LOGICA PER LA PROGRAMMAZIONE - a.a. 2019-2020

Seconda Prova di Verifica Intermedia - 20/12/2019

Soluzioni Proposte

Attenzione: Le soluzioni che seguono sono considerate corrette dai docenti. Per ogni esercizio possono esistere altre soluzioni corrette, anche molto diverse da quelle proposte.

ESERCIZIO 1

Assumendo che P e Q contengano la variabile libera x, si provi che la seguente formula è valida:

$$(\forall x . \neg Q \Rightarrow R) \wedge (\exists x . P \Rightarrow \neg Q) \;\Rightarrow\; \neg(\, (\forall x . P) \wedge (\forall x . \neg R)\,)$$

SOLUZIONE ESERCIZIO 1

Semplifichiamo la conseguenza:

$$\begin{array}{ll}
 & \neg(\, (\forall x . P) \wedge (\forall x . \neg R)\,) \\
\equiv & \{(\text{De Morgan})\} \\
 & \neg(\forall x . P) \vee \neg(\forall x . \neg R) \\
\equiv & \{(\text{De Morgan}), (\text{doppia negazione})\} \\
 & (\exists x . \neg P) \vee (\exists x . R) \\
\equiv & \{(\exists : \vee)\} \\
 & (\exists x . \neg P \vee R) \\
\equiv & \{(\text{elim-}\Rightarrow), \text{ al contrario}\} \\
 & (\exists x . P \Rightarrow R)
\end{array}$$

A questo punto utilizzando la regola della **Skolemizzazione** è sufficiente dimostrare che:

$$(\forall x . \neg Q \Rightarrow R) \wedge (\exists x . P \Rightarrow \neg Q) \wedge (P \Rightarrow \neg Q)[^a/_x] \;\Rightarrow\; (\exists x . P \Rightarrow R)$$

con a costante nuova. Per dimostrare la formula partiamo dalla premessa:

$$\begin{array}{ll}
 & (\forall x . \neg Q \Rightarrow R) \wedge \underline{(\exists x . P \Rightarrow \neg Q)} \wedge (P \Rightarrow \neg Q)[^a/_x] \\
\Rightarrow & \{(\text{sempl-}\wedge), \text{ occor. pos.}\} \\
 & \underline{(\forall x . \neg Q \Rightarrow R)} \wedge (P \Rightarrow \neg Q)[^a/_x] \\
\Rightarrow & \{(\text{elim-}\forall), \text{ occor. pos.}\} \\
 & (\neg Q \Rightarrow R)[^a/_x] \wedge (P \Rightarrow \neg Q)[^a/_x] \\
\equiv & \{\text{ sostituzione }\} \\
 & (\neg Q[^a/_x] \Rightarrow R[^a/_x]) \wedge (P[^a/_x] \Rightarrow \neg Q[^a/_x]) \\
\Rightarrow & \{(\text{transitività}), \text{ occ. pos.}\} \\
 & P[^a/_x] \Rightarrow R[^a/_x] \\
\Rightarrow & \{(\text{intro-}\exists), \text{ occ. pos.}\} \\
 & (\exists x . P \Rightarrow R)
\end{array}$$

ESERCIZIO 2

Assumendo **a: array [0, n) of int** e **b: array [0, m) of int** con $n, m > 1$, si formalizzi il seguente enunciato con la logica del primo ordine:

"L'ultimo elemento dell'array **a** è uguale alla somma degli elementi dispari di **b**,
mentre tutti gli altri elementi di **a** sono uguali al massimo degli elementi pari di **b**."

SOLUZIONE ESERCIZIO 2

$a[n-1] = \big(\Sigma j : j \in [0,m) \wedge Dispari(b[j]) \,.\, b[j]\big) \;\wedge\; \big(\,\forall i \,.\, i \in [0,n-1) \Rightarrow a[i] = (\mathbf{max} j : j \in [0,m) \wedge Pari(b[j]) \,.\, b[j])\,\big)$

ESERCIZIO 3

Si dica se le seguenti triple sono soddisfatte, assumendo **a: array [0, n) of int** e **b: array [0, n) of int**. Se lo è, fornire una dimostrazione formale; se non lo è, fornire un controesempio.

1. $\{x = A \wedge y = B \wedge z = C\}$ $x := x + (y - z)\,;\, y := y + (x - z)$ $\{x = y\}$,
2. $\{x = A \wedge y = B \wedge z = C\}$ $x, y := x + (y - z), y + (x - z)$ $\{x = y\}$,

SOLUZIONE ESERCIZIO 3

1. La tripla non è verificata. Per mostrarlo, forniamo un controesempio, cioè uno stato σ che
   (a) soddisfa la precondizione ($\sigma \models x = A \wedge y = B \wedge z = C$), ma tale che
   (b) l'esecuzione del comando in σ porta in uno stato σ' che non soddisfa la postcondizione, ovvero $x = y$.

