Linguaggi di programmazione: semantica

Esempio di esame 1

1

Dato il frammento di programma:

```
var y : int = 0;
var x : int = 1;
procedure p( var x : int )
    var x : int = 2;
    y := x + 1;
p(4);
```

dire quale e' il valore di y dopo la chiamata di procedura. Motivare la risposta con le rilevanti derivazioni.

1.1 Valutazioni preliminari

Per prima cosa riscrivo il frammento di programma assegnado dei nomi alle singole parti di codice.

d_y	var y : int = 0;
d_x^1	var x : int = 1;
d_p	procedure p(var x : int)
d_x^2	var x : int = 2;
c	y := x + 1;
	p(4);

Il frammento di codice puo' essere quindi scritto come: d_y; d_x^1; d_p; p(4)

1.2 Semantica statica

Verifico la correttezza statica del programma.

1. $$\frac{\emptyset \vdash 0 : int}{\emptyset \vdash d_y : [y = intloc]}$$
   $\Delta_y = [y = intloc]$

2. $$\frac{\Delta_y \vdash 1 : int}{\Delta_y \vdash d_x^1 : [x = intloc]}$$
   $\Delta_x = [x = intloc]$

3. $$\frac{\dfrac{\Delta_y[\Delta_x][\Delta_{form}] \vdash 2 : int}{\Delta_y[\Delta_x][\Delta_{form}] \vdash d_x^2 : [x = intloc]} \quad \dfrac{\dfrac{\Delta_y[\Delta_x][\Delta_{form}][x = intloc] \vdash 1 : int \quad \Delta_y[\Delta_x][\Delta_{form}][x = intloc] \vdash x : int}{\Delta_y[\Delta_x][\Delta_{form}][x = intloc] \vdash x + 1 : \tau_+(int, int) = int}}{\Delta_y[\Delta_x][\Delta_{form}][x = intloc] \vdash c}}{\Delta_y[\Delta_x][\Delta_{form}] \vdash d_x^2; c}$$

   $$\frac{\dfrac{\bullet : \emptyset}{\text{var x : int}, \bullet : \emptyset[x = intloc] = \Delta_{form}} \quad \Delta_y[\Delta_x][\Delta_{form}] \vdash d_x^2; c}{\Delta_y[\Delta_x] \vdash d_p : [p = (int, \bullet)proc]}$$
   $\Delta_p = [p = (int, \bullet)proc]$

4. $$\dfrac{\dfrac{\dfrac{\emptyset \vdash d_y : \Delta_y(\textit{per 1.}) \quad \emptyset[\Delta_y] \vdash d_x^1 : \Delta_x(\textit{per 2.})}{\emptyset \vdash d_y; d_x^1 : \Delta_y[\Delta_x]} \quad \emptyset[\Delta_y][\Delta_x] \vdash d_p : \Delta_p(\textit{per 3.})}{\emptyset \vdash d_y; d_x^1; d_p : [\Delta_y][\Delta_x][\Delta_p]} \quad \dfrac{\dfrac{\emptyset[\Delta_y][\Delta_x][\Delta_p] \vdash \bullet : \bullet}{\emptyset[\Delta_y][\Delta_x][\Delta_p] \vdash 4, \bullet : \text{int}, \bullet}}{\emptyset[\Delta_y][\Delta_x][\Delta_p] \vdash \text{p}(4)} *}{\emptyset \vdash d_y; d_x^1; d_p; \text{p}(4)}$$

$* = \emptyset[\Delta_y][\Delta_x][\Delta_p](p) = (\text{int}, \bullet)\text{proc}$

Questo dimostra che il programma $d_y; d_x^1; d_p;$ p(4) e' staticamente corretto.

1.3 Semantica dinamica -scoping dinamico-

Assumiamo per semplicita' e ove non vi siano problemi di interpretazione che l'ambiente $\rho_1[\rho_2]$ possa essere scritto come $\rho_1\rho_2$.

Costruiamo le derivazioni per il frammento di codice dato.

