Corso di Laurea in Informatica A.A. 2009/2010
Logica per la Programmazione
Secondo appello - 10 febbraio 2010

Soluzioni proposte dai docenti

---

a) **Si provi che la seguente formula del calcolo proposizionale è una tautologia:**

$$\mathbf{(P \wedge S \Rightarrow R) \wedge (\sim R \Rightarrow S) \wedge P \Rightarrow R}$$

Si suggerisce di usare una dimostrazione per casi su S.

Soluzione: Sia **Z** la formula da dimostrare. Mostriamo separatamente che $\mathbf{S \Rightarrow Z}$ e $\mathbf{\sim S \Rightarrow Z}$: questo è sufficiente come dimostrazione della validità di **Z**.

Caso $\mathbf{S \equiv true}$

$(P \wedge S \Rightarrow R) \wedge (\sim R \Rightarrow S) \wedge P$
$\equiv$ { **Ip: S**, Unità }
$(P \Rightarrow R) \wedge (\sim R \Rightarrow S) \wedge P$
$\Rightarrow$ { semplificazione-$\wedge$ }
$(P \Rightarrow R) \wedge P$
$\Rightarrow$ { Modus Ponens }
R

Caso $\mathbf{S \equiv false}$

$(P \wedge S \Rightarrow R) \wedge (\sim R \Rightarrow S) \wedge P$
$\Rightarrow$ { semplificazione-$\wedge$ }
$\sim R \Rightarrow S$
$\equiv$ { contraposizione, doppia negazione }
$\sim S \Rightarrow R$
$\equiv$ { **Ip: ~ S**, Modus Ponens }
R

---

b) **Si provi che la seguente formula della logica del primo ordine è valida, dove P e Q contengono la variabile libera x:**

$$\mathbf{(\forall x.\ P \Rightarrow \sim Q) \ \Rightarrow \ \sim (\exists x.\ P \wedge Q)}$$

Soluzione: $(\forall x.\ P \Rightarrow \sim Q)$
$\Rightarrow$ { eliminazione-$\Rightarrow$ }
$(\forall x.\ \sim P \vee \sim Q)$
$\equiv$ { De Morgan }
$(\forall x.\ \sim (P \wedge Q))$
$\Rightarrow$ { De Morgan }
$\sim (\exists x.\ P \wedge Q)$

---

c) **Utilizzando il calcolo del primo ordine si formalizzi il seguente enunciato dichiarativo, indicando esplicitamente l'interpretazione intesa:**

"Non tutte le coppie di amici hanno un amico in comune"

Soluzione: **Dominio**: le persone. **Predicati**: Amico(x, y) "x e y sono amici"

Formula: $\sim(\forall x, y.\ Amico(x, y) \Rightarrow (\exists z.\ Amico(x, z) \wedge Amico(y, z)))$

---

d) **Assumendo che a : array [0, n) of nat, si formalizzi il seguente enunciato:**

"l'array a contiene i valori x e y nell'ordine

(cioè la posizione di x è strettamente minore di quella di y)"

Soluzione: $(\exists\ i.\ i \in [0,n) \wedge (\exists\ j.\ j \in (i,n).\ a[\ i\ ] = x \wedge a[\ j] = y\ ))$

---

e) **Si dimostri la correttezza della seguente tripla:**

$\{\ x = a \wedge y = b\ \}$

x := y ;

y := x ;

$\{\ x = y\ \}$

Soluzione: Per la regola della sequenza basta trovare una formula R tale che le seguenti triple siano verificate:

(1) $\{\ x = a \wedge y = b\ \}\ x := y\ \{\ R\ \}$

(2) $\{\ R\ \}\ y := x\ \{\ x = y\ \}$

Applicando l'assioma dell'assegnamento a (2) otteniamo che la tripla è soddisfatta per

$R \equiv (\ x = y)[x/y] \wedge def(x)$

Semplificando, otteniamo $R \equiv T$. Applicando a (1) l'assioma dell'assegnamento e la regola (pre) con questo valore per R, otteniamo che la tripla (1) è soddisfatta se la seguente implicazione è valida:

$x = a \wedge y = b \ \Rightarrow T[y/x] \wedge def(x)$

La dimostrazione è immediata perché la conseguenza è vera.

