REGOLE DI INFERENZA PER TRIPLE DI HOARE: ASSEGNAMENTO, SEQUENZA E CONDIZIONALE

Corso di Logica per la Programmazione

RIASSUNTO

- Una tripla {P} C {R} è costituita da precondizione-comando-postcondizione
- La tripla {P} C {R} è **soddisfatta** se: per ogni stato σ che soddisfa P ($\sigma \models P$), l'esecuzione di C a partire da σ **termina**, e lo stato σ' in cui termina soddisfa R ($\sigma' \models R$)
- Obiettivo principale: **verificare che una tripla sia soddisfatta** (come *dimostrare che una proposizione è una tautologia* nel Calcolo Proposizionale, o *dimostrare che una formula è valida* nella Logica del Primo Ordine)
- Diremo anche, più semplicemente: **verificare una tripla**
- Quindi: introduciamo un **proof system** per verificare le triple
  - Assiomi – alcune triple che sono sempre soddisfatte
  - Regole di inferenza – per induzione strutturale
- "Correttezza" di assiomi + regole basata su semantica informale dei comandi

PROOF SYSTEM: REGOLE "PRE-POST"

- Ricordiamo che {P} è l'insieme di stati che soddisfano P
- Quindi $P \Rightarrow P'$ se e solo se $\{P\} \subseteq \{P'\}$

- Da questo e dalla definizione di "tripla soddisfatta" segue subito la correttezza delle sequenti regole già viste:

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

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

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

ASSIOMA PER SKIP

- **Assioma** per il comando vuoto:

$$\{P\}\ \textbf{skip}\ \{P\} \qquad \text{(SKIP)}$$

- Correttezza? Ovvia...

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

ASSIOMA PER ASSEGNAMENTO SEMPLICE

- **Assioma** per l'assegnamento semplice

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

dove *def*(E) è vera in uno stato σ se il valore di E in σ (cioè ***E***(E,σ)) è ben definito (come già visto)

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

- L'espressione *E* deve essere definita...

DEFINIZIONE DI ESPRESSIONI

- La funzione *def(_)*, applicata a un'espressione *E*, restituisce una asserzione tale che $\sigma \models def(E)$ se esiste un v tale che $\boldsymbol{E}(E, \sigma) = v$.
- Serve per garantire che la valutazione di *E* sia definita e termini.

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

CORRETTEZZA DELL'ASSIOMA PER ASSEGNAMENTO SEMPLICE

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

- La correttezza della regola si vede confrontando l'assioma con la semantica informale:
  - L'esecuzione dell'assegnamento **x := E** a partire dallo stato σ porta nello stato $\sigma[\boldsymbol{E}(E,\sigma)/x]$

  e ricordando che
  - per ogni variabile x

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

REGOLA 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:

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

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

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

- Esempio: Verificare la seguente tripla:
  - $\{x = 5\}\ x := x + 1\ \{x > 2\}$

ASSIOMA PER ASSEGNAMENTO MULTIPLO

- Generalizza quello per l'assegnamento singolo

$$\{def(E_1) \wedge \ldots \wedge def(E_k) \wedge P^{E_1,\ldots,E_k}_{x_1,\ldots,x_k}\}\ x_1, \ldots, x_k := E_1, \ldots, E_k\ \{P\} \quad \text{(ASS)}$$

- Tutte le espressioni vengono valutate PRIMA di tutti gli assegnamenti: confrontiamo con la semantica informale
  - L'esecuzione dell'assegnamento **x1, ..., xn := E1, ..., En** a partire dallo stato σ porta nello stato σ[*E(E1,σ)*/x1, ..., *E(En,σ)*/xn]
- Nota che gli assiomi per l'assegnamento consentono di **completare** una tripla di cui si conosce solo il comando (l'assegnamento) e la postcondizione, "propagando" all'indietro l'asserzione.

REGOLA PER LA SEQUENZA DI COMANDI

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

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

- 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.
- Si verifichi che la seguente tripla è soddisfatta
  - {x >= y - 1} x := x+1; y := y - 1 {x > y}

ESEMPIO DI SEQUENZA DI COMANDI

Verificare la tripla: $\{x >= y - 1\}\ x := x+1; y := y - 1\ \{x > y\}$

- Per la **Regola per la Sequenza**, dobbiamo trovare una asserzione **R** e verificare le seguenti triple:

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

2) $\{\mathbf{R}\}\ y := y - 1\ \{x > y\}$

- Per determinare **R**, usiamo l'**Assioma dell'Assegnamento** nella seconda tripla. Sappiamo infatti che la seguente è verificata:

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

- Quindi fissiamo **R** = $def(y-1) \wedge x > y-1$. Resta da verificare:

$$\{x \geq y - 1\}\ x := x+1\ \{def(y-1) \wedge x > y-1\}$$

- Usando la **Regola per l'Assegnamento**, basta dimostrare:

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

- **Esercizio:** completare la dimostrazione

REGOLA PER IL COMANDO CONDIZIONALE

- Ricordiamo il significato informale del comando condizionale:
  - L'esecuzione di **if E then C1 else C2 fi** a partire da σ porta nello stato σ' che si ottiene dall'esecuzione di **C1** in σ, se $E(E,\sigma) = \mathit{tt}$, e dall'esecuzione di **C2** in σ, *se* $E(E,\sigma) = \mathit{ff}$
- Per verificare una tripla che ha come comando un **if-then-else** possiamo usare la regola seguente:

$$\frac{P \Rightarrow \mathit{def}(E) \qquad \{P \wedge E\}\ C1\ \{Q\} \qquad \{P \wedge \sim E\}\ C2\ \{Q\}}{\{P\}\ \textbf{if}\ E\ \textbf{then}\ C1\ \textbf{else}\ C2\ \textbf{fi}\ \{Q\}}\ (IF)$$

- La correttezza della regola segue dal confronto con il significato informale del condizionale

ESERCIZIO

- Si verifichi la seguente tripla:

{m = 0}

if x < y

then m := y

else m := x

fi

{m = max(x, y)}

SOLUZIONE

Verificare la seguente tripla:

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

- Per la **Regola per il Condizionale**, dobbiamo mostrare:

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

- Il punto 1) è facile.
- Per il 2), usando la **Regola per l'Assegnamento** occorre mostrare:
  - $m = 0 \ \wedge x < y \Rightarrow def(y) \wedge (m = max(x, y))[y/m]$
- Analogamente, per il 3) occorre mostrare:
  - $m = 0 \ \wedge x \geq y \Rightarrow def(x) \wedge (m = max(x, y))[x/m]$

- **Esercizio**: completare la dimostrazione

SEQUENZA CON **VARIABILI DI SPECIFICA**

Determinare E in modo che la tripla sia verificata:

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

- Per la **Regola per la Sequenza**, dobbiamo trovare **R1** e **R2** tali che:

1) $\{x = N \wedge y = M\}\ t := E\ \{\mathbf{R1}\}$
2) $\{\mathbf{R1}\}\ x := y\ \{\mathbf{R2}\}$
3) $\{\mathbf{R2}\}\ y := t\ \ \{x = M \wedge y = N\}$

Nota: M e N sono **variabili di specifica**: non possono essere usate nei comandi

- Per determinare **R2**, usiamo l'**assioma (ASS)** in 3):

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

- Fissiamo **R2 =** $(x = M \wedge t = N)$. Per **R1**, usiamo assioma (ASS) in 2):

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

- Fissiamo **R1 =** $(y = M \wedge t = N)$. Resta da verificare:

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

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

$$x = N \wedge y = M \Rightarrow def(E) \wedge (y = M \wedge t = N)[E/t]$$

ESERCIZI

- Verificare la seguente tripla:

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

- La seguente tripla non è soddisfatta. Mostrare formalmente perché.
  - $\{z = 5 \wedge y = 7 \wedge x = 3\}\ x := 0;\ y := z$ **div** $x\ \{z > 4\}$
- Le seguenti triple sono soddisfatte? Perché?
  - $\{x = N \wedge y = M\}\ \ x := y;\ \ y := x\ \ \{x = M \wedge y = N\}$
  - $\{x = N \wedge y = M\}\ \ x, y := y, x\ \ \{x = M \wedge y = N\}$

ESEMPIO

- Si verifichi la seguente tripla:

$\{x >= 0 \wedge y = (\Sigma\ i: i \in [0,x) \wedge i \% 6 = 0.\ i)\}$

if $x \% 6 = 0$

then $y := y + x$

else skip **fi**

$\{y = (\Sigma\ i: i \in [0,x] \wedge i \% 6 = 0.\ i)\}$

- Suggerimento: usare la legge dell'intervallo per Σ:

$(\sum x : x \in [a,b] \wedge P.E) =$

$(\sum x : x \in [a,b) \wedge P.\ E) + E[b/x]$ se $P[b/x]$

$(\sum x : x \in [a,b) \wedge P.\ E)$ se $\sim P[b/x]$

- L'esempio è parte dell'es. 5 del compito del 7/2/2012. La soluzione è scaricabile da http://www.di.unipi.it/~andrea/Didattica/LPP-A-11/

ESERCIZIO
Verificare la seguente tripla:

```
{ x = 5 }
   x := 3 ;
   if  ( x = 3 )
         then      y := 7
         else      y := 5
   fi
{ x = 3  ∧ y = 7 }
```

- Per la **Regola per la Sequenza,** dobbiamo trovare una asserzione **R** tale che:

1) { x = 5 } x := 3 { **R** }
2) {**R**} **if** (x=3) **then** y := 7 **else** y := 5 **fi** {x=3 ∧ 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}. Infatti con questa asserzione (e con un po' di attenzione) sia 1) che 2) sono facilmente dimostrabili.
- **Esercizio**: completare la dimostrazione