LOGICA PER LA PROGRAMMAZIONE - a.a. 2018-2019
Prima prova di verifica intermedia - 2/11/2018

Attenzione: si scrivano **nome**, **cognome**, **matricola** e **corso IN ALTO A DESTRA** su ogni foglio che si consegna.

ESERCIZIO 1

Si dimostri che la seguente proposizione è una tautologia, senza usare tabelle di verità:

$$\neg((Q \Rightarrow \neg(R \vee S) \vee S) \ \Rightarrow \ P) \ \equiv \ (R \wedge \neg S) \vee P \Rightarrow \neg Q \wedge \neg P$$

ESERCIZIO 2

Per ognuna delle seguenti formule si dica se si tratta di una tautologia oppure no. Se è una tautologia si fornisca una dimostrazione altrimenti si fornisca un controesempio.

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

ESERCIZIO 3

Si dica se, la dimostrazione sulla destra prova correttamente che la formula

$$(\neg(R \Rightarrow \neg Q \wedge R) \Rightarrow P) \ \Rightarrow \ (\neg Q \wedge R \Rightarrow \neg P)$$

è una tautologia. In caso negativo si elenchino gli eventuali errori presenti, **senza correggerli**.

Dimostrazione. Usando la formula $\neg(R \Rightarrow \neg Q \wedge R) \Rightarrow P$ come *ipotesi non tautologica* si dimostra $(\neg Q \wedge R \Rightarrow \neg P) \Rightarrow \mathbf{T}$.

$\neg Q \wedge R \ \Rightarrow \ \neg P$

$\Rightarrow$ $\{$ **Ip:** $\neg(R \Rightarrow \neg Q \wedge R) \Rightarrow P$, occ. neg. (Doppia Neg.)$\}$

$\neg Q \wedge R \ \Rightarrow \ (R \Rightarrow \neg Q \wedge R)$

$\equiv$ $\{$(Elim-$\Rightarrow$)$\}$

$\neg Q \wedge R \ \Rightarrow \ (\neg R \vee \neg Q \wedge R)$

$\equiv$ $\{$(Complemento)$\}$

$\neg Q \wedge R \ \Rightarrow \ R \wedge \neg Q$

$\equiv$ $\{$(Riflessività di $\Rightarrow$)*$\}$

$\mathbf{T}$

*La (Riflessività di $\Rightarrow$), ovvero $A \Rightarrow A$, non è presente nella *tavola delle leggi*, ma è comunque valida.

ESERCIZIO 4

Si consideri l'alfabeto del primo ordine con $\mathcal{C} = \emptyset$, $\mathcal{F} = \emptyset$, $\mathcal{P} = \{S, R, C, =\}$, dove i simboli di predicato S e R sono unari mentre C e $=$ sono binari. Si consideri l'interpretazione $\mathcal{I} = (\mathcal{D}, \alpha)$, dove $\mathcal{D}$ è l'insieme di tutti i programmi e α è definita come segue:

- $\alpha(S)(d) = \mathbf{T}$ se e solo se il programma d è un sottoprogramma,
- $\alpha(R)(d) = \mathbf{T}$ se e solo se il programma d è ricorsivo,
- $\alpha(C)(d, d') = \mathbf{T}$ se e solo se il programma d viene chiamato dal programma d'.
- $\alpha(=)(d, d') = \mathbf{T}$ se e solo se d e d' sono lo stesso programma.

Si formalizzi il seguente enunciato:

Un programma è un sottoprogramma se viene chiamato da altri programmi,
ma se viene chiamato da se stesso è ricorsivo.

ESERCIZIO 5

Si calcoli, motivando la risposta, il valore di verità della seguente formula sull'alfabeto del primo ordine con $\mathcal{C} = \emptyset$, $\mathcal{F} = \emptyset$ e $\mathcal{P} = \{P, Q\}$:

$$\phi = (\exists x \,.\, P(x) \wedge (\exists y \,.\, Q(x, y)))$$

nell'interpretazione $\mathcal{I} = (\mathcal{D}, \alpha)$, dove $\mathcal{D} = \{a, b, c\}$ ed α è definita come segue

$$\alpha(P)(x) = \begin{cases} \mathbf{T} & \text{se } x \in \{a, b\} \\ \mathbf{F} & \text{altrimenti} \end{cases} \qquad \alpha(Q)(x, y) = \begin{cases} \mathbf{T} & \text{se } (x, y) \in \{(c, a), (c, b), (b, c)\} \\ \mathbf{F} & \text{altrimenti} \end{cases}$$

Si calcoli cioè $\mathcal{I}_\rho(\phi)$ usando le regole della semantica del primo ordine, dove ρ è un assegnamento arbitrario.