COMPITO di LOGICA PER INFORMATICA (fila 1)
21 settembre 2005

Nome: Matricola:

Esercizio 1.1
Si consideri la seguente scrittura

$$\to (\forall((x)\ \exists((y)\ P(x)(y))))(\exists((y)\ \forall((x)\ P(y)(x))))$$

Supponendo fissate le arietà per $\to$: $0 \to (0 \to 0)$, $\forall : (0 \to 0) \to 0$ e $\exists : (0 \to 0) \to 0$, determinare le arietà delle variabili x e y e della costante P in modo che ne risulti una espressione corretta che denoti una formula nel linguaggio del primo ordine.
Si determini poi una interpretazione che falsifichi la formula così ottenuta.

Soluzione. Affinchè la scrittura considerata risulti una formula è necessario che si tratti di una espressione di arietà 0. Vista l'arietà di $\to$ questo si verifica se e solo se $\forall((x)\ \exists((y)\ P(x)(y)))$ e $\exists((y)\ \forall((x)\ P(y)(x)))$ sono entrambe espressioni di arietà 0. Ora affinchè $\forall((x)\ \exists((y)\ P(x)(y)))$ sia una espressione di arietà 0 bisogna che $(x)\ \exists((y)\ P(x)(y))$ sia una espressione di arietà $0 \to 0$ e quindi x deve essere una variabile di arietà 0 e $\exists((y)\ P(x)(y))$ deve essere una espressione di arietà 0; ma allora $(y)\ P(x)(y)$ deve essere una espressione di arietà $0 \to 0$ e quindi y deve essere una variabile di arietà 0 e $P(x)(y)$ deve essere una espressione di arietà 0 e, vista la scelta fatta per le arietà di x e di y, questo accade se e solo se P è una costante di arietà $0 \to (0 \to 0)$. Per concludere basta ora verificare che con le scelte fatte per le arietà di P, x e y succeda che anche $\exists((y)\ \forall((x)\ P(y)(x)))$ sia una espressione di arietà 0, ma questo è immediato.
La scrittura che abbiamo studiato denota quindi la formula

$$(\forall x.\exists y.P(x, y)) \to (\exists y.\forall x.P(y, x))$$

Vediamo ora una struttura ed una interpretazione che la falsifica. Consideriamo un insieme $U \equiv \{0, 1\}$ costituito di due elementi e la relazione $\mathsf{R} \equiv \{\langle 0, 1\rangle, \langle 1, 1\rangle\}$ ed interpretiamo il segno predicativo nella relazione R. Allora l'antecedente della implicazione viene interpretato in vero mentre il conseguente viene interpretato in falso e quindi l'implicazione risulta falsificata.

