Logica per la Programmazione

Lezione 12

- Sistema di Dimostrazioni per le Triple di Hoare
- Comando Vuoto, Assegnamento, Sequenza, Condizionale

Tripla di Hoare Soddisfatta: richiamo

Data la **tripla di Hoare** $\{Q\}\ C\ \{R\}$

- Q è detta precondizione
- R è detta postcondizione
- La **tripla è soddisfatta** se:
  - *per ogni* stato σ che soddisfa la precondizione Q (ovvero $\sigma \models Q$)
  - l'esecuzione del comando C a partire dallo stato σ
  - **termina** producendo *uno stato* σ'
  - σ' soddisfa la postcondizione R (ovvero $\sigma' \models R$)

Tripla di Hoare Soddisfatta:esempio

- Assumiamo che $\mathbf{a} : array[0, n)$ of int e consideriamo il comando C

```
while x < n do
    if (a[x] > 0) then c := c + 1 else skip fi;
    x := x + 1
endw
```

- La tripla $\{Q\}\ C\ \{R\}$ risulta soddisfatta per

$$Q = (x = 0 \land c = 0)$$

$$R = (c = \#\{j : j \in [0, n) \mid a[j] > 0\})$$

- Come possiamo verificarlo?

Verificare una Tripla di Hoare

- Obiettivo principale: **verificare che una tripla sia soddisfatta** (ovvero più semplicemente: verificare una tripla)
- Per questo introduciamo un **proof system** per verificare le triple **per induzione strutturale sui comandi**
  - Assiomi - alcune triple che sono sempre soddisfatte
  - Regole di inferenza - triple che dipendono da un insieme di condizioni (premesse)
- Correttezza degli assiomi e delle regole basata su semantica informale dei comandi
- Discuteremo la correttezza di ogni assioma e regola rispetto alla semantica

Regola di Inferenza per la precondizione

$$(PRE) \quad \frac{P \Rightarrow P' \qquad \{P'\}\ C\ \{R\}}{\{P\}\ C\ \{R\}}$$

- La **correttezza** segue dalla definizione di "tripla soddisfatta" e dal fatto che $P \Rightarrow P'$ garantisce $\{P\} \subseteq \{P'\}$
- Intuitivamente si può rafforzare la precondizione P'
- Ricordiamo che $\{Q\}$ è l'insieme di stati che soddisfano Q!!!

Regola di Inferenza "Pre-Post"

$$(POST) \qquad \frac{\{P\}\ C\ \{R'\} \qquad R' \Rightarrow R}{\{P\}\ C\ \{R\}}$$

$$(PRE-POST) \qquad \frac{P \Rightarrow P' \qquad \{P'\}\ C\ \{R'\} \qquad R' \Rightarrow R}{\{P\}\ C\ \{R\}}$$

- La **correttezza** segue dalla definizione di "tripla soddisfatta" in modo analogo al caso della regola (PRE)
- Intuitivamente si può rafforzare la precondizione P' ed indebolire la postcondizione R'

Assioma per Skip

- **Assioma** per il comando vuoto:

$$(SKIP) \quad \{P\}\ \mathbf{skip}\ \{P\}$$

- Correttezza? Ovvia...
- Infatti ricordiamo il significato informale:
  - L'esecuzione di **skip** a partire dallo stato σ porta nello stato σ

Assioma per Assegnamento Semplice: commenti

- **Assioma** per l'assegnamento semplice

$$(ASS) \quad \{?????????\}\ x := E\ \{P\}$$

- Ricordiamo la semantica del comando esguito in uno stato σ:
  - L'esecuzione di $x := E$ a partire dallo stato σ porta nello stato $\sigma[\mathcal{E}(E,\sigma)/x]$
- La precondizione deve garantire che l'espressione E sia definita (infatti $\mathcal{E}(E,\sigma)$ **potrebbe non avere valore**)
- Inoltre bisogna garantire che la postcondizione P sia soddisfatta dopo l'assegnameno ovvero dopo che all'identificatore di variabile x è stato sostituito il valore $\mathcal{E}(E,\sigma)$

Definizione di Espressioni

- Introduciamo una funzione $def(\_)$ che applicata a un'espressione E restituisce una asserzione. Intuitivamente dato uno stato σ, se $\sigma \models def(E)$ allora esiste un v tale che $\mathcal{E}(E, \sigma) = v$.

