

# LOGICA PER LA PROGRAMMAZIONE (A,B) - a.a. 2017-2018

## Primo Appello - 18/01/2018

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

### ESERCIZIO 2

Si consideri l'alfabeto del primo ordine  $\mathcal{A}$  con simboli di predicato  $\mathcal{P} = \{stud(-), univ(-), iscr(-, -), =( -, -)\}$  e l'interpretazione  $I = (\mathcal{D}, \alpha)$ , dove  $\mathcal{D}$  è l'insieme studenti e delle università, e

- $\alpha(stud)(p)$  è vera se e solo se  $p$  è uno studente,
- $\alpha(univ)(p)$  è vera se e solo se  $p$  è una università,
- $\alpha(iscr)(p, q)$  è vera se e solo se lo studente  $p$  è iscritto all'università  $q$ ,
- $\alpha(=)(p, q)$  è vera se e solo se  $p$  e  $q$  sono uguali.

Formalizzare il seguente enunciato usando l'alfabeto  $\mathcal{A}$  rispetto all'interpretazione  $I$ :

“Ogni studente è iscritto ad una università e ogni università ha almeno due studenti iscritti”

### ESERCIZIO 3

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

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

### ESERCIZIO 4

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

“Per ogni elemento dell'array **a** esiste un elemento in **b** che è minore della somma degli elementi di **a** che lo precedono ed uno che è uguale al minimo tra gli elementi pari di **a** che lo seguono.”

### ESERCIZIO 5

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

```
{c = 0 ∧ y = 0}
{Inv: y ∈ [0, n] ∧ (c = #{i : i ∈ [0, y] | pari(i) ∧ a[i]2 > b[i]})}{t: n - y}
while y < n do
  if (y mod 2 = 0 ∧ a[y] * a[y] > b[y])
    then c, y := c+1, y+1
    else y := y+1
  fi
endw
{(c = #{i : i ∈ [0, n] | pari(i) ∧ a[i]2 > b[i]})}
```

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

### ESERCIZIO 6

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

```
{h ∈ [1, k] ∧ (∀i. i ∈ [0, h] ⇒ c[i] > (Σy : y ∈ [0, i]. a[y]))}
  c[h] := c[h-1] + a[h] + 1
{(∀i. i ∈ [0, h] ⇒ c[i] > (Σy : y ∈ [0, i]. a[y]))}
```