Esercizio 1.2
Si supponga che

$$A_1 \vee B_1, \dots, A_n \vee B_n, \dots \models C$$

Dimostrare che

1. $$A_1, B_2, \dots, A_{2n+1}, B_{2(n+1)}, \dots \models C$$

2. esiste un numero naturale n tale che

$$A_1, B_2, \dots, A_{2n+1}, B_{2(n+1)} \models C$$

Soluzione. Per risolvere la prima parte dell'esercizio cominciamo notando che

$$A_1 \vee B_1, \dots, A_n \vee B_n, \dots \models C$$

vale se e solo se l'insieme $\{A_1 \vee B_1, \dots, A_n \vee B_n, \dots, \neg C\}$ è non soddisfacibile, vale a dire che per ogni interpretazione σ accade che almeno uno delle formule di tale insieme viene interpretata da σ in falso. Ora se si tratta della formula $\neg C$ l'insieme $\{A_1, B_2, \dots, A_{2n+1}, B_{2(n+1)}, \dots, \neg C\}$ è chiaramente non soddisfatto da σ e d'altra parte se si tratta di una disgiunzione $A_{2n+1} \vee B_{2n+1}$ di posto dispari nella lista di formule considerata allora $(A_{2n+1} \vee B_{2n+1})^\sigma = \perp$ e quindi anche $(A_{2n+1})^\sigma = \perp$ e, analogamente, se si tratta di una disgiunzione $A_{2(n+1)} \vee B_{2(n+1)}$ di posto pari nella lista allora $(A_{2(n+1)} \vee B_{2(n+1)})^\sigma = \perp$ e quindi anche $(B_{2(n+1)})^\sigma = \perp$ e quindi anche in questi due casi l'insieme $\{A_1, B_2, \dots, A_{2n+1}, B_{2(n+1)}, \dots, \neg C\}$ non è soddisfatto da σ. Abbiamo quindi dimostrato che non esiste nessuna interpretazione che soddisfa l'insieme $\{A_1, B_2, \dots, A_{2n+1}, B_{2(n+1)}, \dots, \neg C\}$ e quindi $A_1, B_2, \dots, A_{2n+1}, B_{2(n+1)}, \dots \models C$.
Per risolvere ora la seconda parte dell'esercizio basta ora una semplice applicazione del teorema di compattezza, in quanto, in virtù di tale teorema, dal fatto che l'insieme $\{A_1, B_2, \dots, A_{2n+1}, B_{2(n+1)}, \dots, \neg C\}$ sia non soddisfacibile si ottiene che esiste un numero naturale n tale che l'insieme $\{A_1, B_2, \dots, A_{2n+1}, B_{2(n+1)}, \neg C\}$ è non soddisfacibile che porta subito a $A_1, B_2, \dots, A_{2n+1}, B_{2(n+1)} \models C$.

Esercizio 1.3
Si considerino le seguenti formule

1. $\forall y.(\forall x.(R(x,y) \to P(x)) \to P(y))$
2. $\forall x.((\neg\exists z.R(z,x)) \to P(x))$
3. $\forall x.\forall z.((R(z,x) \to \forall w.\neg R(w,z))$

Si dimostri dapprima semanticamente che le prime due tra esse non sono sufficienti per dimostrare la formula $\forall w.P(w)$ (sugg.: si trovi un modello per l'insieme costituito da tali formule e $\exists w.\neg P(w)$).
Si dimostri poi sintatticamente, utilzzando il metodo di risoluzione o gli alberi di confutazione, che l'insieme costituito da tutte e tre le formule è sufficiente per dimostrare $\forall w.P(w)$.

Soluzione. Per risolvere la prima parte dell'esercizio seguendo il suggerimento dato, basta considerare una struttura costituita da un solo elemento $U \equiv \{*\}$ e interpretare il segno predicativo R nella relazione totale $\mathsf{R} \equiv \{\langle *, * \rangle\}$ e il segno predicativo P nel sottoinsieme vuoto di U, vale a dire in modo che esso risulti sempre falso. Allora chiaramente $\exists w.\neg P(w)$ viene interpretata in vero; inoltre $\forall y.(\forall x.(R(x,y) \to P(x)) \to P(y))$ viene interpretata in vero visto che, per qualsiasi interpretazione della variabile y, $\forall x.(R(x,y) \to P(x))$ viene interpretato in falso, e $\forall x.((\neg\exists z.R(z,x)) \to P(x))$ viene interpretato in vero visto, che per ogni interpretazione della variabile x, $\neg\exists z.R(z,x)$ viene interpretato in falso.
Per quanto riguarda invece la seconda parte dell'esercizio possiamo utilizzare il metodo di risoluzione per verificare che l'insieme costituito dalle formule

$$\begin{array}{l}\{\forall y.(\forall x.(R(x,y) \to P(x)) \to P(y)), \forall x.((\neg\exists z.R(z,x)) \to P(x)),\\ \ \forall x.\forall z.((R(z,x) \to \forall w.\neg R(w,z)), \neg\forall w.P(w)\}\end{array}$$

è non soddisfacibile.
Dopo l'opportuna trasformazione otteniamo il seguente insieme di clasuole in cui come al solito non scriviamo i quantificatori universali e sostituiamo i quantificatori esistenziali con opportune costanti o funzioni di scelta

$$\begin{array}{ll}(1) & \{R(f(y),y) \vee P(y),\\ (2) & \neg P(f(y)) \vee P(y),\\ (3) & R(h(x),x) \vee P(x),\\ (4) & \neg R(z,x) \vee \neg R(w,z)\\ (5) & \neg P(a)\}\end{array}$$

Per ottenere la clausola vuota possiamo ora procedere così:

