Klauzulární logika

Syntaxe klauzulí

Šárka Vavrečková

Ústav informatiky, Filozoficko-přírodovědecká fakulta Slezské univerzity v Opavě

sarka.vavreckova@fpf.slu.cz

19. listopadu 2007

Predikát rovnosti

- vrací true, jestliže jsou oba argumenty (po interpretaci) shodné,
- tvar
  1. postfixový: $=(argument1, argument2)$
  2. infixový: $argument1 = argument2$
- další podobné relační operátory: $<, >$, atd.

Predikát rovnosti

- vrací true, jestliže jsou oba argumenty (po interpretaci) shodné,
- tvar
  1. postfixový: $=(argument1, argument2)$
  2. infixový: $argument1 = argument2$
- další podobné relační operátory: $<, >$, atd.

Predikát rovnosti

- vrací true, jestliže jsou oba argumenty (po interpretaci) shodné,
- tvar
  1. postfixový: $=(argument1, argument2)$
  2. infixový: $argument1 = argument2$
- další podobné relační operátory: $<, >$, atd.

Příklady

1. Slepic je 25.
   $\rightarrow pocet(slepice, 25)$
   nebo $\rightarrow pocet(slepice) = 25$
2. Barva zralých jahod je červená.
   $jahoda(X), zraly(X) \rightarrow barva(X) = cervena$
   nebo $jahoda(X), zraly(X) \rightarrow barva(X, cervena)$
   nebo $X = jahoda, zraly(X) \rightarrow barva(X) = cervena$
3. Kuchařka potřebuje vařechu.
   $kucharka(X) \rightarrow potrebuje(X, varecha)$
   nebo $X = kucharka \rightarrow potrebuje(X, varecha)$
   ŠPATNĚ: $kucharka(X) \rightarrow potrebuje(X) = varecha$

Příklady

1. Slepic je 25.
   $\rightarrow pocet(slepice, 25)$
   nebo $\rightarrow pocet(slepice) = 25$
2. Barva zralých jahod je červená.
   $jahoda(X), zraly(X) \rightarrow barva(X) = cervena$
   nebo $jahoda(X), zraly(X) \rightarrow barva(X, cervena)$
   nebo $X = jahoda, zraly(X) \rightarrow barva(X) = cervena$
3. Kuchařka potřebuje vařechu.
   $kucharka(X) \rightarrow potrebuje(X, varecha)$
   nebo $X = kucharka \rightarrow potrebuje(X, varecha)$
   ŠPATNĚ: $kucharka(X) \rightarrow potrebuje(X) = varecha$

Příklady

1. Slepic je 25.
   $\rightarrow pocet(slepice, 25)$
   nebo $\rightarrow pocet(slepice) = 25$
2. Barva zralých jahod je červená.
   $jahoda(X), zraly(X) \rightarrow barva(X) = cervena$
   nebo $jahoda(X), zraly(X) \rightarrow barva(X, cervena)$
   nebo $X = jahoda, zraly(X) \rightarrow barva(X) = cervena$
3. Kuchařka potřebuje vařechu.
   $kucharka(X) \rightarrow potrebuje(X, varecha)$
   nebo $X = kucharka \rightarrow potrebuje(X, varecha)$
   ŠPATNĚ: $kucharka(X) \rightarrow potrebuje(X) = varecha$

Příklady

4. Pes má uši.
   $pes(X) \rightarrow ma(X, usi)$
   nebo $X = pes \rightarrow ma(X, usi)$
   ŠPATNĚ: $pes(X) \rightarrow ma(X) = usi$

5. Jestliže je nějaké číslo dvojnásobkem jiného celého čísla, pak je sudé.
   Predikátová logika:
   $\forall X((\exists N(cele(N) \,\&\, X = 2 \cdot N)) \rightarrow sude(X))$
   $\Leftrightarrow \forall X(\neg(\exists N(cele(N) \,\&\, X = 2 \cdot N)) \vee sude(X))$
   $\Leftrightarrow \forall X \forall N(\neg cele(N) \vee \neg(X = 2 \cdot N) \vee sude(X))$
   $\Leftrightarrow \forall X \forall N(true \rightarrow \neg cele(N) \vee \neg(X = 2 \cdot N) \vee sude(X))$
   Klauzule:
   $cele(X), X = 2 \cdot N \rightarrow sude(X)$

Příklady

4. Pes má uši.
   $pes(X) \rightarrow ma(X, usi)$
   nebo $X = pes \rightarrow ma(X, usi)$
   ŠPATNĚ: $pes(X) \rightarrow ma(X) = usi$
5. Jestliže je nějaké číslo dvojnásobkem jiného celého čísla, pak je sudé.
   Predikátová logika:
   $\forall X((\exists N(cele(N) \,\&\, X = 2 \cdot N)) \rightarrow sude(X))$
   $\Leftrightarrow \forall X(\neg(\exists N(cele(N) \,\&\, X = 2 \cdot N)) \vee sude(X))$
   $\Leftrightarrow \forall X \forall N(\neg cele(N) \vee \neg(X = 2 \cdot N) \vee sude(X))$
   $\Leftrightarrow \forall X \forall N(true \rightarrow \neg cele(N) \vee \neg(X = 2 \cdot N) \vee sude(X))$
   Klauzule:
   $cele(X), X = 2 \cdot N \rightarrow sude(X)$

Příklady

4. Pes má uši.

   $pes(X) \rightarrow ma(X, usi)$

   nebo $X = pes \rightarrow ma(X, usi)$

   ŠPATNĚ: $pes(X) \rightarrow ma(X) = usi$

5. Jestliže je nějaké číslo dvojnásobkem jiného celého čísla, pak je sudé.

   Predikátová logika:

   $\forall X((\exists N(cele(N) \,\&\, X = 2 \cdot N)) \rightarrow sude(X))$

   $\Leftrightarrow \forall X(\neg(\exists N(cele(N) \,\&\, X = 2 \cdot N)) \vee sude(X))$

   $\Leftrightarrow \forall X \forall N(\neg cele(N) \vee \neg(X = 2 \cdot N) \vee sude(X))$

   $\Leftrightarrow \forall X \forall N(true \rightarrow \neg cele(N) \vee \neg(X = 2 \cdot N) \vee sude(X))$

   Klauzule:

   $cele(X), X = 2 \cdot N \rightarrow sude(X)$

Příklady

4. Pes má uši.
   $pes(X) \rightarrow ma(X, usi)$
   nebo $X = pes \rightarrow ma(X, usi)$
   ŠPATNĚ: $pes(X) \rightarrow ma(X) = usi$

5. Jestliže je nějaké číslo dvojnásobkem jiného celého čísla, pak je sudé.
   Predikátová logika:
   $\forall X((\exists N(cele(N) \,\&\, X = 2 \cdot N)) \rightarrow sude(X))$
   $\Leftrightarrow \forall X(\neg(\exists N(cele(N) \,\&\, X = 2 \cdot N)) \vee sude(X))$
   $\Leftrightarrow \forall X \forall N(\neg cele(N) \vee \neg(X = 2 \cdot N) \vee sude(X))$
   $\Leftrightarrow \forall X \forall N(true \rightarrow \neg cele(N) \vee \neg(X = 2 \cdot N) \vee sude(X))$
   Klauzule:
   $cele(X), X = 2 \cdot N \rightarrow sude(X)$

