PROJECT #3: FIRST-ORDER LOGIC & INTRO TO PLANNING (NO CODE!)

Please take care to complete all parts of this project independently. Do not use any solutions found online or written by other students. If you are struggling please go to office hours, post to Piazza, and/or send us email. Please submit a single PDF file `project3.pdf` to Canvas. The PDF file can contain a scan of handwritten or typewritten/typeset solutions, and any supporting graphics.

1. **R&N Problem 8.10:** (7 points)
   Consider a vocabulary with the following symbols:
   > *Occupation(p, o):* Predicate. Person *p* has occupation *o*.
   > *Customer (p1,p2)*: Predicate. Person *p1* is a customer of person *p2*.
   > *Boss(p1,p2):* Predicate. Person *p1* is a boss of person *p2*.
   > *Doctor , Surgeon, Lawyer , Actor:* Constants denoting occupations.
   > *Emily, Joe:* Constants denoting people.

   Use these symbols to write the following assertions in first-order logic:
   a. Emily is either a surgeon or a lawyer.
   b. Joe is an actor, but he also holds another job.
   c. All surgeons are doctors.
   d. Joe does not have a lawyer (i.e., is not a customer of any lawyer).
   e. Emily has a boss who is a lawyer.
   f. There exists a lawyer all of whose customers are doctors.
   g. Every surgeon has a lawyer.

2. **R&N Problem 8.20:** (6 points)
   Arithmetic assertions can be written in first-order logic with the predicate symbol <, the function symbols + and *, and the constant symbols 0 and 1. Additional predicates can also be defined with biconditionals.
   a. Represent the property "x is an even number."
   b. Represent the property "x is prime."
   c. Golbach's conjecture is the conjecture (unproven as yet) that every even number is equal to the sum of two primes. Represent this conjecture as a logical sentence.
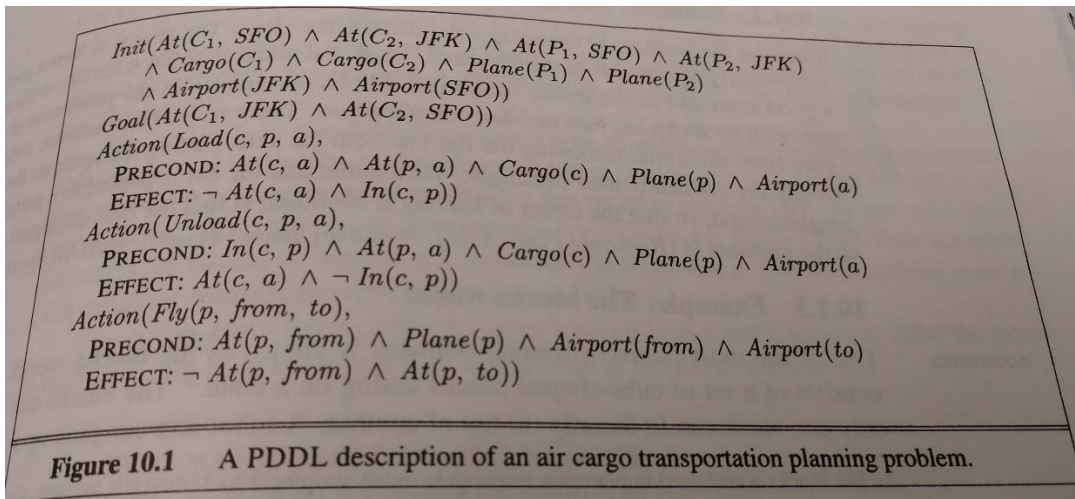
3. **R&N Problem 9.6:** (6 points)
   Write logical representations for the following sentences, suitable for use with Generalized Modus Ponens:
   a. Horses, cows, and pigs are mammals.
   b. An offspring of a horse is a horse.
   c. Bluebeard is a horse.
   d. Bluebeard is Charlie's parent.
   e. Offspring and parent are inverse relations.
   f. Every mammal has a parent.

4. **R&N Problem 9.13:** (10 points) In this exercise, use the sentences you wrote in Exercise 9.6 (Problem 4 of this assignment) to answer a question by using a backward-chaining algorithm.
   a. Draw the proof tree generated by an exhaustive backward-chaining algorithm for the query $\exists h\ Horse(h)$, where clauses are matched in the order given.
   b. What do you notice about this domain?
   c. How many solutions for *h* actually follow from your sentences?
   d. Can you think of a way to find all of them? (Hint: See Smith et al. (1986) – R&N bibliography.)

