LOGICA PER LA PROGRAMMAZIONE – a.a. 2019/20

Esercitazione Pretest Online
Triple di Hoare

A causa dell'emergenza COVID, le modalità di esame del corso di Logica per la Programmazione vengono modificate in modo da permette agli studenti di sostenere l'esame a distanza. La nuova modalità d'esame prevede una prova orale a cui si accede superando un pretest.

Il pretest ha la finalità di accertare le conoscenze di base degli studenti ed alcune elementari capacità deduttive. Il pretest consiste di tre parti:
Calcolo Proposizionale, Logica del Primo Ordine e Triple di Hoare.

Le domande concernenti Calcolo Proposizionale e Logica del Primo Ordine sono analoghe a quelle previste dalla precedente modalità d'esame (prova scritta). Per quanto riguarda le Triple di Hoare invece l'enfasi non è concentrata sul sistema di dimostrazione, come lo era nella precedente modalità d'esame, ma piuttosto sulle nozioni di base, come ad esempio la semantica del linguaggio imperativo, il predicato *def* e la nozione di tripla soddisfatta.

I docenti, preoccupati dal fatto che alcuni studenti possano trovarsi disorientati, hanno preparato queste note che illustrano tutte le tipologie di domande che potrebbero capitare sulle Triple di Hoare, assieme alle loro soluzioni. Le soluzioni sono spesso dettagliate e talvolta illustrano i ragionamenti formali che però non sono richiesti agli studenti per superare il pretest.

Come leggere queste note

Si consiglia di provare prima su Moodle il test "Test di prova su Triple di Hoare" e quindi di confrontare le proprie risposte con le soluzioni spiegate in questo documento.

ESERCIZIO 1 [Controesempio]
La seguente tripla non è soddisfatta:

$\{\mathbf{T}\}$
```
while (x > 0) do
     x := x - 1
endw
```
$\{x = 0\}$

Quali dei seguenti stati iniziali fornisce un controesempio?

- $x \mapsto 5$,
- $x \mapsto 0$.
- $x \mapsto -1$,

SOLUZIONE ESERCIZIO 1
Per risolvere questo esercizio è necessario richiamare la definizione di tripla soddisfatta:
La tripla $\{P\}\, C\, \{Q\}$ **è soddisfatta** se:

1. *per ogni* stato iniziale σ che soddisfa la precondizione P,
2. l'esecuzione del comando C a partire dallo stato σ **termina** producendo *uno stato* σ', *e*
3. σ' soddisfa la postcondizione Q.

Per dimostrare che una tripla **non è soddisfatta** si deve quindi mostrare *uno stato iniziale* σ che soddisfa P, ma per cui *o* l'esecuzione di C non termina, *o* lo stato di arrivo σ' non soddisfa Q.

- Consideriamo lo stato iniziale $x \mapsto 5$. Tale stato chiaramente soddisfa la precondizione $\mathbf{T}$. L'esecuzione del comando a partire da tale stato termina nello stato $x \mapsto 0$ che, chiaramente soddisfa la postcondizione $x = 0$.
- Consideriamo lo stato iniziale $x \mapsto 0$. Tale stato chiaramente soddisfa la precondizione $\mathbf{T}$. L'esecuzione del comando a partire da tale stato termina (infatti il corpo del while non viene mai eseguito) nello stato $x \mapsto 0$ che, chiaramente soddisfa la postcondizione $x = 0$.
- Consideriamo lo stato iniziale $x \mapsto -1$. Tale stato chiaramente soddisfa la precondizione $\mathbf{T}$. L'esecuzione del comando a partire da tale stato termina (infatti il corpo del while non viene mai eseguito) nello stato $x \mapsto -1$ che, chiaramente *non* soddisfa la postcondizione $x = 0$.

L'unico controesempio è quindi lo stato iniziale $x \mapsto -1$.

ESERCIZIO 2 [Controesempio]
La seguente tripla non è soddisfatta:

$\{x = A \ \wedge \ y = B \ \wedge \ y \geq 0\}$
```
while (x > 0) do
     x := x - 1 ;
     y := y + 1
endw
```
$\{x = 0 \ \wedge \ y = A + B\}$

Quali dei seguenti stati iniziali fornisce un controesempio?

