LOGICA PER LA PROGRAMMAZIONE (A,B) - a.a. 2010-2011
SOLUZIONI PROPOSTE
PRIMO APPELLO - 8/02/2011

ESERCIZIO 1
Si provi che la seguente proposizione è una tautologia:

$$\neg R \wedge (P \Rightarrow Q) \wedge (R \vee \neg Q) \Rightarrow (P \Rightarrow \neg Q)$$

Soluzione 1

Partiamo dalla premessa:

$$\begin{array}{ll}
 & \neg R \wedge (P \Rightarrow Q) \wedge (R \vee \neg Q) \\
\equiv & \quad \{ \text{ Elim-}\Rightarrow \text{ al contrario } \} \\
 & \neg R \wedge (P \Rightarrow Q) \wedge (Q \Rightarrow R) \\
\Rightarrow & \quad \{ \text{ Trans-}\Rightarrow \text{ } \} \\
 & \neg R \wedge (P \Rightarrow R) \\
\equiv & \quad \{ \text{ Contropositiva } \} \\
 & \neg R \wedge (\neg R \Rightarrow \neg P) \\
\Rightarrow & \quad \{ \text{ Modus Ponens } \} \\
 & \neg P \\
\Rightarrow & \quad \{ \text{ Intro-}\vee \text{ } \} \\
 & \neg P \vee \neg Q \\
\equiv & \quad \{ \text{ Elim-}\Rightarrow \text{ al contrario } \} \\
 & P \Rightarrow \neg Q
\end{array}$$

Soluzione 2

Partiamo dalla premessa:

$$\begin{array}{ll}
 & \neg R \wedge (P \Rightarrow Q) \wedge (R \vee \neg Q) \\
\equiv & \quad \{ \text{ Elim-}\Rightarrow\text{, Commut. } \} \\
 & (\neg P \vee Q) \wedge \neg R \wedge (R \vee \neg Q) \\
\equiv & \quad \{ \text{ Complemento, Commut. } \} \\
 & (\neg P \vee Q) \wedge \neg Q \wedge \neg R \\
\equiv & \quad \{ \text{ Complemento } \} \\
 & \neg P \wedge \neg Q \wedge \neg R \\
\Rightarrow & \quad \{ \text{ Sempl-}\wedge \} \\
 & \neg P \\
\Rightarrow & \quad \{ \text{ Intro-}\vee \text{ } \} \\
 & \neg P \vee \neg Q \\
\equiv & \quad \{ \text{ Elim-}\Rightarrow \text{ al contrario } \} \\
 & P \Rightarrow \neg Q
\end{array}$$

ESERCIZIO 2
Si provi che la seguente proposizione formula è valida (P, Q e R contengono la variabile libera x)

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

Soluzione Partiamo dalla premessa:

$$\begin{array}{ll}
 & \neg(\forall x.P \vee Q) \vee (\exists x.Q \wedge R) \\
\equiv & \quad \{ \text{ De Morgan, due volte } \} \\
 & (\exists x.\neg P \wedge \neg Q) \vee (\exists x.Q \wedge R) \\
\equiv & \quad \{ \ \exists : \vee \ \} \\
 & (\exists x.(\neg P \wedge \neg Q) \vee (Q \wedge R)) \\
\Rightarrow & \quad \{ \text{ Sempl-}\wedge\text{, due volte, occorrenze positive } \} \\
 & (\exists x.\neg P \vee Q) \\
\equiv & \quad \{ \text{ De Morgan, due volte } \} \\
 & \neg(\forall x.P \wedge \neg Q)
\end{array}$$

ESERCIZIO 3
Utilizzando il calcolo del primo ordine si formalizzi il seguente enunciato dichiarativo, indicando esplicitamente l'interpretazione intesa:

"Non tutti gli studenti che hanno seguito il corso di Logica lo superano,
ma tutti quelli che superano il corso Programmazione lo hanno seguito."

