

5. Rezoluce, Prolog, SLD-stromy ve výrokové logice

Příklad 1. Pomocí LD rezoluce ukažte, že následující množina klauzulí je nespílitelná:

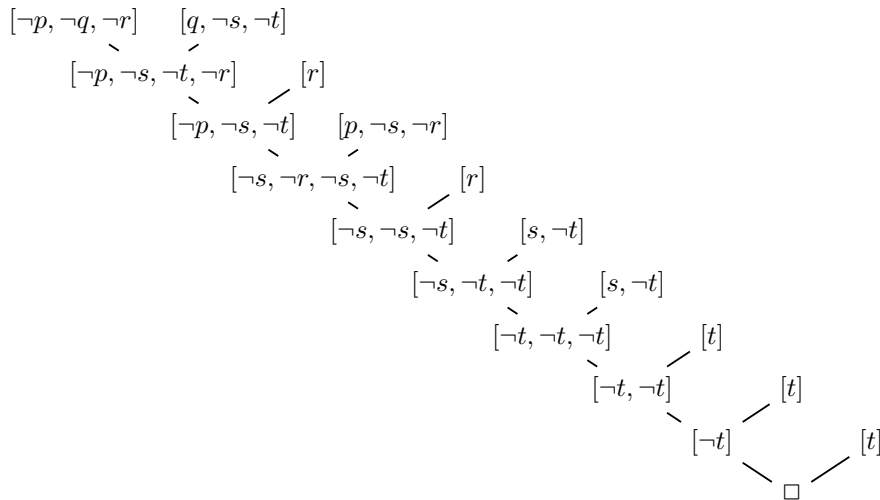
$$S = \{\{\neg s, q, \neg t\}, \{\neg s, \neg r, p\}, \{\neg t, s\}, \{\neg p, \neg q, \neg r\}, \{r\}, \{t\}\}$$

LD (lineární vstupní rezoluce pro uspořádané klauzule): jedná se o podobnou rezoluci jako LI, ale klauzule jsou uspořádané seznamy (nelze automaticky eliminovat duplicitní literály!).

Rezolventou: $[\neg A_1, \dots, \neg A_n]$ a $[B_0, \neg B_1, \dots, \neg B_m]$ a pro $A_i = \neg B_0$ je

$$[\neg A_1, \dots, \neg A_{i-1}, \neg B_1, \dots, \neg B_m, \neg A_{i+1}, \dots, \neg A_n]$$

Řešení 1. Převědeme množinu klauzulí na množinu uspořádaných klauzulí: $S' = \{[q, \neg s, \neg t], [p, \neg s, \neg r], [s, \neg t], [\neg p, \neg q, \neg r], [r], [t]\}$ (pozitivní literál musí být na prvním místě). Najdeme cílovou klauzuli (samé negativní literály) a vytvoříme LD-rezoluční vyvrácení:



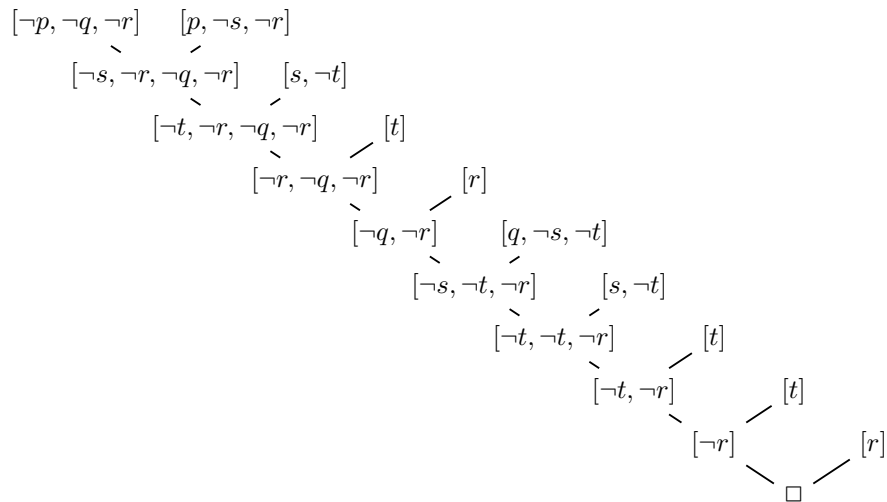
Příklad 2. Pomocí SLD rezoluce ukažte, že následující množina uspořádaných klauzulí je nespílitelná:

$$S = \{[q, \neg s, \neg t], [p, \neg s, \neg r], [s, \neg t], [r], [t]\} \cup \{[\neg p, \neg q, \neg r]\}$$

SLD (lineární vstupní rezoluce pro uspořádané klauzule s výběrovým pravidlem):

- v i -tém kroku se rezolvuje na literálu $R(G_i)$;
- používá se v jazyce Prolog (R vybírá vždy nejlevější pozici).

Řešení 2.



