LOGICA PER LA PROGRAMMAZIONE - a.a. 2017-2018
Seconda Prova di Verifica Intermedia - 21/12/2017 — 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, Q, R e S contengano la variabile libera x, si provi che la seguente formula è valida:

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

SOLUZIONE ESERCIZIO 1

Semplifichiamo la conseguenza:

$\underline{\neg(\forall x\,.\,S \wedge P)}$
$\equiv$ {(De Morgan)}
$(\exists x\,.\,\underline{\neg(S \wedge P)})$
$\equiv$ {(De Morgan)}
$(\exists x\,.\,\neg S \vee \neg P)$

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

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

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

$\underline{(\forall x\,.\,P \;\Rightarrow\; (\neg Q \;\Rightarrow\; R)) \wedge (\exists x\,.\,\neg S \vee ((\neg Q \wedge \neg R) \wedge S)) \wedge (\neg S(a) \vee ((\neg Q(a) \wedge \neg R(a)) \wedge S(a)))}$
$\Rightarrow$ {(sempl-$\wedge$), occor. pos.}
$(\forall x\,.\,P \;\Rightarrow\; (\neg Q \;\Rightarrow\; R)) \wedge \underline{(\neg S(a) \vee (\neg Q(a) \wedge \neg R(a)) \wedge S(a))}$
$\equiv$ {(complemento)}
$\underline{(\forall x\,.\,P \;\Rightarrow\; (\neg Q \;\Rightarrow\; R))} \wedge (\neg S(a) \vee (\neg Q(a) \wedge \neg R(a)))$
$\Rightarrow$ {(elim-$\forall$), occor. pos.}
$(\underline{P(a) \;\Rightarrow\; (\neg Q(a) \;\Rightarrow\; R(a))}) \wedge (\neg S(a) \vee (\neg Q(a) \wedge \neg R(a)))$
$\equiv$ {(elim- $\Rightarrow$)}
$(\neg P(a) \vee \underline{(\neg Q(a) \;\Rightarrow\; R(a))}) \wedge (\neg S(a) \vee (\neg Q(a) \wedge \neg R(a)))$
$\equiv$ {(doppia negazione)}
$(\neg P(a) \vee \neg\underline{\neg(\neg Q(a) \;\Rightarrow\; R(a))}) \wedge (\neg S(a) \vee (\neg Q(a) \wedge \neg R(a)))$
$\equiv$ {($\neg$- $\Rightarrow$)}
$\underline{(\neg P(a) \vee \neg(\neg Q(a) \wedge \neg R(a)) \wedge (\neg S(a) \vee (\neg Q(a) \wedge \neg R(a)))}$
$\Rightarrow$ {(risoluzione), occor. pos.}
$\underline{\neg P(a) \vee \neg S(a)}$
$\Rightarrow$ {(intro-$\exists$), occor. pos. }
$(\exists x\,.\,\neg P \vee \neg S)$

ESERCIZIO 2
Assumendo **a, b: array [0, n) of int**, si formalizzi il seguente enunciato:

"Ogni elemento dell'array **b** è uguale al minimo tra l'elemento corrispondente di **a** e la somma dei valori precedenti di **a**."

SOLUZIONE ESERCIZIO 2

$$(\forall x.\ x \in [0,n)\ \Rightarrow\ b[x] = a[x]\ \textbf{min}\ (\Sigma y : y \in [0,x).a[y]))$$

dove l'operatore binario **min** restituisce il minimo tra i due operandi.

ESERCIZIO 3
Si dica se le seguenti triple sono verificate. Se lo è, fornire una dimostrazione formale; se non lo è, fornire un controesempio.

1. $\{x = A\ \wedge\ y = B\ \wedge w = C\}$ `x := y - w; y := x - w` $\{x = B - C\ \wedge\ y = A - C\}$,
2. $\{x = A\ \wedge\ y = B\ \wedge w = C\}$ `x,y := y - w,x - w` $\{x = B - C\ \wedge\ y = A - C\}$.

