

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

Primo Appello - 20/01/2017

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. $(Q \vee \neg S \Rightarrow \neg(P \wedge \neg R)) \wedge ((P \vee R) \wedge \neg R) \Rightarrow S$
2. $\neg(A \Rightarrow B) \wedge (\neg C \vee (D \wedge C) \Rightarrow \neg(A \wedge \neg B)) \Rightarrow C \wedge \neg A$

ESERCIZIO 2

Si consideri l'alfabeto del primo ordine \mathcal{A} con simboli di costante $\mathcal{C} = \{L, M\}$ e simboli di predicato $\mathcal{P} = \{persona(-), scrittore(-), libro(-), hascritto(-, -)\}$ e l'interpretazione $I = (\mathcal{D}, \alpha)$, dove \mathcal{D} è l'insieme delle persone e dei libri, e

- $\alpha(L)$ è la persona Luca,
- $\alpha(M)$ è la persona Marco,
- $\alpha(persona)(p)$ è vera se e solo se p è una persona,
- $\alpha(scrittore)(p)$ è vera se e solo se p è uno scrittore,
- $\alpha(libro)(p)$ è vera se e solo se p è un libro,
- $\alpha(hascritto)(p, q)$ è vera se e solo se lo scrittore p ha scritto il libro q

Formalizzare i seguenti enunciati usando l'alfabeto \mathcal{A} rispetto all'interpretazione I :

1. "Ogni persona è uno scrittore solo se ha scritto almeno un libro"
2. "Marco e Luca non hanno scritto nessun libro insieme"

ESERCIZIO 3

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

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

ESERCIZIO 4

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

"Ogni elemento dell'array **b** è uguale alla somma degli elementi di **a** che lo precedono strettamente oppure è uguale al massimo tra gli elementi pari di **a** che lo seguono."

ESERCIZIO 5

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

```
{ n > 0 }
y:=n-1; h:=0;
{Inv : y ∈ [-1, n) ∧ h = #{i : i ∈ (y, n) | a[i] > c[i]}}{t: y}
while (y >= 0) do
    if (a[y] > c[y])
        then h:=h+1
        else skip fi;
    y:=y-1
endw
{h = #{i : i ∈ [0, n) | a[i] > c[i]}}
```

Scrivere e dimostrare l'ipotesi di invarianza.

ESERCIZIO 6

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

$$\{k \in (0, n) \wedge (\forall i. i \in [0, k) \Rightarrow c[i] = (\sum y : y \in [0, i]. d[y]) - (\sum x : x \in [0, i]. x))\}$$
$$c[k] := c[k-1] + d[k] - k$$
$$\{(\forall i. i \in [0, k) \Rightarrow c[i] = (\sum y : y \in [0, i]. d[y]) - (\sum x : x \in [0, i]. x))\}$$