**CS1571 Intro to Artificial Intelligence  
Fall 2020 – Week 9  
10/21 Wednesday Worksheet**

**Name: Connor Johnson**

**Groupmates: Caroline Pridgen, George Meisinger**

**I am ok with having my solutions discussed in class on Monday: Yes**/No

**Textbook Reading:** Russell & Norvig, Chapters 9, 11

**Videos:** Week 9, Videos 1-3

**This Week’s Learning Objectives:**

* Explain substitution, unification, and resolution in FOL
* Conduct a proof by resolution in FOL
* Convert a sentence to CNF in FOL
* Describe Horn Clauses and their properties
* Explain forward chaining in propositional logic and FOL
* Explain backward chaining in propositional logic and FOL
* Define planning algorithms
* Create an action representation for a planning algorithm
* Explain a forward planning approach
* Explain a backward planning approach

**Worksheet Activities:**

* Practice manipulating FOL statements and applying the resolution rule in FOL
* Interpret the forward chaining algorithm
* Answer some questions related to forward planning and backwards planning

1. **Resolution in FOL**
2. Consider the following expression:

*Everyone who is kind to all animals is loved by somebody.*

∀*x* (∀*y* Animal(y) ⇒ Kind(x,y)) ⇒ (∃*y* Loves(y,x))

Demonstrate how to convert it to CNF. You should get:   
( Animal(F(x)) v (Loves(G(x),x)) ^ (~Kind(x,F(x)) v Loves(G(x),x))

Step 1: Remove the biconditionals and implications.

∀*x~* (∀*y* ~Animal(y) v Kind(x,y)) v (∃*y* Loves(y,x))

Step 2: Move the negations inward.

∀*x* (∃*y* Animal(y) ^ Kind(x,y)) v (∃*y* Loves(y,x))

Step 3: Standardize variables

∀x ∃y∃z (Animal(y) ^ ~Kind(x,y)) v (Loves(z,x))

Step 4: Skolemize

∀x ∃y∃z (Animal(F(x)) ^ ~Kind(x,F(x))) v (Loves(G(x),x))

Step 5: Drop universal quantifiers

(Animal(F(x)) ^ ~Kind(x,F(x))) v (Loves(G(x),x))

Step 6: Distribute v over ^

(Animal(F(x)) v (Loves(G(x),x)) ^ (~Kind(x,F(x)) v Loves(G(x),x))

1. Consider the following sentence:

( Animal(F(x)) v (Loves(G(x),x)) ^ (~Kind(x,F(x)) v Loves(G(x),x))

What are three different examples of sentences that could be added to this KB that would lead to a sound application of the resolution rule? Write the sentence and then the new sentence that can be inferred. Include at least one sentence with a disjunction, and at least one sentence with ground terms in it.

~Animal(Bob) Animal(F(x) v ~Kind(x,F(x))

-------------------------------- ------------------------------------------------------------

Loves(G(x),x) Animal(F(x)) v Loves(G(x), x) v ~Kind(x,F(x))

Kind(x,F(x))

--------------------------

Loves(G(x),x)

1. Explain in what ways (if any) you needed to use unification and substitution to apply the resolution rule in Question 2.

I used substitution for Animal(Bob). Bob can be substituted for F(x) which would imply Loves(G(x), x).

1. **Forward and Backward Chaining**

The following is pseudocode for forward chaining in FOL. Answer the following questions based on the pseudocode.



1. What leads the algorithm to return false?

If no new sentences are inferred on an iteration.

1. Explain how q’ is inferred, given the pseudocode.

Q’ is inferred by substituting theta for p.

1. Does the algorithm necessarily infer all possible new sentences, given the KB? If yes, explain how you know this. If no, explain how you would modify the algorithm to do this.

Yes because it loops through all rules in the KB and infers all potential new rules using substitution. It then does the same thing with the new rules.

1. **Planning**
2. Formulate the 8-puzzle problem as a planning problem, describing the initial state, goal state, and possible actions.

Initial State: no conditions for the initial state

Action(Move(p,x,y))

Precond: On(p,x) ^ Clear(y) ^ Adjacent(x,y)

Effect On(p,y) ^ Clear(x) ^ ~On(p,x) ^ ~Clear(y)

Goal state: On(Pi, i) for all pi

Each tile is on its correctly numbered position

1. Consider the following spare tire problem described in the textbook. If you are using forward planning, what are the actions that can be applied to the initial state to create a new state in your search tree.



Remove and LeaveOvernight because the initial state satisfies the preconditions for these actions. PutOn requires the flat to not be on the axle and the initial state has the flat on the axle

1. Discuss the following question from the quiz. What do you think the right answer is?

In backward STRIPS planning in the blocks world, what's an example of a precondition of an action not being consistent with a subgoal of G that is NOT an effect of the action. Assume the subgoal is On(B,C).

1. Precondition = Clear(B)
2. Precondition = On(B,C)
3. Precondition = Clear(C)
4. Precondition = ~On(B, C)
5. Imagine you are executing a backward planning algorithm in the following scenario. For each action, say whether you can apply it in backward planning to create a subgoal. If you can apply it, write the subgoal. If you cannot apply it, expalin why.

*Init*(Have(Cake) ^ NotEaten(Cake))

*Goal*(Have(Cake) ^ Eaten(Cake))

*Action*(Eat(x)

Precond: *Have*(x)

Effect: *~Have*(x) ^ *Eaten*(x))

*Action*(Bake(x)

Precond: *NotHave*(x)

Effect: *Have*(x)

1. Eat(x)
2. Bake(x)

**Optional**

1. Explain in your own words how this backwards chaining algorithm works. If it helps, you can reference the proof tree discussed in the video.



