CS1571
Fall 2019
10/16 In-Class Worksheet

Name: Simran Gidwani

Where were you sitting in class today: Back left

A. Pre-Reflection

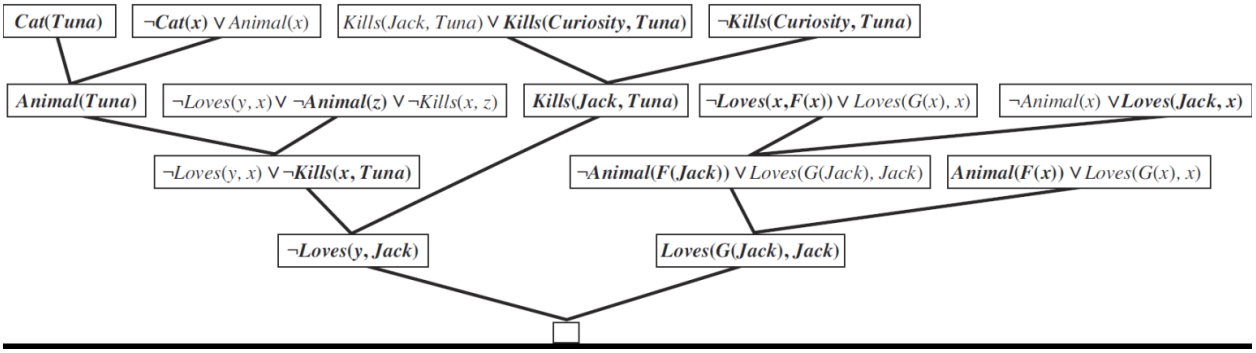
On a scale of 1-5, with 5 being most confident, how well do you think you could execute these learning objectives:

12.1 Demonstrate how to make inferences in FOL using resolution _____
13.1 Analyze the implementation of substitution, unification, and resolution in FOL. _____

B. Prove Curiosity Killed the Cat Using Resolution

Walk through the proof on p.349. Discuss in a group how the diagram maps to the English language proof.

Suppose Curiosity did not kill Tuna. We know that either Jack or Curiosity did; thus Jack must have. Now, Tuna is a cat and cats are animals, so Tuna is an animal. Because anyone who kills an animal is loved by no one, we know that no one loves Jack. On the other hand, Jack loves all animals, so someone loves him; so we have a contradiction. Therefore, Curiosity killed the cat.



B. Explore Code

Download the 10-16-code folder from CourseWeb (this code is taken from https://github.com/aimacode/aima-python). We've already provided some code for you in curiosity.py – you should be working in this file.

1. Call logic.subst to substitute *Curiosity* for *x* in the expression ~Animal(x) v Loves(Jack, x). Copy your line of code here:

```
print(logic.subst({expr('F(x)'): expr('Curiosity')}, expr('~Animal(x) V Loves(Jack, x)')))
```

2. Next, we've created a knowledge base for you in curiosity.py that maps to the KB above (curiosity_kb). Call logic.unify on terms and negations of those terms from the KB. What are some examples of sentences that unify together that should? What are some examples of sentences that don't unify together and shouldn't? Which sentences do not unify together that should?

```
print(logic.unify(expr('Cat(Tuna)'), expr('Cat(x)')))


does not unify:

print(logic.unify(expr('Animal(z)'), expr('Animal(F(x)'))
```

3. Adapt pl_resolution to apply to first-order logic. Assume clauses are already in CNF and standardized. We've provided a fol_resolution function, but you need to adapt pl_resolve to fol_resolve by checking for unification of literals instead of the negation of literals. You also need to make the appropriate substitutions as you add new clauses. Test your code on curiosity_kb_shortened to prove that not everyone loves Jack (~Loves(y, Jack)). Paste your function below.

4. Bonus: Implement Skolem functions in the code.

C. Post-Reflection

On a scale of 1-5, with 5 being most confident, how well do you think you could execute these learning objectives:

12.2 Demonstrate how to make inferences in FOL using resolution _____
13.1 Analyze the implementation of substitution, unification, and resolution in FOL. _____