SOLUZIONE ESERCIZIO 3

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

   Consideriamo lo stato $\sigma = \{(x,3),(y,2),(w,1)\}$. Eseguendo il primo assegnamento `x := y - w` nello stato σ otteniamo lo stato
   $$\sigma_1 = \sigma[^1/_x] = \{(x,1),(y,2),(w,1)\}.$$
   Eseguendo il secondo assegnamento `y := x - w` nello stato σ_1 otteniamo lo stato
   $$\sigma_2 = \sigma_1[^0/_y] = \{(x,1),(y,0),(w,1)\}.$$
   Si noti che lo stato σ_2 non soddisfa la postcondizione $x = B - C\ \wedge\ y = A - C$. Infatti le variabili di specifica A, B e C si riferiscono ai valori di x, y e w nella precondizione, rispettivamente. Quindi si dovrebbe avere $x = 2 - 1 = 1$ e $y = 3 - 1 = 2$
2. La tripla è verificata. Per mostrarlo applichiamo la regola dell' **Assegnamento Multiplo** e ci riduciamo a dimostrare che
   $$x = A\ \wedge\ y = B\ \wedge w = C\ \Rightarrow\ def(y - w) \wedge def(x - w) \wedge \big(x = B - C\ \wedge\ y = A - C\big)[^{y-w,x-w}/_{x,y}]$$
   Partiamo dalla conseguenza, applicando la sostituzione
   $def(y - w) \wedge def(x - w) \wedge \big(x = B - C\ \wedge\ y = A - C\big)[^{y-w,x-w}/_{x,y}]$
   $\equiv$ {definizione di def, sostituzione}
   $\big(y - w = B - C\ \wedge\ x - w = A - C\big)$
   $\equiv$ {**Ip**: $x = A, y = B, w = C$)}
   $\big(B - C = B - C\ \wedge\ A - C = A - C\big)$
   $\equiv$ {(Zero), (Unità)}
   T

ESERCIZIO 4
Assumendo **a,c: array [0, m) of int**, si verifichi la seguente tripla:

$\{k \in [1,m) \ \wedge \ \big(\forall i \,.\, i \in [0,k) \ \Rightarrow \ c[i] = \big(\Sigma j : j \in [0,i] \,.\, (a[j]+1)^2\big)\big)\}$

$\quad c[k] := c[k-1] + a[k]^2 + 2 * a[k] + 1$

$\{\big(\forall i \,.\, i \in [0,k] \ \Rightarrow \ c[i] = \big(\Sigma j : j \in [0,i] \,.\, (a[j]+1)^2\big)\big)\}$

SOLUZIONE ESERCIZIO 4

Applicando la regola dell' **Aggiornamento Selettivo** dobbiamo verificare che:

$$k \in [1,m) \ \wedge \ \big(\forall i \,.\, i \in [0,k) \ \Rightarrow \ c[i] = \big(\Sigma j : j \in [0,i] \,.\, (a[j]+1)^2\big)\big) \Rightarrow$$

$$k \in dom(c) \wedge def(k) \wedge def(c[k-1] + a[k]^2 + 2 * a[k] + 1) \wedge R[^b/_c]$$

dove $b = c[^{c[k-1]+a[k]^2+2*a[k]+1}/_k]$ e $R = \big(\forall i \,.\, i \in [0,k] \ \Rightarrow \ c[i] = \big(\Sigma j : j \in [0,i] \,.\, (a[j]+1)^2\big)\big)$.
Partiamo dalla conseguenza

