

## Prova scritta di Linguaggi - 31.01.2017

Si consideri il linguaggio *Lang*:

$$\begin{aligned}
 E \in \text{Lang} ::= & x \mid l \mid n \mid \{lab_1 = E_1, \dots, lab_k = E_k\} \mid \\
 & \text{ref } E \mid E_1 + E_2 \mid \#lab E \mid !E \mid E_1 := E_2 \mid \\
 & \text{let } x : T = E_1 \text{ in } E_2 \mid E_1; E_2 \mid \text{skip} \mid \\
 & \text{fun}(x : T) \Rightarrow E \mid E_1 E_2 \mid \text{fix}.E
 \end{aligned}$$

1. (3 punti) Formalizzare la *statica* del linguaggio *Lang* fornendo: i) una grammatica dei tipi ammessi nel linguaggio; ii) un sistema per il tipaggio; iii) un sistema per il sottotipaggio.
2. (9 punti) Dire, giustificando *formalmente* la risposta, se, in presenza di sottotipaggio, il termine

$$\begin{aligned}
 & \text{fun}(f : T_1) \Rightarrow \\
 & \quad (\text{fun}(g : \{c : \text{int}, d : \text{int}\}) \Rightarrow \#dg)((\text{fix}.f)(\{a = 5, b = 6\})) \\
 & \quad + \\
 & \quad (\text{fun}(g : \{d : \text{int}\}) \Rightarrow \#dg)((\text{fix}.f)(\{a = 7\}))
 \end{aligned}$$

è ben tipato o meno, al variare del tipo  $T_1$ .

3. (10 punti) Si consideri il seguente linguaggio concorrente:

$$\begin{aligned}
 E \in \text{Lang} ::= & l \mid \text{true} \mid \text{false} \mid !l \mid l := E \\
 & E_1 \parallel E_2 \mid E_1 \oplus E_2 \mid \text{await } E_1 \text{ protect } E_2 \text{ end}
 \end{aligned}$$

dove l'operatore  $\oplus$  denota la scelta non deterministica. Si fornisca type system e semantica operativa per **i soli costrutti concorrenti**. Provare **formalmente** se le seguenti leggi algebriche sono vere o false per  $E_1$  e  $E_2$  arbitrari. Se le leggi non dovessero valere, si fornisca un controesempio, e si mostri formalmente la simulazione è verificata in uno o entrambi i sensi.

- (a)  $\alpha; (\text{skip}; E_1 \oplus \text{skip}; E_2) \approx_{\Gamma} \text{skip}; (\alpha; E_1 \oplus \alpha; E_2)$   
dove  $\alpha$  è un assegnamento arbitrario
- (b)  $\text{await true protect } (\text{skip}; E_1 \oplus \text{skip}; E_2) \text{ end} \approx_{\Gamma}$   
 $(\text{await true protect } E_2 \text{ end}) \oplus (\text{await true protect } E_2 \text{ end})$
- (c)  $\alpha; (E_1 \oplus \text{skip}; E_2) \oplus \alpha; E_1 \approx_{\Gamma} \alpha; (\text{skip}; (E_1 \oplus \text{skip}; E_2)) \oplus (\alpha; E_2)$   
dove  $\alpha$  è un assegnamento arbitrario.

4. (9 punti) Si considerino i seguenti programmi da valutare nello store ini-

ziale  $s = \{1 \mapsto 2, m \mapsto 1\}$ , con una semantica CBN:

$$A \stackrel{\text{def}}{=} \text{let } F : T_F = (\text{fn } f:T \Rightarrow (\text{fn } x:\text{unit} \Rightarrow x \parallel (fx))) \\ \text{in } (\text{fix}.F)(m := !m - 1; l := !m + 1)$$

$$B \stackrel{\text{def}}{=} \text{let } F : T_F = (\text{fn } f:T \Rightarrow (\text{fn } x:\text{unit} \Rightarrow x; (fx))) \\ \text{in } (\text{fix}.F)(m := !m - 1; l := !m + 1)$$

$$C \stackrel{\text{def}}{=} \text{let } F : T_F = (\text{fn } f:T \Rightarrow (\text{fn } x:\text{unit} \Rightarrow (\text{await true protect } x \text{ end}) \parallel (fx))) \\ \text{in } (\text{fix}.F)(m := !m - 1; l := !m + 1)$$

- (a) Si dica per quali valori di  $T_F$  e  $T$  i programmi sono ben tipati.
- (b) Si considerino le possibili coppie di programmi:  $(A, B)$ ,  $(A, C)$  e  $(B, C)$ . Da un'analisi accurata delle tracce dei programmi  $A$ ,  $B$  e  $C$  si mostri, per ciascuna delle 3 coppie, se trattasi di programmi tracce equivalent. Nel caso di programmi tracce equivalent, si argomenti (informalmente) se vale a meno la simulazione, in uno o entrambi i versi, e la bisimulazione.