Rezoluce v predikátové logice

- metoda založená na *vyvracení*, vhodná pro strojové dokazování
- pracujeme s formulemi ve Skolemově normální formě, kvantifikátory nepíšeme
- literály představují atomické formule a jejich negace
- používáme stejnou notaci jako ve výrokové logice:
  - klauzule: množiny reprezentující disjunkci literálů
  - formule: množiny klauzulí reprezentující jejich konjunkci
- příklad: formuli $\forall x \forall y ((P(x, f(x)) \vee \neg Q(y)) \wedge (\neg R(f(x)) \vee \neg Q(y)))$ reprezentujeme jako $\{\{P(x, f(x)), \neg Q(y)\}, \{\neg R(f(x)), \neg Q(y)\}\}$

Standardizace proměnných

- proměnné chápeme jako lokální v dané klauzuli (pozn.: $\forall x(A(x) \wedge B(x)) \Leftrightarrow (\forall x A(x) \wedge \forall x B(x)) \Leftrightarrow (\forall x A(x) \wedge \forall y B(y))$)
- mezi stejně pojmenovanými proměnnými v různých klauzulích není žádná vazba
- syntakticky tuto vlastnost zachytíme *standardizací proměnných*: přejmenujeme proměnné v různých klauzulích tak, aby v nich žádné stejně pojmenované proměnné nevystupovaly
- standardizace proměnných je nezbytná
- příklad: $\{\{P(x)\}, \{\neg P(f(x))\}\}$ je nesplnitelná, ale bez přejmenování proměnných výrazy $P(x)$ a $P(f(x))$ nezunifikujeme a bez unifikace neprovedeme ani rezoluční vyvrácení

Rezoluční pravidlo

- *rezoluční pravidlo pro predikátovou logiku*: mějme klauzule C_1 a C_2 bez společných proměnných (po případném přejmenování) ve tvaru $C_1 = C_1' \sqcup \{P(\vec{x}_1), \ldots, P(\vec{x}_n)\}$, $C_2 = C_2' \sqcup \{\neg P(\vec{y}_1), \ldots, \neg P(\vec{y}_m)\}$. Je-li ϕ mgu množiny $\{P(\vec{x}_1), \ldots, P(\vec{x}_n), P(\vec{y}_1), \ldots, P(\vec{y}_m)\}$, pak *rezolventou* C_1 a C_2 je $C_1'\phi \cup C_2'\phi$.
- *rezoluční důkazy* a *rezoluční vyvrácení* definujeme stejně jako ve výrokové logice, pouze používáme rezoluční pravidlo pro predikátovou logiku:
  - *rezoluční důkaz* C z S je binární strom s listy z S a kořenem C, jehož každý vnitřní uzel je rezolventou svých bezprostředních následníků,
  - *rezoluční vyvrácení* S je rezoluční důkaz $\square$ z S

Rezolventa – příklady I

Př. 1: rezolvujeme klauzule $\{P(x,a)\}$ a $\{\neg P(x,x)\}$

- přejmenujeme proměnné např. v první klauzuli: $\{P(x_1,a)\}$
- $mgu(\{P(x_1,a),P(x,x)\}) = \{x_1/a, x/a\}$
- rezolventa $\square$

Př. 2: rezolvujeme klauzule $\{P(x,y),\neg R(x)\}$ a $\{\neg P(a,b)\}$

- $mgu(\{P(x,y),P(a,b)\}) = \{x/a, y/b\}$
- aplikujeme mgu na $\{\neg R(x)\}$
- rezolventa $\{\neg R(a)\}$

Rezolventa – příklady II

Př. 3: rezolvujeme klauzule $C_1 = \{Q(x), \neg R(y), P(x,y), P(f(z), f(z))\}$ a $C_2 = \{\neg N(u), \neg R(w), \neg P(f(a), f(a)), \neg P(f(w), f(w))\}$

