CS 2209b - Quiz 4 - Solutions

Paul Vrbik

March 28, 2011

Question 1

Translate the following in the quantificational logic:

(a) Aristotle cause nothing evil.

$$\sim (\exists x)\,(Ex \bullet Cax)$$

(b) Everything is caused by something.

$$(x)\,(\exists y)\,Cyx$$

(c) Something evil caused all evil things.

$$(\exists x)\,(Ex)\,(y)\,(Ey \supset Cxy)$$

(d) God caused everything except himself.

$$(x)\,(\sim x = g \supset Cgx) \bullet \sim Cgg$$

Question 2

Question 2a

1	$(x)\,(Pxk \supset Pkx)$
2	$\sim (\exists x)\,(Nx \bullet Pkx)$
3	$\sim (\exists x) \sim (\sim Pxk \supset \sim Nx)$
4	$[\,\therefore \quad \sim Nj$

Solution.

$$\begin{aligned}(x)\,(Pxk \supset Pkx) &\equiv (x)\,(\sim Pxk \vee Pkx)\\ &\equiv (x)\,(\sim Nx \vee \sim Pkx)\\ &\equiv (x)\,(Px \vee \sim Nx)\\ &\to \{\sim Pxk, Pkx\}\end{aligned}$$

$$\begin{aligned}\sim(\exists x)(Nx \bullet Pkx) &\equiv (x)(\sim Nx \vee \sim Pkx)\\ &\equiv \sim Nx \vee \sim Pkx\\ &\rightarrow \{\sim Nx, \sim Pkx\}\end{aligned}$$

$$\begin{aligned}\sim(\exists x) \sim(\sim Pxk \supset \sim Nx) &\equiv \sim(\exists x) \sim(Pxk \vee \sim Nx)\\ &\equiv (x)(Pxk \vee \sim Nx)\\ &\equiv Pzx \vee \sim Nx\\ &\rightarrow \{Pxk, \sim Nx\}\end{aligned}$$

Negated conclusion:

$$\sim\sim Nj \equiv Nj.$$

Resolution proof:

1	$\{\sim Pxk, Pkx\}$	premise
2	$\{\sim Nx, \sim Pkx\}$	premise
3	$\{Pxk, \sim Nx\}$	premise
4	$\{Nj\}$	negated goal
5	$\{\sim Pxk, \sim Nx\}$	: 1, 2
6	$\{\sim Nx\}$	$\{x \leftarrow j\}$: 3, 5
7	$\{\}$	$\{x \leftarrow j\}$: 4, 6

■

Question 2b

1	$\sim(\exists x)(Qx \vee \sim Ex)$
2	$[\therefore \quad (\sim(\exists x)Qx \bullet \sim(\exists x) \sim Ex)$

Solution.

$$\begin{aligned}\sim(\exists x)(Qx \vee \sim Ex) &\equiv (x)(\sim Qx \bullet Ex)\\ &\equiv \sim Qx \bullet Ex\\ &\rightarrow \{\sim Qx\}, \{Ex\}\end{aligned}$$

Negated conclusion:

$$\begin{aligned} \sim (\sim (\exists x)\, Qx \bullet \sim (\exists x) \sim Ex) &\equiv (\exists x)\, Qx \vee (\exists x) \sim Ex \\ &\equiv (\exists x)\, Qx \vee \exists y \sim Ey \\ &\equiv Qa \vee \sim Eb \\ &\rightarrow \{Qa, \sim Eb\} \end{aligned}$$

Resolution proof:

1	$\{\sim Qx\}$	premise
2	$\{Ex\}$	premise
3	$\{Qa, \sim Eb\}$	negated goal
4	$\{\sim Eb\}$	$\{x \leftarrow a\}$: 1, 3
5	$\{\}$	$\{x \leftarrow b\}$: 2, 4

■

Question 3

Write the following facts and rules in Prolog: (1) Mercury orbits around Sun. (2) Venus orbits around Sun. (3) Earth orbits around Sun. (4) Mars orbits around Sun. (5) Moon orbits around Earth. (6) Phobos orbits around Mars. (7) Deimos orbits around Mars. If something orbits around Sun, then it is a planet (8). If something orbits around a planet, then it is a satellite (9).

Solution.

```
orbits(mercury,sun).            %1
orbits(venus,sun).              %2
orbits(earth,sun).              %3
orbits(mars,sun).               %4
orbits(moon,earth).             %5
orbits(phobos,mars).            %6
orbits(deimos,mars).            %7

planet(X) :- orbits(X,sun).     %8

satellite(X) :-
    orbits(X,Y),
    planet(Y).                  %9

%or, one could always be lazy and write:
satellite(X) :- not( orbits(X,sun) )
```

■

Write a Prolog query for each of the question below. How do you get all possible answers for the query? What are all possible answers that Prolog would return for each query?

Solution. To get all answers to each query repeatedly hit ";" until False is returned.

(a) Find what orbits around the sun.

```
?- orbits( X, sun ).
X = mercury ;
X = venus ;
X = earth ;
X = mars ;
false.
```

(b) Find a planet that has a satellite.

```
?- planet(X), orbits(Y,X), satellite(Y).
X = earth,
Y = moon ;
X = mars,
Y = phobos ;
X = mars,
Y = deimos ;
false.
```

(c) Find a planet that has no satellite.

```
?- planet(X), not(orbits(Y,X)).
X = mercury ;
X = venus ;
false.
```

(d) Find a satellite that orbits around earth.

```
?- satellite(X), orbits(X,earth).
X = moon ;
false.
```

■

Question 4

Write a Prolog program for `fact(N,M)` where M is assigned $N!$.

Solution. First observe that

```
factorial(N) = N * factorial(N-1).
```

```
fact(0,1).

fact(N,M) :-
    Nt is N - 1,
    fact( Nt, Mt ),
    M is N * Mt.
```

(By the way $0! = 1$ not 0.) ■