Soluzione

- **Linguaggio**
  - $\mathbf{C} = \{Logica, Programmazione\}$
  - $\mathbf{F} = \emptyset$
  - $\mathbf{P} = \{studente(\cdot), segue(\cdot,\cdot), superaEsame(\cdot,\cdot)\}$
- **Interpretazione**: $\mathbf{I} = (\mathbf{D}, \alpha)$

- $\mathbf{D} = \mathcal{P} \cup \mathbf{C}$, con $\mathcal{P}$ insieme delle persone e $\mathcal{C}$ insieme dei corsi
- $\alpha(Logica)$ = il corso di Logica
- $\alpha(Programmazione)$ = il corso di Programmazione
- $\alpha(studente)(p) \equiv \mathbf{T}$ se e solo la persona p è uno studente
- $\alpha(segue)(p, c) \equiv \mathbf{T}$ se e solo se la persona p ha seguito il corso c
- $\alpha(superaEsame)(p, c) \equiv \mathbf{T}$ se e solo se la persona p supera l'esame del corso c

Una formalizzazione è allora

$$\neg(\forall p.studente(p) \wedge segue(p, Logica) \Rightarrow superaEsame(p, Logica)) \wedge$$
$$(\forall q.studente(q) \wedge superaEsame(q, Programmazione) \Rightarrow segue(q, Programmazione))$$

ESERCIZIO 4
Assumendo **a : array [0, n) of nat** e **b : array [0, m) of nat**, si formalizzi il seguente enunciato:

"Tutti gli elementi dell'array **a** che non occorrono in **b** sono strettamente positivi."

Soluzione
$(\forall i.i \in [0, n) \wedge (\forall j.j \in [0, m) \Rightarrow a[i] \neq b[j]) \Rightarrow a[i] > 0)$

Attenzione: la seguente soluzione non è corretta:
$(\forall i.i \in [0, n) \wedge (\forall j.j \in [0, m) \wedge a[i] \neq b[j] \Rightarrow a[i] > 0))$
Infatti, per esempio, gli array **a = [-1, 2]** e **b = [3, -1]** soddisfano sia l'enunciato che la prima formula, ma non la seconda.

ESERCIZIO 5
Si dimostri la correttezza della seguente tripla:

$\{ x = A \wedge y = B \}$
  z := x - y;
  if (z $\neq$ 0) **then**
    x := x - z;
    y := y + z
  else
    skip
  fi
$\{ x = B \wedge y = A \}$

Soluzione Trattandosi di una sequenza, dobbiamo determinare una asserzione **R** in modo che:

(1) $\{ x = A \wedge y = B \}$
  z := x - y;
$\{ \mathbf{R} \}$

(2) $\{ \mathbf{R} \}$
  if (z $\neq$ 0) **then** x := x - z; y := y + z **else skip fi**
$\{ x = B \wedge y = A \}$

In questo caso non possiamo sfruttare l'assioma dell'assegnamento per determinare **R**. Tuttavia, poiché nei comandi successivi si usano le variabili x, y e z, scegliamo per **R** la seguente asserzione, che comprende tutte le informazioni disponibili sullo stato in quel punto del programma:
$\mathbf{R} \equiv x = A \wedge y = B \wedge z = A - B$
A questo punto per dimostrare (1), usando la Regola per l'Assegnamento, dobbiamo dimostrare la seguente implicazione:

(1.1) $x = A \wedge y = B \Rightarrow def(x - y) \wedge (x = A \wedge y = B \wedge z = A - B)\,[^{x-y}/_{z}]$
Applicando la sostituzione otteniamo:
$x = A \wedge y = B \Rightarrow def(x - y) \wedge (x = A \wedge y = B \wedge x - y = A - B)$
che è banalmente vera (esercizio per il lettore).

La dimostrazione di (2), per la regola per il condizionale, si riduce alle seguenti tre dimostrazioni:

(2.1) $x = A \wedge y = B \wedge z = A - B \Rightarrow def(z \neq 0)$
banalmente vera osservando che $def(z \neq 0) \equiv \mathbf{T}$

(2.2) $\{ x = A \wedge y = B \wedge z = A - B \wedge z \neq 0 \}$
x := x - z; y := y + z
$\{ x = B \wedge y = A \}$

(2.3) $\{ x = A \wedge y = B \wedge z = A - B \wedge z = 0 \}$
skip
$\{ x = B \wedge y = A \}$

(2.2) Trattandosi di una sequenza, dobbiamo determinare una asserzione **Q** in modo che:

(2.2.1) $\{ x = A \wedge y = B \wedge z = A - B \wedge z \neq 0 \}$
x := x - z
$\{ \mathbf{Q} \}$

(2.2.2) $\{ \mathbf{Q} \}$
y := y + z
$\{ x = B \wedge y = A \}$

Usando l'assioma dell'assegnamento, un'asserzione **Q** che garantisce la validità della (2.2.2) è