- $x \mapsto 5, \ y \mapsto 3,$
- $x \mapsto -3, \ y \mapsto -6.$
- $x \mapsto -1, \ y \mapsto 6,$

SOLUZIONE ESERCIZIO 2
Per risolvere questo esercizio è necessario richiamare la definizione di tripla soddisfatta:
La tripla $\{P\}\, C\, \{Q\}$ **è soddisfatta** se:

1. *per ogni* stato iniziale σ che soddisfa la precondizione P,
2. l'esecuzione del comando C a partire dallo stato σ **termina** producendo *uno stato* σ', *e*
3. σ' soddisfa la postcondizione Q.

Per dimostrare che una tripla **non è soddisfatta** si deve quindi mostrare *uno stato iniziale* σ che soddisfa P, ma per cui *o* l'esecuzione di C non termina, *o* lo stato di arrivo σ' non soddisfa Q.

- Consideriamo lo stato iniziale $x \mapsto 5, \ y \mapsto 3$. Tale stato chiaramente soddisfa la precondizione $x = A \ \wedge \ y = B \ \wedge \ y \geq 0$. L'esecuzione del comando a partire da tale stato termina nello stato $x \mapsto 0, y \mapsto 8$ che, chiaramente soddisfa la postcondizione $x = 0 \ \wedge \ y = A + B$.
- Consideriamo lo stato iniziale $x \mapsto -3, \ y \mapsto -6$. Tale stato chiaramente *non* soddisfa la precondizione $x = A \ \wedge \ y = B \ \wedge \ y \geq 0$ perché $y < 0$. Pertanto tale stato non fornisce un controesempio.
- Consideriamo lo stato iniziale $x \mapsto -1, \ y \mapsto 6$. Tale stato chiaramente soddisfa la precondizione $x = A \ \wedge \ y = B \ \wedge \ y \geq 0$. L'esecuzione del comando a partire da tale stato termina (infatti il corpo del while non viene mai eseguito) nello stato $x \mapsto -1, y \mapsto 6$ che, chiaramente *non* soddisfa la postcondizione $x = 0 \ \wedge \ y = A + B$.

L'unico controesempio è quindi lo stato iniziale $x \mapsto -1, \ y \mapsto 6$.

ESERCIZIO 3 [Semantica]
Per quale postcondizione Q la seguente tripla è soddisfatta?

$$\{x > 0 \ \wedge \ y < 0\} \ \mathtt{x, y} := \mathtt{x} + \mathtt{y}, \ \mathtt{x} * \mathtt{y} \ \{Q\}$$

(a) $Q = x \geq 0$

(b) $Q = x > 0 \wedge y \leq 0$

(c) $Q = y < 0$

(d) Nessuna delle precedenti.

SOLUZIONE ESERCIZIO 3

(a) La tripla $\{x > 0 \wedge y < 0\} \, \mathtt{x, y} := \mathtt{x} + \mathtt{y}, \ \mathtt{x} * \mathtt{y} \, \{x \geq 0\}$ non è soddisfatta: per vederlo basta costruire un controesempio (vedi Esercizio 1 per maggiori dettagli): lo stato iniziale $x \mapsto 1, \ y \mapsto -2$ soddisfa la precondizione $\{x > 0 \wedge y < 0\}$ e l'esecuzione del comando $\mathtt{x, y} := \mathtt{x} + \mathtt{y}, \ \mathtt{x} * \mathtt{y}$ a partire da tale stato porta nello stato $x \mapsto -1, \ y \mapsto -2$ che non soddisfa la postcondizione $x \geq 0$.

(b) La tripla $\{x > 0 \wedge y < 0\} \, \mathtt{x, y} := \mathtt{x} + \mathtt{y}, \ \mathtt{x} * \mathtt{y} \, \{x > 0 \wedge y \leq 0\}$ non è soddisfatta: per vederlo basta considerare lo stesso controesempio del punto (a).

(c) La tripla $\{x > 0 \wedge y < 0\} \, \mathtt{x, y} := \mathtt{x} + \mathtt{y}, \ \mathtt{x} * \mathtt{y} \, \{y < 0\}$ è soddisfatta perché il prodotto di un intero positivo per un intero negativo risulta sempre in un intero negativo.

