Fondamenti dei linguaggi di programmazione

Aniello Murano
Università degli Studi di Napoli
"Federico II"

Equivalenza delle semantiche di IMP

Riepilogo

Definizione di un Linguaggio Formale

- Un linguaggio formale, qual è un linguaggio di programmazione, deve essere definito in modo chiaro, preciso e completo,
- Una **specifica** chiara precisa e completa è necessaria sia per chi deve costruire compilatori e/o interpreti del linguaggio, sia per chi deve scrivere programmi.
- Descrivere con precisione un linguaggio significa definirne con precisione sintassi e semantica.
- La **sintassi** specifica sia la struttura lessicale del linguaggio (come costruire le parole a partire dai caratteri dell'alfabeto) sia le regole per la costruzione delle frasi (programmi sintatticamente corretti).
- La **semantica** specifica il significato delle frasi (programmi).

Semantiche considerate

- Nelle lezioni precedenti abbiamo introdotto il linguaggio imperativo IMP e per questo linguaggio abbiamo mostrato una **Sintassi** basata su regole di composizione (BNF) e due tipi di semantiche:
- **Semantica operazionale**
  - associa alle frasi del linguaggio operazioni eseguite da una macchina astratta
- **Semantica denotazionale**
  - associa alle frasi del linguaggio funzioni matematiche
- **Semantica denotazionale (solo accennata)**
  - associare alle frasi del linguaggio formule logiche.

Semantica denotazionale

- Al linguaggio IMP sono associate le seguenti funzioni:
  - $\mathcal{A}$: Aexp $\rightarrow (\Sigma \rightarrow \mathcal{N})$;
  - $\mathcal{B}$: Bexp $\rightarrow (\Sigma \rightarrow \{$true, false$\})$,
  - $\mathcal{C}$: Com $\rightarrow (\Sigma \rightarrow \Sigma)$
- Per esempio, se la denotazione di X è data da $(\sigma, \sigma(X))$, la valutazione di X in uno stato σ è data da $\mathcal{A}[\![X]\!](\sigma)=\sigma(X)$
- Per i comandi, questo ragionamento può dare luogo a difficoltà.
- Per esempio, sia w $\equiv$ while b do c. Abbiamo visto che $\mathcal{C}[\![w]\!] = \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]\sigma = \text{true} \;\&\; (\sigma,\sigma') \in \mathcal{C}[\![w]\!] \circ \mathcal{C}[\![c]\!]\} \cup \{(\sigma,\sigma) \mid \mathcal{B}[\![b]\!]\sigma = \text{false}\}$ non da la denotazione di w, perché $\mathcal{C}[\![w]\!]$ è una funzione ricorsiva.
- In pratica, la denotazione di w è data dalla sua denotazione al prossimo ciclo concatenata alla denotazione del comando c.
- Questo ragionamento suggerisce una funzione continua f: $(\Sigma \rightarrow \Sigma) \rightarrow (\Sigma \rightarrow \Sigma)$ il cui fixpoint (corrispondente all'eventuale termine del ciclo while) fornisce la denotazione di w.

Fixpoint per $\mathcal{C}[\![w]\!]$

Sia $\mathcal{C}[\![w]\!]$ =

$\{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]\sigma = \text{true} \;\&\; (\sigma,\sigma') \in \mathcal{C}[\![w]\!] \circ \mathcal{C}[\![c]\!]\} \cup$
$\{(\sigma,\sigma) \mid \mathcal{B}[\![b]\!]\sigma = \text{false}\}$ =

$\{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]\sigma = \text{true} \;\&\; (\sigma,\sigma'') \in \mathcal{C}[\![c]\!] \;\&\; (\sigma'',\sigma') \in \mathcal{C}[\![w]\!]\} \cup$
$\{(\sigma,\sigma) \mid \mathcal{B}[\![b]\!]\sigma = \text{false}\}$

f è dunque definita sulle seguenti regole

$$\frac{\emptyset}{\sigma\,,\,\sigma} \quad \text{se } B[\![b]\!]\sigma = \text{false}$$

$$\frac{\sigma'',\,\sigma'}{\sigma\,,\,\sigma'} \quad \text{se } B[\![b]\!]\sigma = \text{true} \;\&\; (\sigma,\sigma'') \in C[\![c]\!]$$

