# Prova scritta di Linguaggi - 04.02.2019 (4 esercizi in totale)

Si consideri il linguaggio *Lang*:

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

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

$$\begin{array}{c} \mathsf{let}\, f : T_1 = \mathsf{fun}(x : T_2) \Rightarrow \mathsf{ref}\{p = \#a\, x, q = (\#b\, x) \geq 0, r = \#c\, x\} \;\mathsf{in} \\ \Big( \big(\mathsf{fun}(g : \mathsf{ref}\{p : \mathsf{int}, r : \mathsf{int},\}) \Rightarrow \#q\, g\big)\big(f\{a = 5, b = 6, c = 7\}\big) \\ \wedge \\ \big(\mathsf{fun}(g : \mathsf{ref}\{p : \mathsf{int}\,,\, r : \mathsf{int}\}) \Rightarrow (\#p\, g) \;\leq\; (\#r\, g)\big)\big(f\{a = 7, b = 8\}\big) \Big) \end{array}$$

   è ben tipato o meno, al variare dei 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:

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

   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) $\alpha; (\beta; E_1 \parallel \beta; E_2) \approx_\Gamma \mathsf{skip}; \alpha; \beta; (E_1 \parallel E_2)$
   dove $\alpha$ e $\beta$ sono arbitrari assegnamenti semplici.

   (b) (await $\mathsf{true}$ protect $(E_1 \parallel E_2)$ end)
   $\approx_\Gamma$ (await $\mathsf{true}$ protect $E_2$ end) $\parallel$ (await $\mathsf{true}$ protect $E_1$ end)

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

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

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

Se ne dia una semantica operazionale in stile CBV (ovviamente left-to-right). Si riporti formalmente la proprietà di "Type preservation" per il linguaggio e si fornisca una prova formale di tale proprietà. Nel caso la proprietà non dovesse valere si fornisca un controesempio.