$\mathbf{Q} \equiv def(y + z) \wedge (x = B \wedge y = A)[^{y+z}/_{y}]$

ovvero, osservando che $def(y + z) \equiv \mathbf{T}$

$\mathbf{Q} \equiv (x = B \wedge y + z = A)$

Sostituendo **Q** in (2.2.1) e usando la Regola dell'assegnamento, dobbiamo dimostrare:

$$x = A \wedge y = B \wedge z = A - B \wedge z \neq 0 \quad \Rightarrow \quad def(x - z) \wedge (x = B \wedge y + z = A)[^{x-z}/_{x}]$$

Partiamo dalla conseguenza, usando le premesse come ipotesi:

$def(x - z) \wedge (x = B \wedge y + z = A)[^{x-z}/_{x}]$
$\equiv$ { Sostituzione, def di def }
$x - z = B \wedge y + z = A$
$\equiv$ { **Ip:** $x = A$, $y = B$, $z = A - B$; Leibniz }
$A - (A - B) = B \wedge B + (A - B) = A$
$\equiv$ { Calcolo }
$\mathbf{T}$

(2.3) Dimostrazione: per l'assioma per **skip**, dobbiamo dimostrare

$$x = A \wedge y = B \wedge z = A - B \wedge z = 0 \quad \Rightarrow \quad x = B \wedge y = A$$

Partiamo dalla premessa:

$x = A \wedge y = B \wedge z = A - B \wedge z = 0$
$\equiv$ { Leibniz }
$x = A \wedge y = B \wedge 0 = A - B \wedge z = 0$
$\Rightarrow$ { Sempl-$\wedge$ e calcolo }
$x = A \wedge y = B \wedge A = B$
$\equiv$ { Leibniz, due volte }
$x = B \wedge y = A \wedge A = B$
$\Rightarrow$ { Sempl-$\wedge$ }
$x = B \wedge y = A$

ESERCIZIO 6

Si consideri il seguente programma annotato:

{a : array [0, n) of int $\wedge$ n $\geq$ 0}
x,m := 0, -1;
{Inv : m = (-1 *max* (max k : k $\in$ [0,x) $\wedge$ a[k] > 0. k)) $\wedge$ x $\in$ [0,n] }{t: n-x}
while x<n **do**
if (a[x] > 0) **then** m := x **else skip fi** ;

x := x+1;
endw
{Inv ∧ ¬(x < n)}
{ m = (-1 *max* (max k : k ∈ [0,n) ∧ a[k] > 0. k)) }

Scrivere e dimostrare la proprietà di invarianza.
(Si ricordi che (max k : P. E) = - ∞ se P è vuoto).
Soluzione proposta
La proprietà di invarianza è la seguente

$\{ m = (-1 \ max \ (\max k : k \in [0,x) \wedge a[k] > 0 \,.\, k)) \wedge x \in [0,n] \wedge x < n \}$
if (a[x] > 0) **then** m := x **else skip fi** ;
x := x+1
$\{ m = (-1 \ max \ (\max k : k \in [0,x) \wedge a[k] > 0 \,.\, k)) \wedge x \in [0,n] \wedge \ def(x < n) \}$

Osserviamo che *def*(x <n) ≡ **T** e quindi la omettiamo. Osserviamo poi che
x ∈ [0,n] ∧ x < n ≡ x ∈ [0,n)
La dimostrazione, per l'assioma della sequenza, richiede di determinare una asserzione {**R**} in modo che

(1) $\{ m = (-1 \ max \ (\max k : k \in [0,x) \wedge a[k] > 0 \,.\, k)) \wedge x \in [0,n) \}$
if (a[x] > 0) **then** m := x **else skip fi** ;
{ **R** }

(2) { **R** }
x := x+1;
$\{ m = (-1 \ max \ (\max k : k \in [0,x) \wedge a[k] > 0 \,.\, k)) \wedge x \in [0,n] \}$

Per l'assioma dell'assegnamento, l'asserzione **R** candidata, osservando che $def(x+1) \equiv \mathbf{T}$, è
$\mathbf{R} \equiv m = (-1 \ max \ (\max k : k \in [0,x+1) \wedge a[k] > 0 \,.\, k)) \wedge x+1 \in [0,n]$
A questo punto la dimostrazione di (1), per la regola per il condizionale, si riduce alle seguenti tre dimostrazioni:

