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. $\big(Q \vee \neg S \Rightarrow \neg(P \wedge \neg R)\big) \ \wedge \ \big((P \vee R) \wedge \neg R\big) \ \Rightarrow \ S$
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 $\mathcal{P} = \{\mathit{persona}(\_), \mathit{scrittore}(\_), \mathit{libro}(\_), \mathit{hascritto}(\_,\_)\}$ 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,
- $\alpha(\mathit{persona})(p)$ è vera se e solo se p è una persona,
- $\alpha(\mathit{scrittore})(p)$ è vera se e solo se p è uno scrittore,
- $\alpha(\mathit{libro})(p)$ è vera se e solo se p è un libro,
- $\alpha(\mathit{hascritto})(p, q)$ 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):

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

3 Cominciamo col semplificare la conseguenza:

$$\begin{array}{ll} & \neg(\forall x \,.\, \neg(\forall x \,.\, C \wedge \neg D) \\ \equiv & \{\text{(De Morgan), (doppia negazione)}\} \\ & (\exists x \,.\, \neg\neg(\forall x \,.\, C \wedge \neg D) \\ \equiv & \{\text{(De Morgan), (Doppia negazione)}\} \\ & (\exists x \,.\, \neg S \wedge R) \qquad (\dagger) \end{array}$$

Per Regola della Skolemizzazione, per mostrare che la premessa implica la formula (†) è sufficiente dimostrare la seguente implicazione, dove **d** è una nuova costante:

$$(\forall x \,.\, P \wedge \neg Q) \ \wedge \ (\exists x \,.\, P \Rightarrow (\neg Q \Rightarrow \neg S \wedge R)) \ \wedge \ \big(P \Rightarrow (\neg Q \Rightarrow \neg S \wedge R)\big)[^{\mathbf{d}}/_x] \ \Rightarrow \ (\exists x \,.\, \neg S \wedge R)$$

Partiamo dalla premessa:

$$\begin{array}{ll} & (\forall x \,.\, P \wedge \neg Q) \ \wedge \ (\exists x \,.\, P \Rightarrow (\neg Q \Rightarrow \neg S \wedge R)) \ \wedge \ \big(P \Rightarrow (\neg Q \Rightarrow \neg S \wedge R)\big)[^{\mathbf{d}}/_x] \\ \Rightarrow & \{\text{(semp-}\wedge\text{)}\} \\ & (\forall x \,.\, P \wedge \neg Q) \ \wedge \ \big(P \Rightarrow (\neg Q \Rightarrow \neg S \wedge R)\big)[^{\mathbf{d}}/_x] \\ \Rightarrow & \{\text{(elim-}\forall\text{)}\} \\ & (P \wedge \neg Q)[^{\mathbf{d}}/_x] \wedge \big(P \Rightarrow (\neg Q \Rightarrow \neg S \wedge R)\big)[^{\mathbf{d}}/_x] \\ \equiv & \{\text{Sostituzione}\} \\ & P[^{\mathbf{d}}/_x] \wedge \neg Q[^{\mathbf{d}}/_x] \wedge \big(P[^{\mathbf{d}}/_x] \Rightarrow (\neg Q[^{\mathbf{d}}/_x] \Rightarrow \neg S[^{\mathbf{d}}/_x] \wedge R[^{\mathbf{d}}/_x])\big) \\ \Rightarrow & \{\text{(Modus Ponens)}\} \\ & \neg Q[^{\mathbf{d}}/_x] \wedge (\neg Q[^{\mathbf{d}}/_x] \Rightarrow \neg S[^{\mathbf{d}}/_x] \wedge R[^{\mathbf{d}}/_x]) \end{array}$$

$\Rightarrow$ {(Modus Ponens)}

$\neg S[^{\mathbf{d}}/_x] \wedge R[^{\mathbf{d}}/_x]$

$\Rightarrow$ {(intro-$\Rightarrow$)}

$(\exists x \,.\, \neg S \wedge R)$

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 : $y \in [-1, n) \ \wedge\ h = \#\{i : i \in (y, n) \mid a[i] > c[i]\}$}{t: y}

```
while (y >= 0) do
     if (a[y] > c[y])
          then h:=h+1
          else skip fi;
     y:=y-1
endw
```

$\{h = \#\{i : i \in [0, n) \mid a[i] > c[i]\}\}$

Scrivere e dimostrare l'ipotesi di invarianza.

ESERCIZIO 6

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

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

```
     c[k] := c[k-1] + d[k] - k
```

$\{\ \big(\forall i \,.\, i \in [0, k] \ \Rightarrow\ c[i] = (\Sigma y : y \in [0, i] \,.\, d[y]) \ - (\Sigma x : x \in [0, i] \,.\, x)\big)\}$