Příklady

4. Pes má uši.
   $pes(X) \rightarrow ma(X, usi)$
   nebo $X = pes \rightarrow ma(X, usi)$
   ŠPATNĚ: $pes(X) \rightarrow ma(X) = usi$
5. Jestliže je nějaké číslo dvojnásobkem jiného celého čísla, pak je sudé.
   Predikátová logika:
   $\forall X((\exists N(cele(N) \& X = 2 \cdot N)) \rightarrow sude(X))$
   $\Leftrightarrow \forall X(\neg(\exists N(cele(N) \& X = 2 \cdot N)) \vee sude(X))$
   $\Leftrightarrow \forall X \forall N(\neg cele(N) \vee \neg(X = 2 \cdot N) \vee sude(X))$
   $\Leftrightarrow \forall X \forall N(true \rightarrow \neg cele(N) \vee \neg(X = 2 \cdot N) \vee sude(X))$
   Klauzule:
   $cele(X), X = 2 \cdot N \rightarrow sude(X)$

Příklady

4. Pes má uši.
   $pes(X) \rightarrow ma(X, usi)$
   nebo $X = pes \rightarrow ma(X, usi)$
   ŠPATNĚ: $pes(X) \rightarrow ma(X) = usi$
5. Jestliže je nějaké číslo dvojnásobkem jiného celého čísla, pak je sudé.
   Predikátová logika:
   $\forall X((\exists N(cele(N)\,\&\,X = 2 \cdot N)) \rightarrow sude(X))$
   $\Leftrightarrow \forall X(\neg(\exists N(cele(N)\,\&\,X = 2 \cdot N)) \vee sude(X))$
   $\Leftrightarrow \forall X \forall N(\neg cele(N) \vee \neg(X = 2 \cdot N) \vee sude(X))$
   $\Leftrightarrow \forall X \forall N(true \rightarrow \neg cele(N) \vee \neg(X = 2 \cdot N) \vee sude(X))$
   Klauzule:
   $cele(X), X = 2 \cdot N \rightarrow sude(X)$

(Běžná) substituce

Definice

Substituce termů $t_1, t_2, \ldots, t_n$ *za proměnné* $X_1, X_2, \ldots, X_n$ *do klauzule* $\mathcal{C}$*, kde každý term* t_i *je substituovatelný za proměnnou* X_i*,* $1 \leq i \leq n$*, je dána množinou* $\varphi = \{t_1/X_1, t_2/X_2, \ldots, t_n/X_n\}$*, značíme* $\mathcal{C}[\varphi]$*.*

Co lze dosadit

Za proměnnou dosazujeme

- funktor,
- proměnnou,
- existenční term.

Nahradíme všechny výskyty dané proměnné.

(Běžná) substituce

Definice

Substituce termů $t_1, t_2, \ldots, t_n$ *za proměnné* $X_1, X_2, \ldots, X_n$ *do klauzule* $\mathcal{C}$, *kde každý term* t_i *je substituovatelný za proměnnou* X_i, $1 \leq i \leq n$, *je dána množinou* $\varphi = \{t_1/X_1, t_2/X_2, \ldots, t_n/X_n\}$, *značíme* $\mathcal{C}[\varphi]$.

Co lze dosadit

Za proměnnou dosazujeme

- funktor,
- proměnnou,
- existenční term.

Nahradíme všechny výskyty dané proměnné.

(Běžná) substituce

Definice

Substituce termů $t_1, t_2, \dots, t_n$ *za proměnné* $X_1, X_2, \dots, X_n$ *do klauzule* $\mathcal{C}$*, kde každý term* t_i *je substituovatelný za proměnnou* X_i, $1 \le i \le n$*, je dána množinou* $\varphi = \{t_1/X_1, t_2/X_2, \dots, t_n/X_n\}$*, značíme* $\mathcal{C}[\varphi]$*.*

Co lze dosadit

Za proměnnou dosazujeme

- funktor,
- proměnnou,
- existenční term.

Nahradíme všechny výskyty dané proměnné.

(Běžná) substituce

Definice

Substituce termů $t_1, t_2, \ldots, t_n$ *za proměnné* $X_1, X_2, \ldots, X_n$ *do klauzule* $\mathcal{C}$*, kde každý term* t_i *je substituovatelný za proměnnou* X_i, $1 \leq i \leq n$, *je dána množinou* $\varphi = \{t_1/X_1, t_2/X_2, \ldots, t_n/X_n\}$, *značíme* $\mathcal{C}[\varphi]$.

Co lze dosadit

Za proměnnou dosazujeme

- funktor,
- proměnnou,
- existenční term.

Nahradíme všechny výskyty dané proměnné.

(Běžná) substituce

Definice

Substituce termů $t_1, t_2, \ldots, t_n$ za proměnné $X_1, X_2, \ldots, X_n$ do klauzule $\mathcal{C}$, kde každý term t_i je substituovatelný za proměnnou X_i, $1 \leq i \leq n$, je dána množinou $\varphi = \{t_1/X_1, t_2/X_2, \ldots, t_n/X_n\}$, značíme $\mathcal{C}[\varphi]$.

Co lze dosadit

Za proměnnou dosazujeme

- funktor,
- proměnnou,
- existenční term.

Nahradíme všechny výskyty dané proměnné.

Existenční substituce

Definice

Existenční substituce dosud nepoužitého existenčního termu $@m$ *za bázový term* t *se zapisuje* $\varphi = (@m/t;\ k_1, k_2, \ldots, k_r)$, *a znamená nahrazení* k_1*.,* k_2*., ... a* k_r*. výskytu termu* t *existenčním termem* $@m$ *v dané klauzuli.*

Co lze dosadit

Za bázový term (konstantu nebo funktor, jehož argumenty jsou bázové) dosazujeme

- novou existenční konstantu.

Můžeme si vybrat, které výskyty daného bázového termu nahradíme.

Když je univerzum diskurzu neprázdné, můžeme existenční substituci uplatnit i na proměnnou.

Existenční substituce

Definice

Existenční substituce dosud nepoužitého existenčního termu $@m$ *za bázový term* t *se zapisuje* $\varphi = (@m/t;\ k_1, k_2, \ldots, k_r)$, *a znamená nahrazení* k_1., k_2., ... *a* k_r. *výskytu termu* t *existenčním termem* $@m$ *v dané klauzuli.*

Co lze dosadit

Za bázový term (konstantu nebo funktor, jehož argumenty jsou bázové) dosazujeme

- novou existenční konstantu.

Můžeme si vybrat, které výskyty daného bázového termu nahradíme.

Když je univerzum diskurzu neprázdné, můžeme existenční substituci uplatnit i na proměnnou.

Existenční substituce

Definice

Existenční substituce dosud nepoužitého existenčního termu $@m$ *za bázový term* t *se zapisuje* $\varphi = (@m/t;\ k_1, k_2, \ldots, k_r)$, *a znamená nahrazení* k_1., k_2., ... *a* k_r. *výskytu termu* t *existenčním termem* $@m$ *v dané klauzuli.*

Co lze dosadit

Za bázový term (konstantu nebo funktor, jehož argumenty jsou bázové) dosazujeme

- novou existenční konstantu.

