Scritto di Logica Matematica,
11 febbraio 2009

Ho pubblicato i risultati degli esami su esse3, dopo aver iscritto tutti gli studenti che si sono presentati (anche quelli della specialistica). Dovreste aver ricevuto un messaggio con l'esito. Se così non fosse, mandatemi un messaggio via mail.

1. Siano F, G le due formule seguenti:

$$F = \forall x(\exists y \neg r(x, gy) \to \exists z r(z, x)), \qquad G = \forall x(\forall y r(x, gy) \vee \exists y r(y, x))$$

Dimostrare che $F \models G$ utilizzando il metodo di risoluzione. Nelle risoluzioni, utilizzare la fattorizzazione o spiegare perché non sia possibile utilizzarla.

Soluzione

Dimostriamo che $F \wedge \neg G$ è insoddisfacibile utilizzando il metodo di risoluzione. Per prima cosa trasformiamo $F \wedge \neg G$ in un insieme di clausole equisoddisfacibile:

La formula F è equivalente a $\forall x \exists z \forall y(r(x, gy) \vee r(z, x))$, che a sua volta è equisoddisfacibile a $\forall x \forall y(r(x, gy) \vee r(fx, x))$. Possiamo quindi rappresentare F con la clausola $C_1 = \{r(x, gy), r(fx, x)\}$.

La formula $\neg G$ è equivalente a $\exists x(\neg \forall y r(x, gy) \wedge \neg \exists y r(y, x))$, e quindi a $\exists x(\exists y \neg r(x, gy) \wedge \forall y \neg r(y, x))$. Skolemizzando, otteniamo la formula $\neg r(a, gb) \wedge \neg r(y, a)$. Possiamo quindi rappresentare F con l'insieme di clausole $\{C_2, C_3\}$, dove $C_2 = \{\neg r(a, gb)\}$ e $C_3 = \{\neg r(y, a)\}$.

Risolvendo il letterale $r(x, gy)$ di C_1 con il letterale $\neg r(a, gb)$ di C_2 si ottiene la clausola $\{r(fa, a)\}$. Un ulteriore passo di risoluzione con la clausola C_3 ci porta alla clausola vuota.

La fattorizzazione nel primo passo non è applicabile perché i letterali $r(x, gy), r(fx, x)$ non sono unificabili (cercando di risolvere il sistema corrispondente si arriva all'equazione $x = fx$).

2. Sia $\mathcal{L} = \{r, =\}$, dove r è un simbolo di relazione binario. Considera il seguente enunciato

$$F = \forall x(\exists y r(x, y) \to r(x, x)).$$

a) Date due interpretazioni I, J di L tali che $I \models F$, $J \models \neg F$, per quali numeri naturali n siamo certi che Spoiler abbia una strategia vincente nel gioco di Ehrenfeucht di n passi fra I, J? Descrivi nei dettagli la strategia di Spoiler.

b) Determina un numero $n > 0$ e due interpretazioni I, J di L tali che $I \models F$, $J \models \neg F$ e Duplicator ha una strategia vincente nel gioco di Ehrenfeucht di n passi fra I, J.

Soluzione

a) La formula F ha rango due ed è vera in I ma falsa in J. Questo implica che Spoiler avrà una strategia nel gioco di n passi, se $n \geq 2$.
La formula $\neg F$ è equivalente a $\exists x \exists y(r(x, y) \wedge \neg r(x, x))$.
Siano x_0, y_0 due elemeti in D^J che soddisfano : $(x_0, y_0) \in r^J$, $(x_0, x_0) \notin r^J$.
La strategia di Spoiler nel gioco di due mosse è la seguente:
Spoiler sceglie x_0 in D^J. Duplicator deve replicare con un elemento $x \in D^I$ tale che $(x, x) \notin r^I$, altrimenti ha già perso. Notiamo che un tale x non può essere in relazione con alcun elemento $y \in D^I$, altrimenti la formula F sarebbe falsa in I. Quindi, se Duplicator sceglie un tale x, Spoiler può scegliere y_0 in D^J. Per vincere, Duplicator dovrebbe replicare con $y \in D^I$ con $(x, y) \in r^I$, ma nessun tale y esiste in D^I. La strategia di Spoiler è quindi vincente.

b) Possiamo prendere $n = 1$ e $D^I = \{0, 1\}, r^I = \emptyset$ e $D^J = \{0, 1\}, r^J = \{(0, 1)\}$. Qualsiasi scelta d Spoiler faccia, in D^I o D^J, Duplicator può rispondere con un qualsiasi elemento d' dell'altra struttura: la corrispondenza $d \to d'$ è infatti unisomorfismo locale, visto che né d né d' sono in relazione con sè stessi.

3. a) Sia $\mathcal{L} = \{\epsilon, [\,|\,]\}$ l'usuale linguaggio delle liste. Scrivere un programma Prolog P per descrivere il predicato binario:
$EVENSX(L, L') \Leftrightarrow L'$ è ottenuta da L cancellando tutti gli elementi in posizione dispari, partendo da sinistra; (ad esempio, vogliamo che il goal $: -EVENSX([x, y, z, w], [y, w])$ sia refutabile).

b) Scrivere un altro programma P' per descrivere il predicato binario:
$EVENDX(L, L') \Leftrightarrow L'$ è ottenuta da L cancellando tutti gli elementi in posizione dispari, partendo da destra; (ad esempio, vogliamo che il goal $: -EVENDX([x, y, z, w], [x, z])$ sia refutabile).

c) Sia L un linguaggio, P un programma Prolog e H un'interpretazione di Herbrand di L (vista come sottoinsieme della base di Herbrand). È vero che, se H rende vero P e $A \in H$ (A formula atomica chiusa), allora $P \models A$?

Soluzione

a)

$$\begin{cases} EVENSX(\epsilon, \epsilon') : - \\ EVENSX([x], [x]) : - \\ EVENSX([x, y|L], [y|L']) : -EVENSX(L, L') \end{cases}$$

b) Il programma per $EVENDX$ si ottiene aggiungendo al programma P_{REV}

$$P_{REV} \begin{cases} REV(x, y) : -ACC(x, \epsilon, y) \\ ACC(\epsilon, z, z) : - \\ ACC([x|w], z, y) : -ACC(w, [x|z], y) \end{cases}$$

la clausola seguente:

$$EVENDX(L, L_1) : -REV(L, L_2), EVENSX(L_2, L_3), REV(L_3, L_1),$$

(il programma per il predicato REV è sulle dispense).

c) L'affermazione è valida solo se H è il modello minimo. Ad esempio, se P è dato dalla sola clausola $q(a) : -p(a)$, l'interpretazione $H = \{q(a)\}$ è modello del programma, ma $q(a)$ non è conseguenza logica del programma.