Fondamenti dei linguaggi di programmazione

Aniello Murano
Università degli Studi di Napoli
“Federico II”

Esercitazione sulle semantiche operazionale e denotazionale di IMP

Esercizio 1

- Scrivere la semantica operazionale del seguente comando:

while b_1 do c exit b_2

- con la seguente semantica intuitiva:
  "la semantica standard del costrutto while di IMP viene arricchita controllando alla fine di ogni iterazione di c se la condizione b_2 è soddisfatta. Se b_2 è soddisfatta, il ciclo viene interrotto".

Esercizio 1 - Soluzione

- Sia $w \equiv$ while b_1 do c exit b_2.
- Vogliamo definire la semantica operazionale (regole di derivazione) di w rispetto ad ogni stato $\sigma \in \Sigma$ ($\langle w,\sigma\rangle$).
- Per definire le regole di derivazione di $\langle w,\sigma\rangle$, dobbiamo distinguere i seguenti casi

1. b_1 è valutata "false" in σ, secondo le regole della semantica operazionale di IMP. Allora il comando c non viene eseguito e il ciclo viene interrotto. Dunque, $\langle w,\sigma\rangle$ viene derivata in σ.
2. b_1 è valutata "true" in σ, secondo le regole della semantica operazionale di IMP. Allora il comando c viene eseguito. Sia σ' lo stato raggiunto dopo l'esecuzione di c, dato dalle valutazione di $\langle c,\sigma\rangle$, secondo le regole della semantica operazionale di IMP. Allora dobbiamo distinguere i seguenti due casi:
   1. b_2 è valutata "true" in σ, secondo le regole della semantica operazionale di IMP. Allora il ciclo viene interrotto dopo l'esecuzione di c. Dunque, $\langle w,\sigma\rangle$ viene derivata in σ'.
   2. b_2 è valutata "false" in σ, secondo le regole della semantica operazionale di IMP. Allora viene eseguito nuovamente il ciclo a partire dallo stato σ. Dunque, la valutazione di w, in σ dipende dalla valutazione di w in σ'

Esercizio 1 - Regole

$$\frac{\langle b_1 , \sigma\rangle \to false}{\langle w , \sigma\rangle \to \sigma.}$$

$$\frac{\langle b_1 , \sigma\rangle \to true,\ \langle c , \sigma\rangle \to \sigma' , \langle b_2 , \sigma'\rangle \to true}{\langle w , \sigma\rangle \to \sigma'}$$

$$\frac{\langle b_1 , \sigma\rangle \to true,\ \langle c , \sigma\rangle \to \sigma' , \langle b_2 , \sigma'\rangle \to false, \langle w, \sigma'\rangle \to \sigma''}{\langle w , \sigma\rangle \to \sigma''}$$

Esercizio 2

- Dimostrare l'equivalenza semantica di

  while b do c exit b

  definito nell'esercizio 1, con il costrutto di IMP

  if b then c else skip

- Sia w ≡ while b do c exit b e if ≡ if b then c else skip, vogliamo dimostrare in base alla struttura delle regole di derivazione di IMP che

$$\forall \sigma\ \sigma', \langle w, \sigma\rangle \to \sigma' \Leftrightarrow \langle if, \sigma\rangle \to \sigma'$$

Esercizio 2 – Soluzione

- Per dimostrare l'equivalenza di w $\equiv$ while b do c exit b con if $\equiv$ if b then c else skip, in qualsiasi stato σ, occorre distinguere i seguenti casi:
- b è valutata "false" in σ. In questo caso w e if sono valutati σ.
- b è valutata "true" in σ. Allora occorre eseguire c nello stato attuale σ. Sia σ' lo stato raggiunto dopo l'esecuzione di c in σ. A questo punto bisogna distinguere le seguenti possibilità:
  1. b è valutata "true" in σ'. Allora il ciclo viene interrotto dopo l'esecuzione di c. Dunque, $\langle w,\sigma\rangle$ viene derivata in σ'. Si noti che nel comando if, comunque sia valutato b in σ', esso viene interrotto dopo l'esecuzione di c e dunque in questo caso $\langle w,\sigma\rangle$ è valutato σ' sse $\langle if,\sigma\rangle$ è valutato σ'.
  2. b è valutata "false" in σ'. Allora dopo il comando c viene eseguito nuovamente il comando w nello stato σ'. Ma adesso già sappiamo che b è valutata "false" in σ', e dunque il ciclo while viene interrotto senza eseguire ulteriormente c, proprio come nel comando if. Dunque, anche in questo caso $\langle w,\sigma\rangle$ è valutato σ' sse $\langle if,\sigma\rangle$ è valutato σ'.