Můžeme si vybrat, které výskyty daného bázového termu nahradíme.

Když je univerzum diskurzu neprázdné, můžeme existenční substituci uplatnit i na proměnnou.

Příklady

$$\mathcal{C}_1 = p(a, X), q(f(X), b, Z) \to q(f(a), g(b, a), Y)$$

$\varphi_1 = \{a/X, g(a, Z)/Y, Z/Z\}$
$\varphi_2 = (@c/a; 1, 3)$

$$\mathcal{C}_1[\varphi_1] = p(a, a), q(f(a), b, Z) \to q(f(a), g(b, a), g(a, Z))$$

$$\mathcal{C}_1[\varphi_2] = p(@c, X), q(f(X), b, Z) \to q(f(a), g(b, @c), Y)$$

$$\mathcal{C}_1[\varphi_1][\varphi_2] = p(@c, a), q(f(@c), b, Z) \to q(f(a), g(b, a), g(a, Z))$$

Příklady

$$\mathcal{C}_1 \;=\; p(a, X), q(f(X), b, Z) \to q(f(a), g(b, a), Y)$$

$$\varphi_1 = \{a/X, g(a, Z)/Y, Z/Z\}$$
$$\varphi_2 = (@c/a; 1, 3)$$

$$\mathcal{C}_1[\varphi_1] \;=\; p(a, a), q(f(a), b, Z) \to q(f(a), g(b, a), g(a, Z))$$

$$\mathcal{C}_1[\varphi_2] \;=\; p(@c, X), q(f(X), b, Z) \to q(f(a), g(b, @c), Y)$$

$$\mathcal{C}_1[\varphi_1][\varphi_2] \;=\; p(@c, a), q(f(@c), b, Z) \to q(f(a), g(b, a), g(a, Z))$$

Příklady

$$\mathcal{C}_1 \;=\; p(a,X), q(f(X),b,Z) \to q(f(a),g(b,a),Y)$$

$$\varphi_1 = \{a/X, g(a,Z)/Y, Z/Z\}$$
$$\varphi_2 = (@c/a; 1,3)$$

$$\mathcal{C}_1[\varphi_1] \;=\; p(a,a), q(f(a),b,Z) \to q(f(a),g(b,a),g(a,Z))$$

$$\mathcal{C}_1[\varphi_2] \;=\; p(@c,X), q(f(X),b,Z) \to q(f(a),g(b,@c),Y)$$

$$\mathcal{C}_1[\varphi_1][\varphi_2] \;=\; p(@c,a), q(f(@c),b,Z) \to q(f(a),g(b,a),g(a,Z))$$

Příklady

$$\mathcal{C}_1 \;=\; p(a,X), q(f(X),b,Z) \rightarrow q(f(a),g(b,a),Y)$$

$$\varphi_1 = \{a/X, g(a,Z)/Y, Z/Z\}$$
$$\varphi_2 = (@c/a; 1,3)$$

$$\mathcal{C}_1[\varphi_1] \;=\; p(a,a), q(f(a),b,Z) \rightarrow q(f(a),g(b,a),g(a,Z))$$

$$\mathcal{C}_1[\varphi_2] \;=\; p(@c,X), q(f(X),b,Z) \rightarrow q(f(a),g(b,@c),Y)$$

$$\mathcal{C}_1[\varphi_1][\varphi_2] \;=\; p(@c,a), q(f(@c),b,Z) \rightarrow q(f(a),g(b,a),g(a,Z))$$

Příklady

$$\mathcal{C}_1 = p(a, X), q(f(X), b, Z) \rightarrow q(f(a), g(b, a), Y)$$

$$\varphi_1 = \{a/X, g(a, Z)/Y, Z/Z\}$$
$$\varphi_2 = (@c/a; 1, 3)$$

$$\mathcal{C}_1[\varphi_1] = p(a, a), q(f(a), b, Z) \rightarrow q(f(a), g(b, a), g(a, Z))$$

$$\mathcal{C}_1[\varphi_2] = p(@c, X), q(f(X), b, Z) \rightarrow q(f(a), g(b, @c), Y)$$

$$\mathcal{C}_1[\varphi_1][\varphi_2] = p(@c, a), q(f(@c), b, Z) \rightarrow q(f(a), g(b, a), g(a, Z))$$

Příklady

$\mathcal{C}_2 \;=\; dum(@f(Y)), vlastni(@f(Y), Y) \rightarrow ma(Y, bydliste)$
(Kdo vlastní nějaký dům, má kde bydlet.)

$\varphi_3 = \{X/X, pavel/Y\}$
$\varphi_4 = (@c/pavel; 4)$

$\mathcal{C}_2[\varphi_3] \;=\; dum(@f(pavel)), vlastni(@f(pavel), pavel) \rightarrow ma(pavel, bydliste)$
(Jestliže Pavel vlastní nějaký dům, má kde bydlet.)

$\mathcal{C}_2[\varphi_3][\varphi_4] \;=\; dum(@f(pavel)), vlastni(@f(pavel), pavel) \rightarrow ma(@c, bydliste)$
(Pokud Pavel vlastní nějaký dům, někdo má kde bydlet.)

Příklady

$\mathcal{C}_2 \;=\; dum(@f(Y)), vlastni(@f(Y), Y) \rightarrow ma(Y, bydliste)$
(Kdo vlastní nějaký dům, má kde bydlet.)

$\varphi_3 = \{X/X, pavel/Y\}$
$\varphi_4 = (@c/pavel; 4)$

$\mathcal{C}_2[\varphi_3] \;=\; dum(@f(pavel)), vlastni(@f(pavel), pavel) \rightarrow ma(pavel, bydliste)$
(Jestliže Pavel vlastní nějaký dům, má kde bydlet.)

$\mathcal{C}_2[\varphi_3][\varphi_4] \;=\; dum(@f(pavel)), vlastni(@f(pavel), pavel) \rightarrow ma(@c, bydliste)$
(Pokud Pavel vlastní nějaký dům, někdo má kde bydlet.)

Příklady

$\mathcal{C}_2 \;=\; dum(@f(Y)), vlastni(@f(Y), Y) \rightarrow ma(Y, bydliste)$
(Kdo vlastní nějaký dům, má kde bydlet.)

$\varphi_3 = \{X/X, pavel/Y\}$
$\varphi_4 = (@c/pavel; 4)$

$\mathcal{C}_2[\varphi_3] \;=\; dum(@f(pavel)), vlastni(@f(pavel), pavel) \rightarrow ma(pavel, bydliste)$
(Jestliže Pavel vlastní nějaký dům, má kde bydlet.)

$\mathcal{C}_2[\varphi_3][\varphi_4] \;=\; dum(@f(pavel)), vlastni(@f(pavel), pavel) \rightarrow ma(@c, bydliste)$
(Pokud Pavel vlastní nějaký dům, někdo má kde bydlet.)