- vybereme množinu literálů, na kterých budeme rezolvovat $\{P(x,y), P(f(z), f(z)), P(f(a), f(a)), P(f(w), f(w))\}$
- její mgu $\phi = \{x/f(a), y/f(a), z/a, w/a\}$
- $C_1' = \{Q(x), \neg R(y)\}$, $C_1'\phi = \{Q(f(a)), \neg R(f(a))\}$
- $C_2' = \{\neg N(u), \neg R(w)\}$, $C_2'\phi = \{\neg N(u), \neg R(a)\}$
- výsledná rezolventa $C_1'\phi \cup C_2'\phi = \{Q(f(a)), \neg R(f(a)), \neg N(u), \neg R(a)\}$

Rezoluční důkazy

- obecné schméma důkazu ‚A je důsledkem množiny formulí $\mathbf{T}$‘: všechny prvky $\mathbf{T}$ a $\neg A$ převedeme do klauzulární formy, dokazujeme nesplnitelnost jejich sjednocení
- poznámka: při jednom rezolučním kroku musíme být schopni odstranit několik literálů zároveň (pokud bychom v následujícím příkladu rezolvovali vždy pouze na jediném literálu, množinu rezolučně nevyvrátíme)
- příklad: ukažte rezoluční vyvrácení $\{\{P(x), P(y)\}, \{\neg P(x_1), \neg P(y_1)\}\}$

$$\begin{tikzcd}
\{P(x),P(y)\} \arrow[dr, dash, "x/y_1, y/y_1, x_1/y_1"'] & & \{\neg P(x_1),\neg P(y_1)\} \arrow[dl, dash, "x/y_1, y/y_1, x_1/y_1"] \\
& \square &
\end{tikzcd}$$

Rezoluční důkaz – příklad I

Z předpokladu reflexivity $\forall x P(x,x)$ a vlastnosti $\forall x \forall y \forall z((P(x,y) \wedge P(y,z)) \Rightarrow P(z,x))$ dokažte symetrii $\forall x \forall y(P(x,y) \Rightarrow P(y,x))$.

- převedeme předpoklady do klauzulárního tvaru:

  $S_1 = \{\{P(x,x)\}\}$

  $S_2 = \{\{\neg P(x,y), \neg P(y,z), P(z,x)\}\}$

- dokazovanou formuli negujeme:

  $\exists x \exists y(P(x,y) \wedge \neg P(y,x))$,

  převedeme do klauzulárního tvaru (přes Skolemovu nf):