(a)	$R(f(a),a) \vee P(a)$	istanza della clausola 1 con $y := a$
(b)	$R(f(a),a)$	risoluzione con clausola 5
(c)	$\neg R(h(f(a)),a)$	risoluzione con clausola 4 con $z := f(a), x := a, w := h(f(a))$
(d)	$P(f(a))$	risoluzione con clausola 3 con $x := f(a)$
(e)	$P(a)$	risoluzione con clausola 2 con $y := a$
(f)	$\square$	risoluzione con clausola 5

Esercizio 1.4
Si trovi una funzione recursiva and che quando applicata a due argomenti dia come risultato 0 se e solo se almeno uno dei due argomenti vale 0 e se ne costruisca il *nome* utilizzando i costrutti Z, S, P, C e R. Si trasformi poi tale *nome* in un programma PROLOG.

Soluzione. Una possibile definizione per la funzione and è la seguente definizione ricorsiva

$$\begin{cases} \mathsf{and}(0,y) & = & 0 & = & \mathsf{Z}^1(y) \\ \mathsf{and}(x+1,y) & = & y & = & \mathsf{P}^3_2(x,y,\mathsf{and}(x,y)) \end{cases}$$

per cui otteniamo che $\mathsf{and} \equiv \mathsf{R}[\mathsf{Z},\mathsf{P}^3_2]$.
Se utilizziamo ora la *traduzione automatica* otteniamo il seguente programma PROLOG:

$$\begin{array}{l} \mathsf{AND}(0,y,z) \leftarrow \mathsf{Z}(y,z) \\ \mathsf{AND}(s(x),y,z) \leftarrow \mathsf{AND}(x,y,w), \mathsf{P}^3_2(x,y,w,z) \\ \mathsf{Z}(y,0) \leftarrow \\ \mathsf{P}^3_2(x,y,z,y) \leftarrow \end{array}$$

che si può semplificare in

$$\begin{array}{l} \mathsf{AND}(0,y,0) \leftarrow \\ \mathsf{AND}(s(x),y,y) \leftarrow \end{array}$$

COMPITO di LOGICA PER INFORMATICA (fila 2)
21 settembre 2005

Nome: Matricola:

Esercizio 2.1
Si consideri la seguente scrittura

$$\rightarrow (\forall((x)\ P(f(x))))(\forall((x)\ P(x)))$$

Supponendo fissate le arietà per $\rightarrow$: $0 \rightarrow (0 \rightarrow 0)$ e $\forall : (0 \rightarrow 0) \rightarrow 0$, determinare le arietà della variabile x e delle costanti f e P in modo che ne risulti una espressione corretta che denoti una formula nel linguaggio del primo ordine.
Si determini poi una interpretazione che falsifichi la formula così ottenuta.

Soluzione. Affinchè la scrittura considerata risulti una formula è necessario che si tratti di una espressione di arietà 0. Vista l'arietà di $\rightarrow$ questo si verifica se e solo se $\forall((x)\ P(f(x)))$ e $\forall((x)\ P(x))$ sono entrambe espressioni di arietà 0. Ora affinchè $\forall((x)\ P(f(x)))$ sia una espressione di arietà 0 bisogna che $(x)\ P(f(x))$ sia una espressione di arietà $0 \rightarrow 0$ e quindi x deve essere una variabile di arietà 0 e $P(f(x))$ deve essere una espressione di arietà 0; ma allora, vista la scelta fatta per le arietà di x, affinchè questo accada bisogna che f sia una costante di arietà $0 \rightarrow \alpha$ e P sia una costante di arietà $\alpha \rightarrow 0$ per qualche arietà α. Per concludere basta ora verificare che esista una qualche arietà α che fa si che anche $\forall((x)\ P(x))$ sia una espressione di arietà 0. Affinchè questo accada bisogna che $(x)\ P(x)$ sia una espressione di arietà $0 \rightarrow 0$ e quindi, avendo noi già determinato l'arietà di x bisogna semplicemente che richiediamo che α sia 0.
La scrittura che abbiamo studiato denota quindi la formula

$$(\forall x.P(f(x)) \rightarrow (\forall x.P(x))$$

Vediamo ora una struttura ed una interpretazione che la falsifica. Consideriamo un insieme $U \equiv \{0, 1\}$ costituito di due elementi, la funzione $\mathsf{F} \equiv \{\langle 0, 1\rangle, \langle 1, 1\rangle\}$ in cui interpretiamo il segno funzionale f e il sottoinsieme $R \equiv \{1\}$ di U in cui interpretiamo il segno di predicato P. Allora l'antecedente della implicazione viene interpretato in vero, visto che la funzione F manda ogni elemento in 1 e su tale valore il predicato P viene interpretato in vero, mentre il conseguente viene interpretato in falso, visto che l'interpretazione di P non vale su 0, e quindi l'implicazione risulta falsificata.

