Prova scritta di Linguaggi - 20.09.2019 (4 esercizi in totale)

Si consideri il linguaggio *Lang*:

$$
\begin{array}{llllllllllll}
E \in Lang & ::= & x & | & l & | & n & | & \{lab_1 = E_1, \ldots lab_k = E_k\} & | \\
& & \mathsf{ref}\, E & | & E_1 \,+\, E_2 & | & E_1 \,\wedge\, E_2 & | & E_1 \,\geq\, E_2 & | & \#lab\, E & | \\
& & !E & | & E_1 := E_2 & | & \mathtt{let}\ x : T = E_1\ \mathtt{in}\ E_2 & | & E_1; E_2 & | \\
& & \mathsf{skip} & | & \mathsf{fun}(x:T) \Rightarrow E & | & E_1\, E_2 & | & \mathsf{fix}.E
\end{array}
$$

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** (dare *solo* le regole che userete nel prossimo esercizio).

2. (10 punti) Dire, giustificando *formalmente* la risposta, se, in presenza di sottotipaggio, il termine

$$
\begin{array}{l}
\mathsf{let}\ f : T_1 = (\mathsf{fun}(x:T_2) \Rightarrow \{a = \#a\,!x, b = (\#a\,!x) \geq 0, c = \#b\,!x\})\ \mathsf{in} \\
\Big((\mathsf{fun}(g : \{c : \mathsf{bool}\}) \Rightarrow \#c\, g)(f(\mathsf{ref}\,\{a = 5, b = \mathsf{true}, c = \mathsf{false}\})) \\
\qquad\qquad\qquad\qquad \wedge \\
(\mathsf{fun}(g : \{a : \mathsf{int}\,,\, b : \mathsf{bool}\}) \Rightarrow ((\#a\, g) \geq 0) \vee (\#b\, g))(f(\mathsf{ref}\,\{a = 7, b = \mathsf{false}, c = 8\}))\Big)
\end{array}
$$

   è ben tipato o meno, al variare deri tipi T_1 e T_2. Nel caso sia tipabile fornire il tipo dell'intera espressione. Usare *esclusivamente* le regole di sottotipaggio viste a lezione.

3. (12 punti) Si consideri il seguente linguaggio concorrente:

$$
\begin{array}{lllllllllll}
E \in Lang & ::= & l & | & \mathsf{true} & | & \mathsf{false} & | & !l & | & l := E \\
& & E_1 \,||\, E_2 & | & E_1 \oplus E_2 & | & \text{await } E_1 \text{ protect } E_2 \text{ end} & & & &
\end{array}
$$

   dove l'operatore $\oplus$ denota la scelta non deterministica. Supponendo di aver definito type system e semantica operazionale anche per i 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 se la simulazione è verificata in uno o entrambi i sensi.

   (a) $\alpha; (\mathsf{skip}; E_1 \parallel \alpha; E_2) \approx_\Gamma \mathsf{skip}; \alpha; (\alpha; E_1 \parallel \mathsf{skip}; E_2)$
   dove α è un arbitrario assegnamento semplice.

   (b) (await true protect $(\alpha; E_1 \oplus \beta; E_2)$ end)
   $\approx_\Gamma$ (β;await true protect E_1 end) $\parallel$ (α;await true protect E_2 end)
   dove α e β sono assegnamenti arbitari. Nel caso, in cui l'equivalenza non valga mostrare se esistono assegnamenti α e β, $\alpha \neq \beta$ tali che l'equivalenza valga.

(c) $\alpha; \alpha; (\mathsf{skip}; E_1 \oplus \beta; E_2) \oplus (\alpha; E_1) \approx_\Gamma \alpha; (\mathsf{skip}; (\alpha; E_1 \oplus \beta; E_2)) \oplus (\beta; E_2)$ dove α e β sono assegnamenti arbitari. Nel caso, in cui l'equivalenza non valga mostrare se esistono assegnamenti α e β, $\alpha \neq \beta$ tali che l'equivalenza valga.

4. (6 punti) Si consideri il seguente linguaggio:

$$E \in \mathit{Lang} \quad ::= \quad x \quad | \quad n \quad | \quad E_1 + E_2 \quad | \quad \mathsf{fun}(x : T) \Rightarrow E \quad | \quad E_1\, E_2$$

Se ne dia un sistema di tipi e una semantica operazionale in stile CBV. Dopodiché si provi formalmente il teorema di type preservation per il linguaggio in questione.