Fondamenti dei linguaggi di programmazione

Aniello Murano
Università degli Studi di Napoli
"Federico II"

Esercitazione sulla semantica di IMP e sulle tecniche di prova

Argomenti di Esercitazione

- Seconda lezione: Sintassi e semantica operazionale del linguaggio imperativo IMP
  1. Utile per definire sintassi e semantica di nuovi operatori di un linguaggio
  2. Utile per provare l'equivalenza semantica di due elementi di un linguaggio
- Terza lezione: Tecniche di prova per induzione
  3. Utile per provare delle proprietà semantiche degli elementi di un linguaggio

1. Definizione di un nuovo operatore

- Supponiamo di voler estendere le espressioni aritmetiche di IMP includendo un nuovo operatore "^" con il significato di elevamento a potenza.
- La sintassi di Aexp di IMP verrà estesa includendo anche $a_0 \hat{} a_1$ dove a_0 e a_1 sono elementi di Aexp. Dunque per Aexp avremo:
- $a ::= n \mid X \mid a_0 + a_1 \mid a_0 - a_1 \mid a_0 \times a_1 \mid a_0 \hat{} a_1.$
- Valutazione del nuovo operatore

$$\frac{\langle a_0,\sigma\rangle \to n_0 \qquad \langle a_1,\sigma\rangle \to n_1}{\langle a_0 \hat{} a_1,\sigma\rangle \to n}$$

- dove n è uguale a n_0 elevato a n_1

2. Equivalenza semantica di due espressioni

- Utilizzando le regole di valutazione introdotte per le espressioni aritmetiche è possibile mostrare l'equivalenza di due espressioni
- Per esempio, è possibile mostrare formalmente che
- " y + y " equivale all'espressione " y * 2 ", cioè $(y + y) \sim (y * 2)$
- Formalmente, si vuole provare che

$$\langle (y + y),\sigma\rangle \to m \ \text{ iff } \langle (y * 2),\sigma\rangle \to m, \text{ per qualsiasi } \sigma \text{ e } m$$

"⇒" se $\langle (y + y), \sigma\rangle \to m$ allora deve esistere un albero di derivazione con y valutato m_0 e conclusione $\langle (y + y), \sigma\rangle \to m=m_0+m_0$. Dunque m è due volte la valutazione di y. Allora possiamo definire un albero di derivazione per $\langle (y * 2), \sigma\rangle$ con conclusione $\langle (y * 2), \sigma\rangle \to m'=2 \times m_0= m_0+m_0=m$

"⇐" L'inverso del discorso precedente

2. Equivalenza semantica di due comandi

- Utilizzando le regole di valutazione introdotte per i comandi è possibile mostrare l'equivalenza di due comandi
- Per esempio, è possibile mostrare formalmente che

if y=3 then y:=y+1 else y:=y-1

$\sim$

if ¬(y=3) then y:=y-1 else y:=y+1

- Formalmente, si vuole provare che

  $\langle$if y=3 then y:=y+1 else y=y-1, $\sigma\rangle \to \sigma'$ iff $\langle$if ¬(y=3) then y:=y-1 else y:=y+1,$\sigma\rangle \to \sigma'$, per ogni scelta degli stati σ, σ'

"⇒" $\langle$if y=3 then y:=y+1 else y:=y-1, $\sigma\rangle \to \sigma'$ allora deve esistere un albero di derivazione in cui σ' dipende dalla valutazione di y=3. Allora possiamo definire un albero di derivazione per $\langle$if ¬(y=3) then y:=y-1 else y:=y+1,$\sigma\rangle$ con conclusione $\langle$if ¬(y=3) then y:=y-1 else y:=y+1,$\sigma\rangle \to \sigma''$ dove $\sigma''=\sigma'$

"⇐" L'inverso del discorso precedente

3. $\rightarrow_{Bexp}$ è deterministica

- Usando l'induzione strutturale proviamo che
- $P(b)=\forall\sigma \in Loc$ e $\forall w,w' \in \{true,false\}$. $<b,\sigma> \rightarrow w$ & $<b,\sigma> \rightarrow w' \Rightarrow w=w'$
- Se **b** $\equiv$ **true/false**. Allora c'è una sola regola per true/false che si può applicare e dunque w=w'.
- Se **b** $\equiv$ $\mathbf{a_0 = a_1}$ oppure **b** $\equiv$ $\mathbf{a_0} \leq \mathbf{a_1}$ allora i valori w e w' dipendono dalle regole di valutazione delle espressioni aritmetiche. Il risultato segue dal fatto che $\rightarrow_{Aexp}$ è deterministica.
- Se **b** $\equiv$ $\mathbf{b_0} \wedge \mathbf{b_1}$ oppure **b** $\equiv$ $\mathbf{b_0} \vee \mathbf{b_1}$ oppure **b** $\equiv$ $\neg\mathbf{b_0}$, il risultato segue per ipotesi induttiva.

Esercizio

- Provare che $\rightarrow_{Bexp}$ è totale utilizzando il metodo di induzione strutturale