

1. Nadefinujte funkci *forEach* tak, aby vrátila akci, která postupně projde všechny prvky sekvence zadané v prvním parametru funkce *forEach* a na každém z nich vykoná akci zadanou v druhém parametru funkce *forEach*.

$$forEach : A^* \times (A \times State \rightarrow State) \rightarrow (State \rightarrow State)$$

2. Dokažte následující tvrzení: pokud $(\vdash t : T) \wedge (t \Rightarrow t')$, pak $\vdash t' : T$.

$$\begin{array}{l} t ::= \text{true} \mid \text{false} \mid \\ 0 \mid \text{succ } t \mid \\ \text{if } t \text{ then } t \text{ else } t \end{array} \quad (1)$$

$$\frac{}{\text{if true then } t_1 \text{ else } t_2 \Rightarrow t_1} \quad (2)$$

$$\frac{}{\text{if false then } t_1 \text{ else } t_2 \Rightarrow t_2} \quad (3)$$

$$\frac{t_1 \Rightarrow t'_1}{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \Rightarrow \text{if } t'_1 \text{ then } t_2 \text{ else } t_3} \quad (4)$$

$$\frac{t_1 \Rightarrow t'_1}{\text{succ } t_1 \Rightarrow \text{succ } t'_1} \quad (5)$$

$$\frac{}{\vdash \text{true} : Bool} \quad (6)$$

$$\frac{}{\vdash \text{false} : Bool} \quad (7)$$

$$\frac{}{\vdash 0 : Nat} \quad (8)$$

$$\frac{\vdash t : Nat}{\vdash \text{succ } t : Nat} \quad (9)$$

$$\frac{\vdash t_1 : Bool \quad \vdash t_2 : T \quad \vdash t_3 : T}{\vdash \text{if } t_1 \text{ then } t_2 \text{ else } t_3 : T} \quad (10)$$

3. Dokažte, že přepisovací relace \Rightarrow definovaná v předcházejícím příkladě je silně normalizující (vždy terminuje).

4. Sémantika S má vstupní funkci $\lambda z, e.(z, e)$, výstupní funkci $\lambda(z, z').z'$, množinu konfigurací $Z \times Expr$, množinu finálních konfigurací $Z \times Z$ a přepisovací relaci \Rightarrow . Napište ekvivalentní denotační sémantiku, jako sémantickou doménu použijte $Z \rightarrow Z$.

$$Expr ::= Num \mid \Delta Expr \mid Expr \odot Expr \mid \square \quad (11)$$

$$\overline{(z, \Delta z') \Rightarrow (z, -z')} \quad (12)$$

$$\overline{(z, z' \odot z'') \Rightarrow (z, z' + z'')} \quad (13)$$

$$\frac{(z, e) \Rightarrow (z, e')}{(z, \Delta e) \Rightarrow (z, \Delta e')} \quad (14)$$

$$\frac{(z, e) \Rightarrow (z, e')}{(z, e \odot e'') \Rightarrow (z, e' \odot e'')} \quad (15)$$

$$\frac{(z, e'') \Rightarrow (z, e')}{(z, e \odot e'') \Rightarrow (z, e \odot e')} \quad (16)$$

$$\overline{(z, \square) \Rightarrow (z, z)} \quad (17)$$

5. Pravidlo 18 je pravidlo pro podtypování rekurzivních typů. Na příkladu vysvětlíte, proč by pravidlo 19 nefungovalo. Svoje zdůvodnění opřete o vlastnosti programovacích jazyků jako je soundness, progress apod.

$$\frac{\Gamma \vdash \mu X.A \quad \Gamma \vdash \mu Y.B \quad \Gamma \cup \{Y <: Top, X <: Y\} \vdash A <: B}{\Gamma \vdash \mu X.A <: \mu Y.B} \quad (18)$$

$$\frac{\Gamma \vdash \mu X.A \quad \Gamma \vdash \mu Y.B \quad \Gamma \cup \{Y <: Top, X <: Top\} \vdash A <: B}{\Gamma \vdash \mu X.A <: \mu Y.B} \quad (19)$$