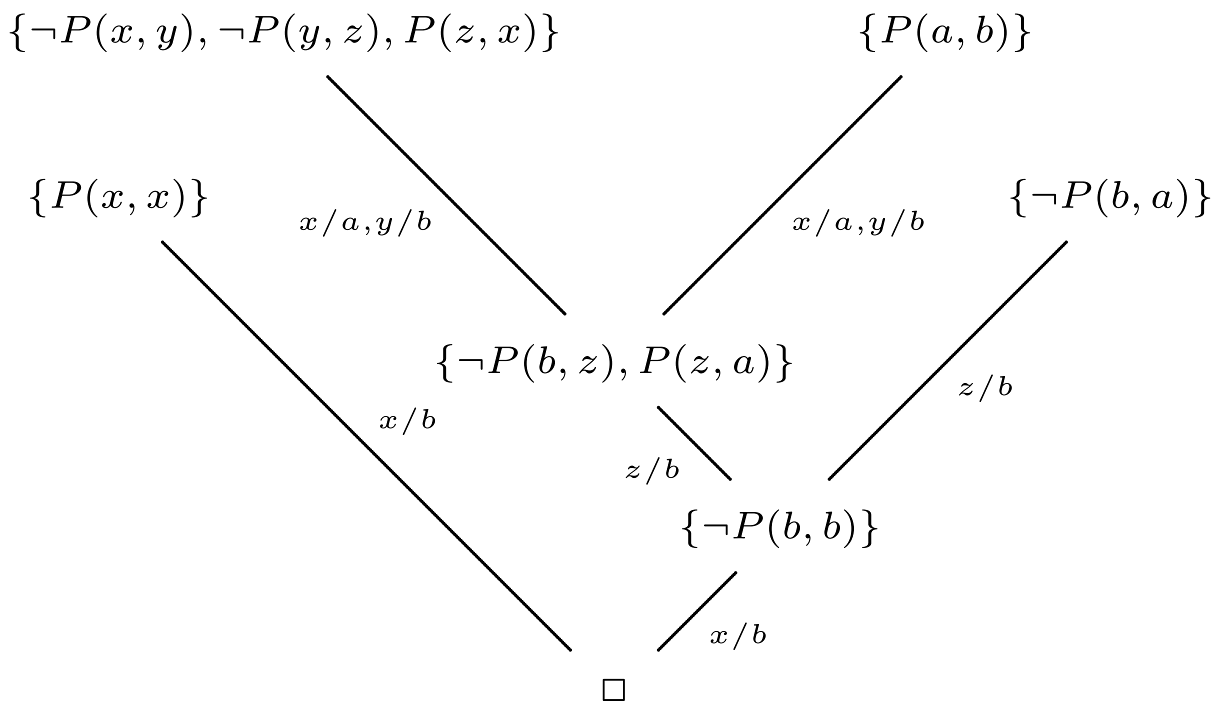
  $P(a,b) \wedge \neg P(b,a)$

  $S = \{\{P(a,b)\}, \{\neg P(b,a)\}\}$

- dokazujeme nesplnitelnost $S_1 \cup S_2 \cup S$

Rezoluční důkaz – příklad II

- dokazujeme nesplnitelnost množiny klauzulí
  $\{\{P(x,x)\},\{\neg P(x,y),\neg P(y,z),P(z,x)\},\{P(a,b)\},\{\neg P(b,a)\}\}$
- strom rezolučního vyvrácení (jeden z možných):

