A.A. 2000-01

Linguaggi di Programmazione I: Paradigmi
Prova scritta del 21 febbraio 2001

Esercizio 1. Una sostituzione di tipi di ML è una funzione *totale* V che mappa variabili in tipi primitivi e tale che solo per un numero finito di variabili X si ha $V(X) \neq X$. Se σ è uno schema, $V(\sigma)$ è il tipo ottenuto sostituendo in σ ogni occorrenza libera di ciascuna variabile X con $V(X)$. La composizione di due sostituzioni V e W è la sostituzione $W \circ V$ che mappa X in $W(V(X))$. Scriviamo $V < W$ se esiste una sostituzione U tale che $U \circ V = W$. Trovare, se esistono, due sostituzioni *diverse* V e W tali che $V < W$ e $W < V$. Altrimenti dimostrare che tali sostituzioni non possono esistere.

Esercizio 2. Trovare nel sistema dei tipi di ML un contesto Γ in cui il termine $x(x(x))$ sia tipabile. Inferire il suo tipo mostrando l'albero di derivazione *oppure* applicando l'algoritmo W.

Esercizio 3. Sia t una espressione di tipo T in Java, sia $m(x)$ un metodo con parametro formale x di tipo T_1 ed $n(y)$ uno con parametro formale y di tipo T_2. E' possibile che T_1 e T_2 siano tali che né T_1 è sottotipo di T_2 né T_2 è sottotipo di T_1, eppure risultino legali tanto la chiamata $m(t)$ che $n(t)$? Motivare la risposta (in caso affermativo fornendo un esempio).

Esercizio 4. Definire in Modula2 un modulo che implementi un tipo di dato astratto YIN-YANG sul quale siano definite una procedura di inizializzazione *init* ed una procedura *switch*: una variabile x di tipo YIN-YANG può essere inizializzata invocando $init(x)$ ed il suo valore può essere cambiato nel suo "opposto" con $switch(x)$. Due invocazioni consecutive di $switch(x)$ riportano x nello stesso stato. L'implementazione deve rimanere nascosta.