ESERCIZIO 4 [Semantica]
Per quale postcondizione Q la seguente tripla è soddisfatta?

$$\{x > 0 \wedge y < 0\} \, \mathtt{x} := \mathtt{x} + \mathtt{y} \ ; \ \mathtt{y} := \mathtt{x} * \mathtt{y} \, \{Q\}$$

(a) $Q = x \geq 0$

(b) $Q = x > 0 \wedge y \leq 0$

(c) $Q = y < 0$

(d) Nessuna delle precedenti.

SOLUZIONE ESERCIZIO 4

(a) La tripla $\{x > 0 \wedge y < 0\} \, \mathtt{x} := \mathtt{x} + \mathtt{y} \ ; \ \mathtt{y} := \mathtt{x} * \mathtt{y} \, \{x \geq 0\}$ non è soddisfatta: per vederlo basta costruire un controesempio (vedi Esercizio 1 per maggiori dettagli): lo stato iniziale $x \mapsto 1, \ y \mapsto -2$ soddisfa la precondizione $\{x > 0 \wedge y < 0\}$ e l'esecuzione del comando $\mathtt{x} := \mathtt{x} + \mathtt{y} \ ; \ \mathtt{y} := \mathtt{x} * \mathtt{y}$ a partire da tale stato porta nello stato $x \mapsto -1, \ y \mapsto 2$ che non soddisfa la postcondizione $x \geq 0$.

(b) La tripla $\{x > 0 \wedge y < 0\} \, \mathtt{x} := \mathtt{x} + \mathtt{y} \ ; \ \mathtt{y} := \mathtt{x} * \mathtt{y} \, \{x > 0 \wedge y \leq 0\}$ non è soddisfatta: per vederlo basta considerare lo stesso controesempio del punto (a).

(c) La tripla $\{x > 0 \wedge y < 0\} \, \mathtt{x} := \mathtt{x} + \mathtt{y} \ ; \ \mathtt{y} := \mathtt{x} * \mathtt{y} \, \{y < 0\}$ non è soddisfatta: per vederlo basta considerare lo stesso controesempio del punto (a).

Quindi la risposta corretta la (d).

ESERCIZIO 5 [Specifica]
Si consideri la seguente tripla:

$$\{(\forall i.i \in [0,n) \ \wedge \ a[i] > b[i]) \ \wedge \ x = n\}$$
$$\texttt{if} \ \ E \ \ \texttt{then} \ \ C \ \ \texttt{else x:= x+1 fi}$$
$$\{(\forall i.i \in [0,n] \ \wedge \ a[i] > b[i]) \ \wedge \ x = n+1\}$$

dove $dom(\mathbf{a}) = dom(\mathbf{b}) > n$. Per quale espressione E e quale comando C la tripla è soddisfatta?

E: (a) $a[x] > b[x]$, (b) $a[x] = b[x]$, (c) $a[x] \leq b[x]$

C: (d) `a[x]:= b[x]; x:= x+1`, (e) `a[x],x := b[x]+1, x+1`, (f) `b[x]:= a[x]`

SOLUZIONE ESERCIZIO 5
La soluzione corretta è (c) $E = a[x] \leq b[x]$ e (e) $C = (a[x], x := b[x] + 1, x + 1)$.

Per arrivare a tale conclusione conviene dapprima esaminare pre- e postcondizione. Da un loro confronto risulta che il comando condizionale deve garantire che la proprietà $a[i] > b[i]$, che nello stato iniziale vale per $i \in [0, n)$, valga alla fine per $i \in [0, n]$, quindi in particolare deve garantire che $a[n] > b[n]$, dove $n = x$ per la precondizione. Quindi il comando dovrà necessariamente modificare $a[x]$ oppure $b[x]$ se nello stato iniziale vale $a[x] \leq b[x]$. Ragioniamo ora per esclusione.

