Linguaggi di Programmazione
Esame – 22 Giugno 2022

Domanda

Qual e' la formula per esprimere l'equivalenza nella semantica denotazionale?

Esercizio 1

Consideriamo il programma IMP

$$w \stackrel{\text{def}}{=} \textbf{while } \neg(x = 0) \vee \neg(y = 0) \textbf{ do } (x := x - 1; y := y - 1)$$

Definire l'insieme delle memorie $T = \{\sigma \mid ...\}$ per le quali il programma w termina e:

1. provare formalmente che per ogni memoria $\sigma \in T$ abbiamo che $\langle w, \sigma\rangle \to \sigma[0/x, 0/y]$.
   (Suggerimento: usare l'induzione ben fondata su T)
2. provare formalmente (usando il metodo visto a lezione) che $\langle w, \sigma\rangle \not\to$ per ogni memoria $\sigma \notin T$.

Esercizio 2

1. Definire in FSharp il tipo `Nat` per i numeri naturali che usi il caso base "zero" e l'operazione unaria "successore".
2. Definire ricorsivamente la somma fra due numeri naturali (hint: bastano due casi).
3. Dare un esempio (corto).
4. Definire una funzione che dato un intero di FSharp (cioè un valore di tipo `int`) restituisce un valore di tipo `Nat` che rappresenta lo stesso numero se positivo, e fallisce altrimenti.

Esercizio 3

Consideriamo il termine HOFL:

$$t \stackrel{\text{def}}{=} \textbf{rec } f.\lambda x.\ \lambda y.\ \textbf{if } x \times y \textbf{ then } x \textbf{ else } (fx)((fx)y)$$

Derivare il tipo, la forma canonica e la semantica denotazionale di t.

Esercizio 4

Estendere l'interprete dato a lezione con il costrutto `CIterate(ide,pseq,expr1,expr2)` che prende un identificatore, una sequenza di comandi, e due espressioni.

In questo costrutto, `ide` deve denotare una variabile, `expr1` e `expr2` devono dare come risultato rispettivamente `i1` e `i2` di tipo intero. Se una di queste condizioni non vale, si deve restituire un errore.

Se le condizioni sono rispettate, la semantica intesa è la seguente: per ogni valore `x` compreso nell'intervallo chiuso `[i1,i2]`, alla variabile `ide` viene assegnato il valore `x` e in queste condizioni viene eseguita la sequenza `pseq`.

Alla fine dell'esecuzione, prima di eseguire il resto del programma, la variabile `ide` conserva il valore che ha assunto dopo l'ultima esecuzione di `pseq`. Le modifiche allo stato vanno preservate nello stato di uscita, ma non le modifiche all'ambiente (usiamo il binding statico). Opzionale: fare in modo che le locazioni eventualmente allocate in `pseq` siano cancellate quando non sono più utili.

Se l'intervallo chiuso `[i1,i2]` è vuoto, non si deve modificare lo stato (quindi la variabile `ide` conserva il suo valore iniziale).

1. Si fornisca il caso aggiuntivo della funzione di valutazione semantica dei comandi, che ha il seguente tipo e struttura:

```
let rec esem: exp -> env -> store -> eval =
    fun e ev st ->
        match e with
        | Eint i ->
        ...

and csem: com -> env -> store -> (env * store) =
    fun c ev st ->
        match c with
        | Cassign (i, e) ->
        ...

and pssem: pseq -> env -> store -> (env * store) =
    fun s ev st ->
    ...

and ...
```

2. Si fornisca un semplice esempio in sintassi astratta e lo stato finale.

Esercizio 5

Consideriamo i seguenti processi CCS

$$p \stackrel{\text{def}}{=} \textbf{rec}\ X.\ a.(b.b'.X\ +\ c.c'.X) \qquad r \stackrel{\text{def}}{=} \textbf{rec}\ Z.\ (\overline{b}.Z + \overline{c}.Z)$$
$$q \stackrel{\text{def}}{=} \textbf{rec}\ Y.\ a.b.(b'.Y\ +\ c'.Y) \qquad s \stackrel{\text{def}}{=} (\textbf{rec}\ V.\ \overline{b}.V)\ |\ (\textbf{rec}\ W.\ \overline{c}.W)$$

1. I processi r e s sono fortemente bisimili?
2. Disegnare gli LTS dei processi $(p|r)\backslash b\backslash c$ e $(q|s)\backslash b\backslash c$.
3. I processi $(p|r)\backslash b\backslash c$ e $(q|s)\backslash b\backslash c$ sono debolmente bisimili?