LOGICA PER LA PROGRAMMAZIONE (A,B) - a.a. 2010-2011
SOLUZIONI PROPOSTE
QUARTO APPELLO - 29/06/2011

ESERCIZIO 1
Si provi che la seguente proposizione è una tautologia:

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

Soluzione 1

Dimostriamo la conclusione $P \Rightarrow Q$ usando le premesse come ipotesi:

P
$\Rightarrow$ { **Ip:** $P \Rightarrow R \vee S$ }
$R \vee S$
$\Rightarrow$ { **Ip:** $R \Rightarrow \neg Q$, occ. positiva}
$\neg Q \vee S$
$\equiv$ { **Ip:** Q}
$(\neg Q \vee S) \wedge Q$
$\equiv$ { Complemento }
$S \wedge Q$
$\Rightarrow$ { Sempl-$\wedge$ }
S

Soluzione 2

Partiamo dalla premessa:

$Q \wedge (P \Rightarrow R \vee S) \wedge (R \Rightarrow \neg Q)$
$\equiv$ { Contraposizione }
$Q \wedge (P \Rightarrow R \vee S) \wedge (Q \Rightarrow \neg R)$
$\Rightarrow$ { Modus Ponens }
$(P \Rightarrow R \vee S) \wedge \neg R$
$\equiv$ { Elim-$\Rightarrow$ }
$(\neg P \vee R \vee S) \wedge \neg R$
$\equiv$ { Complemento }
$(\neg P \vee S) \wedge \neg R$
$\Rightarrow$ { Sempl-$\wedge$ }
$\neg P \vee S$
$\equiv$ { Elim-$\Rightarrow$ }
$P \Rightarrow S$

ESERCIZIO 2
Si provi che la seguente formula è valida (P, Q e R contengono la variabile libera x)

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

Soluzione Partiamo dalla premessa, indicando con d un generico elemento del dominio che testimonia $(\exists x.R \wedge P)$ (quindi usiamo la Regola di Skolemizzazione):

$(R[d/x] \wedge P[d/x]) \wedge (\forall x.P \Rightarrow Q)$
$\Rightarrow$ { Elim-$\forall$ }
$R[d/x] \wedge P[d/x] \wedge (P[d/x] \Rightarrow Q[d/x])$
$\Rightarrow$ { Modus Ponens }
$R[d/x] \wedge Q[d/x])$
$\Rightarrow$ { Intro-$\exists$ }
$(\exists x.R \wedge Q)$
$\equiv$ { De Morgan }
$\neg(\forall x.\neg(R \wedge Q))$
$\equiv$ { De Morgan }
$\neg(\forall x.\neg R \vee \neg Q))$
$\equiv$ { Elim-$\Rightarrow$ }
$\neg(\forall x.Q \Rightarrow \neg R))$

ESERCIZIO 3
Utilizzando il calcolo del primo ordine si formalizzi il seguente enunciato dichiarativo, indicando esplicitamente l'interpretazione intesa, *in tutte le sue componenti*:

"Non tutti i membri del Parlamento italiano conoscono la Costituzione, ma tutti dicono di conoscerla!"

Soluzione 1

- **Linguaggio**

- $\mathbf{C} = \{Costituzione\}$
- $\mathbf{F} = \emptyset$
- $\mathbf{P} = \{parlamentare(\cdot), conosceLegge(\cdot,\cdot), diceConoscereLegge(\cdot,\cdot)\}$

- **Interpretazione**: $\mathbf{I} = (\mathbf{D}, \alpha)$
  - $\mathbf{D} = \mathcal{P} \cup \mathcal{L}$, con $\mathcal{P}$ insieme delle persone e $\mathcal{L}$ insieme delle leggi
  - $\alpha(Costituzione)$ = la legge fondamentale: la Costituzione italiana
  - $\alpha(parlamentare)(p) \equiv \mathbf{T}$ se e solo la persona p è un membro del Parlamento
  - $\alpha(conosceLegge)(p, l) \equiv \mathbf{T}$ se e solo se la persona p conosce la legge l
  - $\alpha(diceConoscereLegge)(p, l) \equiv \mathbf{T}$ se e solo se la persona p dice di conoscere la legge l