- Per (a) $E = a[x] > b[x]$, l'esecuzione del comando `if` a partire da uno stato in cui $a[x] \leq b[x]$ consiste nell'esecuzione del ramo `else` che non modifica né $a[x]$ né $b[x]$. Pertanto nello stato di arrivo vale ancora $a[n] \leq b[n]$ e quindi tale stato non soddisfa la postcondizione.
- Per (b) $E = a[x] = b[x]$, si può ragionare in modo analogo: (si parta da uno stato in cui $a[x] < b[x]$).
- Per (c) $E = a[x] \leq b[x]$, l'esecuzione del ramo `else` porta invece in uno stato che soddisfa la postcondizione: $a[n] > b[n]$ e $x = n+1$. Ci possiamo quindi occupare del comando C nel ramo `then`, cioè a partire da uno stato iniziale in cui $a[x] \leq b[x]$.
  - Per (d) $C =$ `a[x]:= b[x]; x:= x+1`, l'esecuzione porta in uno stato in cui $a[n] = b[n]$ e quindi non soddisfa la postcondizone.
  - Per (e) $C =$ `a[x],x := b[x]+1, x+1`, l'esecuzione porta in uno stato in cui $a[n] = b[n]+1$ e $x = n + 1$ che quindi soddisfa la postcondizione.
  - Per (f) $C =$ `b[x]:= a[x]`, l'esecuzione porta in uno stato in cui $a[n] = b[n]$ e quindi non soddisfa la postcondizone.

ESERCIZIO 6 [Specifica]
Assumendo che $dom(\mathbf{a}) > n$, si consideri la seguente tripla:

$$\{(\sum i : i \in [0,x)\,.\,a[i]) = y \ \wedge \ x < n\} \ \ C \ \ \{(\sum i : i \in [0,x)\,.\,a[i]) = y\}$$

Quale dei seguenti comandi deve essere sostituito a C perché la tripla sia soddisfatta?

(a) `x:= x+1`

(b) `x:= x+1; y:=a[x]`

(c) `x,y := x+1,a[x]`

(d) `x:= x+1; y:=a[x]+y`

(e) `x,y := x+1,a[x]+y`

(f) Nessuno dei precedenti

SOLUZIONE ESERCIZIO 6
La soluzione corretta è (e) $C =$ `x,y := x+1,a[x]+y`.

Infatti confrontando la pre- e la postcondizione si vede che il comando deve preservare la proprietà $(\sum i : i \in [0,x)\,.\,a[i]) = y$. D'altro canto tutti i comandi proposti incrementano la x di una unità, quindi la sommatoria nello stato finale è estesa ad un elemento in più dell'array, e precisamente all'elemento $a[m]$ dove m è il valore di x nello stato iniziale. Ragioniamo quindi per esclusione fissando uno stato iniziale σ in cui x vale m.

(a) L'esecuzione del comando `x:= x+1` porta in uno stato in cui x vale $m+1$. Dalle precondizioni sappiamo che $(\sum i : i \in [0,m).a[i]) = y$, ma niente garantisce che $(\sum i : i \in [0,m+1).a[i]) = y$.

(b) L'esecuzione del comando `x:= x+1; y=a[x]` porta in uno stato in cui x vale $m+1$ e y vale $a[m+1]$ e non $(\sum i : i \in [0,m+1).a[i])$.

(c) L'esecuzione del comando `x,y := x+1,a[x]` porta in uno stato in cui x vale $m+1$ e y vale $a[m]$ e non $(\sum i : i \in [0,m+1).a[i])$.

(d) L'esecuzione del comando `x:= x+1; y=a[x]+y` porta in uno stato in cui x vale $m+1$ e y vale $a[m+1] + (\sum i : i \in [0,m).a[i])$ che non è uguale a $(\sum i : i \in [0,m+1).a[i])$.

(e) L'esecuzione del comando `x,y := x+1,a[x]+y` porta in uno stato in cui x vale $m+1$ e y vale $a[m] + (\sum i : i \in [0,m).a[i])$ che è uguale a $(\sum i : i \in [0,m+1).a[i])$.