Equivalenza delle semantiche

- In questa lezione mostriamo l'equivalenza della semantica operazionale con quella denotazionale per il linguaggio IMP.
- Per provare l'equivalenza delle due semantiche occorre dimostrare che
  - Per ogni $a \in Aexp$, $(\sigma, n) \in \mathcal{A}[\![a]\!] \Leftrightarrow \langle a, \sigma \rangle \to n$
  - Per ogni $b \in Bexp$, $(\sigma, t) \in \mathcal{B}[\![b]\!] \Leftrightarrow \langle b, \sigma \rangle \to t$
  - Per ogni $c \in Com$, $(\sigma, \sigma') \in \mathcal{C}[\![c]\!] \Leftrightarrow \langle c, \sigma \rangle \to \sigma'$

Equivalenza per Aexp

- Dimostriamo per <u>induzione strutturale</u> che per ogni $a \in Aexp$ vale la proprietà P(a)

$$(\sigma, n) \in \mathcal{A}[\![a]\!] \Leftrightarrow \langle a, \sigma \rangle \to n$$

- Sia $a \equiv m$. Allora $(\sigma, n) \in \mathcal{A}[\![m]\!] \Leftrightarrow m = n \Leftrightarrow \langle m, \sigma \rangle \to n$
- Sia $a \equiv X$. Allora $(\sigma, n) \in \mathcal{A}[\![X]\!] \Leftrightarrow n = \sigma(X) \Leftrightarrow \langle X, \sigma \rangle \to n$
- Sia $a \equiv a_1 + a_2$.
  - Se $(\sigma, n) \in \mathcal{A}[\![a_1 + a_2]\!]$ allora esistono $(\sigma, n_1) \in \mathcal{A}[\![a_1]\!]$ & $(\sigma, n_2) \in \mathcal{A}[\![a_2]\!]$ tale che $n=n_1+n_2$. Allora per induzione $\langle a_1, \sigma \rangle \to n_1$ e $\langle a_2, \sigma \rangle \to n_2$, dunque $\langle a_1 + a_2, \sigma \rangle \to n$.
  - Se $\langle a_1 + a_2, \sigma \rangle \to n$ allora esiste una derivazione con sottoderivazioni $\langle a_1, \sigma \rangle \to n_1$ e $\langle a_2, \sigma \rangle \to n_2$ tale che $n=n_1+n_2$. Allora, per induzione $(\sigma, n_1) \in \mathcal{A}[\![a_1]\!]$ & $(\sigma, n_2) \in \mathcal{A}[\![a_2]\!]$, dunque $(\sigma, n) \in \mathcal{A}[\![a_2 + a_2]\!]$.
- Per $a \equiv a_1 - a_2$ e $a \equiv a_1 * a_2$ ragionamento analogo al precedente.

Equivalenza per Bexp (1)

- Dimostriamo per <u>induzione strutturale</u> che per ogni $b \in$ Bexp vale la proprietà P(b)

$$(\sigma, t) \in \mathcal{B}[\![b]\!] \Leftrightarrow \langle b, \sigma \rangle \to t$$