Příklady

$\mathcal{C}_2 \;=\; dum(@f(Y)), vlastni(@f(Y), Y) \rightarrow ma(Y, bydliste)$
(Kdo vlastní nějaký dům, má kde bydlet.)

$\varphi_3 = \{X/X, pavel/Y\}$
$\varphi_4 = (@c/pavel; 4)$

$\mathcal{C}_2[\varphi_3] \;=\; dum(@f(pavel)), vlastni(@f(pavel), pavel) \rightarrow ma(pavel, bydliste)$
(Jestliže Pavel vlastní nějaký dům, má kde bydlet.)

$\mathcal{C}_2[\varphi_3][\varphi_4] \;=\; dum(@f(pavel)), vlastni(@f(pavel), pavel) \rightarrow ma(@c, bydliste)$
(Pokud Pavel vlastní nějaký dům, někdo má kde bydlet.)

Příklady

$\mathcal{C}_3 = \text{clovek}(X) \rightarrow \text{umi}(X, zpivat), \text{hraje\_na}(X, @nastroj(X))$
(Každý člověk umí zpívat nebo hrát na nějaký hudební nástroj.)

$\varphi_5 = \{ucitel(X, hudebni\_v)/X\}$
$\varphi_6 = \{honza/X\}$
$\varphi_7 = (@c/ucitel(honza, hudebni\_v); 1 - 4)$

Příklady

$\mathcal{C}_3 \;=\; \mathsf{clovek}(X) \rightarrow \mathsf{umi}(X, zpivat), \mathsf{hraje\_na}(X, @nastroj(X))$
(Každý člověk umí zpívat nebo hrát na nějaký hudební nástroj.)

$\varphi_5 = \{ucitel(X, hudebni\_v)/X\}$
$\varphi_6 = \{honza/X\}$
$\varphi_7 = (@c/ucitel(honza, hudebni\_v); 1-4)$

$\mathcal{C}_3[\varphi_5] \;=\; \mathsf{clovek}(ucitel(X, hudebni\_v)) \rightarrow$
$\mathsf{umi}(ucitel(X, hudebni\_v), zpivat),$
$\mathsf{hraje\_na}(ucitel(X, hudebni\_v),$
$@nastroj(ucitel(X, hudebni\_v)))$
(Pro každého člověka, který je něčí učitel v předmětu Hudební výchova, platí, že buď umí zpívat nebo hraje na nějaký hudební nástroj.)

Příklady

$\mathcal{C}_3 = \text{clovek}(X) \rightarrow \text{umi}(X, zpivat), \text{hraje\_na}(X, @nastroj(X))$
(Každý člověk umí zpívat nebo hrát na nějaký hudební nástroj.)

$\varphi_5 = \{ucitel(X, hudebni\_v)/X\}$
$\varphi_6 = \{honza/X\}$
$\varphi_7 = (@c/ucitel(honza, hudebni\_v); 1 - 4)$

$$\begin{aligned} \mathcal{C}_3[\varphi_5][\varphi_6] = {} & \text{clovek}(ucitel(honza, hudebni\_v)) \\ & \rightarrow \text{umi}(ucitel(honza, hudebni\_v), zpivat), \\ & \text{hraje\_na}(ucitel(honza, hudebni\_v), \\ & @nastroj(ucitel(honza, hudebni\_v))) \end{aligned}$$

(Každý člověk, kdo je Honzův učitel v předmětu Hudební výchova, buď umí zpívat nebo hraje na nějaký hudební nástroj.)

Příklady

$\mathcal{C}_3 \ = \ \text{clovek}(X) \rightarrow \text{umi}(X, zpivat), \text{hraje\_na}(X, @nastroj(X))$
(Každý člověk umí zpívat nebo hrát na nějaký hudební nástroj.)

$\varphi_5 = \{ucitel(X, hudebni\_v)/X\}$
$\varphi_6 = \{honza/X\}$
$\varphi_7 = (@c/ucitel(honza, hudebni\_v); 1 - 4)$

$\mathcal{C}_3[\varphi_5][\varphi_6][\varphi_7] \ =$
$= \ \text{clovek}(@c) \rightarrow \text{umi}(@c, zpivat), \text{hraje\_na}(@c, @nastroj(@c))$
(Existuje člověk, který umí zpívat nebo hraje na nějaký hudební nástroj.)

Definice (Rezoluční odvozovací pravidlo)

$$A_1, p \to K_1, \quad A_2 \to p, K_2 \quad \vdash \quad A_1, A_2 \to K_1, K_2$$

Rezoluční odvozovací pravidlo

Společný atom p „odřízneme“, proto se toto pravidlo také nazývá *odvozovací pravidlo rezolučního řezu*.

Definice (Rezoluční odvozovací pravidlo)

$$A_1, p \to K_1, \quad A_2 \to p, K_2 \quad \vdash \quad A_1, A_2 \to K_1, K_2$$

Rezoluční odvozovací pravidlo

Společný atom p „odřízneme“, proto se toto pravidlo také nazývá *odvozovací pravidlo rezolučního řezu*.

Důkaz (z predikátové logiky):

$$\begin{array}{ccccc}
C \vee \neg p,\ \ p \vee D & \models & C & \vee & D \\
(\neg A_1 \vee K_1) \vee \neg p,\ \ p \vee (\neg A_2 \vee K_2) & \models & (\neg A_1 \vee K_1) & \vee & (\neg A_2 \vee K_2) \\
\neg A_1 \vee \neg p \vee K_1,\ \ \neg A_2 \vee p \vee K_2 & \models & \neg A_1 \vee \neg A_2 & \vee & K_1 \vee K_2 \\
A_1 \,\&\, p \rightarrow K_1,\ \ A_2 \rightarrow p \vee K_2 & \models & A_1 \,\&\, A_2 & \rightarrow & K_1 \vee K_2
\end{array}$$

Příklad

Vyvoďte závěr *Monika je matkou Honzy.*

1. $otec(pavel, honza), manzele(pavel, monika) \rightarrow matka(monika, honza)$
2. $\rightarrow otec(pavel, honza)$
3. $\rightarrow manzele(pavel, monika)$
4. $manzele(pavel, monika) \rightarrow matka(monika, honza)$ R(1,2)
5. $\rightarrow matka(monika, honza)$ R(3,4)

Příklad

Vyvoďte závěr *Monika je matkou Honzy.*

1. $otec(pavel, honza), manzele(pavel, monika) \rightarrow matka(monika, honza)$
2. $\rightarrow otec(pavel, honza)$
3. $\rightarrow manzele(pavel, monika)$
4. $manzele(pavel, monika) \rightarrow matka(monika, honza)$ R(1,2)
5. $\rightarrow matka(monika, honza)$ R(3,4)

Příklad

Vyvoďte závěr *Monika je matkou Honzy.*

1. $otec(pavel, honza), manzele(pavel, monika) \rightarrow matka(monika, honza)$
2. $\rightarrow otec(pavel, honza)$
3. $\rightarrow manzele(pavel, monika)$

4. $manzele(pavel, monika) \rightarrow matka(monika, honza)$ R(1,2)
5. $\rightarrow matka(monika, honza)$ R(3,4)

Postup uplatnění rezolučního pravidla

1. pokud je to nutné, přejmenujeme v jedné klauzuli proměnné a existenční konstanty, které mají v obou klauzulích stejné názvy,
2. najdeme *vhodnou substituci*, kterou lze použít na obě klauzule,
3. aplikujeme tuto substituci,
4. provedeme rezoluční odvození.