$$\begin{array}{ll}
 & < d_y; d_x^1; d_p; \text{p}(4), \emptyset > \\
\xrightarrow{(1)} & < \rho_y = [y = l_y^1]; d_x^1; d_p; \text{p}(4), [l_y^1 = 0] > \\
\xrightarrow{(2)} & < \rho_y; \rho_x^1 = [x = l_x^1]; d_p; \text{p}(4), [l_y^1 = 0, l_x^1 = 1] > \\
\xrightarrow{(3)} & < \rho_y\rho_x^1; d_p; \text{p}(4), [l_y^1 = 0, l_x^1 = 1] > \\
\xrightarrow{(4)} & < \rho_y\rho_x^1; \rho_p = [p = \lambda(\text{var x : int}, \bullet).d_x^2; c]; \text{p}(4), [l_y^1 = 0, l_x^1 = 1] > \\
\xrightarrow{(5)} & < \rho_y\rho_x^1\rho_p; \text{p}(4), [l_y^1 = 0, l_x^1 = 1] > \\
\xrightarrow{(6)} & < \rho_y\rho_x^1\rho_p; (\text{var x : int}, \bullet) = (4, \bullet); d_x^2; c, [l_y^1 = 0, l_x^1 = 1] > \\
\xrightarrow{(7)} & < \rho_y\rho_x^1\rho_p; \rho_f = [x = l_x^f]; d_x^2; c, [l_y^1 = 0, l_x^1 = 1, l_x^f = 4] > \\
\xrightarrow{(8)} & < \rho_y\rho_x^1\rho_p\rho_f; d_x^2; c, [l_y^1 = 0, l_x^1 = 1, l_x^f = 4] > \\
\xrightarrow{(9)} & < \rho_y\rho_x^1\rho_p\rho_f; \rho_x^2 = [x = l_x^2]; c, [l_y^1 = 0, l_x^1 = 1, l_x^f = 4] > \\
\xrightarrow{(10)} & < \rho_y\rho_x^1\rho_p\rho_f\rho_x^2; c, [l_y^1 = 0, l_x^1 = 1, l_x^f = 4, l_x^2 = 2] > \\
\xrightarrow{(11)} & [l_y^1 = 3, l_x^1 = 1, l_x^f = 4, l_x^2 = 2]
\end{array}$$

Evidenziamo alcuni passaggi importanti:

(4) elaboriamo la dichiarazione della procedura p.

$$\dfrac{\dfrac{\rho_y\rho_x^1 \vdash < d_p, [l_y^1 = 0, l_x^1 = 1] > \rightarrow_d < \rho_p = [p = \lambda(\text{var x : int}, \bullet).d_x^2; c], [l_y^1 = 0, l_x^1 = 1] >}{\rho_y\rho_x^1 \vdash < d_p; \text{p}(4), [l_y^1 = 0, l_x^1 = 1] > \rightarrow_c < \rho_p; \text{p}(4), [l_y^1 = 0, l_x^1 = 1] >}}{\emptyset \vdash < \rho_y\rho_x^1; d_p; \text{p}(4), [l_y^1 = 0, l_x^1 = 1] > \rightarrow_c < \rho_y\rho_x^1; \rho_p; \text{p}(4), [l_y^1 = 0, l_x^1 = 1] >}$$

(6) elaboriamo la chiamata alla procedura p.

$$\dfrac{\rho_y\rho_x^1\rho_p \vdash < \text{p}(4), [l_y^1 = 0, l_x^1 = 1] > \rightarrow_c < (\text{var x : int}, \bullet) = (4, \bullet); d_x^2; c, [l_y^1 = 0, l_x^1 = 1] >}{\emptyset \vdash < \rho_y\rho_x^1\rho_p; \text{p}(4), [l_y^1 = 0, l_x^1 = 1] > \rightarrow_c < \rho_y\rho_x^1\rho_p; (\text{var x : int}, \bullet) = (4, \bullet); d_x^2; c, [l_y^1 = 0, l_x^1 = 1] >}$$

(7) elaboriamo la creazione dell'ambiente dei parametri formali.

$$\dfrac{\dfrac{\dfrac{\dfrac{\bullet, \{l_x^f\} \vdash \bullet : \emptyset, \emptyset}{(4, \bullet), \emptyset \vdash \text{var x : int}, \bullet : [x = l_x^f], [l_x^f 4]}}{\rho_y\rho_x^1\rho_p \vdash < (\text{var x : int}, \bullet) = (4, \bullet), [l_y^1 = 0, l_x^1 = 1] > \rightarrow_d < [x = l_x^f], [l_y^1 = 0, l_x^1 = 1, l_x^f 4] >}}{\rho_y\rho_x^1\rho_p \vdash < (\text{var x : int}, \bullet) = (4, \bullet); d_x^2; c, [l_y^1 = 0, l_x^1 = 1] > \rightarrow_d < [x = l_x^f]; d_x^2; c, [l_y^1 = 0, l_x^1 = 1, l_x^f 4] >}}{\emptyset \vdash < \rho_y\rho_x^1\rho_p; (\text{var x : int}, \bullet) = (4, \bullet); d_x^2; c, [l_y^1 = 0, l_x^1 = 1] > \rightarrow_d < \rho_y\rho_x^1\rho_p; \rho_f = [x = l_x^f]; d_x^2; c, [l_y^1 = 0, l_x^1 = 1, l_x^f 4] >}$$

(11) elaboriamo l'esecuzione del comando c nel corpo della procedura p. Al termine di questo comando nella locazione di memoria l_y^1 posso leggere il valore per la y.

$$\dfrac{\dfrac{\rho_y\rho_x^1\rho_p\rho_f\rho_x^2 \vdash < x + 1, [l_y^1 = 0, l_x^1 = 1, l_x^f = 4, l_x^2 = 2] > \rightarrow_e^* < 3, [l_y^1 = 0, l_x^1 = 1, l_x^f = 4, l_x^2 = 2] >}{\rho_y\rho_x^1\rho_p\rho_f\rho_x^2 \vdash < c, [l_y^1 = 0, l_x^1 = 1, l_x^f = 4, l_x^2 = 2] > \rightarrow_c [l_y^1 = 3, l_x^1 = 1, l_x^f = 4, l_x^2 = 2]}}{\emptyset \vdash < \rho_y\rho_x^1\rho_p\rho_f\rho_x^2; c, [l_y^1 = 0, l_x^1 = 1, l_x^f = 4, l_x^2 = 2] > \rightarrow_c [l_y^1 = 3, l_x^1 = 1, l_x^f = 4, l_x^2 = 2]}$$

Otteniamo, nel caso di scoping dinamico, un valore per la variabile y pari a 3.