Esercizio 3

- Dimostrare l'equivalenza semantica di

  $$w_1 \equiv \textbf{while b do c exit } \neg b$$

  definito nell'esercizio 1, con il costrutto di IMP

  $$w_2 \equiv \textbf{while b do c}$$

- Formalmente, vogliamo dimostrare che

$$\forall\sigma\ \sigma',\ \langle w_1,\ \sigma\rangle \to \sigma' \Leftrightarrow \langle w_2,\ \sigma\rangle \to \sigma'$$

- Per provare questa asserzione, utilizziamo una dimostrazione per <u>induzione sulle sottoderivazioni proprie</u> fra le derivazioni per l'esecuzione dei comandi
- Dunque, provare l'enunciato precedente equivale a provare che per ogni derivazione d, la proprietà P(d) seguente

$$P(d) \Leftrightarrow \forall\sigma,\sigma',\ \mathbf{d} \Vdash \langle w_1,\sigma\rangle \to \sigma' \ \&\ \Vdash \langle w_2,\sigma\rangle \to \sigma' \Rightarrow \sigma=\sigma'$$

Esercizio 3 – Soluzione

$P(d) \Leftrightarrow \forall \sigma,\sigma'$, $d \Vdash \langle w_1,\sigma\rangle \rightarrow \sigma'$ & $\Vdash \langle w_2,\sigma\rangle \rightarrow \sigma' \Rightarrow \sigma=\sigma'$

- Passo base: $\sigma = \sigma'$, cioè b è valutata "false" in σ.
  - In questo caso sia w_1 che w_2 sono valutati entrambi σ.
- Passo induttivo:
- Sia b valutata "true" in σ. Allora occorre eseguire c in σ, in entrambi i comandi w_1 e w_2. Sia σ'' lo stato raggiunto dopo l'esecuzione di c in σ. A questo punto consideriamo:
  - b valutata false in σ'' ($\neg$b valutata true in σ''). Allora entrambi i cicli si fermano (si noti che ci sono le stesse sottoderivazioni in entrambi i casi. Dunque, $\langle w_1,\sigma\rangle$ è derivato $\sigma' = \sigma''$ sse $\langle w_2, \sigma\rangle$ è derivato σ'.
  - b valutata true in σ'', ($\neg$b valutata false in σ''). Allora dopo il comando c vengono eseguiti nuovamente i comandi w_1 e w_2 su σ''. Per ipotesi induttiva, la proprietà vale sulle sottoderivazioni e dunque $\langle w_1,\sigma''\rangle$ è derivato σ' sse $\langle w_2, \sigma''\rangle$ è derivato σ'. Per le regole di derivazione dei comandi w_1 e w_2 si ha che $\langle w_1,\sigma\rangle$ è derivato σ' sse $\langle w_2, \sigma\rangle$ è derivato σ'

Esercizio 4

- Scrivere la semantica operazionale e denotazionale del seguente costrutto iterativo che viene aggiunto nel linguaggio IMP:

  DO c_1 check b_1; c_2 check b_2 OD

  con b_1, b_2 espressioni booleane e c_1, c_2 comandi.
- Intuitivamente, ad ogni iterazione del costrutto i comandi c_1 check b_1 e c_2 check b_2 vengono eseguiti in modo sequenziale.
- Il commando c_i check b_i ha il seguente effetto: sullo stato corrente viene eseguito il comando c_i e la condizione b_i viene valutata sullo stato ottenuto dall'esecuzione di c_i; se la valutazione della condizione è "false", l'effetto dell'esecuzione di c_i viene annullato (abort di c_1 check b_1).
- Se entrambi i comandi c_i check b_i provocano una azione di abort il ciclo viene interrotto.

Esercizio 4 – Soluzione (Sem.op.)