Postup uplatnění rezolučního pravidla

1. pokud je to nutné, přejmenujeme v jedné klauzuli proměnné a existenční konstanty, které mají v obou klauzulích stejné názvy,
2. najdeme *vhodnou substituci*, kterou lze použít na obě klauzule,
3. aplikujeme tuto substituci,
4. provedeme rezoluční odvození.

Postup uplatnění rezolučního pravidla

1. pokud je to nutné, přejmenujeme v jedné klauzuli proměnné a existenční konstanty, které mají v obou klauzulích stejné názvy,
2. najdeme *vhodnou substituci*, kterou lze použít na obě klauzule,
3. aplikujeme tuto substituci,
4. provedeme rezoluční odvození.

Postup uplatnění rezolučního pravidla

1. pokud je to nutné, přejmenujeme v jedné klauzuli proměnné a existenční konstanty, které mají v obou klauzulích stejné názvy,
2. najdeme *vhodnou substituci*, kterou lze použít na obě klauzule,
3. aplikujeme tuto substituci,
4. provedeme rezoluční odvození.

Unifikátor dvojice klauzulí

Unifikátor dvojice klauzulí je substituce splňující tyto podmínky:

- je aplikovatelná na obě klauzule,
- po provedení substituce na obě klauzule je možné provést rezoluční odvození z těchto klauzulí.

Obecnost unifikátorů

Unifikátor φ je obecnější než unifikátor σ, pokud σ lze dostat tak, že na klauzuli po aplikaci φ uplatníme ještě nějakou další substituci – unifikaci (tj. čím méně konstant – konkrétních hodnot, tím obecnější).
Účelem je navrhnout co nejobecnější unifikátor.

Unifikátor dvojice klauzulí

Unifikátor dvojice klauzulí je substituce splňující tyto podmínky:

- je aplikovatelná na obě klauzule,
- po provedení substituce na obě klauzule je možné provést rezoluční odvození z těchto klauzulí.

Obecnost unifikátorů

Unifikátor φ je obecnější než unifikátor σ, pokud σ lze dostat tak, že na klauzuli po aplikaci φ uplatníme ještě nějakou další substituci – unifikaci (tj. čím méně konstant – konkrétních hodnot, tím obecnější).
Účelem je navrhnout co nejobecnější unifikátor.

Definice

Nechť σ a φ jsou unifikátory dvojice klauzulí.
φ je obecnější než σ, jestliže existuje substituce λ taková, že platí

$$\varphi \circ \lambda = \sigma,$$

tedy je možno σ odvodit z φ.

Definice

φ je nejobecnější unifikátor, jestliže pro každý unifikátor γ existuje substituce λ taková, že platí

$$\varphi \circ \lambda = \gamma$$

– jakýkoliv unifikátor pro stejnou dvojici klauzulí je možno odvodit z φ.

Definice

Nechť σ a φ jsou unifikátory dvojice klauzulí.
φ je obecnější než σ, jestliže existuje substituce λ taková, že platí

$$\varphi \circ \lambda = \sigma,$$

tedy je možno σ odvodit z φ.

Definice

φ je nejobecnější unifikátor, jestliže pro každý unifikátor γ existuje substituce λ taková, že platí

$$\varphi \circ \lambda = \gamma$$

– jakýkoliv unifikátor pro stejnou dvojici klauzulí je možno odvodit z φ.

Algoritmus zjištění nejobecnějšího unifikátoru

1. Vytvoříme *množinu neshod*
   Z obou klauzulí, které chceme unifikovat, vezmeme atomy, přes které má být provedeno rezoluční odvození:

$$p(t_1, t_2, \ldots, t_n),\ p(r_1, r_2, \ldots, r_n)$$

   a vytvoříme z nich rovnice:
   $t_1 = r_1$
   $t_2 = r_2$
   $\ldots$
   $t_n = r_n$

   Množina neshod je množinou rovnic vytvořených podle odpovídajících si argumentů unifikovaných atomů.

Algoritmus zjištění nejobecnějšího unifikátoru

1. Vytvoříme *množinu neshod*
   Z obou klauzulí, které chceme unifikovat, vezmeme atomy, přes které má být provedeno rezoluční odvození:

$$p(t_1, t_2, \ldots, t_n),\; p(r_1, r_2, \ldots, r_n)$$

a vytvoříme z nich rovnice:

$t_1 = r_1$

$t_2 = r_2$

$\ldots$

$t_n = r_n$

Množina neshod je množinou rovnic vytvořených podle odpovídajících si argumentů unifikovaných atomů.

Algoritmus zjištění nejobecnějšího unifikátoru

1. Vytvoříme *množinu neshod*
   Z obou klauzulí, které chceme unifikovat, vezmeme atomy, přes které má být provedeno rezoluční odvození:

   $$p(t_1,t_2,\ldots,t_n),\ p(r_1,r_2,\ldots,r_n)$$

   a vytvoříme z nich rovnice:

   $t_1 = r_1$

   $t_2 = r_2$

   $\ldots$

   $t_n = r_n$

   Množina neshod je množinou rovnic vytvořených podle odpovídajících si argumentů unifikovaných atomů.

Algoritmus zjištění nejobecnějšího unifikátoru

1. Vytvoříme *množinu neshod*
   Z obou klauzulí, které chceme unifikovat, vezmeme atomy, přes které má být provedeno rezoluční odvození:

$$p(t_1, t_2, \ldots, t_n), \; p(r_1, r_2, \ldots, r_n)$$

   a vytvoříme z nich rovnice:

   $t_1 = r_1$

   $t_2 = r_2$

   $\ldots$

   $t_n = r_n$

   Množina neshod je množinou rovnic vytvořených podle odpovídajících si argumentů unifikovaných atomů.

Algoritmus zjištění nejobecnějšího unifikátoru

1. Vytvoříme *množinu neshod*

   Z obou klauzulí, které chceme unifikovat, vezmeme atomy, přes které má být provedeno rezoluční odvození:

   $$p(t_1, t_2, \ldots, t_n),\ p(r_1, r_2, \ldots, r_n)$$

   a vytvoříme z nich rovnice:

   $t_1 = r_1$

   $t_2 = r_2$

   $\ldots$

   $t_n = r_n$

   Množina neshod je množinou rovnic vytvořených podle odpovídajících si argumentů unifikovaných atomů.

Algoritmus zjištění nejobecnějšího unifikátoru

2. Následující body provádíme tak dlouho, dokud rovnice nejsou ve tvaru *Proměnná=výraz*, kde
   - *Proměnná* se nevyskytuje na pravé straně žádné rovnice množiny,
   - pro každou *Proměnnou* je zde nejvýše jeden *výraz*.