$$\begin{array}{ll}
 & k \in dom(c) \wedge \underline{def(k) \wedge def(c[k-1] + a[k]^2 + 2 * a[k] + 1)} \wedge R[^b/_c] \\
\equiv & \{\text{definizione di } def\} \\
 & \underline{k \in dom(c) \wedge k \in dom(a) \wedge k - 1 \in dom(c)} \wedge R[^b/_c] \\
\equiv & \{\mathbf{Ip}\text{: } k \in [1,m) \wedge dom(a) = dom(c) = [0,m)\} \\
 & \underline{\big(\big(\forall i \,.\, i \in [0,k] \ \Rightarrow \ c[i] = \big(\Sigma j : j \in [0,i] \,.\, (a[j]+1)^2\big)\big)\big)[^b/_c]} \\
\equiv & \{\text{sostituzione}\} \\
 & \underline{\big(\forall i \,.\, i \in [0,k] \ \Rightarrow \ b[i] = \big(\Sigma j : j \in [0,i] \,.\, (a[j]+1)^2\big)\big)} \\
\equiv & \{(\text{Intervallo-}\forall), \mathbf{Ip}\text{: } k > 0\} \\
 & \underline{\big(\forall i \,.\, i \in [0,k) \ \Rightarrow \ b[i] = \big(\Sigma j : j \in [0,i] \,.\, (a[j]+1)^2\big)\big) \wedge b[k] = \big(\Sigma j : j \in [0,k] \,.\, (a[j]+1)^2\big)} \\
\equiv & \Big\{\text{definizione di } b = c[^{c[k-1]+a[k]^2+2*a[k]+1}/_k]\Big\} \\
 & \underline{\big(\forall i \,.\, i \in [0,k) \ \Rightarrow \ c[i] = \big(\Sigma j : j \in [0,i] \,.\, (a[j]+1)^2\big)\big) \wedge c[k-1] + a[k]^2 + 2 * a[k] + 1 = \big(\Sigma j : j \in [0,k] \,.\, (a[j]+1)^2\big)} \\
\equiv & \big\{\mathbf{Ip}\text{: } \big(\forall i \,.\, i \in [0,k) \ \Rightarrow \ c[i] = \big(\Sigma j : j \in [0,i] \,.\, (a[j]+1)^2\big)\big)\big\} \\
 & c[k-1] + a[k]^2 + 2 * a[k] + 1 = \underline{\big(\Sigma j : j \in [0,k] \,.\, (a[j]+1)^2\big)} \\
\equiv & \{(\text{Intervallo-}\Sigma)\} \\
 & \underline{c[k-1] + a[k]^2 + 2 * a[k] + 1 = \big(\Sigma j : j \in [0,k) \,.\, (a[j]+1)^2\big) + (a[k]+1)^2} \\
\equiv & \{\text{calcolo}\} \\
 & \underline{c[k-1]} = \big(\Sigma j : j \in [0,k) \,.\, (a[j]+1)^2\big) \\
\equiv & \big\{\mathbf{Ip}\text{: } c[k-1] = \big(\Sigma j : j \in [0,k) \,.\, (a[j]+1)^2\big)\big\} \\
 & \mathbf{T}
\end{array}$$

ESERCIZIO 5

Assumendo **a**, **b: array [0, n) of int**, si consideri il seguente frammento di programma annotato, in cui l'operatore binario `max` restituisce il maggiore tra i due operandi:

$\{cond = \mathbf{true} \ \wedge \ w = 0\}$

$\{\mathsf{Inv}\text{: } w \in [0,n] \ \wedge \ \big(cond \equiv (\forall i.i \in [0,w) \Rightarrow a[i] = b[i] \ \mathbf{max} \ i)\big)\}\{\mathrm{t}\text{: } n - w\}$

```
while ((w < n) and cond) do
     if (a[w] = b[w] max w)
          then w := w + 1
          else cond,w := false,w+1
     fi
endw
```

$\{cond \equiv (\forall i.i \in [0,n) \Rightarrow a[i] = b[i] \ \mathbf{max} \ i)\}$

Si scrivano le ipotesi di progresso ed invarianza. Inoltre si dimostri l'ipotesi di invarianza.

SOLUZIONE ESERCIZIO 5

Invariante $Inv : w \in [0,n] \ \wedge \ \big(cond \equiv (\forall i.i \in [0,w) \Rightarrow a[i] = b[i] \ \mathbf{max} \ i)\big)$
Funzione di terminazione $t : n - w$

1. **Ipotesi di Invarianza**:

$$\{w \in [0,n] \ \wedge \ \big(cond \equiv (\forall i.i \in [0,w) \Rightarrow a[i] = b[i] \ \mathbf{max} \ i)\big) \wedge w < n \wedge cond\}$$

```
if (a[w] = b[w] max w)  then w := w + 1 else  cond,w := false,w+1 fi
```

$$\{w \in [0,n] \ \wedge \ \big(cond \equiv (\forall i.i \in [0,w) \Rightarrow a[i] = b[i] \ \mathbf{max} \ i)\big) \ \wedge \ def(w < n \wedge cond)\}$$

2. **Ipotesi di Progresso**:

$$\{w \in [0,n] \ \wedge \ \big(cond \equiv (\forall i.i \in [0,w) \Rightarrow a[i] = b[i] \ \mathbf{max} \ i)\big) \wedge w < n \wedge cond \wedge n - w = V\}$$

```
if (a[w] = b[w] max w)  then w := w + 1 else  cond,w := false,w+1 fi
```

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

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

(5.1.1) $Inv \wedge w < n \wedge cond \Rightarrow def(a[w] = b[w] \ \texttt{max} \ w)$