   Consideriamo lo stato σ tale che $\sigma(x) = 0$, $\sigma(y) = 0$ e $\sigma(z) = 1$. Eseguendo il primo assegnamento otteniamo lo stato σ'' definito come $\sigma''(x) = -1$, $\sigma''(y) = 0$ e $\sigma''(z) = 1$. Eseguendo il secondo assegnamento, otteniamo lo stato σ' definito come $\sigma''(x) = -1$, $\sigma''(y) = -2$ e $\sigma''(z) = 1$. Chiaramente σ' non soddisfa la postcondizione.

2. La tripla è verificata. Infatti, grazie alla regola dell'assegnamento multiplo è sufficiente dimostrare che
   $x = A \wedge y = B \wedge z = C \;\Rightarrow\; def(\,x + (y - z)\,) \wedge def(\,y + (x - z)\,) \wedge x = y[^{x+(y-z), y+(x-z)}/_{x,y}]$.
   Si dimostra partendo dalla conclusione:

   $def(\,x + (y - z)\,) \wedge def(\,y + (x - z)\,) \wedge x = y[^{x+(y-z), y+(x-z)}/_{x,y}]$

   $\equiv$ {definizione di *def*, sostituzione}

   $x + (y - z) = y + (x - z)$

   $\equiv$ {calcolo}

   $\mathbf{T}$

ESERCIZIO 4

Assumendo **a: array [0, n) of int** e **b: array [0, m) of int** con $m \geq n$, si verifichi la seguente tripla:

$\{x \in [1,n) \;\wedge\; \big(\forall i \,.\, i \in [0,x) \;\Rightarrow\; (b[i] > 0 \wedge a[i] > \big(\mathbf{max}\, j : j \in [0,i) \,.\, b[j]\big)\big)\}$

$a[x] := a[x-1] + b[x-1] + 1\,;$

$x := x + 1$

$\{\big(\forall i \,.\, i \in [0,x) \;\Rightarrow\; (a[i] > \big(\mathbf{max}\, j : j \in [0,i) \,.\, b[j]\big)\big)\}$

SOLUZIONE ESERCIZIO 4

Applicando la regola della **Composizione Sequenziale** dobbiamo trovare una asserzione R tale che le seguenti due triple siano soddisfatte.

1. $\{x \in [1,n) \;\wedge\; \big(\forall i \,.\, i \in [0,x) \;\Rightarrow\; (b[i] > 0 \wedge a[i] > (\mathbf{max}\, j : j \in [0,i) \,.\, b[j])\big)\}$ $a[x] := a[x-1] + b[x-1] + 1$ $\{R\}$
2. $\{R\}$ $x := x + 1$ $\{\big(\forall i \,.\, i \in [0,x) \;\Rightarrow\; (a[i] > (\mathbf{max}\, j : j \in [0,i) \,.\, b[j])\big)\}$

Per calcolare R, utilizziamo l'assioma dell' **Assegnamento** applicato alla seconda tripla.

$$R = def(x+1) \wedge \big(\forall i \,.\, i \in [0,x) \;\Rightarrow\; (a[i] > (\mathbf{max}\, j : j \in [0,i) \,.\, b[j]))\big)[^{x+1}/_x].$$

Per tale R, la seconda tripla è verificata. Per verificare la prima tripla, utilizzando la regola dell' **Aggionamento Selettivo**, si deve dimostrare che:

$$x \in [1,n) \;\wedge\; \big(\forall i \,.\, i \in [0,x) \;\Rightarrow\; (b[i] > 0 \;\wedge\; a[i] > (\mathbf{max}\, j : j \in [0,i) \,.\, b[j]))\big) \;\Rightarrow$$
$$def(x) \wedge def(a[x-1] + b[x-1] + 1) \wedge x \in dom(a) \wedge R[^c/_a]$$

dove $\mathbf{c} = \mathbf{a}[^{a[x-1]+b[x-1]+1}/_x]$.