Nejobecnější unifikátor je pak

$$\varphi = \{vyraz_1/Prom_1, vyraz_2/Prom_2, \dots, vyraz_k/Prom_k\}.$$

Algoritmus zjištění nejobecnějšího unifikátoru

2. Následující body provádíme tak dlouho, dokud rovnice nejsou ve tvaru *Proměnná*=*výraz*, kde
   - *Proměnná* se nevyskytuje na pravé straně žádné rovnice množiny,
   - pro každou *Proměnnou* je zde nejvýše jeden *výraz*.

Nejobecnější unifikátor je pak

$$\varphi = \{vyraz_1/Prom_1, vyraz_2/Prom_2, \ldots, vyraz_k/Prom_k\}.$$

Algoritmus zjištění nejobecnějšího unifikátoru

Opakujeme následující kroky:

- Vyřadíme rovnice, které mají stejnou levou a pravou stranu (např. $X = X$ nebo $f(a, Y) = f(a, Y)$).
- Rovnice ve tvaru *výraz=Proměnná* nahradíme rovnicemi *Proměnná=výraz* (přehodíme Proměnnou doleva).
- Rovnice ve tvaru $t_1 = t_2$, kde t_1 a t_2 nejsou proměnné:
  1. jsou to různé funkční symboly (např. $f(X) = g(Y)$) $\Rightarrow$ KONEC, množina neshod nemá řešení.
  2. jinak: tuto rovnici nahradíme množinou neshod pro tyto termy. Např. $p(X, a) = p(f(Y, b), Y)$ nahradíme $X = f(Y, b)$, $a = Y$.
- Rovnice ve tvaru *Proměnná=t*, *t* je jakýkoliv term:
  1. jestliže se *Proměnná* vyskytuje v t (např. jako parametr funktoru) $\Rightarrow$ KONEC, množina neshod nemá řešení.
  2. jinak všechny výskyty *Proměnné* v ostatních rovnicích nahradíme termem t, samotnou rovnici neměníme.

Algoritmus zjištění nejobecnějšího unifikátoru

Opakujeme následující kroky:

- Vyřadíme rovnice, které mají stejnou levou a pravou stranu (např. $X = X$ nebo $f(a, Y) = f(a, Y)$).
- Rovnice ve tvaru *výraz=Proměnná* nahradíme rovnicemi *Proměnná=výraz* (přehodíme Proměnnou doleva).
- Rovnice ve tvaru $t_1 = t_2$, kde t_1 a t_2 nejsou proměnné:
  1. jsou to různé funkční symboly (např. $f(X) = g(Y)$) $\Rightarrow$ KONEC, množina neshod nemá řešení.
  2. jinak: tuto rovnici nahradíme množinou neshod pro tyto termy. Např. $p(X, a) = p(f(Y, b), Y)$ nahradíme $X = f(Y, b)$, $a = Y$.
- Rovnice ve tvaru *Proměnná=t*, *t* je jakýkoliv term:
  1. jestliže se *Proměnná* vyskytuje v *t* (např. jako parametr funktoru) $\Rightarrow$ KONEC, množina neshod nemá řešení.
  2. jinak všechny výskyty *Proměnné* v ostatních rovnicích nahradíme termem *t*, samotnou rovnici neměníme.

Algoritmus zjištění nejobecnějšího unifikátoru

Opakujeme následující kroky:

- Vyřadíme rovnice, které mají stejnou levou a pravou stranu (např. $X = X$ nebo $f(a, Y) = f(a, Y)$).
- Rovnice ve tvaru *výraz=Proměnná* nahradíme rovnicemi *Proměnná=výraz* (přehodíme Proměnnou doleva).
- Rovnice ve tvaru $t_1 = t_2$, kde t_1 a t_2 nejsou proměnné:
  1. jsou to různé funkční symboly (např. $f(X) = g(Y)$) $\Rightarrow$ KONEC, množina neshod nemá řešení.
  2. jinak: tuto rovnici nahradíme množinou neshod pro tyto termy. Např. $p(X, a) = p(f(Y, b), Y)$ nahradíme $X = f(Y, b)$, $a = Y$.
- Rovnice ve tvaru *Proměnná=t*, *t* je jakýkoliv term:
  1. jestliže se *Proměnná* vyskytuje v *t* (např. jako parametr funktoru) $\Rightarrow$ KONEC, množina neshod nemá řešení.
  2. jinak všechny výskyty *Proměnné* v ostatních rovnicích nahradíme termem *t*, samotnou rovnici neměníme.

Algoritmus zjištění nejobecnějšího unifikátoru

Opakujeme následující kroky:

- Vyřadíme rovnice, které mají stejnou levou a pravou stranu (např. $X = X$ nebo $f(a, Y) = f(a, Y)$).
- Rovnice ve tvaru *výraz=Proměnná* nahradíme rovnicemi *Proměnná=výraz* (přehodíme Proměnnou doleva).
- Rovnice ve tvaru $t_1 = t_2$, kde t_1 a t_2 nejsou proměnné:
  1. jsou to různé funkční symboly (např. $f(X) = g(Y)$) $\Rightarrow$ KONEC, množina neshod nemá řešení.
  2. jinak: tuto rovnici nahradíme množinou neshod pro tyto termy. Např. $p(X, a) = p(f(Y, b), Y)$ nahradíme $X = f(Y, b)$, $a = Y$.
- Rovnice ve tvaru *Proměnná=t*, *t* je jakýkoliv term:
  1. jestliže se *Proměnná* vyskytuje v *t* (např. jako parametr funktoru) $\Rightarrow$ KONEC, množina neshod nemá řešení.
  2. jinak všechny výskyty *Proměnné* v ostatních rovnicích nahradíme termem *t*, samotnou rovnici neměníme.

Algoritmus zjištění nejobecnějšího unifikátoru

Opakujeme následující kroky:

- Vyřadíme rovnice, které mají stejnou levou a pravou stranu (např. $X = X$ nebo $f(a, Y) = f(a, Y)$).
- Rovnice ve tvaru *výraz=Proměnná* nahradíme rovnicemi *Proměnná=výraz* (přehodíme Proměnnou doleva).
- Rovnice ve tvaru $t_1 = t_2$, kde t_1 a t_2 nejsou proměnné:
  1. jsou to různé funkční symboly (např. $f(X) = g(Y)$) $\Rightarrow$ KONEC, množina neshod nemá řešení.
  2. jinak: tuto rovnici nahradíme množinou neshod pro tyto termy. Např. $p(X, a) = p(f(Y, b), Y)$ nahradíme $X = f(Y, b),\ a = Y$.
- Rovnice ve tvaru *Proměnná=t*, *t* je jakýkoliv term:
  1. jestliže se *Proměnná* vyskytuje v *t* (např. jako parametr funktoru) $\Rightarrow$ KONEC, množina neshod nemá řešení.
  2. jinak všechny výskyty *Proměnné* v ostatních rovnicích nahradíme termem *t*, samotnou rovnici neměníme.