Una formalizzazione è allora

$$\neg(\forall p.parlamentare(p) \Rightarrow conosceLegge(p, Costituzione)) \wedge \\ (\forall q.parlamentare(q) \Rightarrow diceConoscereLegge(q, Costituzione))$$

Soluzione 2 È anche accettabile la seguente soluzione più semplice:

- **Linguaggio**
  - $\mathbf{C} = \emptyset$
  - $\mathbf{F} = \emptyset$
  - $\mathbf{P} = \{parlamentare(\cdot), conosceCostituzione(\cdot), diceConoscereCostituzione(\cdot)\}$
- **Interpretazione**: $\mathbf{I} = (\mathbf{D}, \alpha)$
  - $\mathbf{D} = \mathcal{P}$, con $\mathcal{P}$ insieme delle persone
  - $\alpha(parlamentare)(p) \equiv \mathbf{T}$ se e solo la persona p è un membro del Parlamento
  - $\alpha(conosceCostituzione)(p) \equiv \mathbf{T}$ se e solo se la persona p conosce la Costituzione
  - $\alpha(diceConoscereCostituzione)(p) \equiv \mathbf{T}$ se e solo se la persona p dice di conoscere la Costituzione

Una formalizzazione è allora

$$\neg(\forall p.parlamentare(p) \Rightarrow conosceCostituzione(p)) \wedge \ (\forall q.parlamentare(q) \Rightarrow diceConoscereCostituzione(q))$$

ESERCIZIO 4
Assumendo **a : array [0, n) of nat**, si formalizzi il seguente enunciato:

"Almeno due elementi consecutivi di **a** sono uguali tra loro
ed esattamente due elementi di **a** sono maggiori di 20"

Soluzione
$(\exists i.i \in [0, n-1) \wedge a[i] = a[i+1]) \wedge \#\{j \mid j \in [0, n) \wedge a[j] > 20\} = 2$

ESERCIZIO 5
Si determini una espressione E in modo che la seguente tripla sia verificata e si dimostri formalmente la correttezza della soluzione proposta.

$\{x = A \wedge y = B\}$
if $x \neq y$ **then** x := E; **else** x := x+1; **fi**
$\{x > y\}$

Soluzione Una soluzione corretta è $E = y + 1$, o in generale $E = y + k$ con $k > 0$. L'esercizio si riduce quindi a una semplice applicazione della Regola del Condizionale.
Attenzione: Soluzioni come $E = B + 1$ oppure $E = B + k$ con $k > 0$ non sono corrette, perché B è una variable di specifica e non può comparire in un comando.

ESERCIZIO 6
Si consideri il seguente programma annotato, dove '*' è l'operatore di moltiplicazione:

{Inv : y = x * n ∧ x ∈ [0, n] }{t: n - x}
while x≠n **do**
x,y := x+1, y+n
endw
{y = n^2}

- Scrivere e dimostrare le proprietà di invarianza
- Dire perché
  t: x + 2
  non è una buona funzione di terminazione, giustificando formalmente la risposta.

Soluzione La proprietà di invarianza è la tripla

$\{y = x * n \wedge x \in [0, n] \wedge x \neq n\}$
x,y := x+1, y+n
$\{y = x * n \wedge x \in [0, n] \wedge def(x \neq n)\}$

che si risolve con una semplice applicazione della Regola per l'Assegnamento.

Per il secondo punto, $t : x + 2$ non è una buona funzione di terminazione perché non soddisfa l'ipotesi di progresso. Infatti è facile mostrare che la seguente tripla non è verificata:

$\{y = x * n \wedge x \in [0, n] \wedge x + 2 = V\}$
x,y := x+1, y+n
$\{x + 2 < V\}$