- Sia $b \equiv$ true. Allora $(\sigma, t) \in \mathcal{B}[\![true]\!] \Leftrightarrow t = true \Leftrightarrow \langle true, \sigma \rangle \to t$
- Sia $b \equiv$ false. Ragionamento analogo al precedente.
- Sia $b \equiv a_0 = a_1$.
  - Se $(\sigma, t) \in \mathcal{B}[\![a_0 = a_1]\!]$ allora esistono n_0 e n_1 tale che $\mathcal{A}[\![a_0]\!]\sigma = n_0$, $\mathcal{A}[\![a_1]\!]\sigma = n_1$ e "$n_0 = n_1$ & t = true" oppure "$n_0 \neq n_1$ & t = false". Per ipotesi induttiva, $\langle a_0, \sigma \rangle \to n_0$ e $\langle a_1, \sigma \rangle \to n_1$ e, in entrambi i casi $\langle a_0 = a_1, \sigma \rangle \to t$.
  - Se $\langle a_0 = a_1, \sigma \rangle \to t$ allora esiste una derivazione con sottoderivazioni $\langle a_0, \sigma \rangle \to n_0$ e $\langle a_1, \sigma \rangle \to n_1$ e tale che o "$n_0 = n_1$ e t=true" oppure "$n_0 \neq n_1$ e t=false". Per ipotesi induttiva $\mathcal{A}[\![a_0]\!]\sigma = n_0$, $\mathcal{A}[\![a_1]\!]\sigma = n_1$, e in entrambi i casi $(\sigma, t) \in \mathcal{B}[\![a_0 = a_1]\!]$
- Per $b \equiv a_0 \leq a_1$ ragionamento analogo al precedente

Equivalenza per Bexp (2)

- Sia $b \equiv b_0 \wedge b_1$.
  - Se $(\sigma, t) \in \mathcal{B}[\![b_0 \wedge b_1]\!]$ allora esistono t_0 e t_1 tale che $\mathcal{B}[\![b_0]\!]\sigma = t_0$, $\mathcal{B}[\![b_1]\!]\sigma = t_1$ e $t_0 \wedge t_1 = t$. Per ipotesi induttiva, $\langle b_0, \sigma \rangle \to t_0$ e $\langle b_1, \sigma \rangle \to t_1$. Dunque $\langle b_0 \wedge b_1, \sigma \rangle \to t$.
  - Se $\langle b_0 \wedge b_1, \sigma \rangle \to t$ allora esiste una derivazione con sottoderivazioni $\langle b_0, \sigma \rangle \to t_0$ e $\langle b_1, \sigma \rangle \to t_1$ e $t_0 \wedge t_1 = t$. Per ipotesi induttiva $\mathcal{B}[\![b_0]\!]\sigma = t_0$, $\mathcal{B}[\![b_1]\!]\sigma = t_1$. Dunque per definizione $(\sigma, t) \in \mathcal{B}[\![b_0 \wedge b_1]\!]$
- Per $b \equiv b_0 \vee b_1$ ragionamento analogo al precedente.
- Per $b \equiv \neg b'$ (esercizio!)

Equivalenza per Com (1)

- Dimostriamo che per ogni $c \in$ Com vale la proprietà P(c)
  $$(\sigma, \sigma') \in \mathcal{C}[\![c]\!] \Leftrightarrow \langle c, \sigma \rangle \to \sigma'$$
- Iniziamo con il verso "$\Leftarrow$".
- Per questo verso usiamo una induzione sulle derivazioni.
- Sia **c** $\equiv$ **skip**. Per gli assiomi, $\langle$skip$, \sigma \rangle \to \sigma$ e $(\sigma, \sigma) \in \mathcal{C}[\![$skip$]\!]$
- Sia **c** $\equiv$ **X:=a**. Se $\langle X:=a, \sigma \rangle \to \sigma'$, allora esiste una derivazione

$$\cfrac{\cfrac{\dots}{\langle a, \sigma \rangle \to n}}{\langle X:=a, \sigma \rangle \to \sigma'}$$

con $\sigma' = \sigma[n/X]$.
Per l'equivalenza sulle espressioni di Aexp, segue $(\sigma, n) \in \mathcal{A}[\![a]\!]$.
Dato che $\mathcal{C}[\![X:=a]\!] = \{(\sigma, \sigma[n/X]) \mid \mathcal{A}[\![a_0]\!]\sigma = n\}$, segue $(\sigma, \sigma') \in \mathcal{C}[\![c]\!]$

Equivalenza per Com di ";" per "$\Leftarrow$"

- Sia $c=c_0; c_1$
  Se $\langle c_0; c_1, \sigma\rangle \to \sigma'$ allora esiste la seguente derivazione

$$\frac{\langle c_0, \sigma\rangle \to \sigma'' \quad \langle c_1, \sigma'' \rangle \to \sigma'}{\langle c_0; c_1, \sigma\rangle \to \sigma'}$$

Per ipotesi induttiva, abbiamo che $\mathcal{C}[\![c_0]\!]\sigma=\sigma''$ e $\mathcal{C}[\![c_1]\!]\sigma''=\sigma'$.