$$
\begin{array}{rcll}
def(c) & = & tt & se \quad c \in Num \ o \ c \in Bool \\
def(x) & = & tt & se \quad x \in Ide \\
def(E \ op \ E') & = & def(E) \wedge def(E') & se \quad op \in \left\{ \begin{array}{l} +, -, =, \neq, <, >, \\ \leq, \geq, and, or \end{array} \right\} \\
def(E \ op \ E') & = & def(E) \wedge def(E') \wedge E' \neq 0 & se \quad op \in \{div, mod\} \\
def(not \ E) & = & def(E) & \\
def((E)) & = & def(E) &
\end{array}
$$

Definizione di Espressioni: commenti

- L'asserzione $def(E)$ serve a garantire che la valutazione di E sia definita e termini
- Esempio:

$$\begin{aligned} def(x \ div \ (z-y)) &= def(x) \wedge def(y) \wedge def(z) \wedge (z-y) \neq 0 \\ &\equiv (z-y) \neq 0 \equiv z \neq y \end{aligned}$$

- Ricordiamo che manca un controllo dei tipi dei dati manipolati dalle operazioni:

$$def(x+y) = def(x) \wedge def(y) \equiv \mathbf{T}$$

- Per semplicità supponiamo che il controllo dei tipi sia realizzato in una fase preliminare. Quindi tutte le espressioni che compaiono nei nostri programmi sono sempre ben tipate

Stati come Modelli: proprietà

Sia E una espressione e P un'asserzione. Valgono le seguenti proprietà:

- Dati due stati σ e σ' tali che $\sigma(x) = \sigma'(x)$, per ogni $x \in \mathit{free}(E)$, allora
  - $\mathcal{E}(E, \sigma) = \mathcal{E}(E, \sigma')$
  - $\sigma \models P$ se e solo se $\sigma' \models P$
- Per ogni variabile x abbiamo

$$\sigma[\mathcal{E}(E, \sigma)/x] \models P \text{ se e solo se } \sigma \models P[E/x]$$

  dove $P[E/x]$ denota l'asserzione P in cui E viene sostituito ad x
- Esempio: $\sigma[5/x] \models (y = x)$ se e solo se

$$\sigma \models (y = x)[5/x]$$
$$\sigma \models (y = 5)$$

Assioma per Assegnamento Semplice

- **Assioma** per l'assegnamento semplice

$$(ASS) \quad \{def(E) \wedge P[E/x]\}\ x := E\ \{P\}$$

  dove $P[E/x]$ denota l'asserzione P in cui E viene sostituito ad x
- La **correttezza** dell'assioma si vede confrontando l'assioma con la semantica informale:
  - L'esecuzione dell'assegnamento $x := E$ a partire dallo stato σ porta nello stato $\sigma[\mathcal{E}(E,\sigma)/x]$
- e ricordando che
  - $def(E)$ garantisce che l'espressione E sia definita
  - inoltre

$$\sigma[\mathcal{E}(E,\sigma)/x] \models P \text{ se e solo se } \sigma \models P[E/x]$$

- L'idea è che affinchè la postcondizione P sia soddisfatta dopo l'assegnamento, la stessa P deve essere soddisfatta dallo stato precedente l'assegnamento quando all'identificatore di variabile x è sostituita l'espressione E

Assegnamento Semplice: Esempi

Verificare le triple:

- $\{\mathbf{T}\}\ x := 5\ \{x = 5\}$
  - Dall'assioma (ASS) otteniamo la precondizione

$$def(5) \wedge (x = 5)[5/x] \equiv (5 = 5) \equiv \mathbf{T}$$

- $\{x > 2\}\ x := 5\ \{x = 5\}$
  - Dall'assioma (ASS) otteniamo la precondizione precedente
  - Vale??? Come la possiamo dimostrare????
  - Dalla regola (PRE) notando che

$$x > 2 \ \Rightarrow\ \mathbf{T}$$

- $\{x = 5\}\ x := x + 1\ \{x > 2\}$
  - Analogamente da (ASS) e (PRE) osservando che

$$(x > 2)[x + 1/x] = (x + 1 > 2) \equiv (x > 1)$$

$$x = 5 \ \Rightarrow\ x > 1$$

Regola di Inferenza per l'Assegnamento

- Combinando la regola (PRE) con l'assioma per l'assegnamento semplice, si ottiene la seguente regola, utile per verificare che una tripla data sia soddisfatta:

$$(ASS) \quad \frac{R \Rightarrow def(E) \wedge P[E/x]}{\{R\}\ x := E\ \{P\}}$$

- Infatti abbiamo la seguente istanza di (PRE), in cui la seconda premessa scompare perché è un assioma:

$$\frac{R \Rightarrow def(E) \wedge P[E/x] \qquad \{def(E) \wedge P[E/x]\}\ x := E\ \{P\}}{\{R\}\ x := E\ \{P\}}$$

