

LOGICA PER LA PROGRAMMAZIONE - a.a. 2018-2019

Secondo Appello - 11/02/2019

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 che rende la formula falsa.

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

ESERCIZIO 2

Si consideri l'alfabeto del primo ordine \mathcal{A} con simboli di predicato $\mathcal{P} = \{F(-), T(-), Friend(-, -), Follow(-, -)\}$ e l'interpretazione $I = (\mathcal{D}, \alpha)$, dove \mathcal{D} è un insieme di persone, e

- $\alpha(F)(x)$ è vera se e solo se la persona x è un utente di Facebook,
- $\alpha(T)(x)$ è vera se e solo se la persona x è un utente di Twitter,
- $\alpha(Friend)(x, y)$ è vera se e solo se la persona x è *amica* su Facebook della persona y ,
- $\alpha(Follow)(x, y)$ è vera se e solo se la persona x *segue* su Twitter la persona y .

Formalizzare usando l'alfabeto \mathcal{A} e l'interpretazione I , il fatto che la relazione di *amicizia* su Facebook è simmetrica, mentre la relazione *segue* su Twitter non lo è. Più precisamente, formalizzare il seguente enunciato:

“Dati due utenti qualunque di Facebook, se il primo è amico del secondo, anche il secondo è amico del primo.
Invece su Twitter ci sono almeno due utenti tali che il primo segue il secondo, ma non viceversa.”

ESERCIZIO 3

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

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

ESERCIZIO 4

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

“Ogni elemento pari dell'array **a** compare almeno due volte nell'array **b**”

ESERCIZIO 5

Assumendo **a: array [0, n) of int**, si consideri il seguente frammento di programma annotato,

```
{x = 0 ∧ s = a[0]}
{Inv: x ∈ [0, n) ∧ s = a[x] + x2 + x }{t: n - x}
while x < n - 1 do
  x := x + 1;
  s := s + a[x] - a[x-1] + 2*x
endw
{s = a[n - 1] + n2 - n}
```

Si scrivano le ipotesi di progresso ed invarianza. Inoltre si dimostri l'ipotesi di invarianza.

ESERCIZIO 6

Si verifichi **solo la tripla relativa al ramo “then”** della seguente tripla di Hoare (assumendo **b: array [0, n) of int**):

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