1.4 Semantica dinamica -scoping statico-

Nel caso di scoping statico abbiamo che il passaggio (4) deve essere modificato per salvare l'ambiente al momento della dichiarazione. In particolare, viene salvato l'ambiente ρ_y.

$$\begin{array}{ll} & < \rho_y \rho_x^1; d_p; \mathrm{p}(4), [l_y^1 = 0, l_x^1 = 1] > \\ \xrightarrow{(4)} & < \rho_y \rho_x^1; \rho_p = [p = \lambda(\text{var x : int}, \bullet).\rho_{\mathbf{y}}; d_x^2; c]; \mathrm{p}(4), [l_y^1 = 0, l_x^1 = 1] > \end{array}$$

Questa modfica diventa importante nel passaggio (11) quando, per eseguire l'assegnamento di valore alla variabile y dobbiamo cercare a quale locazione di memoria fa riferimento y.

$$\begin{array}{ll} & < \rho_y \rho_x^1 \rho_p \rho_f \rho_y \rho_x^2; c, [l_y^1 = 0, l_x^1 = 1, l_x^f = 4, l_x^2 = 2] > \\ \xrightarrow{(11)} & [l_y^1 = 3, l_x^1 = 1, l_x^f = 4, l_x^2 = 2] \end{array}$$

Si puo' notare come anche in questo caso y faccia riferimento alla locazione l_y^1. Anche nel caso di scoping statico il valore della y e' pari a 3.

2

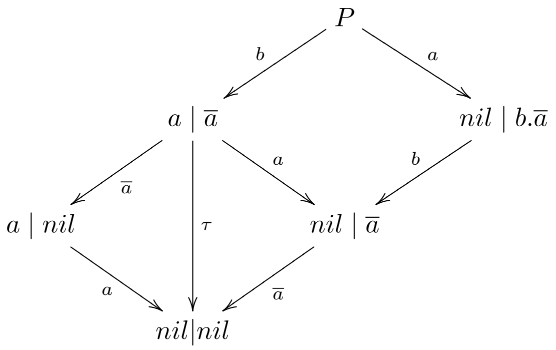
(NOTA: Per comodita' omettiamo il nil alla fine del processo, ove non crei confusione.)
Riscrivere il termine CCS

$$P = a \mid b.\overline{a}$$

senza usare l'operatore di composizione parallela e in modo da ottenere da entrambi i termini lo stesso sistema di transizione. Motivare la risposta.

2.1

Costruisco il sitema di transizione del processo P.

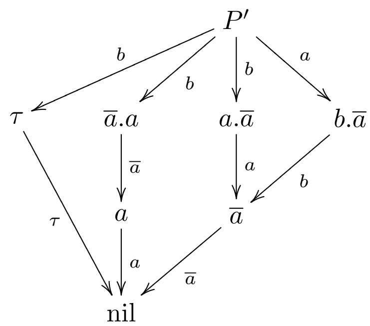


2.2

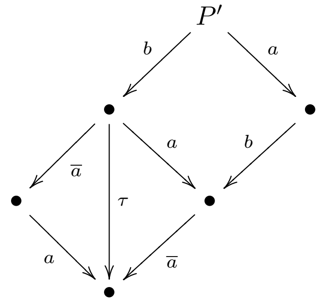
Uso una proprieta' dei sistemi di transizione tale per cui il sistema di transizione di $a \mid \overline{a}$ e' uguale al sistema di transizione del processo $a.\overline{a} + \overline{a}.a + \tau$.
Costruisco il processo P' che catturi tutte le tracce del sistema di transizione di P.

$$\begin{array}{lll} P' = & & b.\overline{a}.a \\ & + & b.\tau \\ & + & b.a.\overline{a} \\ & + & a.b.\overline{a} \end{array}$$

Se si va a costruire il sistema di transizione di P' si ottiene un sistema di transizione diverso da quello di P.



Per risolvere l'esercizio e' sufficiente non considere i nodi come etichettati con i processi. In questo modo le tre transizione etichettate con b collassano in una sola, in quanto i nodi di arrivo non sono piu' distinguibili. Quello che risulta e' il seguente:



3

Scrivere e discutere le regole di ricorsione in FUN.

3.1

Per risolvere l'esercizione e' necessario presentare la semantica statica delle dichiarazioni rec. In, particolare si deve discutere il perche' sono stati introdotti due gruppi di regole (ambienti e validita').

In seguito riportare le regole della tabella 8.6 del libro. Il commento e' scritto a pagina 146.