Dunque:

$\mathcal{C}[\![c_0; c_1]\!]\sigma = \mathcal{C}[\![c_1]\!] (\mathcal{C}[\![c_0]\!]\sigma) = \mathcal{C}[\![c_1]\!]\sigma'' = \sigma'$, come richiesto

Equivalenza per Com: while per "$\Leftarrow$"

- Sia $w \equiv$ while b do c. Proviamo che $(\sigma, \sigma') \in \mathcal{C}[\![w]\!] \Leftarrow \langle w, \sigma \rangle \rightarrow \sigma'$.
- Se $\langle w, \sigma \rangle \rightarrow \sigma'$, allora esistono due possibili derivazioni dipendenti dalla valutazione "true" o "false" di b. Nel primo caso:

$$\frac{\langle b, \sigma \rangle \rightarrow \text{true} \qquad d_1 ::= \langle c, \sigma \rangle \rightarrow \sigma'' \quad d_2 ::= \langle w, \sigma'' \rangle \rightarrow \sigma'}{\langle w, \sigma \rangle \rightarrow \sigma'}$$

1. Per l'equivalenza provata su Bexp, $(\sigma, \text{true}) \in \mathcal{B}[\![b]\!]$.
2. Per ipotesi induttiva su d_1, da $\langle c, \sigma \rangle \rightarrow \sigma''$ segue che $(\sigma, \sigma'') \in \mathcal{C}[\![c]\!]$
3. Per ipotesi induttiva su d_2, da $\langle w, \sigma'' \rangle \rightarrow \sigma'$ segue che $(\sigma, \sigma') \in \mathcal{C}[\![w]\!]$

- Ricordiamo che la denotazione di w (nel caso $(\sigma, \text{true}) \in \mathcal{B}[\![b]\!]$) è

$$\mathcal{C}[\![w]\!] = \{(\sigma, \sigma_1) \mid (\sigma, \sigma_1) \in \mathcal{C}[\![w]\!] \circ \mathcal{C}[\![c]\!]\}$$

il cui risultato è dato dal suo fixpoint.
- Siccome $\mathcal{C}[\![w]\!]\sigma = (\mathcal{C}[\![w]\!] \circ \mathcal{C}[\![c]\!])\sigma = (\mathcal{C}[\![c;w]\!])\sigma = \mathcal{C}[\![w]\!](\mathcal{C}[\![c]\!])\sigma$, segue

$$\mathcal{C}[\![\mathbf{w}]\!]\sigma = \text{fix } \mathcal{C}[\![w]\!]\sigma = \text{fix } \mathcal{C}[\![w]\!](\mathcal{C}[\![c]\!]\sigma) = \text{fix } \mathcal{C}[\![w]\!]\sigma'' = \mathcal{C}[\![w]\!]\sigma'' = \sigma'$$

Dunque $(\sigma, \sigma') \in \mathcal{C}[\![w]\!]$

- Esercizio: Completare con $\langle b, \sigma \rangle \rightarrow$ false è "$\Leftarrow$" per gli altri comandi.

Equivalenza per Com "$\Rightarrow$"

- Dobbiamo dimostrare che

$$(\sigma, \sigma') \in \mathcal{C}[\![c]\!] \Rightarrow \langle c, \sigma \rangle \rightarrow \sigma'$$

- L'equivalenza sui comandi

Skip, X:=a, $c_0;c_1$, if b then c_0 else c_1

è lasciata per esercizio.

