LOGICA PER LA PROGRAMMAZIONE - a.a. 2019-2020
Primo Appello - 10/01/2019

Attenzione: si scrivano **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. $\neg(A \Rightarrow (B \vee C) \wedge \neg D) \;\Rightarrow\; (\neg D \Rightarrow C)$
2. $(A \wedge \neg C \Rightarrow B) \wedge (\neg D \Rightarrow \neg B) \wedge \neg D \;\Rightarrow\; (A \Rightarrow C)$

ESERCIZIO 2
Si consideri l'alfabeto del primo ordine $\mathcal{A}$ con simboli di predicato $\mathcal{P} = \{L(-), K(-), A(-,-)\}$ e l'interpretazione $I = (\mathcal{D}, \alpha)$, dove $\mathcal{D}$ è l'insieme di tutti i lucchetti e tutte le chiavi, e

- $\alpha(L)(d)$ è vera se e solo se d è un lucchetto
- $\alpha(K)(d)$ è vera se e solo se d è una chiave
- $\alpha(A)(d_1, d_2)$ è vera se e solo se il d_2 è un lucchetto e d_1 è una chiave che lo apre.

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

"Ogni lucchetto ha almeno una chiave che lo apre, ma non esiste una chiave che apra tutti i lucchetti."

ESERCIZIO 3
Si provi che la seguente formula è valida (P, Q e R contengono la variabile libera x):

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

Suggerimento: potrebbe essere utile la legge $A \Rightarrow B \vee C \;\equiv\; A \wedge \neg B \Rightarrow C$

ESERCIZIO 4
Si formalizzi il seguente enunciato (assumendo **a: array [0, n) of int** e **b: array [0, k) of int**):

"Ogni elemento di **a** compare anche in **b**, ma un numero diverso di volte."

ESERCIZIO 5
Assumendo **a: array [0, n) of int**, si consideri il seguente frammento di programma annotato,

$\{c = 0 \;\wedge\; y = 0\}$
$\{\mathsf{Inv}\colon\; y \in [0, n] \;\wedge\; \big(c = \#\{i\colon i \in [0, y) \mid a[i] > 0 \wedge \mathit{dispari}(a[i])\}\big)\}\{\mathrm{t}\colon\; n - y\}$

```
while y < n do
     if (a[y] > 0)
          then c, y := c + a[y] mod 2, y+1
          else y := y+1
     fi
endw
```

$\{c = \#\{i\colon i \in [0, n) \mid a[i] > 0 \wedge \mathit{dispari}(a[i])\}\}$

Si scrivano le ipotesi di progresso ed invarianza. Inoltre si dimostri l'ipotesi di invarianza **limitatamente alle due prime condizioni della regola del comando condizionale (si ignori il ramo else)**.

ESERCIZIO 6
Si verifichi la seguente tripla di Hoare (assumendo **a: array [0, n) of int**):

$\{\; x \in [0, n) \;\wedge\; (\Sigma i\colon i \in [0, x) \,.\, a[i]) = x^2\}$

```
a[x] := 2 * x + 1 ;
x := x + 1
```

$\{(\Sigma i\colon i \in [0, x) \,.\, a[i]) = x^2\}$