- Sia $C \equiv$ DO c_1 check b_1; c_2 check b_2 OD.
- Vogliamo definire la semantica operazionale (regole di derivazione) di C rispetto ad ogni stato $\sigma \in \Sigma$ ($\langle C,\sigma\rangle$).
- Per definire le regole di derivazione di $\langle C,\sigma\rangle$, dobbiamo distinguere i seguenti casi

1. b_1 e b_2 sono valutate entrambe "false" dopo l'esecuzione di c_1 e c_2, rispettivamente. In questo caso, il ciclo si interrompe, vengono annullati entrambi i comandi c_1 e c_2 e la valutazione del comando C equivale a quella di uno "skip"
2. Almeno una tra b_1 e b_2 è valutata "true", dopo l'esecuzione di c_1 e c_2. In questo caso, il ciclo non si interrompe, e sarà eventualmente annullato il comando c_i corrispondente a b_i valutata "false". La valutazione di C è data dalla valutazione del comando sullo stato derivato dall'esecuzione di uno dei comandi c_1 e c_2 (o eventualmente entrambi) .

Esercizio 4 - Soluzione (Regole)

- Regole di derivazione per $D \equiv$ DO c_1 check b_1; c_2 check b_2 OD:

$$\frac{\langle c_1,\sigma\rangle \longrightarrow \sigma' \;\; \langle b_1,\sigma'\rangle \longrightarrow False \;\; \langle c_2,\sigma\rangle \longrightarrow \sigma'' \;\; \langle b_2,\sigma''\rangle \longrightarrow False}{\langle D',\sigma\rangle \longrightarrow \sigma}$$

$$\frac{\langle c_1,\sigma\rangle \longrightarrow \sigma' \;\; \langle b_1,\sigma'\rangle \longrightarrow True \;\; \langle c_2,\sigma'\rangle \longrightarrow \sigma'' \;\; \langle b_2,\sigma''\rangle \longrightarrow False \;\; \langle D',\sigma'\rangle \longrightarrow \sigma'''}{\langle D,\sigma\rangle \longrightarrow \sigma'''}$$

$$\frac{\langle c_1,\sigma\rangle \longrightarrow \sigma' \;\; \langle b_1,\sigma'\rangle \longrightarrow False \;\; \langle c_2,\sigma\rangle \longrightarrow \sigma'' \;\; \langle b_2,\sigma''\rangle \longrightarrow True \;\; \langle D',\sigma''\rangle \longrightarrow \sigma'''}{\langle D',\sigma\rangle \longrightarrow \sigma'''}$$

$$\frac{\langle c_1,\sigma\rangle \longrightarrow \sigma' \;\; \langle b_1,\sigma'\rangle \longrightarrow True \;\; \langle c_2,\sigma'\rangle \longrightarrow \sigma'' \;\; \langle b_2,\sigma''\rangle \longrightarrow True \;\; \langle D',\sigma''\rangle \longrightarrow \sigma'''}{\langle D',\sigma\rangle \longrightarrow \sigma'''}$$

Esercizio 4 – Soluzione (Sem.den.)

- Per la denotazione di $D \equiv$ DO c_1 check b_1; c_2 check b_2 OD, ricordiamo che la semantica denotazionale di un comando iterativo c è data utilizzando dal minimo punto fisso di una funzione f definita su $\mathcal{C}[\![c]\!] : \Sigma \rightarrow \Sigma$. Nel nostro caso la funzione $\mathcal{C}[\![D]\!]$ è così definita
- $\mathcal{C}[\![D]\!] =$

  $\{(\sigma,\sigma) \mid (\sigma,\sigma') \in \mathcal{C}[\![c_1]\!], (\sigma',\text{false}) \in \mathcal{B}[\![b_1]\!], (\sigma,\sigma') \in \mathcal{C}[\![c_2]\!], (\sigma',\text{false}) \in \mathcal{B}[\![b_2]\!]\} \cup$

  $\{(\sigma,\sigma''') \mid (\sigma,\sigma') \in \mathcal{C}[\![c_1]\!], (\sigma',\text{false}) \in \mathcal{B}[\![b_1]\!], (\sigma,\sigma'') \in \mathcal{C}[\![c_2]\!], (\sigma'',\text{true}) \in \mathcal{B}[\![b_2]\!], (\sigma'',\sigma''') \in \mathcal{C}[\![D]\!]\} \cup$

