

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

Terzo Appello - 19/06/2017

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

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 di tutte le città e di tutti gli stati, 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 .

“Tutti gli stati hanno 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):

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

ESERCIZIO 4

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

“Ogni elemento dell'array **b** contiene la somma degli elementi dell'array **a** con indice maggiore o uguale al suo che si trovano in posizione pari.”

ESERCIZIO 5

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

```
{k > 0}
  y := 1; m := a[0];
{Inv : y ∈ (0, k) ∧ m = (max i : i ∈ [0, y]. a[i])}{t: k - y}
  while (y < k) do
    if (a[y] > m)
      then m := a[y]
      else skip fi;
    y := y + 1
  endw
{m = (max i : i ∈ [0, k]. a[i])}
```

Scrivere e dimostrare l'ipotesi di invarianza.

ESERCIZIO 6

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

```
{y ∈ [1, m) ∧ (∀i. i ∈ [1, y] ⇒ a[i] ≥ 3 * a[i - 1])}
  a[y] := 3 * a[y - 1] + 2
{(∀i. i ∈ [1, y] ⇒ a[i] ≥ 3 * a[i - 1])}
```