5. **R&N Problem 9.23:** (6 points) From "Horses are animals," it follows that "The head of a horse is the head of an animal." Demonstrate that this inference is valid by carrying out the following steps:
   a. Translate the premise and the conclusion into the language of first-order logic. Use three predicates: *HeadOf(h,x)* (meaning "*h* is the head of *x*"), *Horse(x)*, and *Animal(x).*

6. **Write the following statements in first-order logic. Complete all steps of FOL resolution to prove the conclusion:** (15 points)
   a. All hounds howl at night.
   b. Anyone who has any cats will not have any mice.
   c. Light sleepers do not have anything that howls at night.
   d. Sam has either a cat or a hound.
   e. (Conclusion) If Sam is a light sleeper, then Sam does not have any mice.

7. **Write the following statements in first-order logic. Complete all steps of FOL resolution to prove the conclusion:** (15 points)
   a. Everyone who feels warm either is drunk, or every costume they have is warm.
   b. Every costume that is warm is furry.
   c. Every AI student is a CS student.
   d. Every AI student has some robot costume.
   e. No robot costume is furry.
   f. (Conclusion) If every CS student feels warm, then every AI student is drunk.

8. **Write the following statements in first-order logic. Complete all steps of FOL resolution to prove the conclusion:** (15 points)
   a. Every child loves every candy.
   b. Anyone who loves some candy is not a nutrition fanatic.
   c. Anyone who eats any pumpkin is a nutrition fanatic.
   d. Anyone who buys any pumpkin either carves it or eats it.
   e. Stuart buys a pumpkin.
   f. Lifesavers is a candy.
   g. (Conclusion) If Stuart is a child, then Stuart carves some pumpkin.

9. **R&N 10.2 (Flying)** (5 points)
   Given the action schemas and initial state from R&N Figure 10.1 (reproduced below), what are all the applicable concrete instances of *Fly(p, from, to)* in the state described by:

   $$At(P1, JFK) \land At(P2, SFO) \land Plane(P1) \land Plane(P2) \land Airport(JFK) \land Airport(SFO)$$


   ```
   Init(At(C₁, SFO) ∧ At(C₂, JFK) ∧ At(P₁, SFO) ∧ At(P₂, JFK)
        ∧ Cargo(C₁) ∧ Cargo(C₂) ∧ Plane(P₁) ∧ Plane(P₂)
        ∧ Airport(JFK) ∧ Airport(SFO))
   Goal(At(C₁, JFK) ∧ At(C₂, SFO))
   Action(Load(c, p, a),
       PRECOND: At(c, a) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)
       EFFECT: ¬ At(c, a) ∧ In(c, p))
   Action(Unload(c, p, a),
       PRECOND: In(c, p) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)
       EFFECT: At(c, a) ∧ ¬ In(c, p))
   Action(Fly(p, from, to),
       PRECOND: At(p, from) ∧ Plane(p) ∧ Airport(from) ∧ Airport(to)
       EFFECT: ¬ At(p, from) ∧ At(p, to))
   ```

   Figure 10.1 A PDDL description of an air cargo transportation planning problem.

10. **R&N 10.4 (Shakey the Robot)** (15 points)

The original STRIPS planner was designed to control Shakey the robot. Figure 10.14 (from R&N, reproduced below) shows a version of Shakey's world consisting of four rooms lined up along a corridor, where each room has a door and a light switch. The actions in Shakey's world include moving from place to place, pushing movable objects (such as boxes), climbing onto and down from rigid objects (such as boxes), and turning light switches on and off. The robot itself could not climb on a box or toggle a switch, but the planner was capable of finding and printing out plans that were beyond the robot's abilities. Shakey's six actions are the following:

- *Go(x,y,r),* which requires that Shakey be *At x* and that *x* and *y* are locations *In* the same room *r*. By convention a door between two rooms is in both of the rooms.
- *Push(b,x,y,r):* Push a box *b* from location *x* to location *y* within the same room *r.* You need the predicate *Box* and constants for the boxes.
- *ClimbUp(x,b):* Climb onto a box *b* from position *x*; requires predicate *On(x,y)* (*x* is on *y*) and constant *Floor*.
- *ClimbDown(b,x):* Climb down from a box *b* to position *x*.
- *TurnOn(s,b), TurnOff(s,b):* Turn a light switch *s* on (or off). To successfully achieve a light switch action, Shakey must be on top of a box *b* at the light switch's location (room).

Write PDDL sentences for Shakey's six actions and the initial state from Figure 10.14. Manually construct a plan for Shakey to get *Box2* into *Room2*.

Figure 10.14   Shakey's world. Shakey can move between landmarks within a room, can pass through the door between rooms, can climb climbable objects and push pushable objects, and can flip light switches.