Prova scritta di Linguaggi - 30.01.2020 (4 esercizi in totale)

Si consideri il linguaggio *Lang*:

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

e si assuma di aver definito per esso regole di tipaggio e di sottotipaggio.

1. (10 punti) Dire, giustificando *formalmente* la risposta, se, in presenza di sottotipaggio, il termine

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

   è ben tipato o meno, al variare dei tipi T_1 e T_2. Nel caso sia tipabile fornire l'albero di derivazione con il tipo dell'intera espressione e dei valori ben precisi per i tipi T_1 e T_2. Nel caso non fosse tipabile, indicare i punti di contraddizione nell'albero di derivazione costruito.

2. (3 punti) Dire, in presenza di sottotipaggio, se la variante

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

   è ben tipata o meno, al variare dei tipi T_1 e T_2. Nel caso sia tipabile fornire valori ben precisi per i tipi T_1 e T_2 e i punti in cui avviene il sottotipaggio. Nel caso non fosse tipabile indicare nel programma i punti di contraddizione nell'inferenza dei tipi (per questo esercizio non è richiesto l'albero di derivazione).

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

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

   dove l'operatore $\oplus$ denota la scelta non deterministica. Supponendo di aver definito type system e semantica operazionale anche per i costrutti 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 indicare se esistono assegnamenti α e β, $\alpha \neq \beta$ tali che l'equivalenza vale.

(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 indicare se esistono assegnamenti α e β, $\alpha \neq \beta$ tali che l'equivalenza vale.

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 **big-step** in stile **CBV**. Si provi *formalmente* se vale la seguente proprietà:
Theorem. If $E \Downarrow v$ and $E \Downarrow w$ then $v = w$.
Indicare chiaramente che tecnica di prova viene usata e poi, ovviamente, usarla in modo consistente.