Partiamo dalla conseguenza

$$
\begin{array}{ll}
 & def(x) \wedge def(a[x-1] + b[x-1] + 1) \wedge x \in dom(a) \wedge R[^c/_a] \\
\equiv & \{\text{definizione di } def.\} \\
 & x - 1 \in dom(a) \wedge x - 1 \in dom(b) \wedge x \in dom(a) \wedge R[^c/_a] \\
\equiv & \{\textbf{Ip}\text{: } x \in [1,n),\ dom(a) = [0,n),\ dom(b) = [0,m),\ m \geq n\} \\
 & R[^c/_a] \\
\equiv & \{\text{sostituzione}\} \\
 & def(x+1) \wedge \big(\forall i \,.\, i \in [0,x] \;\Rightarrow\; (c[i] > \big(\mathbf{max}\, j : j \in [0,i) \,.\, b[j]\big))\big) \\
\equiv & \{\text{definizione di } def.\} \\
 & \big(\forall i \,.\, i \in [0,x] \;\Rightarrow\; (c[i] > \big(\mathbf{max}\, j : j \in [0,i) \,.\, b[j]\big))\big) \\
\equiv & \{(\text{Intervallo-}\forall)\ \} \\
 & \big(\forall i \,.\, i \in [0,x) \;\Rightarrow\; (c[i] > \big(\mathbf{max}\, j : j \in [0,i) \,.\, b[j]\big))\big) \wedge c[x] > \big(\mathbf{max}\, j : j \in [0,x) \,.\, b[j]\big) \\
\equiv & \{\text{definizione di } \mathbf{c}\text{, quindi } (\forall i \,.\, i \in [0,k) \Rightarrow c[i] = a[i])\} \\
 & \big(\forall i \,.\, i \in [0,x) \;\Rightarrow\; (a[i] > \big(\mathbf{max}\, j : j \in [0,i) \,.\, b[j]\big))\big) \;\wedge\; a[x-1] + b[x-1] + 1 > \big(\mathbf{max}\, j : j \in [0,x) \,.\, b[j]\big) \\
\equiv & \big\{\textbf{Ip}\text{: } \big(\forall i \,.\, i \in [0,x) \;\Rightarrow\; (a[i] > \big(\mathbf{max}\, j : j \in [0,i) \,.\, b[j]\big))\big)\big\} \\
 & a[x-1] + b[x-1] + 1 > \big(\mathbf{max}\, j : j \in [0,x) \,.\, b[j]\big) \\
\equiv & \{(\text{Intervallo-}\mathbf{max})\} \\
 & a[x-1] + b[x-1] + 1 > \big(\mathbf{max}\, j : j \in [0,x-1) \,.\, b[j]\big)\ max\ b[x-1]
\end{array}
$$

Per dimostrare quest'ultima disuguaglianza, mostriamo le seguenti:

(a) $a[x-1] + b[x-1] + 1 > \big(\mathbf{max}\, j : j \in [0,x-1) \,.\, b[j]\big)$

(b) $a[x-1] + b[x-1] + 1 > b[x-1]$

Per la (a) osserviamo che per ipotesi vale $\big(\forall i \,.\, i \in [0,x) \;\Rightarrow\; (b[i] > 0 \wedge a[i] > (\mathbf{max}\, j : j \in [0,i) \,.\, b[j])\big)$, e quindi in particolare $a[x-1] > \big(\mathbf{max}\, j : j \in [0,x-1) \,.\, b[j]\big)$, da cui la (a) segue osservando che $b[x-1] > 0$, e quindi $a[x-1] + b[x-1] + 1 > a[x-1]$.

Per la (b) è sufficiente osservare che dalle ipotesi segue che $a[x-1]$ è maggiore di 0.

ESERCIZIO 5

Assumendo **a : array [0, n) of int**, si consideri il seguente frammento di programma annotato:

$\{s = 0, x = 0\}$

$\{\mathsf{Inv}\text{: } x \in [0,n] \;\wedge\; s = \big(\Sigma i : i \in [0,x) \wedge a[i]\%2 = 0 \,.\, a[i] * 2^{x-1-i}\big)\}\{\text{t: } n - x\}$

```
while (x < n) do
    if (a[x] % 2 = 0)
        then s, x := 2 * s + a[x], x + 1
        else s, x := 2 * s, x + 1
    fi
endw
```

$\{s = \big(\Sigma i : i \in [0,n) \wedge a[i]\%2 = 0 \,.\, a[i] * 2^{n-1-i}\big)\ \}$

Si scrivano le ipotesi di progresso ed invarianza. Inoltre si dimostri l'ipotesi di invarianza **limitatamente alle due prime condizioni della regola del comando condizionale (si ignori il ramo else)**.

