CSE560: Homework 6

Disjuntive and Negative Knowledge

Question 1

This is a continuation of a question from the previous homework.

If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.

Use the translation of the above into clauses from the answers to the previous homework.

Part c: Prove that the unicorn is horned using a bottom-up proof strategy. Show all of your steps. How is the bottom-up proof procedure you used different from the one for datalog?

Part d: Prove that the unicorn is horned using a top-down proof strategy. Show all of your steps.

Explain what step that you did which is not allowed for the top-down proof procedure for datalog.

Question 2

This is a continuation of a question from the previous homework.

Consider the following KB:

poor(X) $\leftarrow$ *student(X)*
student(john) $\vee$ *student(tim)*

Part b: What is the answer for *?poor(X)*.
Show the derivation using a top-down approach.

Question 3

This is a continuation of a question from the previous homework.

Consider the following clauses:

daughter(Daughter,Mother) $\leftarrow \neg$*male(Daughter)* $\wedge$ *mother(Mother,Daughter)*
mother(mary,nancy)
$\neg$*male(nancy)*

Part d: Show the steps of a top-down proof that daughter(nancy,mary)

Complete World Assumption

Question 4

Consider the following KB.

bird(tweety)
fly(X) $\leftarrow$ *bird(X)* $\wedge$ $\neg$*abnormal(X)*
abnormal(X) $\leftarrow$ *toy(X)*

abnormal(X) $\leftarrow$ *dead(X)*
toy(gun)
dead(elvis)

Part a: Without assuming the CWA (just using normal disjunctive and negative knowledge), can you prove that tweety can fly? If yes, give such a proof, if not, show, using either top-down or bottom-up why the proof fails.

Part b: Can we prove that tweety can fly if we use negation as failure? If so, give the proof? Make sure you show all of the details of the proof.

Question 5

Clark completion is sometimes employed on just some of the predicates. We will be doing that with the KB of the previous question for **abnormal**, **toy** and **dead**.

Part a: Write the Clark normal form of each of the 3 predicates.

Part b: Write the completion assumption of each (which is just the Clark normal form but with the implication reversed).

Part c: Write each of the completion assumptions in conjunctive normal form.

Part d: Add the completion assumptions to the database and prove using the top-down proof procedure for disjunctive knowledge that tweety flies. Note that you can also assume the Unique Name assumption (that equality is the same as unification).

First Order Predicate Calculus

Question 6: Precedence

In arithmetic, '*' is of higher precedence than + and so (6*2)+3 can be written as 6*2+3; whereas for 6*(2+3), the brackets cannot be removed.

In class, we assumed a set of precedence rules for first order predicate calculus. Here are the rules in order.
1. $\neg$ is the highest level of precedence
2. $\vee$ and $\wedge$ are the next highest
3. $\leftarrow$
4. Quantifiers: $\forall$ and $\exists$

Use the rules to add in the missing brackets from each.

Part a: $\forall X \forall Y p(X,Z) \rightarrow q(X,Y) \vee r(Y,Z)$

Part b: $\neg \exists Y \forall X \neg p(X,Z) \rightarrow \neg q(X,Y) \wedge r(Y,Z)$

Question 7

Write the following expressions in First Order logic (using $\forall$, $\exists$, $\wedge$, $\vee$, $\leftarrow$). Make sure you use capital letters for variables. Also, you can use the above precedence rules in order not to show all brackets.

Everybody is a butcher.
Nobody is a butcher.
There is a male butcher.
No man is a butcher.

Programming

Question 8

In this question, you will implement a theorem prover for horne clauses that uses refutation and unit resolution. The knowledge base will not use function symbols, thus quarenteeing that it always halts.

Write the following in conjunctive normal form:

```
ostrich(sam)
canary(tweety)
bird(X) <- ostrich(X)
bird(X) <- canary(X)
fly(X) <- bird(X) ^ normal(X)
not normal(X) <- ostrich(X)
normal(X) <- canary(X)
```

Encode the conjunctive normal form in Tcl. For negative literals, use **not**. So $\neg bird(X) \vee ostrich(X)$ would be encoded as $\{\{not\ \{bird\ X\}\}\ \{ostrich\ X\}\}$.

Here is how your proof procedure should work (if you want to do it a different way, please explain how it works in a paragraph). The proof procedure is a bottom-up one. Let your consequent set be the initial KB. For each pair of clauses, if one of them is a unit clause and they can be resolved and the result is not already in the consequent set, then add it in. Keep doing this until you are not able to add anything to your consequent set.

Please use hw4standard in your code, and make sure your code is easy to read. As your program builds its conclusions, print each out. You can use the following code to see if two clauses are the same (except for variable renaming).

```
proc sameclause {a b} {
    set avars [findvariables $a {}]
    set bvars [findvariables $b {}]
    set sublist {}
    foreach av $avars bv $bvars {
        lappend sublist $av $bv
    }
    set newa [substitute $a $sublist]
    if {$newa == $b} {return 1}
    return 0
}
```

Show the output of your program in trying to prove *fly(sam)* and *fly(tweety)*. Note that for proving each one, you need to add its negation to the knowledge base and see if you can derive false (which will be an empty list).