Esercizio 2.2
Si supponga che

$$A_1 \vee B_1, \ldots, A_n \vee B_n, \ldots \models C$$

Dimostrare che

1. $$B_1, A_2, \ldots, B_{2n+1}, A_{2(n+1)}, \ldots \models C$$

2. esiste un numero naturale n tale che

$$B_1, A_2, \ldots, B_{2n+1}, A_{2(n+1)} \models C$$

Soluzione. Per risolvere la prima parte dell'esercizio cominciamo notando che

$$A_1 \vee B_1, \ldots, A_n \vee B_n, \ldots \models C$$

vale se e solo se l'insieme $\{A_1 \vee B_1, \ldots, A_n \vee B_n, \ldots, \neg C\}$ è non soddisfacibile, vale a dire che per ogni interpretazione σ accade che almeno uno delle formule di tale insieme viene interpretata da σ in falso. Ora se si tratta della formula $\neg C$ l'insieme $\{B_1, A_2, \ldots, B_{2n+1}, A_{2(n+1)}, \ldots, \neg C\}$ è chiaramente non soddisfatto da σ e d'altra parte se si tratta di una disgiunzione $A_{2n+1} \vee B_{2n+1}$ di posto dispari nella lista di formule considerata allora $(A_{2n+1} \vee B_{2n+1})^\sigma = \perp$ e quindi anche $(B_{2n+1})^\sigma = \perp$ e, analogamente, se si tratta di una disgiunzione $A_{2(n+1)} \vee B_{2(n+1)}$ di posto pari nella lista allora $(A_{2(n+1)} \vee B_{2(n+1)})^\sigma = \perp$ e quindi anche $(A_{2(n+1)})^\sigma = \perp$ e quindi anche in questi due casi l'insieme $\{B_1, A_2, \ldots, B_{2n+1}, A_{2(n+1)}, \ldots, \neg C\}$ non è soddisfatto da σ. Abbiamo quindi dimostrato che non esiste nessuna interpretazione che soddisfa l'insieme $\{B_1, A_2, \ldots, B_{2n+1}, A_{2(n+1)}, \ldots, \neg C\}$ e quindi $B_1, A_2, \ldots, B_{2n+1}, A_{2(n+1)}, \ldots \models C$.
Per risolvere ora la seconda parte dell'esercizio basta ora una semplice applicazione del teorema di compattezza, in quanto, in virtù di tale teorema, dal fatto che l'insieme $\{B_1, A_2, \ldots, B_{2n+1}, A_{2(n+1)}, \ldots, \neg C\}$ sia non soddisfacibile si ottiene che esiste un numero naturale n tale che l'insieme $\{B_1, A_2, \ldots, B_{2n+1}, A_{2(n+1)}, \neg C\}$ è non soddisfacibile che porta subito a $B_1, A_2, \ldots, B_{2n+1}, A_{2(n+1)} \models C$.

Esercizio 2.3
Si considerino le seguenti formule

1. $\forall y.(Q(y) \to \exists z.R(z, y))$
2. $\forall x.(Q(x) \to \exists z.(R(z, x) \ \&\ Q(z)))$
3. $\forall x.\forall z.((R(z, x) \to \forall w.\neg R(w, z))$

Si dimostri dapprima semanticamente che le prime due tra esse non sono sufficienti per dimostrare la formula $\neg\exists w.Q(w)$ (sugg.: si trovi un modello per l'insieme costituito da tali formule e $\exists w.Q(w)$).
Si dimostri poi sintatticamente, utilzzando il metodo di risoluzione o gli alberi di confutazione, che l'insieme costituito da tutte e tre le formule è sufficiente per dimostrare $\neg\exists w.Q(w)$.