- Proviamo il verso "$\Rightarrow$" solo per il comando while che è anche il più complesso

Equivalenza per Com: while per "$\Rightarrow$"

- Sia dunque **w** $\equiv$ **while b do c** e vogliamo provare che

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

  supponendo che l'equivalenza delle semantiche <u>**sia vera per c**</u>
- Si ricordi che $\mathcal{C}[\![w]\!] = \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]\sigma = \text{true} \;\&\; (\sigma,\sigma') \in \mathcal{C}[\![w]\!] \circ \mathcal{C}[\![c]\!]\} \cup \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!]\sigma = \text{false}\}$, la cui soluzione è data dal fix point
- A questo scopo, nella lezione precedente, abbiamo introdotto
  - $\mathcal{C}_0[\![w]\!] \equiv \perp$ come una funzione indefinita
  - e $\forall n \geq 1$, $\mathcal{C}_n[\![w]\!]$ la valutazione di while con al più n valutazioni di b.
- Inoltre abbiamo introdotto una funzioni f su $\mathcal{C}[\![w]\!]$ tale che
  - $f^0(\perp) = \perp = \mathcal{C}_0[\![w]\!]$, $f^1(\perp) = f(f^0(\perp)) = \mathcal{C}_1[\![w]\!]$,
  - $\forall n > 1$, $f^n(\perp)) = f(f^{n-1}(\perp)) = \mathcal{C}_n[\![w]\!]$
  - $\mathcal{C}[\![w]\!] = fix(f) = (\bigsqcup_{n\in\omega} f^n(\perp))$
- In pratica, $\mathcal{C}_n[\![w]\!] = \{(\sigma,\sigma') \mid \mathcal{B}[\![b]\!]\sigma = \text{true} \;\&\; (\sigma,\sigma') \in \mathcal{C}_{n-1}[\![w]\!] \circ \mathcal{C}[\![c]\!]\} \cup \{(\sigma, \sigma) \mid \mathcal{B}[\![b]\!]\sigma = \text{false}\}$
- Se si dimostra che $\forall \sigma,\sigma' \in \Sigma.\ (\sigma,\sigma') \in \mathcal{C}_n[\![w]\!] \Rightarrow \langle w, \sigma\rangle \to \sigma'$ allora per il fixpoint abbiamo che $(\sigma,\sigma') \in \mathcal{C}[\![w]\!] \Rightarrow \langle w, \sigma\rangle \to \sigma'$.

Equivalenza per Com: while per "$\Rightarrow$" (2)

- Dimostriamo per <u>induzione matematica</u> che

  $$\forall \sigma,\sigma' \in \Sigma.\ (\sigma,\sigma') \in \mathcal{C}_n[\![w]\!] \Rightarrow \langle w,\sigma\rangle \to \sigma'$$
- Caso base: n = 0. Semplice
- Supposto vero per un arbitrario $n \in \omega$, dimostriamo che $(\sigma,\sigma') \in \mathcal{C}_{n+1}[\![w]\!] \Rightarrow (w,\sigma) \to \sigma'$ per qualsiasi coppia di stati
- Si assuma $(\sigma,\sigma') \in \mathcal{C}_{n+1}[\![w]\!]$. Allora
  1. . $\mathcal{B}[\![b]\!]\sigma = \text{true}$ e $(\sigma,\sigma') \in \mathcal{C}_n[\![w]\!] \circ \mathcal{C}[\![c]\!]$ oppure
  2. . $\mathcal{B}[\![b]\!]\sigma = \text{false}$ e $\sigma = \sigma'$.
- Per il caso 2, il risultato segue dalla semantica operazionale di w
- Per il caso 1, per l'equivalenza su Bexp segue $\langle b,\sigma\rangle \to \text{true}$, inoltre $(\sigma,\sigma'') \in \mathcal{C}[\![c]\!]$ implica $\langle c,\sigma\rangle \to \sigma''$ e per ipotesi induttiva, $(\sigma'', \sigma') \in \mathcal{C}_n[\![w]\!]$ implica $\langle w, \sigma''\rangle \to \sigma'$.
- Per la regola di derivazione del while otteniamo $\langle w,\sigma\rangle \to \sigma'$