Assioma per Assegnamento Multiplo

- Generalizzando quello per l'assegnamento singolo otteniamo il seguente assioma (ASS):

$$\{def(E_1) \wedge \ldots \wedge def(E_k) \wedge P[E_1/x_1, \ldots, E_k/x_k]\}\ x_1, \ldots, x_k := E_1, \ldots, E_k\ \{P\}$$

- Tutte le espressioni vengono valutate **prima** di tutti gli assegnamenti: confrontiamo con la semantica informale
  - L'esecuzione dell'assegnamento $x_1, \ldots, x_k := E_1, \ldots, E_k$ a partire dallo stato σ porta nello stato $\sigma[\mathcal{E}(E_1, \sigma)/x_1, \ldots, \mathcal{E}(E_k, \sigma)/x_k]$

Regola per la Sequenza di Comandi

- Regola per verifica di una tripla in cui il comando è una **sequenza** (per induzione strutturale):

$$(SEQ) \quad \frac{\{P\}\ C\ \{R\} \qquad \{R\}\ C'\ \{Q\}}{\{P\}\ C; C'\ \{Q\}}$$

- Convinciamoci della correttezza confrontandola con la semantica informale:
  - L'esecuzione di $C; C'$ a partire dallo stato σ porta nello stato σ' ottenuto eseguendo C' a partire dallo stato σ'' ottenuto dall'esecuzione di C nello stato σ
- R è una asserzione intermedia che rappresenta tutti gli stati σ'' ottenuti il primo comando in uno stato σ che soddisfa la precondizione

Esempio di Sequenza di Comandi

Verifichiamo la tripla

$$\{x \geq y - 1\}\ x := x + 1; y := y - 1\ \{x > y\}$$

- Per la(SEQ) dobbiamo trovare una asserzione intermedia R e verificare le seguenti triple:
  1) $\{x \geq y - 1\}\ x := x + 1\ \{R\}$
  2) $\{R\}\ y := y - 1\ \{x > y\}$
- Per determinare R usiamo l'**assioma** (ASS) nella 2). Quindi la seguente è verificata:

$$\{def(y - 1) \wedge (x > y)[y - 1/y]\}\ y := y - 1\ \{x > y\}$$

- Fissando $R = def(y - 1) \wedge x > y - 1$ resta da verificare la 1):

$$\{x \geq y - 1\}\ x := x + 1\ \{R\}$$

- Usando la **regola** (ASS) basta dimostrare (per esercizio):

$$x \geq y - 1 \Rightarrow def(x + 1) \wedge (def(y - 1) \wedge (x > y - 1))[x + 1/x]$$

Regola per il Comando Condizionale

- Regola per verifica di una tripla in cui il comando è un **condizionale** (per induzione strutturale):

$$(COND) \quad \frac{P \Rightarrow def(E) \quad \{P \wedge E\}\ C_1\ \{Q\} \quad \{P \wedge \neg E\}\ C_2\ \{Q\}}{\{P\}\ \textbf{if}\ E\ \textbf{then}\ C_1\ \textbf{else}\ C_2\ \textbf{fi}\ \{Q\}}$$

- La correttezza della regola segue dal confronto con il significato informale del condizionale:
  - L'esecuzione di **if** E **then** C_1 **else** C_2 **fi** a partire da σ porta nello stato σ':
    - che si ottiene dall'esecuzione di C_1 in σ, se $\mathcal{E}(E, \sigma) = \textbf{tt}$
    - che si ottiene dall'esecuzione di C_2 in σ, se $\mathcal{E}(E, \sigma) = \textbf{ff}$

Esempio: comando condizionale

Verificare la seguente tripla:

$$
\begin{array}{l}
\{m = 0\} \\
\quad \textbf{if } x < y \\
\qquad \textbf{then } m := y \\
\qquad \textbf{else } m := x \\
\quad \textbf{fi} \\
\{m = max(x, y)\}
\end{array}
$$