Soluzione. Per risolvere la prima parte dell'esercizio seguendo il suggerimento dato, basta considerare una struttura costituita da un solo elemento $U \equiv \{*\}$ e interpretare il segno predicativo R nella relazione totale $\mathsf{R} \equiv \{\langle *, * \rangle\}$ e il segno predicativo Q nell'intero set U, vale a dire in modo che esso risulti sempre vero. Allora chiaramente $\exists w.Q(w)$ viene interpretata in vero; inoltre $\forall y.(Q(y) \to \exists z.R(z, y))$ viene interpretata in vero visto che, per qualsiasi interpretazione della variabile y, $\exists z.R(z, y)$ viene interpretato in vero, e $\forall x.(Q(x) \to \exists z.(R(z, x) \ \&\ Q(z)))$ viene interpretato in vero visto, che per ogni interpretazione della variabile x, $\exists z.(R(z, x) \ \&\ Q(z))$ viene interpretato in vero.
Per quanto riguarda invece la seconda parte dell'esercizio possiamo utilizzare il metodo di risoluzione per verificare che l'insieme costituito dalle formule

$$\begin{array}{l}\{\forall y.(Q(y) \to \exists z.R(z, y)), \forall x.(Q(x) \to \exists z.(R(z, x) \ \&\ Q(z))), \\ \ \forall x.\forall z.((R(z, x) \to \forall w.\neg R(w, z)), \exists w.Q(w)\}\end{array}$$

è non soddisfacibile.
Dopo l'opportuna trasformazione otteniamo il seguente insieme di clasuole in cui come al solito non scriviamo i quantificatori universali e sostituiamo i quantificatori esistenziali con opportune costanti o funzioni di scelta

$$\begin{array}{ll}(1) & \{\neg Q(y) \vee R(h(y), y), \\ (2) & \neg Q(x) \vee R(f(x), x), \\ (3) & \neg Q(x) \vee Q(f(x)), \\ (4) & \neg R(z, x) \vee \neg R(w, z) \\ (5) & Q(a)\}\end{array}$$

Per ottenere la clausola vuota possiamo ora procedere così:

(a)	$\neg Q(a) \vee R(f(a), a)$	istanza della clausola 2 con $x := a$
(b)	$R(f(a), a)$	risoluzione con clausola 5
(c)	$\neg R(h(f(a)), f(a))$	risoluzione con clausola 4 con $z := f(a), x := a, w := h(f(a))$
(d)	$\neg Q(f(a))$	risoluzione con clausola 1 con $y := f(a)$
(e)	$\neg Q(a)$	risoluzione con clausola 3 con $x := a$
(f)	$\square$	risoluzione con clausola 5

Esercizio 2.4
Si trovi una funzione recursiva or che quando applicata a due argomenti dia come risultato 0 se e solo se entrambi i due argomenti valgono 0 e se ne costruisca il *nome* utilizzando i costrutti Z, S, P, C e R. Si trasformi poi tale *nome* in un programma PROLOG.

Soluzione. Una possibile definizione per la funzione or è la seguente definizione ricorsiva

$$\begin{cases} \mathsf{or}(0, y) & = & y & = & \mathsf{P}^1_1(y) \\ \mathsf{or}(x+1, y) & = & x+1 & = & \mathsf{C}[\mathsf{S}, \mathsf{P}^3_1](x, y, \mathsf{or}(x, y)) \end{cases}$$

per cui otteniamo che $\mathsf{or} \equiv \mathsf{R}[\mathsf{P}^1_1, \mathsf{C}[\mathsf{S}, \mathsf{P}^3_1]]$.
Se utilizziamo ora la *traduzione automatica* otteniamo il seguente programma PROLOG:

$$\begin{array}{l} \mathsf{OR}(0, y, z) \leftarrow \mathsf{P}^1_1(y, z) \\ \mathsf{OR}(s(x), y, z) \leftarrow \mathsf{OR}(x, y, w), \mathsf{P}^3_1(x, y, w, t), \mathsf{S}(t, z) \\ \mathsf{S}(x, s(x)) \leftarrow \\ \mathsf{P}^1_1(y, y) \leftarrow \\ \mathsf{P}^3_1(x, y, z, x) \leftarrow \end{array}$$

che si può semplificare in

$$\begin{array}{l} \mathsf{OR}(0, y, y) \leftarrow \\ \mathsf{OR}(s(x), y, s(x)) \leftarrow \end{array}$$