Logica e algebra

7 luglio 2016

Laboratorio

Chiamiamo preordine su un insieme X una relazione binaria R su X che sia riflessiva e transitiva. Per ogni $x,y\in X$, definiamo sup(x,y) rispetto ad R in modo analogo a quanto fatto per una relazione d'ordine (ovvero sup(x,y) è un elemento $z\in X$ tale che $(x,z)\in R$, $(y,z)\in R$ e, per ogni $v\in X$, se $(x,v)\in R$ e $(y,v)\in R$ allora $(z,v)\in R$). Diciamo che una funzione $f:X\rightarrow X$ preserva Il preordine R se per ogni $x,y\in X$, $(x,y)\in R$ implica $(f(x),f(y))\in R$, e diciamo che f preserva il sup se supposto z=sup(x,y) si ha f(z)=sup(f(x),f(y)).

Scelto un opportuno linguaggio del I ordine, formalizzare il seguente enunciato.

Sia R *un preordine su* X, *siano* f *e* g *funzioni da* X *ad* X *che preservano* R *tali che* $(f(x),y)\in R$ *se e solo se* $(x,g(y))\in R$. *Allora* f *preserva anche il* sup.

Scrivere poi un programma SPASS per verificare l'enunciato.

Soluzione

Il nostro linguaggio deve contenere una lettera predicativa R di arità 2 e due lettere funzionali f, g di arità 1. E' utile far uso delle seguenti abbreviazioni per formalizzare il problema:

$U(x,y,z)\equiv R(x,z)\wedge R(y,z)$ per indicare che z è un "maggiorante" di x ed y rispetto ad R
$S(x,y,z)\equiv(U(x,y,z)\wedge\forall w(U(x,y,w)\Rightarrow R(z,w)$ per indicare che z=sup(x,y)

Utilizzando anche questa estensione del linguaggio l'enunciato può essere espresso dalle seguenti formule:

R è un preordine : $\forall xR(x,x)$; $\forall x\forall y\forall z(R(x,y)\wedge R(y,z)\Rightarrow R(x,z)$
f e g preservano il preordine : $\forall x\forall y(R(x,y)\Rightarrow R(f(x),f(y))$, $\forall x\forall y(R(x,y)\Rightarrow R(g(x),g(y))$
legame fra f e g: $\forall x\forall y(R(f(x),y)\Leftrightarrow R(x,g(y))$
f preserva il sup: $\forall x\forall y\forall z(S(x,y,z)\Rightarrow S(f(x),f(y),f(z))$

Il codice SPASS, rimuovendo le abbreviazione, è.

```
begin_problem(POA).

list_of_descriptions.
name({**}).
author({**}).
status(unknown).
description({**}).
end_of_list.
```

```
list_of_symbols.
functions[(f,1),(g,1)].
predicates[(R,2)].
end_of_list.

list_of_formulae(axioms).
formula(forall([x],R(x,x))).
formula(forall([x,y,z],implies(and(R(x,y),R(y,z)),R(x,z)))).
formula(forall([x,y],implies(R(x,y),R(f(x),f(y))))).
formula(forall([x,y],implies(R(x,y),R(g(x),g(y))))).
formula(forall([x,y],equiv(R(f(x),y),R(x,g(y))))).
end_of_list.

list_of_formulae(conjectures).
formula(forall([x,y,z],implies(
and(R(x,z), R(y,z), forall([w],implies(and(R(x,w),R(y,w)),R(z,w)))),
and(
R(f(x),f(z)),
R(f(y),f(z)),
forall([w],implies(and(R(f(x),w),R(f(y),w)),R(f(z),w))))
))).
end_of_list.

end_problem.
```