

## Exerciții de seminar

### 1 Programare imperativă

#### 1.1 Semantica operațională

Fie  $X, Y, Z \in L$ , distincte două câte două. Să se descrie formal execuția următoarelor programe IMP din stările inițiale indicate, folosind semanticile operaționale big-step și small-step:

1. **if**  $X \leq 7$  **then**  $Z := X$  **else**  $Z := Y$ , dintr-o stare inițială  $\sigma$  cu  $\sigma(X) = 0$ ,  $\sigma(Y) = 1$ ,  $\sigma(Z) = 7$ ;
2. **while**  $4 \leq X$  **do**  $X := X - Y$ , dintr-o stare inițială  $\sigma$  cu  $\sigma(X) = 6$ ,  $\sigma(Y) = 3$ ;
3. **while**  $\neg(Y = 0)$  **do**  $(Y := Y - 1; X := 2 * X)$ , dintr-o stare inițială  $\sigma$  cu  $\sigma(X) = 1$ ,  $\sigma(Y) = 3$ ;
4. **while**  $\neg(X = Y)$  **do** (**if**  $X \leq Y$  **then**  $Y := Y - X$  **else**  $X := X - Y$ ), dintr-o stare inițială  $\sigma$  cu  $\sigma(X) = 9$ ,  $\sigma(Y) = 12$ .

### 2 Programare logică

#### 2.1 Unificare

Considerăm

- $x, y, z, u, v, w$  variabile,
- $a, b, c$  constante,
- $h, g$  simboluri de funcție de aritate 1,
- $f$  simbol de funcție de aritate 2,
- $p$  simbol de funcție de aritate 3.

Aplicați algoritmul de unificare din curs pentru a găsi un unificator pentru termenii:

1.  $f(x, y)$ ,  $f(h(x), x)$  și  $f(x, b)$
2.  $f(x, f(x, g(y)))$ ,  $f(u, z)$  și  $f(g(y), y)$
3.  $f(f(x, y), x)$ ,  $f(g(y), z)$  și  $f(u, h(z))$
4.  $f(f(x, y), x)$ ,  $f(v, u)$  și  $f(u, h(z))$
5.  $f(f(x, y), x)$ ,  $f(v, u)$  și  $f(u, z)$
6.  $f(f(g(x), h(y)), h(z))$ ,  $f(f(u, h(h(x))), h(y))$  și  $f(v, w)$
7.  $p(x, x, z)$ ,  $p(f(a, a), y, y)$  și  $p(f(x, a), b, z)$
8.  $p(x, x, z)$ ,  $p(f(a, a), y, y)$  și  $p(x, b, z)$
9.  $p(x, x, z)$ ,  $p(f(a, a), y, y)$  și  $p(x, f(a, a), z)$
10.  $p(f(x, a), g(y), z)$ ,  $p(f(a, a), z, u)$  și  $p(v, u, z)$

## 2.2 Rezoluție

Găsiți o SLD-respingere pentru următoarele programe Prolog și ținte:

1.

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

2.

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

3.

```
p(X) :- q(X,f(Y)), r(a).      r(X) :- q(X,Y).      ?- p(X), q(Y,Z).
p(X) :- r(X).                  r(f(b)).
q(X,Y) :- p(Y).
```

## 3 Programare funcțională

### 3.1 Lambda-calcul fără tipuri

Reduceți următorii termeni până la o formă normală:

1.  $((\lambda z.z)(\lambda q.(qq)))(\lambda s.(sa));$
2.  $((\lambda z.z)(\lambda z.(zz)))(\lambda z.(zq));$
3.  $((\lambda s.\lambda q.(sq)))(\lambda a.a)b;$
4.  $((\lambda s.\lambda q.(sq)))(\lambda q.q)q;$
5.  $((\lambda s.(ss)))(\lambda q.q)(\lambda q.q).$

### 3.2 Lambda-calcul cu tipuri

Considerăm următorii termeni:

1.  $\lambda xyz.(x(yz));$
2.  $\lambda xy.(xy(\lambda z.y));$
3.  $(\lambda xyz.zxy)(\lambda xyz.y)(\lambda xy.y).$

Pentru fiecare dintre ei, aplicați algoritmul de inferență a tipurilor și prezentați o deducție în sistemul de deducție corespunzător care să arate că termenului i se poate aloca tipul obținut prin algoritm.