  $\{(\sigma,\sigma''') \mid (\sigma,\sigma') \in \mathcal{C}[\![c_1]\!], (\sigma',\text{true}) \in \mathcal{B}[\![b_1]\!], (\sigma',\sigma'') \in \mathcal{C}[\![c_2]\!], (\sigma'',\text{false}) \in \mathcal{B}[\![b_2]\!], (\sigma',\sigma''') \in \mathcal{C}[\![D]\!]\} \cup$

  $\{(\sigma,\sigma''') \mid (\sigma,\sigma') \in \mathcal{C}[\![c_1]\!], (\sigma',\text{true}) \in \mathcal{B}[\![b_1]\!], (\sigma',\sigma'') \in \mathcal{C}[\![c_2]\!], (\sigma'',\text{true}) \in \mathcal{B}[\![b_2]\!], (\sigma'',\sigma''') \in \mathcal{C}[\![D]\!]\} \cup$

Esercizio 4 – Soluzione (Sem.den.)

- La denotazionale di D è data dunque dal minimo fix point della funzione f: $(\Sigma \rightarrow \Sigma) \rightarrow (\Sigma \rightarrow \Sigma)$ così definita:

$\dfrac{}{(\sigma,\sigma)}$ if $(\sigma,\sigma') \in \mathcal{C}[\![c_1]\!], (\sigma',\text{false}) \in \mathcal{B}[\![b_1]\!], (\sigma,\sigma') \in \mathcal{C}[\![c_2]\!], (\sigma',\text{false}) \in \mathcal{B}[\![b_2]\!]$

$\dfrac{\sigma'',\sigma'''}{\sigma,\sigma'''}$ if $(\sigma,\sigma') \in \mathcal{C}[\![c_1]\!], (\sigma',\text{false}) \in \mathcal{B}[\![b_1]\!], (\sigma,\sigma'') \in \mathcal{C}[\![c_2]\!], (\sigma'',\text{true}) \in \mathcal{B}[\![b_2]\!]$

$\dfrac{\sigma',\sigma'''}{\sigma,\sigma'''}$ if $(\sigma,\sigma') \in \mathcal{C}[\![c_1]\!], (\sigma',\text{true}) \in \mathcal{B}[\![b_1]\!], (\sigma',\sigma'') \in \mathcal{C}[\![c_2]\!], (\sigma'',\text{false}) \in \mathcal{B}[\![b_2]\!]$

$\dfrac{\sigma'',\sigma'''}{\sigma,\sigma'''}$ if $(\sigma,\sigma') \in \mathcal{C}[\![c_1]\!], (\sigma',\text{true}) \in \mathcal{B}[\![b_1]\!], (\sigma',\sigma'') \in \mathcal{C}[\![c_2]\!], (\sigma'',\text{true}) \in \mathcal{B}[\![b_2]\!]$

- Siccome f è una funzione continua, il minimo punto fisso di f esiste e dunque $\mathcal{C}[\![D]\!] = \text{fix}(f) = \cup_{i \in \omega} f^i(\perp)$.

Esercizio 5

- Con riferimento al comando definito nell'esercizio precedente

$$D \equiv \mathbf{DO}\ c_1\ \mathbf{check}\ b_1;\ c_2\ \mathbf{check}\ b_2\ \mathbf{OD}$$

  dimostrare che la semantica operazionale e denotazionale del comando D sono equivalenti
- Suggerimento: Bisogna provare che

$$(\sigma, \sigma') \in \mathcal{C}[\![w]\!] \Leftrightarrow \langle w, \sigma \rangle \to \sigma'$$

- L'enunciato può essere provato mostrando:
- $(\sigma, \sigma') \in \mathcal{C}[\![w]\!] \Rightarrow \langle w, \sigma \rangle \to \sigma'$. Per induzione sulle derivazioni
- $(\sigma, \sigma') \in \mathcal{C}[\![w]\!] \Rightarrow \langle w, \sigma \rangle \to \sigma'$. Mostrando per induzione matematica che $\forall \sigma,\sigma' \in \Sigma.\ (\sigma,\sigma') \in \mathcal{C}_n[\![D]\!] \Rightarrow \langle D, \sigma\rangle \to \sigma'$