

LOGICA PER LA PROGRAMMAZIONE (A,B) - a.a. 2016-2017

Secondo Appello - 10/02/2017

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

ESERCIZIO 1

Si dica se le seguenti proposizioni sono tautologie oppure no. Se una proposizione è una tautologia, lo si deve dimostrare senza usare le tabelle di verità; altrimenti va prodotto un controesempio mostrando esplicitamente che rende la formula falsa.

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

ESERCIZIO 2

Si formalizzi il seguente enunciato usando l'alfabeto con simboli di predicato $\mathcal{P} = \{citta(-), stato(-), capitale(-, -)\}$, rispetto all'interpretazione fissata (\mathcal{D}, α) , dove \mathcal{D} è l'insieme gli stati e di tutte le città, e

- $\alpha(citta)(d)$ è vera se e solo se d è una città,
- $\alpha(stato)(d)$ è vera se e solo se d è uno stato,
- $\alpha(capitale)(b, d)$ è vera se e solo se b è la capitale di d .

“Ogni stato ha una città che è capitale ma esiste una città che non è capitale di nessuno stato.”

ESERCIZIO 3

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

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

ESERCIZIO 4

Si formalizzi il seguente enunciato (assumendo **a, b: array [0, n] of int**):

“Ogni elemento dell'array **b** contiene il numero degli elementi dell'array **a** con indice minore o uguale al suo che sono maggiori della somma degli elementi pari di **a**.”

ESERCIZIO 5

Si consideri il seguente programma annotato (assumendo **a: array [0, n] of int**):

```
{ n > 0 }
z := 0; m := 0
{Inv : z ∈ [0, n) ∧ m = (∑y : y ∈ [0, z). a[y]) + 2 * (∑y : y ∈ [0, z). y)}{t: n - z}
while (z < n) do
    z, m := z + 1, m + a[z] + 2 * z
endw
{m = (∑y : y ∈ [0, n). a[y]) + 2 * (∑y : y ∈ [0, n). y)}
```

Scrivere e dimostrare l'ipotesi di invarianza.

ESERCIZIO 6

Si verifichi la seguente tripla di Hoare (assumendo **b: array [0, m] of int**):

```
{x ∈ [1, m) ∧ (∀i. i ∈ [1, x) ⇒ b[i] ≥ 2 * b[i - 1])}
if (b[x] >= 2 * b[x - 1])
    then skip
    else b[x] := 2 * b[x - 1] + 1
fi
{(∀i. i ∈ [1, x] ⇒ b[i] ≥ 2 * b[i - 1])}
```