Corso di Laurea in Informatica A.A. 2009/2010
Logica per la Programmazione
Primo appello - 13 gennaio 2010

---

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

$$(P \Rightarrow Q) \wedge (\sim Q \wedge R) \Rightarrow \sim P \wedge (P \Rightarrow R)$$

Soluzione: $(P \Rightarrow Q) \wedge (\sim Q \wedge R)$

$\equiv$ { contropositiva }

$(\sim Q \Rightarrow \sim P) \wedge \sim Q \wedge R$

$\Rightarrow$ { Modus Ponens }

$\sim P \wedge R$

$\Rightarrow$ { introduzione-$\vee$ }

$\sim P \wedge (\sim P \vee R)$

$\equiv$ { eliminazione-$\Rightarrow$, al contrario }

$\sim P \wedge (P \Rightarrow R)$

---

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

$$(\forall x.\ P \vee Q) \wedge \sim P[a/x] \Rightarrow (\exists x.\ Q)$$

Soluzione: $(\forall x.\ P \vee Q) \wedge \sim P[a/x]$

$\Rightarrow$ { eliminazione-$\forall$ }

$(P[a/x] \vee Q[a/x]) \wedge \sim P[a/x]$

$\equiv$ { complemento }

$Q[a/x] \wedge \sim P[a/x]$

$\Rightarrow$ { semplificazione-$\wedge$ }

$Q[a/x]$

$\Rightarrow$ { introduzione-$\exists$ }

$(\exists x.\ Q)$

---

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

"Non tutti i cardinali diventano santi, ma qualcuno diventa papa"

Soluzione: **Dominio**: le persone. **Predicati**:

- Cardinale(x) "x è un cardinale"
- Santo(x) "x diventa santo"
- Papa(x) "x diventa papa"

Formula: $\sim(\forall x.\ Cardinale(x) \Rightarrow Santo(x)) \wedge (\exists x.\ Cardinale(x) \wedge Papa(x))$

---

d) **Assumendo che a : array [0, n) of nat e b : array [0, m) of nat, con n > m, si formalizzi il seguente enunciato:**

"l'array b è un sottoarray dell'array a"

Soluzione: $(\exists\ k.\ k \in [0,n) \wedge (\forall\ i \in [0,m).\ b[i] = a[k+i]))$

La formula formalizza l'enunciato "esiste una posizione dell'array **a** a partire dalla quale gli **m** elementi successivi sono uguali agli elementi di **b**, nell'ordine".

---

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

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

if (x = y) then x := 2 * y
else x := x + 1
fi ;
y := 1

$\{ x = 2 * a \wedge y = 1 \}$

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

(1) $\{ x = a \wedge y = a \}$
if (x = y) **then** x := 2 * y
else x := x + 1
fi
$\{ R \}$

(2) $\{ R \}$ y := 1 $\{ x = 2 * a \wedge y = 1 \}$

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

$R \equiv (x = 2 * a \wedge y = 1)[1/y] \wedge def(1)$

Semplificando, otteniamo $R \equiv (x = 2 * a)$. 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) $x = a \wedge y = a \Rightarrow def(x = y)$

(b) $\{ x = a \wedge y = a \wedge x = y \}$ x := 2 * y $\{ x = 2 * a \}$

(c) $\{ x = a \wedge y = a \wedge x \neq y \}$ x := x + 1 $\{ x = 2 * a \}$

La dimostrazione di (a) è ovvia, perché la conseguenza è vera per la definizione di *def()*.

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

(b') $(x = a \wedge y = a \wedge x = y) \Rightarrow (x = 2 * a)[2 * y / x] \wedge def(2 * y)$

Vediamolo:

$(x = 2 * a)[2 * y / x] \wedge def(2 * y)$

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

$2 * y = 2 * a$

$\equiv$ { Ip: y = a, riflessività di = }

T

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

(c') $(x = a \wedge y = a \wedge x \neq y) \Rightarrow (x = 2 * a)[x + 1/ x] \wedge def(x + 1)$

Osserviamo che la premessa vale F (è una contraddizione), e quindi (c') vale T. Infatti,

$x = a \wedge y = a \wedge x \neq y$

$\equiv$ { Leibnitz }

$x = y \wedge y = a \wedge x \neq y$

$\equiv$ { contraddizione, zero }

F

---

f) Si dimostri la correttezza della seguente tripla:

$$\{ d = (\Sigma\, j : j \in [0, i)\, .\, a[j+1] - a[j]) \}$$
$$d := d + a[i+1] - a[i]\,;$$
$$i := i + 1$$
$$\{ d = (\Sigma\, j : j \in [0, i)\, .\, a[j+1] - a[j]) \}$$

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

(1) $\{ d = (\Sigma\, j : j \in [0, i)\, .\, a[j+1] - a[j]) \}\ d := d + a[i+1] - a[i]\ \{ R \}$

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

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

$R \equiv (d = (\Sigma\, j : j \in [0, i)\, .\, a[j+1] - a[j]))[i+1 / i] \wedge def(i+1)$ cioè

$R \equiv d = (\Sigma\, j : j \in [0, i]\, .\, a[j+1] - a[j])$

Con questo valore per R, usando l'assioma dell'assegnamento e la regola (pre) per (1), otteniamo che la tripla è soddisfatta se

$(d = (\Sigma\, j : j \in [0, i)\, .\, a[j+1] - a[j])) \Rightarrow$

$(d = (\Sigma\, j : j \in [0, i]\, .\, a[j+1] - a[j]))[d + a[i+1] - a[i] / d] \wedge def(d + a[i+1] - a[i])$

Dimostriamo l'implicazione partendo dalla conseguenza:

$(d = (\Sigma\, j : j \in [0, i]\, .\, a[j+1] - a[j]))[d + a[i+1] - a[i] / d] \wedge def(d + a[i+1] - a[i])$

$\equiv$ { applicazione della sostituzione }

$(d + a[i+1] - a[i] = (\Sigma\, j : j \in [0, i]\, .\, a[j+1] - a[j])) \wedge def(d + a[i+1] - a[i])$

$\equiv$ { definizione di *def()*, ipotizzando che a[i +1] ed a[i] siano elementi legittimi dell'array }

$(d + a[i+1] - a[i] = (\Sigma\, j : j \in [0, i]\, .\, a[j+1] - a[j]))$

$\equiv$ { legge dell'intervallo per Σ }

$(d + a[i+1] - a[i] = (\Sigma\, j : j \in [0, i)\, .\, a[j+1] - a[j]) + a[i+1] - a[i])$

$\equiv$ { calcolo }

$d = (\Sigma\, j : j \in [0, i)\, .\, a[j+1] - a[j])$

$\equiv$ { Ip: $d = (\Sigma\, j : j \in [0, i)\, .\, a[j+1] - a[j])$ }

T