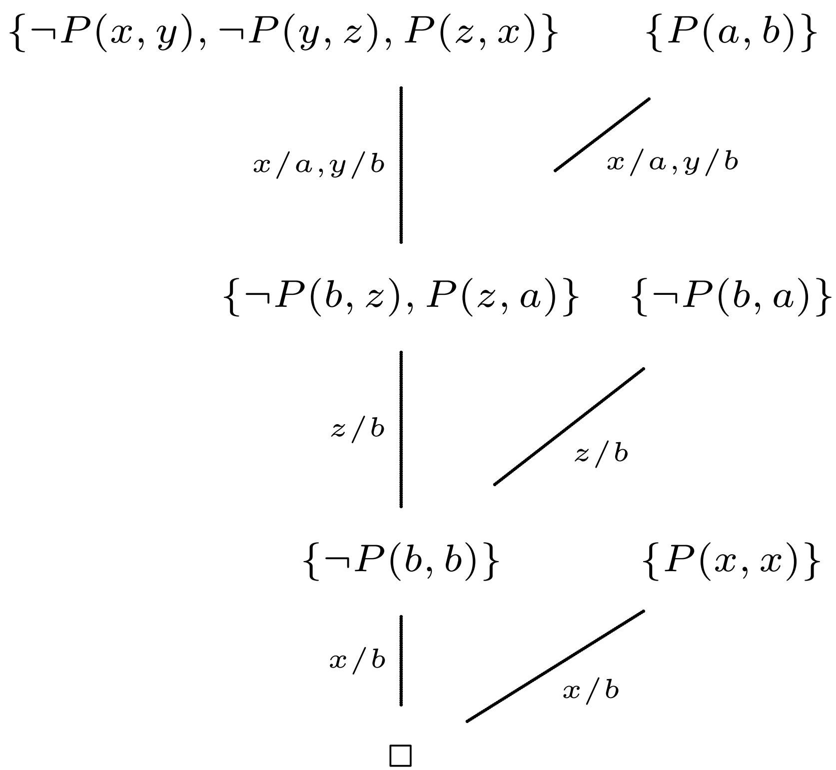


Rezoluce – vlastnosti

- stejně jako ve výrokové logice je rezoluce v predikátové logice korektní a úplná
- stejný problém jako ve výrokové logice: strategie generování rezolvent (příliš velký prohledávaný prostor)
- lze použít všechny metody zjemnění uvedené pro výrokovou logiku (T-rezoluce, sémantická rezoluce atd.)
- uvedeme pouze příklady variant rezoluce postupně směřujících k SLD-rezoluci používané v Prologu; jejich vlastnosti jsou stejné jako ve výrokové logice

Lineární rezoluce

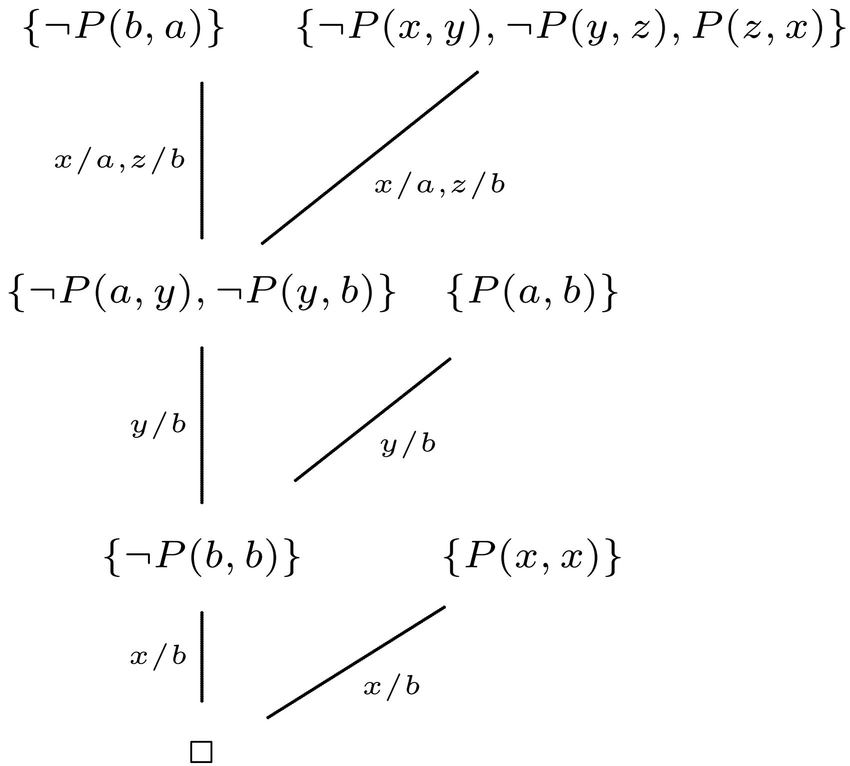
- lineární struktura důkazu, rezolvujeme vždy předchozí rezolventu s klauzulí z vyvracené množiny nebo dříve odvozenou rezolventou; korektní a úplná
- příklad: lineární rezoluční vyvrácení množiny $\{\{P(x,x)\},\{\neg P(x,y),\neg P(y,z),P(z,x)\},\{P(a,b)\},\{\neg P(b,a)\}\}$



LI-rezoluce (lineární vstupní rezoluce)

- lineární rezoluce začínající cílovou klauzulí (žádný pozitivní literál), rezolvujeme vždy předchozí rezolventu (výhradně) s klauzulí z vyvracené množiny. Korektní, obecně není úplná; úplná pro Hornovy klauzule.

