

9. Rezoluce v predikátové logice, Prolog

Příklad 1. Vyvráťte pomocí SLD rezoluce následující množinu klauzulí:

$$S = \{\{P(x), \neg Q(x, f(y)), \neg R(a)\}, \{R(x), \neg Q(x, y)\}, \{\neg P(x), \neg Q(y, z)\}, \\ \{P(x), \neg R(x)\}, \{R(f(b))\}, \{Q(x, y), \neg P(y)\}\}$$

Příklad 2.

a) Předpokládejte, že platí následující tři tvrzení:

- Existuje drak ($D/1$).
- Draci spí ($S/1$) nebo loví ($L/1$).
- Když jsou draci hladoví ($H/1$), tak nespí.

Dokažte pomocí rezoluce, že když jsou draci hladoví, tak loví.

b) Předpokládejte, že platí následující dvě tvrzení:

- Všichni holiči ($B/1$) holí ($S/2$) každého, kdo se neholí sám.
- Žádný holič neholí někoho, kdo se holí sám.

Dokažte pomocí rezoluce, že holiči neexistují.

Řešení 2.

- a) $\exists x D(x)$
 $\forall x (D(x) \Rightarrow (S(x) \vee L(x)))$
 $\forall x ((D(x) \wedge H(x)) \Rightarrow \neg S(x))$
 Závěr: $\forall x ((D(x) \wedge H(x)) \Rightarrow L(x))$

Všechny předpoklady a negaci závěru převedeme na klauzulární tvar a dokazujeme nesplnitelnost takto vzniklé množiny klauzulí:

$$\{\{D(a)\}, \\ \{\neg D(x), S(x), L(x)\}, \\ \{\neg D(x), \neg H(x), \neg S(x)\}, \\ \{D(b)\}, \{H(b)\}, \{\neg L(b)\}\}$$

- b) $\forall x \forall y ((B(x) \wedge \neg S(y, y)) \Rightarrow S(x, y))$
 $\forall y (S(y, y) \Rightarrow \neg \exists x (B(x) \wedge S(x, y)))$
 Závěr: $\neg \exists x B(x)$

Všechny předpoklady a negaci závěru převedeme na klauzulární tvar a dokazujeme nesplnitelnost takto vzniklé množiny klauzulí:

$$\{\{\neg B(x), S(y, y), S(x, y)\}, \\ \{\neg S(y, y), \neg B(x), \neg S(x, y)\}, \\ \{B(a)\}\}$$

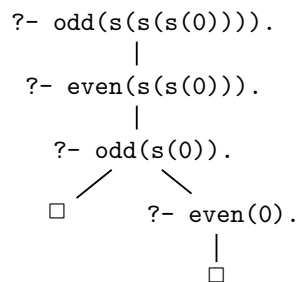
Příklad 3. Mějme následující program v jazyce Prolog:

```
even(0).
even(s(X)):- odd(X).
odd(s(0)).
odd(s(X)):- even(X).
```

- Vytvořte SLD strom pro cíl `?- odd(s(s(s(0))))`.
- Jaké výsledky Prolog postupně vrátí při systematickém procházení SLD stromu (pomocí hledání alternativních důkazů zadáním středníku) pro dotaz `?- odd(X).`?
- Jak docílíme toho, aby každý výsledek vrátil jenom jednou?

Řešení 3.

a)



b) `?- odd(X).`

```

X = s(0) ;
X = s(0) ;
X = s(s(s(0))) ;
X = s(s(s(0))) ;
X = s(s(s(s(s(0)))))) ;
X = s(s(s(s(s(0)))))) ;
X = s(s(s(s(s(s(s(0)))))))) ;
X = s(s(s(s(s(s(s(0)))))))) ;
...
```

c) Vynecháním faktu `odd(s(0)).`

Příklad 4. Co dělá následující predikát v jazyce Prolog?

```
f(0,1).
f(N,F):- N1 is N-1, f(N1,F1), F is N*F1.
```

Jak dopadne volání cíle `?- f(-3,F).`?

Řešení 4. Predikát ve druhém argumentu vrátí faktoriál nezáporného čísla zadaného jako první argument.

Pro daný dotaz běží výpočet do nekonečna (resp. do vyčerpání lokálního zásobníku).