Soluzione: Verifica della Tripla

Per la **regola** $(COND)$ dobbiamo mostrare che:

1. $m = 0 \;\Rightarrow\; def(x < y)$
2. $\{m = 0 \wedge x < y\}\ m := y\ \{m = max(x, y)\}$
3. $\{m = 0 \wedge x \geq y\}\ m := x\ \{m = max(x, y)\}$

1. Banale applicando la definizione di def:
   $def(x < y) = def(x) \wedge def(y) \equiv \mathbf{T}$
2. Usando la **regola** (ASS) ci riduciamo a mostrare:

   $$m = 0 \wedge x < y \;\Rightarrow\; def(y) \wedge (m = max(x, y))[y/m]$$

   ovvero $m = 0 \wedge x < y \;\Rightarrow\; y = max(x, y)$
3. Analogamente, ci riduciamo a mostrare:

   $$m = 0 \wedge x \geq y \;\Rightarrow\; x = max(x, y)$$

Esercizi

- La seguente tripla **non è soddisfatta** Mostrare formalmente perchè.

$$\{z = 5 \wedge y = 7 \wedge x = 3\}\ x := 0;\ y := z\ div\ x\ \{z > 4\}$$

- Le seguenti triple sono soddisfatte? Perché?
  - $\{x = N \wedge y = M\}\ x := y; y := x\ \{x = M \wedge y = N\}$
  - $\{x = N \wedge y = M\}\ x, y := y, x\ \{x = M \wedge y = N\}$
- Verificare la seguente tripla:

$$\{s = (\Sigma\ i : i \in [0, x).i)\}$$
$$s, x := s + x; x + 1$$
$$\{s = (\Sigma\ i : i \in [0, x).i)\}$$

Esercizi

Verificare la seguente tripla:

$$\{x = 5\}$$
$$x := 3;$$
$$\textbf{if } (x = 3) \textbf{ then } y := 7 \textbf{ else } y := 5 \textbf{ fi}$$
$$\{x = 3 \wedge y = 7\}$$

- Per la **regola** (SEQ) dobbiamo trovare una asserzione intermedia R che soddisfi le seguenti triple:
  1. $\{x = 5\}\ x := 3\ \{R\}$
  2. $\{R\}$ **if** $(x = 3)$ **then** $y := 7$ **else** $y := 5$ **if** $\{x = 3 \wedge y = 7\}$
- A differenza di altri esempi, qui non possiamo usare l'assioma dell'assegnamento per trovare R.
- Un candidato naturale per R è $x = 3$.
- **Esercizio**: completare la dimostrazione di 1) e 2)

Sequenza con **Variabili di Specifica** (1)

- Determinare E in modo che la tripla sia verificata:

$$\{x = N \wedge y = M\}\ t := E; x := y; y := t\ \{x = M \wedge y = N\}$$

- Notare che M e N sono **variabili di specifica**: non possono essere usate nei comandi
- Candidati per E???

Sequenza con **Variabili di Specifica** (2)

- Per la **regola** (*SEQ*) dobbiamo trovare R_1 e R_2 tali che:
  1. $\{x = N \wedge y = M\}\ t := E\ \{R_1\}$
  2. $\{R_1\}\ x := y\ \{R_2\}$
  3. $\{R_2\}\ y := t\ \{x = M \wedge y = N\}$
- Per determinare R_2, usiamo l' **assioma** (*ASS*) in 3):

$$\{def(t) \wedge (x = M \wedge y = N)[t/y]\}\ y := t\ \{x = M \wedge y = N\}$$

- Quindi fissiamo $R_2 = (x = M \wedge t = N)$
- Analogamente per R_1, usiamo l' **assioma** (*ASS*) in 2):

$$\{def(y) \wedge (x = M \wedge t = N)[y/x]\}\ x := y\ \{x = M \wedge t = N\}$$

- Quindi fissiamo $R_1 = (y = M \wedge t = N)$
- Resta da verificare la 1):

$$\{x = N \wedge y = M\}\ t := E\ \{y = M \wedge t = N\}$$

- Usando la **regola** (*ASS*), basta trovare un E tale che:

$$x = N \wedge y = M \ \Rightarrow\ def(E) \wedge (y = M \wedge E = N)$$