

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

Quarto Appello - 11/07/2017

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. $\neg(((A \Rightarrow B) \Rightarrow C) \Rightarrow \neg D) \wedge \neg(C \wedge D) \Rightarrow A \wedge \neg B$
2. $\neg(((A \Rightarrow B) \Rightarrow C) \Rightarrow D) \wedge \neg(C \wedge D) \Rightarrow A \wedge \neg B$

ESERCIZIO 2

Si formalizzi il seguente enunciato usando l'alfabeto con simboli di costante $\mathcal{C} = \{A\}$ e simboli di predicato $\mathcal{P} = \{\text{malato}(-), \text{vaAlCinema}(-), \text{fratello}(-, -), \text{genitore}(-, -)\}$, rispetto all'interpretazione fissata (\mathcal{D}, α) , dove \mathcal{D} è l'insieme di tutte le persone e

- $\alpha(A)$ è la persona Antonio,
- $\alpha(\text{genitore})(p, q)$ è vera se e solo se p è genitore di q ,
- $\alpha(\text{fratello})(p, q)$ è vera se e solo se p e q sono fratelli,
- $\alpha(\text{vaAlCinema})(p)$ è vera se e solo se p va al cinema,
- $\alpha(\text{malato})(p)$ è vera se e solo se p è malato.

“Se un genitore di Antonio è malato, allora Antonio va al cinema solo se uno dei suoi fratelli non ci va.”

ESERCIZIO 3

Si provi che la seguente formula è valida (A, B, C e D contengono la variabile libera x):

$$(\forall x. A \Rightarrow B) \wedge \neg(\exists x. \neg B \vee D \Rightarrow C \wedge D) \Rightarrow \neg(\exists x. A \wedge C)$$

ESERCIZIO 4

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

“Gli elementi dell'array **a** sono tutti distinti e ognuno è il quadrato di un elemento dell'array **b**.”

ESERCIZIO 5

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

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

Scrivere e dimostrare sia l'ipotesi di invarianza che l'ipotesi di progresso .

ESERCIZIO 6

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

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