

Linguaggi di Programmazione

Simulazione di prova d'esame

Domanda

Discutere brevemente la differenza tra semantica operativa small e big steps.

Esercizio 1

1. Dimostrare che, secondo la semantica operativa di IMP, per ogni espressione booleana b , comando c e memorie σ, σ'

$$\langle \mathbf{while} \ b \ \mathbf{do} \ c, \sigma \rangle \rightarrow \sigma' \text{ implies } \langle b, \sigma' \rangle \rightarrow \mathit{false}$$

Spiegate quale principio di induzione sfruttate nella prova.

2. Usare l'induzione strutturale per dimostrare che un'espressione aritmetica a ha lo stesso valore se valutata su due stati che concordano sulle variabili libere presenti in a .

Esercizio 2

Sia t il termine HOFL

$$t \stackrel{\text{def}}{=} \mathbf{fst}(\ (\lambda x.x) \ (\ 1 \ , \ ((\mathbf{rec}f. \ \lambda y.(f \ y)) \ 2) \) \) \).$$

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

Esercizio 3

1. Consideriamo i processi CCS

$$p \stackrel{\text{def}}{=} \mathbf{rec} \ x. \ (a.x + a.\mathbf{nil}) \quad q \stackrel{\text{def}}{=} \mathbf{rec} \ y. \ (a.a.y + a.\mathbf{nil})$$

Disegnare l'LTS per p e q e dimostrare che $p \not\sim q$ mostrando una formula in HM-logica che distingue tra i due. (Suggerimento: sfruttare la sottoformula $[a]\mathit{false}$ e spiegarne il significato).

2. Consideriamo i processi CCS

$$\begin{aligned} r &\stackrel{\text{def}}{=} a.(b.c.\mathbf{nil} + b.\tau.c.\mathbf{nil} + \tau.b.\mathbf{nil} + b.\mathbf{nil}) \\ s &\stackrel{\text{def}}{=} a.(b.c.\tau.\mathbf{nil} + \tau.b.\mathbf{nil}) + a.b.\mathbf{nil} \end{aligned}$$

Disegnare l'LTS per r e s e dimostrare che sono debolmente congruenti sfruttando le tecniche presentate nel corso.

Esercizio 4

1. Definire il tipo `Pila<'a>` in fsharp, vale a dire una struttura dati in cui si può inserire ed estrarre un elemento alla volta.
2. Definire le operazioni (funzioni, non ricorsive) `push(x,pila)` che inserisce `x` e restituisce la pila modificata, e `pop(pila)` che estrae un elemento e restituisce l'elemento estratto, e la pila modificata. L'ordine di estrazione è "last in, first out", vale a dire l'ultimo elemento inserito deve essere il primo estratto (e così via ricorsivamente).
3. Definire una operazione in più, tramite una funzione ricorsiva, che restituisce l'elemento massimo di una pila.

Esercizio 5

1. Aggiungere all'interprete sviluppato durante il corso, partendo dalla sintassi astratta data nel prontuario, il **comando** `CFor of ide * expr * expr * psseq` con la seguente interpretazione intesa: `CFor(i,e1,e2,body)` richiede che `e1` e `e2` siano espressioni di tipo intero con valore `n` e `m` rispettivamente. Se questo è vero, il corpo (sequenza di comandi `body`) viene valutato `m-n` volte. Alla `k`-esima iterazione (iniziando a contare da zero), `body` deve essere valutato in un ambiente in cui all'identificatore `i` è associato il valore `n+k`.
2. Esibire un programma di esempio in cui si vede come viene usato l'identificatore passato come primo argomento a `CFor`.