

Model1-Answer

Question1

Consider the following set of axioms: (6Marks)

1. forall x [equal(x,x)]
2. forall y,z [equal(y,z) -> equal(z,y)]
3. forall w,s,t [equal(w,s) and equal (s,t) -> equal(w,t)]
4. equal(B,A)
5. equal(B,C)

And the conclusion:

- $\text{equal}(C,A)$

Where A, B and C denote constants; and x, y, z, w, s, and t denote variables.

Prove the conclusion from the axioms by refutation using resolution (state explicitly which substitutions are made).

Solution

1- Translate the axioms and the negation of the conclusion into clausal form: (3)

- ```

1. equal(x,x)
2. !equal(y,z) ∨ equal(z,y)
3. ![equal(w,s) ^ equal(s,t)] ∨ equal(w,t)
4. !equal(w,s) ∨ !equal(s,t) ∨ equal(w,t)
5. equal(B,A)
6. equal(B,C)
7. (negated conclusion:) !equal(C,A)

```

2- Apply resolution (state explicitly which substitutions are made: (3)

2.  $\neg \text{equal}(y,z) \vee \text{equal}(z,y)$
5.  $\text{equal}(B,C)$

-----

- ```

7.  equal(C, B)                substituting y by b, and z by c.
7.  equal(C, B)
3.  !equal(w,s) ∨ !equal(s,t) ∨ equal(w,t)

```

8. $\neg \text{equal}(B,t) \vee \text{equal}(C,t)$ substituting w by c, and s by b.

- $$\begin{array}{l} 8. \text{!equal}(B,t) \vee \text{equal}(C,t) \\ 4. \text{equal}(B, A) \end{array}$$

9. $\text{equal}(C, A)$ substituting t by a .

- ```
9. equal(C, A)
6. !equal(C, A)
```

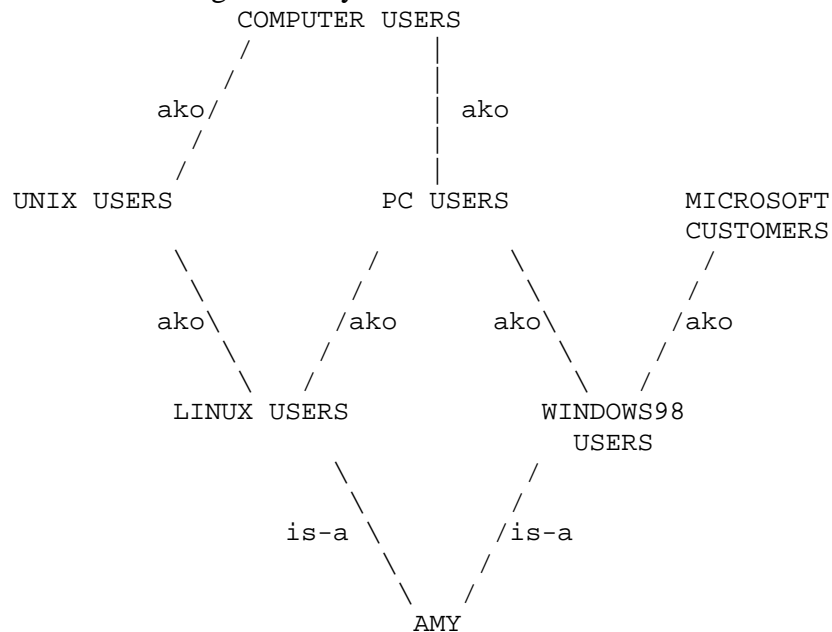
-----

EMPTY CLAUSE

## Question2

(4Marks)

Consider the following hierarchy of frames.



1. Give the class-precedence list for Amy that would be obtained by applying the topological-sorting algorithm to the previous graph (2Marks)

### Class Precedence list :

Amy -Linux Users-Unix Users- Windows98 Users -PC Users- Computer Users  
Microsoft Customers

2. Suppose that each of the classes *Unix users*, *PC users* and *Computer Users* contains a *favorite programming language* slot. The default value for this slot is:
  - Fortran, for the *Computer Users* class.
  - C, for the *Unix Users* class.
  - C++, for the *PC Users* class.

What is the value obtained for Amy's favorite programming language according to the class-precedence list you constructed above? Explain you answer. (2Marks)

According to the class-precedence list

Amy

Linux Users

Unix Users

- Use C

Windows98 Users

PC Users

- Use C++

Computer Users

- Use Fortran

Microsoft Customers

**Amy's favorite programming language is C**

3) Imagine that this program is consulted by the Prolog interpreter: (5Marks)

```
foo([], []).
foo([H|T], [X|Y]) :-
 H = X,
 foo(T, Y).
```

What will be the outcome of each of the following queries?

- A.** ?- foo([a,b,c],A). \_\_\_\_\_ **A = [a,b,c]**  
**B.** ?- foo([c,a,t],[c,u,t]). \_\_\_\_\_ **no**  
**C.** ?- foo(X,[b,o,o]). \_\_\_\_\_ **X = [b,o,o]**  
**D.** ?- foo([p|L],[F|[a,b]]). \_\_\_\_\_ **F = p, L = [a,b]**  
**E.** ?- foo([X,Y],[d,o,g]). \_\_\_\_\_ **no**

Write a program that corresponds to the following recursive functions:

a. Factorial

```
fact(0) = 1
fact(n) = n*fact(n-1), when n>0
```

**Answer:**

```
fact (0, 1). % base case
fact (N, Result) :- % recursion step
 N > 0,
 N1 is N - 1,
 fact (N1, Result1),
 Result is Result1 * N.
```