Algoritmus zjištění nejobecnějšího unifikátoru

Opakujeme následující kroky:

- Vyřadíme rovnice, které mají stejnou levou a pravou stranu (např. $X = X$ nebo $f(a, Y) = f(a, Y)$).
- Rovnice ve tvaru *výraz=Proměnná* nahradíme rovnicemi *Proměnná=výraz* (přehodíme Proměnnou doleva).
- Rovnice ve tvaru $t_1 = t_2$, kde t_1 a t_2 nejsou proměnné:
  1. jsou to různé funkční symboly (např. $f(X) = g(Y)$) $\Rightarrow$ KONEC, množina neshod nemá řešení.
  2. jinak: tuto rovnici nahradíme množinou neshod pro tyto termy. Např. $p(X, a) = p(f(Y, b), Y)$ nahradíme $X = f(Y, b),\ a = Y$.
- Rovnice ve tvaru *Proměnná=t*, *t* je jakýkoliv term:
  1. jestliže se *Proměnná* vyskytuje v *t* (např. jako parametr funktoru) $\Rightarrow$ KONEC, množina neshod nemá řešení.
  2. jinak všechny výskyty *Proměnné* v ostatních rovnicích nahradíme termem *t*, samotnou rovnici neměníme.

Algoritmus zjištění nejobecnějšího unifikátoru

Opakujeme následující kroky:

- Vyřadíme rovnice, které mají stejnou levou a pravou stranu (např. $X = X$ nebo $f(a, Y) = f(a, Y)$).
- Rovnice ve tvaru *výraz=Proměnná* nahradíme rovnicemi *Proměnná=výraz* (přehodíme Proměnnou doleva).
- Rovnice ve tvaru $t_1 = t_2$, kde t_1 a t_2 nejsou proměnné:
  1. jsou to různé funkční symboly (např. $f(X) = g(Y)$) $\Rightarrow$ KONEC, množina neshod nemá řešení.
  2. jinak: tuto rovnici nahradíme množinou neshod pro tyto termy. Např. $p(X, a) = p(f(Y, b), Y)$ nahradíme $X = f(Y, b)$, $a = Y$.
- Rovnice ve tvaru *Proměnná=t*, *t* je jakýkoliv term:
  1. jestliže se *Proměnná* vyskytuje v *t* (např. jako parametr funktoru) $\Rightarrow$ KONEC, množina neshod nemá řešení.
  2. jinak všechny výskyty *Proměnné* v ostatních rovnicích nahradíme termem *t*, samotnou rovnici neměníme.

Algoritmus zjištění nejobecnějšího unifikátoru

Opakujeme následující kroky:

- Vyřadíme rovnice, které mají stejnou levou a pravou stranu (např. $X = X$ nebo $f(a, Y) = f(a, Y)$).
- Rovnice ve tvaru *výraz=Proměnná* nahradíme rovnicemi *Proměnná=výraz* (přehodíme Proměnnou doleva).
- Rovnice ve tvaru $t_1 = t_2$, kde t_1 a t_2 nejsou proměnné:
  1. jsou to různé funkční symboly (např. $f(X) = g(Y)$) $\Rightarrow$ KONEC, množina neshod nemá řešení.
  2. jinak: tuto rovnici nahradíme množinou neshod pro tyto termy. Např. $p(X, a) = p(f(Y, b), Y)$ nahradíme $X = f(Y, b),\ a = Y$.
- Rovnice ve tvaru *Proměnná=t*, *t* je jakýkoliv term:
  1. jestliže se *Proměnná* vyskytuje v *t* (např. jako parametr funktoru) $\Rightarrow$ KONEC, množina neshod nemá řešení.
  2. jinak všechny výskyty *Proměnné* v ostatních rovnicích nahradíme termem *t*, samotnou rovnici neměníme.

Příklad

$\mathsf{p}(f(X, g(X), Y), g(h(X), Y)),\ \mathsf{r}(X, f(Y, X, Y)) \to \mathsf{s}(h(X), a))$
$\mathsf{q}(a, f(a, Z, g(M, b))) \to \mathsf{p}(f(h(Z), R, b), M)$

Množina neshod:

$f(X, g(X), Y) = f(h(Z), R, b)$
$g(h(X), Y) = M$

Úpravy:

$X = h(Z)$	$X = h(Z)$	$X = h(Z)$
$g(X) = R$	$R = g(X)$	$R = g(h(Z))$
$Y = b$	$Y = b$	$Y = b$
$g(h(X), Y) = M$	$M = g(h(X), Y)$	$M = g(h(h(Z)), b)$

Unifikátor:

$\varphi = \{h(Z) \,/\, X,\ b \,/\, Y,\ Z \,/\, Z,\ g(h(Z)) \,/\, R,\ g(h(h(Z)), b) \,/\, M\}$

Příklad

$\underline{p(f(X, g(X), Y), g(h(X), Y))}, \; r(X, f(Y, X, Y)) \rightarrow s(h(X), a))$

$q(a, f(a, Z, g(M, b))) \rightarrow \underline{p(f(h(Z), R, b), M)}$

Množina neshod:

$f(X, g(X), Y) = f(h(Z), R, b)$
$g(h(X), Y) = M$

Úpravy:

$X = h(Z)$	$X = h(Z)$	$X = h(Z)$
$g(X) = R$	$R = g(X)$	$R = g(h(Z))$
$Y = b$	$Y = b$	$Y = b$
$g(h(X), Y) = M$	$M = g(h(X), Y)$	$M = g(h(h(Z)), b)$

Unifikátor:

$\varphi = \{h(Z) / X, \; b / Y, \; Z / Z, \; g(h(Z)) / R, \; g(h(h(Z)), b) / M\}$

Příklad

$\underline{\mathsf{p}(f(X,g(X),Y),g(h(X),Y))}$, $\mathsf{r}(X,f(Y,X,Y)) \rightarrow \mathsf{s}(h(X),a))$

$\mathsf{q}(a,f(a,Z,g(M,b))) \rightarrow \underline{\mathsf{p}(f(h(Z),R,b),M)}$

Množina neshod:

$f(X,g(X),Y) = f(h(Z),R,b)$
$g(h(X),Y) = M$

Úpravy:

$X = h(Z)$	$X = h(Z)$	$X = h(Z)$
$g(X) = R$	$R = g(X)$	$R = g(h(Z))$
$Y = b$	$Y = b$	$Y = b$
$g(h(X),Y) = M$	$M = g(h(X),Y)$	$M = g(h(h(Z)),b)$

Unifikátor:

$\varphi = \{h(Z)\,/\,X,\ b\,/\,Y,\ Z\,/\,Z,\ g(h(Z))\,/\,R,\ g(h(h(Z)),b)\,/\,M\}$

Příklad

$\underline{\mathsf{p}(f(X, g(X), Y), g(h(X), Y))}, \ \mathsf{r}(X, f(Y, X, Y)) \to \mathsf{s}(h(X), a))$

$\mathsf{q}(a, f(a, Z, g(M, b))) \to \underline{\mathsf{p}(f(h(Z), R, b), M)}$

Množina neshod:

$f(X, g(X), Y) = f(h(Z), R, b)$
$g(h(X), Y) = M$

Úpravy:

$X = h(Z)$	$X = h(Z)$	$X = h(Z)$
$g(X) = R$	$R = g(X)$	$R = g(h(Z))$
$Y = b$	$Y = b$	$Y = b$
$g(h(X), Y) = M$	$M = g(h(X), Y)$	$M = g(h(h(Z)), b)$

