

Logica per la Programmazione

Lezione 13

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

Tripla di Hoare Soddisfatta: richiamo

Data una **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 $a : \text{array}[0, n)$ 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

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

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

- ▶ 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 sul comando**
 - ▶ **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 sulla semantica informale dei comandi
- ▶ Discuteremo la **correttezza** di ogni assioma e regola rispetto alla semantica

Regola di Inferenza per la preconditione

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

- ▶ La **correttezza** segue dalla definizione di “tripla soddisfatta” e dal fatto che $P \Rightarrow P'$ garantisce $\{P\} \subseteq \{P'\}$
- ▶ Intuitivamente si può **rafforzare la preconditione**

Regola di Inferenza “Pre-Post”

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

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

- ▶ La **correttezza** segue dalla definizione di “tripla soddisfatta” in modo analogo al caso della regola **(PRE)**
- ▶ Intuitivamente si può **rafforzare la preconditione** ed **indebolire la postcondizione**

Assioma per Skip

- ▶ Assioma per il **comando vuoto**:

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

- ▶ **Correttezza?** Ovvvia...
- ▶ 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** eseguito in uno stato σ :
 - ▶ L'esecuzione di $x := E$ porta nello stato σ' dove $\sigma' = \sigma[\mathcal{E}(E, \sigma)/x]$
- ▶ 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'assegnamento ovvero dopo che il valore $\mathcal{E}(E, \sigma)$ è stato sostituito alla variabile x

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 $\mathcal{E}(E, \sigma) = v$
- ▶ La definizione $def(-)$ procede per induzione su E
 - ▶ $def(c) = tt$ se $c \in Num$ o $c \in Bool$
 - ▶ $def(x) = tt$ e $x \in Ide$
 - ▶ $def(E \text{ op } E') = def(E) \wedge def(E')$ se

$$op \in \left\{ \begin{array}{l} +, -, =, \neq, <, >, \\ \leq, \geq, \text{and}, \text{or} \end{array} \right\}$$

- ▶ $def(E \text{ op } E') = def(E) \wedge def(E') \wedge E' \neq 0$ se $op \in \{div, mod\}$
- ▶ $def(not E) = def(E)$
- ▶ $def((E)) = def(E)$

Definizione di Espressioni: commenti

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

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

- ▶ 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 $x \in \text{free}(E)$, 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: il **comando** $x := E$ eseguito in uno stato σ porta nello stato σ' dove $\sigma' = \sigma[\mathcal{E}(E, \sigma)/x]$
- ▶ e ricordando che
 - ▶ $def(E)$ garantisce che l'espressione E sia definita
 - ▶ inoltre

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

Assegnamento Semplice: Esempi

Verificare le triple:

▶ $\{\mathbf{T}\} x := 5 \{x = 5\}$

- ▶ Dall'assioma (*ASS*) otteniamo la **precondizione**

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

▶ $\{x > 2\} x := 5 \{x = 5\}$

- ▶ 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

$$(x > 2)[x + 1/x] = (x + 1 > 2) \equiv (x > 1) \quad x = 5 \Rightarrow x > 1$$

Regola di Inferenza per l'Assegnamento

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

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

- ▶ Si basa sulla seguente istanza di (PRE), in cui la seconda premessa scomparedato che è un **assioma**:

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

Assioma per Assegnamento Multiplo

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

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

- ▶ Tutte le espressioni vengono valutate prima di tutti gli assegnamenti
- ▶ La correttezza si prova in modo analogo a quello dell'assegnamento semplice

Regola per la Sequenza di Comandi

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

$$(SEQ) \frac{\{P\} C_1 \{R\} \quad \{R\} C_2 \{Q\}}{\{P\} C_1; C_2 \{Q\}}$$

- ▶ Convinciamoci della correttezza confrontandola con la semantica informale:
 - ▶ L'esecuzione del comando $C_1; C_2$ a partire dallo stato σ porta nello stato σ' ottenuto eseguendo C_2 a partire dallo stato σ'' ottenuto dall'esecuzione di C_1 nello stato σ
 - ▶ R è una **asserzione intermedia** che rappresenta tutti gli stati σ'' ottenuti eseguendo il primo comando C_1 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 regola (**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 verificare una tripla in cui il comando è un **condizionale** (per induzione strutturale):

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

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

Esempio: comando condizionale

Verificare la seguente tripla:

```
{  $m = 0$  }  
  if  $x < y$   
    then  $m := y$   
    else  $m := x$   
  fi  
{  $m = \max(x, y)$  }
```

Soluzione: Verifica della Tripla

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

1. $m = 0 \Rightarrow \text{def}(x < y)$
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*:

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

2. Usando la **regola** (*ASS*) ci riduciamo a mostrare:

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

ovvero $m = 0 \wedge x < y \Rightarrow y = \max(x, y)$

3. Analogamente al caso 2)

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

Esercizi

- ▶ Verificare la seguente tripla:

$$\{s = (\sum i : i \in [0, x).i)\}$$

$$s, x := s + x; x + 1$$

$$\{s = (\sum i : i \in [0, x).i)\}$$

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

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

- ▶ Le seguenti triple sono soddisfatte? Perché?

- ▶ $\{x = N \wedge y = M\} x := y; y := x \{x = M \wedge y = N\}$
- ▶ $\{x = N \wedge y = M\} x, y := y, x \{x = M \wedge y = N\}$

Esercizi

Verificare la seguente tripla:

$$\{x = 5\}$$

$$x := 3;$$

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

$$\{x = 3 \wedge y = 7\}$$

- ▶ Per la **regola (SEQ)** dobbiamo trovare una **asserzione intermedia R** che soddisfi le seguenti triple:
 1. $\{x = 5\} x := 3 \{R\}$
 2. $\{R\} \text{if}(x = 3) \text{ then } y := 7 \text{ else } y := 5 \text{ fi } \{x = 3 \wedge y = 7\}$
- ▶ 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 conviene usare 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:

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