SOLUZIONE ESERCIZIO 5

Invariante $Inv : x \in [0, n] \ \wedge \ s = \big(\Sigma i : i \in [0, x) \wedge a[i]\%2 = 0 \,.\, a[i] * 2^{x-1-i}\big)$
Funzione di terminazione $t : n - x$

1. **Ipotesi di Invarianza**:

$\{ \ x \in [0, n] \ \wedge \ s = \big(\Sigma i : i \in [0, x) \wedge a[i]\%2 = 0 \,.\, a[i] * 2^{x-1-i}\big) \ \wedge \ x < n\}$

```
if (a[x] % 2 = 0)
    then s, x := 2 * s + a[x], x + 1
    else s, x := 2 * s, x + 1
fi
```

$\{ \ x \in [0, n] \ \wedge \ s = \big(\Sigma i : i \in [0, x) \wedge a[i]\%2 = 0 \,.\, a[i] * 2^{x-1-i}\big) \ \wedge \ def(x < n) \ \}$

2. **Ipotesi di Progresso**:

$\{ \ x \in [0, n] \ \wedge \ s = \big(\Sigma i : i \in [0, x) \wedge a[i]\%2 = 0 \,.\, a[i] * 2^{x-1-i}\big) \ \wedge \ x < n \ \wedge \ n - x = V\}$

```
if (a[x] % 2 = 0)
    then s, x := 2 * s + a[x], x + 1
    else s, x := 2 * s, x + 1
fi
```

$\{ \ n - x < V \ \}$

Dimostriamo l'ipotesi di invarianza applicando la regola del **Condizionale**. Quindi dobbiamo verificare che

(5.1.1) $Inv \wedge x < n \Rightarrow def(a[x]\%2 = 0)$

(5.1.2) $\{Inv \wedge x < n \wedge (a[x]\%2 = 0)\}$ `s, x := 2 * s + a[x], x + 1` $\{Inv \wedge def(x < n)\}$

(5.1.3) $\{Inv \wedge x < n \wedge \neg(a[x]\%2 = 0)\}$ `s, x := 2 * s, x + 1` $\{Inv \wedge def(x < n)\}$

(5.1.1) Abbiamo che

$def(a[x]\%2 = 0)$

$\equiv$ {definizione di def}

$x \in dom(a)$

$\equiv$ {**Ip**: $dom(a) = [0, n), x \in [0, n], x < n$}

T

(5.1.2) Per dimostrare la tripla applichiamo la regola dell' **Assegnamento Multiplo** e ci riduciamo a dimostrare

$$Inv \wedge x < n \wedge (a[x]\%2 = 0) \ \Rightarrow \\ def(2 * s + a[x]) \wedge def(x + 1) \wedge \big(Inv \wedge def(x < n)\big)\big[^{2*s+a[x],x+1}/_{s,x}\big]$$

Partiamo dalla conseguenza

$def(2 * s + a[x]) \wedge def(x + 1) \wedge \underline{\big(Inv \wedge def(x < n)\big)\big[^{2*s+a[x],x+1}/_{s,x}\big]}$

$\equiv$ {sostituzione}

$\underline{def(2 * s + a[x])} \wedge \underline{def(x + 1)} \wedge Inv\big[^{2*s+a[x],x+1}/_{s,x}\big] \wedge \underline{def(x + 1 < n)}$

$\equiv$ {definizione di def}

$x \in dom(a) \wedge Inv\big[^{2*s+a[x],x+1}/_{s,x}\big]$

$\equiv$ {sostituzione, **Ip**: $dom(a) = [0, n), x \in [0, n], x < n$ }

$\underline{x+1 \in [0,n]} \ \wedge \ 2*s+a[x] = \big(\Sigma i : \underline{i \in [0,x+1)} \wedge a[i]\%2=0 \,.\, a[i]*\underline{2^{(x+1)-1-i}}\big)$

$\equiv$ { **Ip**: $x \in [0,n], x < n$ calcolo}

$2*s+a[x] = \big(\Sigma i : i \in [0,x] \wedge a[i]\%2=0 \,.\, a[i]*2^{x-i}\big)$

$\equiv$ {(Interv-Σ), **Ip**: (a[x]%2=0) }

$2*s+\underline{a[x]} = \big(\Sigma i : i \in [0,x) \wedge a[i]\%2=0 \,.\, a[i]*2^{x-i}\big) + \underline{a[x]}$

$\equiv$ $\big\{$calcolo, **Ip**: $s = \big(\Sigma i : i \in [0,x) \wedge a[i]\%2=0 \,.\, a[i]*2^{x-1-i}\big)$ $\big\}$

$\underline{2*}\big(\Sigma i : i \in [0,x) \wedge a[i]\%2=0 \,.\, a[i]*\underline{2^{x-1-i}}\big) = \big(\Sigma i : i \in [0,x) \wedge a[i]\%2=0 \,.\, a[i]*\underline{2^{x-i}}\big)$

$\equiv$ {calcolo (distributività di $*$ rispetto a $+$)}

T

(5.1.3) Non richiesta dal testo dell'esercizio, ma del tutto analoga alla (5.1.2).