Př.: $\{\{P(x,x)\},\{\neg P(x,y),\neg P(y,z),P(z,x)\},\{P(a,b)\},\{\neg P(b,a)\}\}$

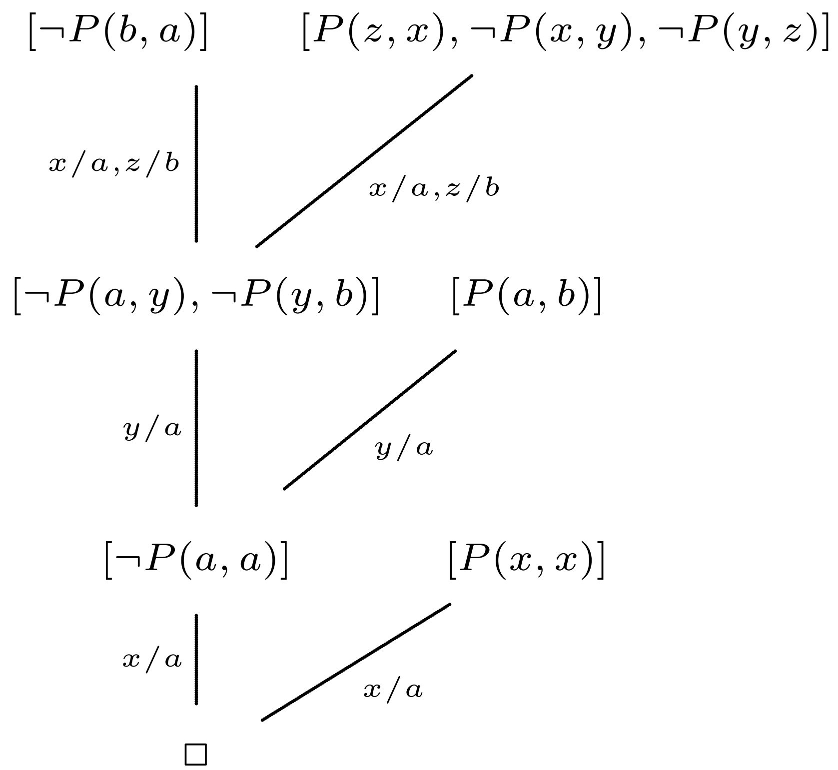


LD-rezoluce

- vychází z LI-rezoluce, směřujeme k implementaci
- pracujeme výhradně s Hornovými klauzulemi (nejvýše jeden pozitivní literál), pojmy přebíráme z výrokové logiky (fakt, cíl, pravidlo, programová klauzule apod.)
- klauzule opět nahradíme *uspořádanými klauzulemi*; změna notace z $\{P(x), \neg R(x, f(y)), \neg Q(a)\}$ na $[P(x), \neg R(x, f(y)), \neg Q(a)]$
- rezoluce pro uspořádané klauzule v predikátové logice: mějme uspořádané klauzule bez společných proměnných (po případném přejmenování)
  $G = [\neg A_1, \neg A_2, \dots, \neg A_n]$ a
  $H = [B_0, \neg B_1, \neg B_2, \dots, \neg B_m]$,
  rezolventou G a H pro $\phi = mgu(B_0, A_i)$ bude uspořádaná klauzule
  $[\neg A_1\phi, \neg A_2\phi, \dots, \neg A_{i-1}\phi, \neg B_1\phi, \neg B_2\phi, \dots, \neg B_m\phi, \neg A_{i+1}\phi, \dots, \neg A_n\phi]$

LD-rezoluce: příklad

- LD-rezoluční vyvrácení množiny uspořádaných klauzulí
  $\{[P(x,x)],[P(z,x),\neg P(x,y),\neg P(y,z)],[P(a,b)],[\neg P(b,a)]\}$