Příklad 3. Převeďte následující množinu uspořádaných klauzulí na program v jazyce Prolog a nalezněte jeho SLD-rezoluční vyvrácení:

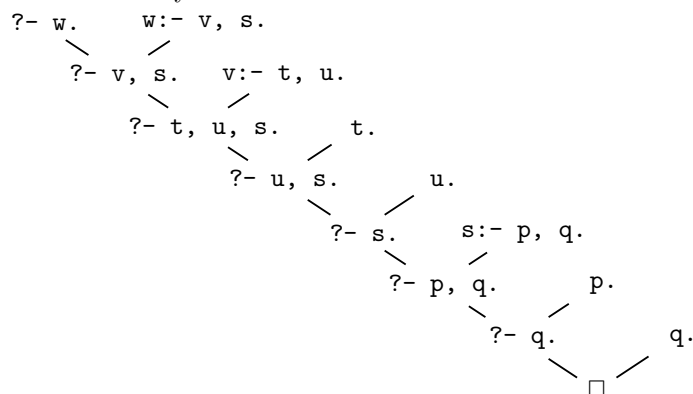
$S = \{[r, \neg p, \neg q], [s, \neg p, \neg q], [v, \neg t, \neg u], [w, \neg v, \neg s], [t], [q], [u], [p], [\neg w]\}$

Řešení 3.

Prologovský program:

```
r:- p, q.
s:- p, q.
v:- t, u.
w:- v, s.
t.
q.
u.
p.
?- w.
```

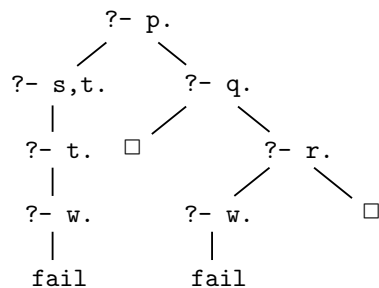
SLD-rezoluční vyvrácení:



Příklad 4. Vytvořte SLD-strom pro následující program v jazyce Prolog:

```
p:- s, t.
p:- q.
q.
q:- r.
r:- w.
r.
s.
t:- w.
?- p.
```

Řešení 4.

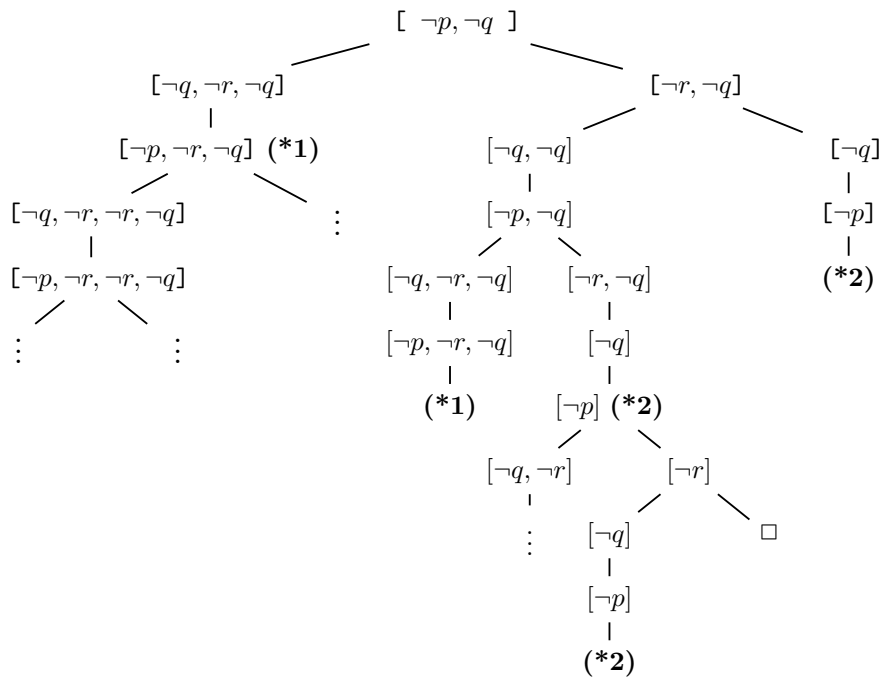


Příklad 5. Srovnajte SLD-stromy pro dotaz $?- p, q.$ a následující dva programy v jazyce Prolog, které se liší pouze pořadím klauzulí:

Program 1:	Program 2:
p:- q, r.	p:- r.
p:- r.	p:- q, r.
q:- p.	q:- p.
r:- q.	r.
r.	r:- q.

Řešení 5. Stromy jsou rozsáhlé a některé větve mají nekonečné. Každá větev se nachází v obou stromech, ale na jiném místě. Například nejlevější nekonečná větev stromu pro první program odpovídá nejpravější větvi pro druhý program. Vzhledem k tomu, že Prolog prohledává strom do hloubky (zleva doprava), záleží na pořadí klauzulí v programu a je potřeba mu při psaní programů věnovat náležitou pozornost.

Program 1:



Program 2:

