Prova scritta di Linguaggi - 17.02.2020 (4 esercizi in totale)

Si consideri il linguaggio *Lang*:

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

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

$$
\begin{array}{l}
\mathsf{let}\, f : T_1 = (\mathsf{fun}(x : T_2) \Rightarrow \mathsf{ref}\,\{a = \#c\,\cancel{!}x, b = (\#c\,\cancel{!}x) \geq 0, c = \#b\,\cancel{!}x\})\ \ \mathsf{in} \\
\quad \Big((\mathsf{fun}(g : \mathsf{ref}\,\{c : \mathsf{bool}\}) \Rightarrow \#c\,!g\big)\big(\mathsf{ref}\,(!(f\{a = \mathsf{false}, b = \mathsf{true}, c = 2\})))\big) \\
\qquad\qquad\qquad\qquad \wedge \\
\quad \big(\mathsf{fun}(g : \mathsf{ref}\,\{a : \mathsf{int}\}) \Rightarrow ((\#a\,!g) \geq 0)\big)\big(\mathsf{ref}\,(!(f\{a = 7, b = \mathsf{false}, c = 8\})))\big) \Big)
\end{array}
$$

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

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

   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) $\alpha; (\mathsf{skip}; E_1 \parallel \beta; E_2) \approx_\Gamma \mathsf{skip}; \alpha; (\beta; E_1 \parallel \mathsf{skip}; E_2)$
   dove α e β sono *assegnamenti semplici arbitrari.* Nel caso, in cui l'equivalenza non valga indicare se esistono assegnamenti semplici α e β, $\alpha \neq \beta$, per cui la bisimulazione/simulazione vale. In tal caso, *dimostrare formalmente* la simulazione/bisimulazione esistente.

   (b) (await true protect $(\gamma; \alpha; E_1 \oplus \gamma; \beta; E_2)$ end)
   $\approx_\Gamma$ (await true protect $\beta;\alpha;E_1$ end) $\oplus$ (await true protect $\alpha;\beta;E_2$ end)
   dove α, β e γ sono *assegnamenti semplici arbitrari.* Nel caso, in cui l'equivalenza non valga indicare se esistono assegnamenti semplici α, β e γ, tutti distinti tra loro, per cui la bisimulazione/simulazione vale. In tal caso, *dimostrare formalmente* la simulazione/bisimulazione esistente.

(c) $\alpha; (\mathsf{skip}; (\alpha; E_1 \oplus \beta; E_2)) \oplus (\beta; E_2) \;\; \approx_\Gamma \;\; \alpha; (\mathsf{skip}; E_1 \oplus \beta; E_2) \oplus \alpha; E_1$
dove α e β sono *assegnamenti semplici arbitrari*. Nel caso, in cui l'equivalenza non valga indicare se esistono assegnamenti α e β, $\alpha \neq \beta$ per cui la bisimulazione/simulazione vale. In tal caso, *dimostrare formalmente* la simulazione/bisimulazione esistente.

3. (9 punti) Supponendo che i tre termini siano ben tipati, si considerino i seguenti programmi da valutare nello store iniziale $s = \{\mathrm{l} \mapsto 3, \mathrm{m} \mapsto 2\}$, con una semantica CBN:

$$A \stackrel{\text{def}}{=} \begin{array}{l} \text{let } F : T_{\mathrm{F}} = \big(\text{fn } f{:}T \Rightarrow (\text{fn } x{:}\mathsf{unit} \Rightarrow x \,||\, (f x))\big) \\ \text{in } (\mathsf{fix}.F)(\mathrm{m} := !\mathrm{l} - 2; \mathrm{l} := !\mathrm{m} + 3) \end{array}$$

$$B \stackrel{\text{def}}{=} \begin{array}{l} \text{let } F : T_{\mathrm{F}} = \big(\text{fn } f{:}T \Rightarrow (\text{fn } x{:}\mathsf{unit} \Rightarrow x; (f x))\big) \\ \text{in } (\mathsf{fix}.F)(\mathrm{m} := !\mathrm{l} - 2; \mathrm{l} := !\mathrm{m} + 3) \end{array}$$

$$C \stackrel{\text{def}}{=} \begin{array}{l} \text{let } F : T_{\mathrm{F}} = \big(\text{fn } f{:}T \Rightarrow (\text{fn } x{:}\mathsf{unit} \Rightarrow (\mathsf{await\ true}\ \text{protect } x \text{ end}) \,||\, (f x))\big) \\ \text{in } (\mathsf{fix}.F)(\mathrm{m} := !\mathrm{l} - 2; \mathrm{l} := !\mathrm{m} + 3) \end{array}$$

(a) Supponendo che i tre programmi siano tutti ben tipati. Si argomenti, attraverso uno studio sulla memoria associata alle tracce di esecuzione, se valgono le seguenti relazioni:

- $A \sqsubseteq B$
- $B \sqsubseteq A$
- $A \sqsubseteq C$
- $C \sqsubseteq A$
- $B \sqsubseteq C$
- $C \sqsubseteq B$.

Per quei casi in cui si ritiene valga la relazione si indichi, argomentando la risposta, l'invariante che lega i volori associati alle due locazioni.