

LOGICA PER LA PROGRAMMAZIONE (A,B) - a.a. 2015-2016

Primo Appello - 19/01/2016

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

ESERCIZIO 2

Si formalizzi il seguente enunciato usando l'alfabeto con simboli di costante $\mathcal{C} = \{C\}$ e simboli di predicato $\mathcal{P} = \{\text{amico}(-, -), \text{abitano}(-, -)\}$ rispetto all'interpretazione fissata (\mathcal{D}, α) , dove \mathcal{D} è l'insieme di tutti gli esseri viventi, e

- $\alpha(C)$ è la persona Carlo,
- $\alpha(\text{amico})(p, q)$ è vera se e solo se p e q sono amici,
- $\alpha(\text{abitano})(p, q)$ è vera se e solo se p e q abitano nella stessa città.

“Tutte le persone che abitano nella stessa città di Carlo hanno almeno un amico in comune .”

ESERCIZIO 3

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

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

ESERCIZIO 4

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

“Ogni elemento dell'array **b** è maggiore della somma dei quadrati degli elementi di **a** con indice minore o uguale al suo.”

ESERCIZIO 5

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

```
{x ∈ (0, n) ∧ (∀i. i ∈ (0, x) ⇒ b[i] > b[i - 1])}
  if (b[x] > b[x-1] )
    then skip
    else b[x] := b[x-1]+1
  fi
{(∀i. i ∈ (0, x] ⇒ b[i] > b[i - 1])}
```

ESERCIZIO 6

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

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

Scrivere e dimostrare l'ipotesi di invarianza.