- LD-rezoluce je korektní a úplná pro Hornovy klauzule

SLD-rezoluce

- pomocí *selekčního pravidla* algoritmizuje výběr literálu z cílové klauzule, na kterém se bude rezolvovat
- SLD-rezoluce (s libovolným selekčním pravidlem) je korektní a úplná pro Hornovy klauzule
- budeme používat selekční pravidlo, které vybírá nejlevější literál
- generování rezolvent pro uvedené pravidlo:
  $G = [\neg A_1, \neg A_2, \ldots, \neg A_n]$,
  $H = [B_0, \neg B_1, \neg B_2, \ldots, \neg B_m]$,
  rezolventou G a H pro $\phi = mgu(B_0, A_1)$ je
  $[\neg B_1\phi, \neg B_2\phi, \ldots, \neg B_m\phi, \neg A_2\phi, \ldots, \neg A_n\phi]$

SLD-rezoluce: příklad

- příklad: SLD-rezoluční vyvrácení se zvoleným selekčním pravidlem (vybírajícím vždy nejlevější literál)

$$
\begin{tikzcd}
{[\neg P(b,a)]} \arrow[d, dash, "x/a{,}z/b"'] & {[P(z,x),\neg P(x,y),\neg P(y,z)]} \arrow[dl, dash, "x/a{,}z/b"] \\
{[\neg P(a,y),\neg P(y,b)]} \arrow[d, dash, "y/b"'] & {[P(a,b)]} \arrow[dl, dash, "y/b"] \\
{[\neg P(b,b)]} \arrow[d, dash, "x/b"'] & {[P(x,x)]} \arrow[dl, dash, "x/b"] \\
\square &
\end{tikzcd}
$$

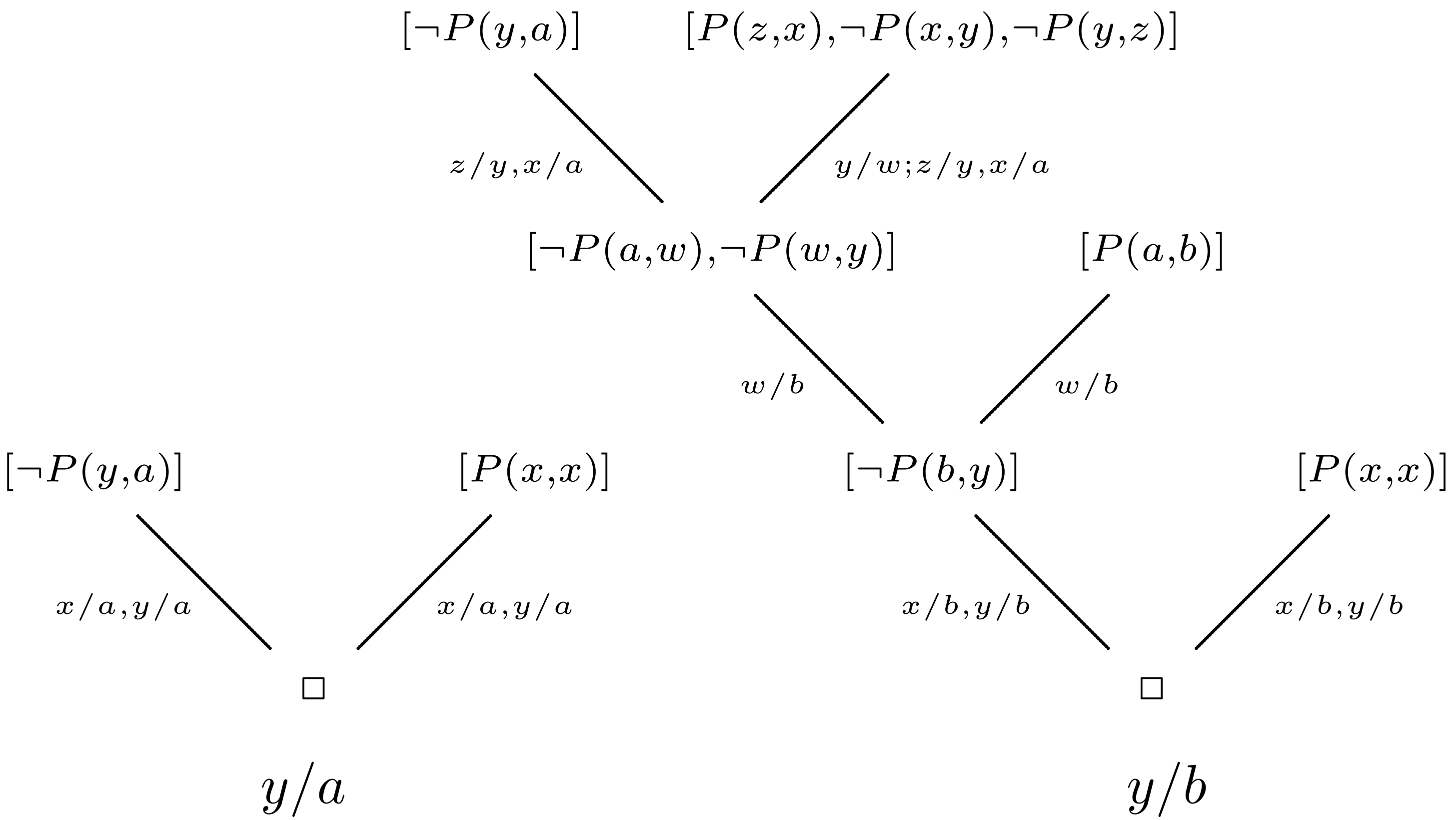
- srovnání s LD-rezolucí: ve druhém kroku musíme rezolvovat na $\neg P(a,y)$ (v LD-rezoluci bylo možné vybrat libovolný z literálů $\neg P(a,y), \neg P(y,b)$)

