Prova scritta di Linguaggi - 26.06.2018

Si consideri il linguaggio *Lang*:

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

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

$$
\begin{array}{l}
\mathsf{let}\, f : T_1 = \mathsf{fun}(y : T_2) \Rightarrow \mathsf{fun}(x : T_3) \Rightarrow \{a = \mathsf{true}, b = \mathsf{false}, c = \mathsf{true}, d = \mathsf{true}\}\ \mathsf{in} \\
\Big(\big(\mathsf{let}\, g : \{a : \mathsf{bool}, b : \mathsf{bool}\} \;=\; (\mathsf{fix}\, f)(\{a = 5, b = 6\})\ \ \mathsf{in}\ \ \#b\, g\big) \\
\qquad\qquad\qquad\qquad \wedge \\
\big(\mathsf{let}\, g : \{a : \mathsf{bool}\,,\, c : \mathsf{bool}\} \;=\; (\mathsf{fix}\, f)(\{a = 7, c = 8\})\ \ \mathsf{in}\ \ \#a\, g \wedge \#c\, g\big)
\end{array}
$$

   è ben tipato o meno, al variare deri tipi T_1 e T_2. Usare *esclusivamente* le regole di sottotipaggio viste a lezione.

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

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

   dove l'operatore $\oplus$ denota la scelta non deterministica. Supponendo di aver definito type system e semantica operazionale anche per i costrutti concorrent. 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) $\alpha; (\beta;E_1 \parallel \beta;E_2) \approx_\Gamma \mathsf{skip}; \alpha; \beta; (E_1 \parallel E_2)$
   dove α e β sono assegnamenti arbitrari "semplici" ($l := 5$, $l := \mathsf{true}$ ma non $l := E$, con E espressione arbitraria) su locazioni diverse associate;

   (b) $\Big($(await true protect E_1 end); (await true protect E_2 end)
   $\oplus$
   (await true protect E_2 end); (await true protect E_1 end)$\Big)$
   $\approx_\Gamma$
   (await true protect E_1 end) $\parallel$ (await true protect E_2 end)

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

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

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

Se ne dia una semantica operazionale in stile CBV (ovviamente left-to-right). Si provi *formalmente* se vale la seguente proprietà:
Theorem. If $E \to F$ and $E \to G$ then $F = G$.
Indicare chiaramente che tecnica di prova si usa e poi, ovviamente, usarla in modo consistente.