LOGICA PER LA PROGRAMMAZIONE (A,B) - a.a. 2013-2014

Primo Appello [Es. 1-6] o Recupero II Compitino [Es. 3-6] - 16/01/2014

Attenzione: Scrivere **nome**, **cognome**, **matricola** e **corso** in alto a destra su ogni foglio che si consegna.

ESERCIZIO 1
Si provi che la seguente proposizione è una tautologia, senza usare le tabelle di verità né dimostrazioni per casi:

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

ESERCIZIO 2
Calcolare, motivando la risposta, il valore di verità della formula

$$\Phi = (\forall x \,.\, R(x) \vee (\exists y \,.\, S(y,x) \wedge R(y)))$$

nell'interpretazione $I = (D, \alpha)$ dove $D = \{*, \#, o\}$ ed α è definita come segue:

$$\alpha(R)(z) = \begin{cases} T & \text{se } z \in \{*, \#\}, \\ F & \text{altrimenti.} \end{cases} \qquad \alpha(S)(z,v) = \begin{cases} T & \text{se } (z,v) \in \{(*,*), (\#, o), (\#, *)\} \\ F & \text{altrimenti.} \end{cases}$$

Calcolare cioè $I_{\rho_0}(\Phi)$ usando le regole della semantica del primo ordine, dove ρ_0 è un assegnamento arbitrario.

ESERCIZIO 3
Si provi che la seguente formula è valida (P, Q, R e S contengono la variabile libera x):

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

ESERCIZIO 4
Assumendo **a: array [0, n) of nat** con $n > 0$ si formalizzi il seguente enunciato:

"Nell'array **a** c'è un solo elemento in posizione pari
che è maggiore della somma degli elementi in posizione dispari che lo seguono."

ESERCIZIO 5
Si verifichi la seguente tripla di Hoare (assumendo **a: array [0, n) of int**):

$\{k > 0 \;\wedge\; x \in [0, n) \;\wedge\; (\forall i \,.\, i \in [0, x) \Rightarrow a[i] \leq k)\}$
  if $a[x] > k$
    then $a[x] := 0$
    else skip
  fi
$\{(\forall i \,.\, i \in [0, x] \Rightarrow a[i] \leq k)\}$

ESERCIZIO 6
Si assuma che il linguaggio di programmazione introdotto a lezione sia esteso con un operatore binario **max**, che applicato a due valori interi restituisce il massimo fra i due, e tale che $def(\mathbf{max}(E_1, E_2)) \equiv def(E_1) \wedge def(E_2)$. Si consideri il seguente programma annotato, dove **a: array [0, n) of int**:

$\{n > 0\}$
  $w := a[0]$; $x := 1$;
$\{\mathsf{Inv} : x \in [1, n] \;\wedge\; w = (max\ i : i \in [0, x) \,.\, a[i])\}\{\mathsf{t}\text{: } n - x\}$
  while $(x < n)$ **do**
    $w := \mathbf{max}(w, a[x])$; $x := x + 1$
  endw
$\{w = (max\ i : i \in [0, n) \,.\, a[i])\}$

Scrivere e dimostrare l'ipotesi di invarianza.