

LOGICA PER LA PROGRAMMAZIONE - a.a. 2019-2020

Seconda Prova di Verifica Intermedia - 20/12/2019

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

ESERCIZIO 1

Assumendo che P e Q contengano la variabile libera x , si provi che la seguente formula è valida:

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

ESERCIZIO 2

Assumendo **a: array [0, n) of int** e **b: array [0, m) of int** con $n, m > 1$, si formalizzi il seguente enunciato con la logica del primo ordine:

“L’ultimo elemento dell’array **a** è uguale alla somma degli elementi dispari di **b**, mentre tutti gli altri elementi di **a** sono uguali al massimo degli elementi pari di **b**.”

ESERCIZIO 3

Si dica se le seguenti triple sono soddisfatte, assumendo **a: array [0, n) of int** e **b: array [0, n) of int**. Se lo è, fornire una dimostrazione formale; se non lo è, fornire un controesempio.

1. $\{x = A \wedge y = B \wedge z = C\} x := x + (y - z); y := y + (x - z) \{x = y\}$,
2. $\{x = A \wedge y = B \wedge z = C\} x, y := x + (y - z), y + (x - z) \{x = y\}$,

ESERCIZIO 4

Assumendo **a: array [0, n) of int** e **b: array [0, m) of int** con $m \geq n$, si verifichi la seguente tripla:

$$\begin{aligned} & \{x \in [1, n) \wedge (\forall i. i \in [0, x) \Rightarrow (b[i] > 0 \wedge a[i] > (\max j : j \in [0, i). b[j]]))\} \\ & a[x] := a[x - 1] + b[x - 1] + 1; \\ & x := x + 1 \\ & \{(\forall i. i \in [0, x) \Rightarrow (a[i] > (\max j : j \in [0, i). b[j]]))\} \end{aligned}$$

ESERCIZIO 5

Assumendo **a: array [0, n) of int**, si consideri il seguente frammento di programma annotato:

```
{s = 0, x = 0}
{Inv: x ∈ [0, n] ∧ s = (∑i : i ∈ [0, x) ∧ a[i] % 2 = 0 . a[i] * 2x-1-i)} {t: n - x}
while (x < n) do
  if (a[x] % 2 = 0)
    then s, x := 2 * s + a[x], x + 1
    else s, x := 2 * s, x + 1
  fi
endw
{s = (∑i : i ∈ [0, n) ∧ a[i] % 2 = 0 . a[i] * 2n-1-i) }
```

Si scrivano le ipotesi di progresso ed invarianza. Inoltre si dimostri l'ipotesi di invarianza **limitatamente alle due prime condizioni della regola del comando condizionale (si ignori il ramo else)**.