Corso di Laurea in Informatica A.A. 2009/2010
Logica per la Programmazione
Seconda prova in itinere - 17 dicembre 2009

Soluzioni proposte dai docenti

---

a) Si provi che la seguente formula è valida (P e Q contengono la variabile libera x):

$$(\forall x.\ P \Rightarrow Q) \land \sim(\exists x.\ Q) \Rightarrow (\forall x.\ \sim P)$$

$(\forall x.\ P \Rightarrow Q) \land \sim(\exists x.\ Q)$

$\equiv$ { De Morgan }

$(\forall x.\ P \Rightarrow Q) \land (\forall x.\ \sim Q)$

$\equiv$ { $\forall-\land$, contraposizione }

$(\forall x.\ (\sim Q \Rightarrow \sim P) \land \sim Q))$

$\Rightarrow$ { Modus Ponens }

$(\forall x.\ \sim P)$

---

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

"Gli studenti che non hanno superato il test di ingresso ma
superano l'esame di LPP sono ammessi al secondo anno"

Dominio: gli studenti

Predicati:

- TestSuperato(x) "x ha superato il test di ingresso"
- SuperatoLPP(x) "x ha superato LPP"
- AmmessoAlSecondo(x) "x e' ammesso al secondo anno"

$$(\forall x.\ \sim TestSuperato(x) \land SuperatoLPP(x) \Rightarrow AmmessoAlSecondo(x))$$

---

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

"Il numero dei multipli di 2 nell'array **a** è strettamente maggiore del numero dei multipli di 3 nello stesso array"

$$\# \{ i : i \in [0, n) \mid a[i] \% 2 = 0 \} > \#\{ i : i \in [0, n) \mid a[i] \% 3 = 0\}$$

---

d) Si dimostri la correttezza del seguente programma annotato:

$\{ x = a \land y = b \land a > 0\}$

$\quad x := x + y;$

$\{ x = a + b \land y = b \land a > 0\}$

$\quad$ **if** $(x > y)$ **then** $\quad x := 0$

$\quad\quad\quad$ **else** $\quad$ **skip**

$\{ x \leq 0\}$

Possiamo dimostrare separatamente la correttezza delle due triple seguenti:

1) $\{ x = a \land y = b \land a > 0\}\ x := x + y\ \{ x = a + b \land y = b \land a > 0\}$

2) $\{ x = a + b \land y = b \land a > 0\}$ **if** $(x > y)$ **then** $x := 0$ **else skip** $\{ x \leq 0\}$

Per 1), dall'assioma dell'assegnamento otteniamo che la seguente tripla e' corretta:

$$\{def(x+y) \land (x = a + b \land y = b \land a > 0)[x+y / x]\}\ x := x + y\ \{ x = a + b \land y = b \land a > 0\}$$

Semplificando $(def(x+y) = T)$ e applicando la sostituzione, otteniamo

1a) $\{x + y = a + b \land y = b \land a > 0\}\ x := x + y\ \{ x = a + b \land y = b \land a > 0\}$

Possiamo completare la dimostrazione di 1) applicando la regola (pre) e dimostrando che

1b) $x = a \land y = b \land a > 0 \quad x + y = a + b \land y = b \land a > 0$

La dimostrazione di 1b) e' ovvia.

Per la tripla 2), dalla regola del condizionale dobbiamo dimostrare la correttezza di:

2a) $x = a + b \land y = b \land a > 0 \Rightarrow def(x > y)$

2b) $\{ x = a + b \land y = b \land a > 0 \land x > y \}\ x := 0\ \{ x \leq 0\}$

2c) $\{ x = a + b \land y = b \land a > 0 \land x \leq y \}$ **skip** $\{ x \leq 0\}$

La prova di 2a) e' banale, perche' $def(x > y) \equiv T$.

Per 2b) sfruttando l'assioma dell'assegnamento e la regola (pre) basta dimostrare

$$x = a + b \land y = b \land a > 0 \land x > y \Rightarrow (x \leq 0)[0/x] \land def(0)$$

Ma la conseguenza $(0 \leq 0 \land def(0))$ e' banalmente vera, e quindi 2b) e' verificata.

Per 2c), si osservi che la premessa e' falsa, e quindi e' anch'essa verificata. Infatti,

$x = a + b \wedge y = b \wedge a > 0 \wedge x \leq y$

$\equiv$ { Leibniz }

$x = a + b \wedge y = b \wedge a > 0 \wedge a + b \leq b$

$\equiv$ $x = a + b \wedge y = b \wedge a > 0 \wedge a \leq 0$

$\equiv$ {def di > e ≤ }

F

---

e) Si dimostri la validità della seguente tripla, assumendo che il calcolo della condizione dell'**if** non dia errore:

{ count = #(j | j ∈ [0, i) ∧ a[j] = a[j + 1]) }

if (a[i] = a[i + 1])

then count := count + 1

else **skip**

fi ;

i := i + 1

{ count = #(j | j ∈ [0, i) ∧ a[j] = a[j + 1]) }

Chiamiamo P ≡ count = #(j | j ∈ [0, i) ∧ a[j] = a[j + 1]).

Per la legge della sequenza, dobbiamo trovare un R tale che

1) {P} **if** (a[i] = a[i + 1]) **then** count := count + 1 **else** skip **fi** {R}

2) {R} i := i + 1 {P}

Dalla legge dell'assegnamento otteniamo che la tripla 2) e' corretta per
R ≡ (count = #(j | j ∈ [0, i) ∧ a[j] = a[j + 1]))[i+1/i] ∧ def(i + 1)
≡ count = #(j | j ∈ [0, i] ∧ a[j] = a[j + 1])

Per la tripla 1), dalla regola del condizionale dobbiamo dimostrare la correttezza di:

1a) P ⇒ def(a[i] = a[i + 1])

1b) { P ∧ a[i] = a[i + 1] } count := count + 1 { R }

1c) { P ∧ a[i] ≠ a[i + 1] } **skip** { R }

L'implicazione 1a) e' soddisfatta per ipotesi. Per 1b) dobbiamo dimostrare

P ∧ a[i] = a[i + 1] ⇒ R[count + 1 / count] ∧ def(count + 1)

Vediamolo:

$R[count + 1 / count] \wedge def(count + 1)$

$\equiv$ { def di R, sostituzione, def() }

$count + 1 = \#(j \mid j \in [0, i] \wedge a[j] = a[j + 1])$

$\equiv$ { intervallo, Ip: $a[i] = a[i + 1]$ }

$count + 1 = \#(j \mid j \in [0, i) \wedge a[j] = a[j + 1]) + 1$

$\equiv$ { Ip: P }

T

La dimostrazione di 1c) e' del tutto analoga, sfruttando la legge dell'intervallo per # e il fatto che $a[i] \neq a[i + 1]$