(1.1) $m = (-1 \ max \ (\max k : k \in [0,x) \wedge a[k] > 0 \,.\, k)) \wedge x \in [0,n) \quad \Rightarrow \quad def(a[x] > 0)$

(1.2) $\{ m = (-1 \ max \ (\max k : k \in [0,x) \wedge a[k] > 0 \,.\, k)) \wedge x \in [0,n) \wedge a[x] > 0 \}$
m := x
$\{ m = (-1 \ max \ (\max k : k \in [0,x] \wedge a[k] > 0 \,.\, k)) \wedge x+1 \in [0,n] \}$

(1.3) $\{ m = (-1 \ max \ (\max k : k \in [0,x) \wedge a[k] > 0 \,.\, k)) \wedge x \in [0,n) \wedge a[x] \leq 0 \}$
skip
$\{ m = (-1 \ max \ (\max k : k \in [0,x] \wedge a[k] > 0 \,.\, k)) \wedge x+1 \in [0,n] \}$

(1.1) Dimostrazione:

$m = (-1 \ max \ (\max k : k \in [0,x) \wedge a[k] > 0 \,.\, k)) \wedge x \in [0,n)$
⇒ {Sempl-∧}
$x \in [0,n)$
≡ {*def*(a[x]>0) ≡ x ∈ [0,n)}
$def(a[x] > 0)$

(1.2) Dimostrazione: Per l'assioma dell'assegnamento dobbiamo dimostrare

$m = (-1 \ max \ (\max k : k \in [0,x) \wedge a[k] > 0 \,.\, k)) \wedge x \in [0,n) \wedge a[x] > 0$
⇒
$def(x) \wedge x = (-1 \ max \ (\max k : k \in [0,x] \wedge a[k] > 0 \,.\, k)) \wedge x+1 \in [0,n]$

Partiamo dalla conclusione:

$def(x) \wedge x = (-1 \ max \ (\max k : k \in [0,x] \wedge a[k] > 0 \,.\, k)) \wedge x+1 \in [0,n]$
≡ { Def di *def*, Intervallo-*max*, **Ip:** $a[x] > 0$, $x \in [0,n)$; $x \in [0,n) \Rightarrow [0,x]$ non vuoto }
$x = (-1 \ max \ (\max k : k \in [0,x) \wedge a[k] > 0 \,.\, k) \ max \ x) \wedge x+1 \in [0,n]$
≡ { Osservando che $(-1 \ max \ (\max k : k \in [0,x) \wedge a[k] > 0 \,.\, k))$ è minore di x }
$x = x \wedge x+1 \in [0,n]$
≡ { **Ip:** $x \in [0,n)$; $x \in [0,n) \Rightarrow x+1 \in [0,n]$ }}
T

(1.3) Dimostrazione:

Per l'assioma del comando vuoto dobbiamo dimostrare

$$
\begin{array}{ll}
 & m = (-1 \ max \ (\max k : k \in [0, x) \wedge a[k] > 0 \,.\, k)) \wedge x \in [0, n) \wedge a[x] \leq 0 \\
\Rightarrow & \\
 & m = (-1 \ max \ (\max k : k \in [0, x] \wedge a[k] > 0 \,.\, k)) \wedge x + 1 \in [0, n]
\end{array}
$$

Partiamo dalla conclusione:

$$
\begin{array}{ll}
 & m = (-1 \ max \ (\max k : k \in [0, x] \wedge a[k] > 0 \,.\, k)) \wedge x + 1 \in [0, n] \\
\equiv & \quad \{\text{Intervallo-}max\text{, } \mathbf{Ip:}\ a[x] \leq 0,\ x \in [0, n);\ x \in [0, n) \Rightarrow [0, x] \text{ non vuoto}\} \\
 & m = (-1 \ max \ (\max k : k \in [0, x) \wedge a[k] > 0 \,.\, k)) \wedge x + 1 \in [0, n] \\
\equiv & \quad \{\mathbf{Ip:}\ m = (-1 \ max \ (\max k : k \in [0, x) \wedge a[k] > 0 \,.\, k))\ \} \\
 & m = m \wedge x + 1 \in [0, n] \\
\equiv & \quad \{\ \mathbf{Ip:}\ x \in [0, n);\ x \in [0, n) \Rightarrow x + 1 \in [0, n]\ \}\} \\
 & \mathbf{T}
\end{array}
$$