SLD-rezoluce: význam

- máme množinu programových klauzulí P a cílovou klauzuli $G = [\neg A_1(\vec{x}), \ldots, \neg A_n(\vec{x})]$
- dokazujeme nesplnitelnost $P \cup \{G\}$, tj. $P \wedge \forall \vec{x}(\neg A_1(\vec{x}) \vee \ldots \vee \neg A_n(\vec{x}))$
- uvedená nesplnitelnost je ekvivalentní $P \vdash \neg G$, tj. $P \vdash \exists \vec{x}(A_1(\vec{x}) \wedge \ldots \wedge A_n(\vec{x}))$
- zadáním cíle G tedy chceme zjistit, zda existují nějaké objekty (případně jaké), které na základě P splňují formuli $A_1(\vec{x}) \wedge \ldots \wedge A_n(\vec{x})$
- aplikujeme-li kompozici všech mgu postupně použitých při SLD-odvození na jednotlivé proměnné vektoru $\vec{x}$, získáme konkrétní příklady zmíněných objektů (termů) splňujících danou formuli

Příklad: je dán program

$P = \{[P(a,b)], [P(x,x)], [P(z,x), \neg P(x,y), \neg P(y,z)]\}$, hledáme konkrétní možná řešení (substituce proměnných) cíle $[\neg P(y,a)]$



SLD-stromy

Prostor všech možných SLD-derivací při vyhodnocování daného cíle G pro program P dokážeme zachytit *SLD-stromem*. Příklad:

1. $[P(x,y), \neg Q(x,z), \neg R(z,y)]$
2. $[P(x,x), \neg S(x)]$
3. $[Q(x,b)]$
4. $[Q(b,a)]$
5. $[Q(x,a), \neg R(a,x)]$
6. $[R(b,a)]$
7. $[S(x), \neg T(x,a)]$
8. $[S(x), \neg T(x,b)]$
9. $[S(x), \neg T(x,x)]$
10. $[T(a,b)]$
11. $[T(b,a)]$

cíl: $[\neg P(x,x)]$