Unifikátor:

$\varphi = \{h(Z) \,/\, X,\ b \,/\, Y,\ Z \,/\, Z,\ g(h(Z)) \,/\, R,\ g(h(h(Z)), b) \,/\, M\}$

Příklad

$\underline{\mathsf{p}(f(X, g(X), Y), g(h(X), Y))}, \ \mathsf{r}(X, f(Y, X, Y)) \rightarrow \mathsf{s}(h(X), a))$

$\mathsf{q}(a, f(a, Z, g(M, b))) \rightarrow \underline{\mathsf{p}(f(h(Z), R, b), M)}$

Množina neshod:

$f(X, g(X), Y) = f(h(Z), R, b)$
$g(h(X), Y) = M$

Úpravy:

$X = h(Z)$	$X = h(Z)$	$X = h(Z)$
$g(X) = R$	$R = g(X)$	$R = g(h(Z))$
$Y = b$	$Y = b$	$Y = b$
$g(h(X), Y) = M$	$M = g(h(X), Y)$	$M = g(h(h(Z)), b)$

Unifikátor:

$\varphi = \{h(Z) / X, \ b / Y, \ Z / Z, \ g(h(Z)) / R, \ g(h(h(Z)), b) / M\}$

Příklad

$\underline{\mathsf{p}(f(X,g(X),Y),g(h(X),Y))}$, $\mathsf{r}(X,f(Y,X,Y)) \to \mathsf{s}(h(X),a))$
$\mathsf{q}(a,f(a,Z,g(M,b))) \to \underline{\mathsf{p}(f(h(Z),R,b),M)}$

Množina neshod:

$f(X,g(X),Y) = f(h(Z),R,b)$
$g(h(X),Y) = M$

Úpravy:

$X = h(Z)$	$X = h(Z)$	$X = h(Z)$
$g(X) = R$	$R = g(X)$	$R = g(h(Z))$
$Y = b$	$Y = b$	$Y = b$
$g(h(X),Y) = M$	$M = g(h(X),Y)$	$M = g(h(h(Z)),b)$

Unifikátor:

$\varphi = \{h(Z)\,/\,X,\ b\,/\,Y,\ Z\,/\,Z,\ g(h(Z))\,/\,R,\ g(h(h(Z)),b)\,/\,M\}$

Příklad

$\underline{\mathsf{p}(f(X,g(X),Y),g(h(X),Y))}$, $\mathsf{r}(X,f(Y,X,Y)) \to \mathsf{s}(h(X),a))$

$\mathsf{q}(a,f(a,Z,g(M,b))) \to \underline{\mathsf{p}(f(h(Z),R,b),M)}$

Množina neshod:

$f(X,g(X),Y) = f(h(Z),R,b)$
$g(h(X),Y) = M$

Úpravy:

$X = h(Z)$	$X = h(Z)$	$X = h(Z)$
$g(X) = R$	$R = g(X)$	$R = g(h(Z))$
$Y = b$	$Y = b$	$Y = b$
$g(h(X),Y) = M$	$M = g(h(X),Y)$	$M = g(h(h(Z)),b)$

Unifikátor:

$$\varphi = \{h(Z)\,/\,X,\ b\,/\,Y,\ Z\,/\,Z,\ g(h(Z))\,/\,R,\ g(h(h(Z)),b)\,/\,M\}$$

Příklad

$\underline{\mathsf{p}(f(X,g(X),Y),g(h(X),Y))},\ \mathsf{r}(X,f(Y,X,Y)) \rightarrow \mathsf{s}(h(X),a))$

$\mathsf{q}(a,f(a,Z,g(M,b))) \rightarrow \underline{\mathsf{p}(f(h(Z),R,b),M)}$

Unifikátor:

$\varphi = \{h(Z) \,/\, X,\ b \,/\, Y,\ Z \,/\, Z,\ g(h(Z)) \,/\, R,\ g(h(h(Z)),b) \,/\, M\}$

Unifikované klauzule:

$\underline{\mathsf{p}(f(h(Z),g(h(Z)),b),g(h(h(Z)),b))},\ \mathsf{r}(h(Z),f(b,h(Z),b)) \rightarrow \mathsf{s}(h(h(Z)),a))$

$\mathsf{q}(a,f(a,Z,g(g(h(h(Z)),b),b))) \rightarrow \underline{\mathsf{p}(f(h(Z),g(h(Z)),b),g(h(h(Z)),b))}$

Uplatnění rezolučního pravidla:

$\mathsf{r}(h(Z),f(b,h(Z),b)),\ \mathsf{q}(a,f(a,Z,g(g(h(h(Z)),b),b))) \rightarrow \mathsf{s}(h(h(Z)),a))$

Příklad

$\underline{p(f(X, g(X), Y), g(h(X), Y))}, \; r(X, f(Y, X, Y)) \rightarrow s(h(X), a))$

$q(a, f(a, Z, g(M, b))) \rightarrow \underline{p(f(h(Z), R, b), M)}$

Unifikátor:

$\varphi = \{h(Z) \,/\, X, \; b \,/\, Y, \; Z \,/\, Z, \; g(h(Z)) \,/\, R, \; g(h(h(Z)), b) \,/\, M\}$

Unifikované klauzule:

$\underline{p(f(h(Z), g(h(Z)), b), g(h(h(Z)), b))}, \; r(h(Z), f(b, h(Z), b)) \rightarrow s(h(h(Z)), a))$

$q(a, f(a, Z, g(g(h(h(Z)), b), b))) \rightarrow \underline{p(f(h(Z), g(h(Z)), b), g(h(h(Z)), b))}$

Uplatnění rezolučního pravidla:

$r(h(Z), f(b, h(Z), b)), \; q(a, f(a, Z, g(g(h(h(Z)), b), b))) \rightarrow s(h(h(Z)), a))$

Příklad

$\underline{\mathsf{p}(f(X,g(X),Y),g(h(X),Y))},\ \mathsf{r}(X,f(Y,X,Y)) \to \mathsf{s}(h(X),a))$

$\mathsf{q}(a,f(a,Z,g(M,b))) \to \underline{\mathsf{p}(f(h(Z),R,b),M)}$

Unifikátor:

$$\varphi = \{h(Z)\,/\,X,\ b\,/\,Y,\ Z\,/\,Z,\ g(h(Z))\,/\,R,\ g(h(h(Z)),b)\,/\,M\}$$

Unifikované klauzule:

$\underline{\mathsf{p}(f(h(Z),g(h(Z)),b),g(h(h(Z)),b))},\ \mathsf{r}(h(Z),f(b,h(Z),b)) \to \mathsf{s}(h(h(Z)),a))$

$\mathsf{q}(a,f(a,Z,g(g(h(h(Z)),b),b))) \to \underline{\mathsf{p}(f(h(Z),g(h(Z)),b),g(h(h(Z)),b))}$

Uplatnění rezolučního pravidla:

$\mathsf{r}(h(Z),f(b,h(Z),b)),\ \mathsf{q}(a,f(a,Z,g(g(h(h(Z)),b),b))) \to \mathsf{s}(h(h(Z)),a))$