Z důsledku Lifting lemmatu dostáváme klauzuli C a základní substituci σ takové, že $C\sigma = \square$ a $S \vdash_R C$ ('na úrovni PL').

Ale protože prázdná klauzule \square je instancí C , musí být $C = \square$.
Tím jsme našli rezoluční zamítnutí $S \vdash_R \square$. □