FMI, Info, Anul II, 2021-2022
Fundamentele Limbajelor de Programare

Seminar 4
Rezoluţie SLD

Teorie pentru S4.1:

- O *clauză definită* este o formulă de forma:
  - $P(t_1, \ldots, t_n)$ (formulă atomică), unde P este un simbol de predicat, iar $t_1, \ldots, t_n$ termeni
  - $P_1 \wedge \ldots \wedge P_n \to Q$, unde toate P_i, Q sunt formule atomice.
- O regulă din Prolog $\mathtt{Q :- P_1, \ldots, P_n}$ este o clauză $P_1 \wedge \ldots \wedge P_n \to Q$, iar un fapt din Prolog $\mathtt{P(t_1, \ldots, t_n)}$ este o formulă atomică $P(t_1, \ldots, t_n)$.
- O clauză definită $P_1 \wedge \ldots \wedge P_n \to Q$ poate fi gândită ca formula $Q \vee \neg P_1 \vee \ldots \vee \neg P_n$.
- Pentru o mulţime de clauze definite T, *regula rezolutiei SLD* este

$$\text{SLD}\ \boxed{\frac{\neg P_1 \vee \cdots \vee \neg P_i \vee \cdots \vee \neg P_n}{(\neg P_1 \vee \cdots \vee \neg Q_1 \vee \cdots \vee \neg Q_m \vee \cdots \vee \neg P_n)\theta}}$$

  unde $Q \vee \neg Q_1 \vee \cdots \vee \neg Q_m$ este o clauză definită din T (în care toate variabilele au fost redenumite) şi θ este c.g.u pentru P_i şi Q.
- Fie T o mulţime de clauze definite şi $P_1 \wedge \ldots \wedge P_m$ o ţintă, unde P_i sunt formule atomice. O derivare din T prin rezoluţie SLD este o secvenţă $G_0 := \neg P_1 \vee \ldots \vee \neg P_m$, G_1, …, G_k, … în care G_{i+1} se obţine din G_i prin regula SLD. Dacă există un k cu $G_k = \square$ (clauza vidă), atunci derivarea se numeşte *SLD-respingere*.

Teorema 1 (Completitudinea SLD-rezoluţiei)**.** *Sunt echivalente:*

(i) există o SLD-respingere a lui $P_1 \wedge \ldots \wedge P_m$ din T,

(ii) $T \vDash P_1 \wedge \cdots \wedge P_m$.

(S4.1) Găsiţi o SLD-respingere pentru următoarele programe Prolog şi ţinte:

(a)
```
1.  r :- p,q.     5.  t.      ?- w.
2.  s :- p,q.     6.  q.
3.  v :- t,u.     7.  u.
4.  w :- v,s.     8.  p.
```

(b)
```
1.  q(X,Y) :- q(Y,X), q(Y,f(f(Y))).     ?- q(f(Z),a).
2.  q(a,f(f(X))).
```

(c)

```
1. p(X) :- q(X,f(Y)), r(a).      4. r(X) :- q(X,Y).      ?- p(X), q(Y,Z).
2. p(X) :- r(X).                 5. r(f(b)).
3. q(X,Y) :- p(Y).
```

Teorie pentru S4.2:

Fie T o mulţime de clauze definite şi o ţintă $G_0 = \neg P_1 \vee \ldots \vee \neg P_m$. Un *arbore SLD* este definit astfel:

- Fiecare nod al arborelui este o ţintă (posibil vidă)
- Rădăcina este G_0
- Dacă arborele are un nod G_i, iar G_{i+1} se obţine din G_i folosind regula SLD folosind o clauză $C_i \in T$, atunci nodul G_i are copilul G_{i+1}. Muchia dintre G_i şi G_{i+1} este etichetată cu C_i.

Dacă un arbore SLD cu rădăcina G_0 are o frunză $\square$ (clauza vidă), atunci există o SLD-respingere a lui G_0 din T.

(S4.2) Desenaţi arborele SLD pentru programul Prolog de mai jos şi ţinta `?- p(X,X).`

```
1. p(X,Y) :- q(X,Z), r(Z,Y).     7. s(X) :- t(X,a).
2. p(X,X) :- s(X).               8. s(X) :- t(X,b).
3. q(X,b).                       9. s(X) :- t(X,X).
4. q(b,a).                       10. t(a,b).
5. q(X,a) :- r(a,X).             11. t(b,a).
6. r(b,a).
```