(5.1.2) $\{Inv \wedge w < n \wedge cond \wedge (a[w] = b[w] \ \texttt{max} \ w)\}$ `w := w + 1` $\{Inv \wedge def(w < n \wedge cond)\}$

(5.1.3) $\{Inv \wedge w < n \wedge cond \wedge \neg(a[w] = b[w] \ \texttt{max} \ w)\}$ `cond,w := false,w + 1` $\{Inv \wedge def(w < n \wedge cond)\}$

(5.1.1) Abbiamo che

$def(a[w] = b[w] \ \texttt{max} \ w)$

$\equiv$ {definizione di *def*}

$w \in dom(a) \wedge w \in dom(b)$

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

T

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

$$Inv \wedge w < n \wedge cond \wedge (a[w] = b[w] \ \texttt{max} \ w) \ \Rightarrow \\ def(w+1) \wedge \big(Inv \wedge def(w < n \wedge cond)\big)[^{w+1}/_{w}]$$

Partiamo dalla conseguenza

$def(w+1) \wedge \big(Inv \wedge def(w < n \wedge cond)\big)[^{w+1}/_{w}]$

$\equiv$ {sostituzione}

$\underline{def(w+1)} \wedge w + 1 \in [0,n] \wedge \big(cond \equiv (\forall i.i \in [0,w+1) \Rightarrow a[i] = b[i] \ \mathbf{max} \ i)\big) \wedge \underline{def(w+1 < n \wedge cond)}$

$\equiv$ {definizione di *def*}

$\underline{w+1 \in [0,n]} \wedge \big(cond \equiv (\forall i.i \in [0,w+1) \Rightarrow a[i] = b[i] \ \mathbf{max} \ i)\big)$

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

$\big(cond \equiv \underline{(\forall i.i \in [0,w+1) \Rightarrow a[i] = b[i] \ \mathbf{max} \ i)}\big)$

$\equiv$ {(Intervallo-$\forall$)}

$\big(cond \equiv \underline{\big((\forall i.i \in [0,w) \Rightarrow a[i] = b[i] \ \mathbf{max} \ i)\big) \wedge (a[w] = b[w] \ \mathbf{max} \ w)}\big)$

$\equiv$ {**Ip**: $(a[w] = b[w] \ \texttt{max} \ w)$, (Unita')}

$\big(\underline{cond} \equiv (\forall i.i \in [0,w) \Rightarrow a[i] = b[i] \textbf{ max } i)\big)$

$\equiv$ $\big\{\textbf{Ip}\text{: } \big(cond \equiv (\forall i.i \in [0,w) \Rightarrow a[i] = b[i] \textbf{ max } i)\big)\big\}$

$\mathbf{T}$

(5.1.3) Applicando la regola dell' **Assegnamento Multiplo** ci riduciamo a dimostrare che

$$Inv \wedge w < n \wedge cond \wedge \neg(a[w] = b[w] \texttt{ max } w) \;\Rightarrow\; \underline{def(false) \wedge def(w+1)} \wedge \big(Inv \wedge def(w < n \wedge cond)\big)[^{w+1,false}/_{w,cond}]$$

Partiamo dalla conseguenza, applicando la sostituzione

$def(false) \wedge def(w+1) \wedge w+1 \in [0,n] \wedge \big(\mathbf{F} \equiv (\forall i.i \in [0,w+1) \Rightarrow a[i] = b[i] \textbf{ max } i)\big) \wedge \underline{def(w+1 < n \wedge cond)}$

$\equiv$ {definizione di def}

$\underline{w+1 \in [0,n]} \wedge \big(\mathbf{F} \equiv (\forall i.i \in [0,w+1) \Rightarrow a[i] = b[i] \textbf{ max } i)\big)$

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

$\mathbf{F} \equiv \underline{(\forall i.i \in [0,w+1) \Rightarrow a[i] = b[i] \textbf{ max } i)}$

$\equiv$ {(Intervallo-$\forall$)}

$\mathbf{F} \equiv \underline{\big((\forall i.i \in [0,w) \Rightarrow a[i] = b[i] \textbf{ max } i) \wedge (a[w] = b[w] \textbf{ max } w)\big)}$

$\equiv$ {**Ip**: $\neg(a[w] = b[w] \texttt{ max } w)$, (zero)}

$\mathbf{F} \equiv \mathbf{F}$

$\equiv$ {(rifl-$\equiv$)}

$\mathbf{T}$