ESERCIZIO 7 [Precondizione]
Per quale precondizione P la seguente tripla è soddisfatta?

$$\{P\}\, \mathtt{x} := \mathtt{x} + \mathtt{y}\ ;\ \mathtt{a}[\mathtt{x}] := \mathtt{a}[\mathtt{x} - \mathtt{1}] * \mathtt{2}\, \{\mathbf{T}\}$$

(a) $P \equiv a[x-1] * 2 \in dom(\mathbf{a})$

(b) $P \equiv x \in dom(\mathbf{a}) \ \wedge\ x - 1 \in dom(\mathbf{a})$

(c) $P \equiv x + y \in dom(\mathbf{a}) \ \wedge\ x + y - 1 \in dom(\mathbf{a})$

(d) Nessuna delle precedenti.

SOLUZIONE ESERCIZIO 7
Visto che la postcondizione è $\mathbf{T}$, la tripla è soddisfatta se per ogni stato σ che soddisfa la precondizione P, l'esecuzione del comando $\mathtt{x} := \mathtt{x} + \mathtt{y}\ ;\ \mathtt{a}[\mathtt{x}] := \mathtt{a}[\mathtt{x} - \mathtt{1}] * \mathtt{2}$ termina correttamente. Si deve quindi verificare che

1. l'esecuzione di $\mathtt{x} := \mathtt{x} + \mathtt{y}$ nello stato σ porti ad un nuovo stato σ';
2. l'esecuzione di $\mathtt{a}[\mathtt{x}] := \mathtt{a}[\mathtt{x} - \mathtt{1}] * \mathtt{2}$ nello stato σ porti ad un nuovo stato σ''.

Per 1., non c'è niente da verificare visto che l'espressione $x + y$ è sempre ben definita per ogni stato σ. Invece per 2., è necessario accertarsi che la valutazione di x e di $x - 1$ in σ' appartengano al dominio dell'array $\mathbf{a}$. Si noti che per la definizione della semantica $\sigma'(x) = \sigma(x) + \sigma(y)$. Si deve quindi richiedere che nello stato iniziale σ, $x + y \in dom(\mathbf{a})$ e $x + y - 1 \in dom(\mathbf{a})$.

Dunque la risposta corretta è la (c).

ESERCIZIO 8 [Precondizione]
Sia $dom(\mathbf{a}) = [\mathbf{0}, \mathbf{n})$. Per quale precondizione P la seguente tripla è soddisfatta?

$$\{P\}\, \mathtt{a}[\mathtt{x}] := \mathtt{a}[(\mathtt{x} + \mathtt{1})\, \mathtt{mod}\, \mathtt{y}] * \mathtt{2}\, \{\mathbf{T}\}$$

(a) $P \equiv y \neq 0 \ \wedge\ x \in dom(\mathbf{a})$

(b) $P \equiv x \in dom(\mathbf{a}) \ \wedge\ y \in (0, n]$

(c) $P \equiv y \neq 0 \ \wedge\ x + 1 \in dom(\mathbf{a})$

(d) Nessuna delle precedenti.

SOLUZIONE ESERCIZIO 8
Visto che la postcondizione è $\mathbf{T}$, la tripla è soddisfatta se per ogni stato σ che soddisfa la precondizione P, l'esecuzione del comando $\mathtt{a}[\mathtt{x}] := \mathtt{a}[(\mathtt{x} + \mathtt{1})\, \mathtt{mod}\, \mathtt{y}] * \mathtt{2}$ termina correttamente. Si deve quindi verificare che, nello stato σ, (1) la valutazione di x appartenga a $dom(\mathbf{a})$ e (2) l'espressione $a[(x+1)\, mod\, y] * 2$ sia ben definita. Per (1), è sufficiente richiedere che $x \in dom(\mathbf{a})$. Per (2), che:

(2.a) $y \neq 0$,

(2.b) la valutazione di $(x + 1)\ mod\ y$ nello stato σ appartenga a $dom(\mathbf{a})$.

Si osservi che il vincolo (1) non viene imposto dalla precondizione (c) che quindi non soddisfa la tripla. Il vincolo (2b) non viene imposto dalla precondizione (a) che quindi non soddisfa la tripla. Infine la precondizione (b) soddisfa tutti e tre i vincoli, e quindi la tripla. In particolare, per il (2b) si osservi che se $y \in (0, n]$ allora $x + 1\, mod\, y \in [0, n) = dom(\mathbf{a})$ indipendentemente dal valore di x.