---

f) **Si dimostri la correttezza della seguente tripla, assumendo che il calcolo della guardia dell'if non dia errore:**

$\{\ c = \#\{\ j \in\ [\ 1, i\ )\ .\ a[\ j\ ] = a[\ 0\ ]\}\ \}$

if (a[i] = a[0])

then c := c + 1

else skip

fi ;

i := i + 1

$\{\ c = \#\{\ j \in\ [\ 1, i\ )\ .\ a[\ j\ ] = a[\ 0\ ]\}\ \}$

Soluzione: Per la regola della sequenza basta trovare una formula R tale che le seguenti triple siano verificate:

(1) $\{ c = \#\{ j \in [1, i) . a[j] = a[0]\} \}$

if $(a[i] = a[0])$

then $c := c + 1$

else skip

fi

$\{ R \}$

(2) $\{ R \}$ $i := i + 1$ $\{ c = \#\{ j \in [1, i) . a[j] = a[0]\} \}$

Applicando l'assioma dell'assegnamento a (2) otteniamo che la tripla è soddisfatta per

$R \equiv (c = \#\{ j \in [1, i) . a[j] = a[0]\} [i + 1 / i] \wedge def(i + 1)$ cioè

$R \equiv c = \#\{ j \in [1, i] . a[j] = a[0]\}$

Con questo valore per R, per verificare la tripla (1) usando la regola del condizionale, dobbiamo dimostrare la seguente formula (a) e verificare le seguenti triple (b) e (c):

(a) $c = \#\{ j \in [1, i) . a[j] = a[0]\} \Rightarrow def(a[i] = a[0])$

(b) $\{ c = \#\{ j \in [1, i) . a[j] = a[0]\} \wedge a[i] = a[0] \}$ $c := c + 1$ $\{ c = \#\{ j \in [1, i] . a[j] = a[0]\} \}$

(c) $\{ c = \#\{ j \in [1, i) . a[j] = a[0]\} \wedge a[i] \neq a[0] \}$ **skip** $\{ c = \#\{ j \in [1, i] . a[j] = a[0]\} \}$

L'assunzione che il calcolo della guardia dell'**if** non dia errore implica che **def(a[i] = a[0]) ≡ T**, quindi (a) è banalmente valida.

Per (b), usando l'assioma dell'assegnamento e la regola (pre), è sufficiente dimostrare:

(b') $(c = \#\{ j \in [1, i) . a[j] = a[0]\}) \wedge a[i] = a[0]) \Rightarrow$

$(c = \#\{ j \in [1, i] . a[j] = a[0]\}[c+1/c] \wedge def(1))$

Vediamolo, partendo dalla conseguenza:

$c = \#\{ j \in [1, i] . a[j] = a[0]\}[c+1/c] \wedge def(1)$

$\equiv$ { applicazione di sostituzione, definizione di *def()*, unità }

$c + 1 = \#\{ j \in [1, i] . a[j] = a[0]\}$

$\equiv$ { Legge dell'intervallo per #, Ip: $a[i] = a[0]$ }

$c + 1 = \#\{ j \in [1, i) . a[j] = a[0]\} + 1$

$\equiv$ { Ip: $c = \#\{ j \in [1, i) . a[j] = a[0]\}$, riflessività di = }

T

Per (c), usando l'assioma (skip) e la regola (pre), è sufficiente dimostrare:

(c') $(c = \#\{ j \in [1, i) . a[j] = a[0]\} \wedge a[i] \neq a[0]) \Rightarrow (c = \#\{ j \in [1, i] . a[j] = a[0]\})$

Infatti, partendo dalla conseguenza abbiamo:

$c = \#\{ j \in [1, i] . a[j] = a[0]\}$

$\equiv$ { Legge dell'intervallo per #, Ip: $a[i] \neq a[0]$ }

$c = \#\{ j \in [1, i) . a[j] = a[0]\}$

$\equiv$ { Ip: $c = \#\{ j \in [1, i) . a[j] = a[0]\}$ }

T