CMSC 421 Homework 3

Due Date: Tuesday, November 1 at the start of class

1. **[15 points] Resolution Theorem Proving**

   (a) Representing the following knowledge base in first-order logic. Use the predicates `attend(x)`, `fair(t)`, `pass(x,t)`, `prepared(x)`, `smart(x)`, `study(x)`, `umd-student(x)`, where arguments `x` have the domain of all students, and arguments `t` have the domain of all tests.

      i. Everyone who is smart, studies, and attends class will be prepared.
      ii. Everyone who is prepared will pass a test if it is fair.
      iii. Every UMD student is smart.
      iv. If a test isn't fair, no one will pass the test.
      v. Chris is a UMD student.
      vi. At least one student passed the 421-exam.
      vii. Chris attends class.

   (b) Convert this knowledge base into conjunctive normal form.

   (c) Prove that

   $$\text{study(Chris)} \Rightarrow \text{pass(Chris,421-exam)}$$

   using resolution refutation. Show your proof as a series of sentences to be added to the KB. Clearly show which sentences were resolved to produce each new sentence, and what the unifier was for each resolution step.

2. **[15 points] Reasoning about action and change**

   Computer Deliveries Inc. (CDI) is delivering two new machines to AV Williams. CDI uses a delivery truck to carry such purchases. The CDI driver is initially at the CDI shipping depot where the two new machines are waiting to be shipped. The first machine, *Machine1*, is already on the truck, but the second machine, *Machine2*, is not. The CDI driver has three actions available to him: loading an object onto the truck, driving the truck, and unloading an object from the truck. In this scenario, the driver needs only to load the second machine on the truck, and drive the truck to AV Williams.

   The scenario can be partially formalized using the following axioms:

   $\forall x, l.(Action(t) = Drive(x, l) \wedge Drivable(x)) \rightarrow At(x, l, t + 1)$
   $\forall x, y.(Action(t) = Load(x, y) \wedge \exists l\ (At(x, l, t) \wedge At(y, l, t))) \rightarrow On(x, y, t + 1)$
   $At(Truck, Depot, 0)$
   $On(Machine1, Truck, 0)$
   $At(Machine2, Depot, 0)$
   $\neg On(Machine2, Truck, 0)$
   $Action(0) = Load(Machine2, Truck)$
   $Action(1) = Drive(Truck, AVW)$
   $Drivable(Truck)$
   $\neg Drivable(Machine1)$
   $\neg Drivable(Machine2)$

In our intended interpretation, *At* represents the overall geographical location of any physical object (drivable or otherwise), where locations are domain entities such as *AVW*. Thus, objects have locations even if they are loaded on other objects, and they may change location even if they themselves do not drive.

(a) For each ground predicate determine whether it is necessarily True ("T"), necessarily False ("F"), or undetermined ("?") at $t = 0, 1, 2$. Your response should be a table, in the format below:

	$t = 0$	$t = 1$	$t = 2$
Action(t)	$Action(0) = Load(...)$	$Action(1) = Drive(...)$	
At(Truck, Depot, t)			
At(Truck, AVW, t)			
At(Machine1, Depot, t)			
At(Machine1, AVW, t)			
On(Machine1, Truck, t)			
At(Machine2, Depot, t)			
At(Machine2, AVW, t)			
On(Machine2, Truck, t)			

(b) Give enough general axioms to fully determine every entry in the table for $t = 0$ in the preferred way. Briefly explain every axiom. Make sure to write general axioms and not ground facts (i.e., answers such as $\neg At(Truck, AVW, 0)$ will not be accepted). [Hint: you will need two general axioms.]

(c) Consider a new predicate *Moving(x,l,t)*, which is true precisely when an object x (whether it drives autonomously or not) is moving from some location at time t to another location l, in which it will be at time $t + 1$. Write a formula that defines precisely when (i.e., an if-and-only-if logical formula) this predicate is true in terms of the original vocabulary.

(d) Write enough frame axioms such that together with the general axioms you wrote before there will be no ambiguity for $t = 1, 2$. Briefly explain every axiom. You must write your frame axioms in a general way such that adding irrelevant actions will not influence your axioms. For example, if we add the action *DrinkCoffee*, which does not change anything related to the machines or the truck, your frame axioms should remain exactly the same.

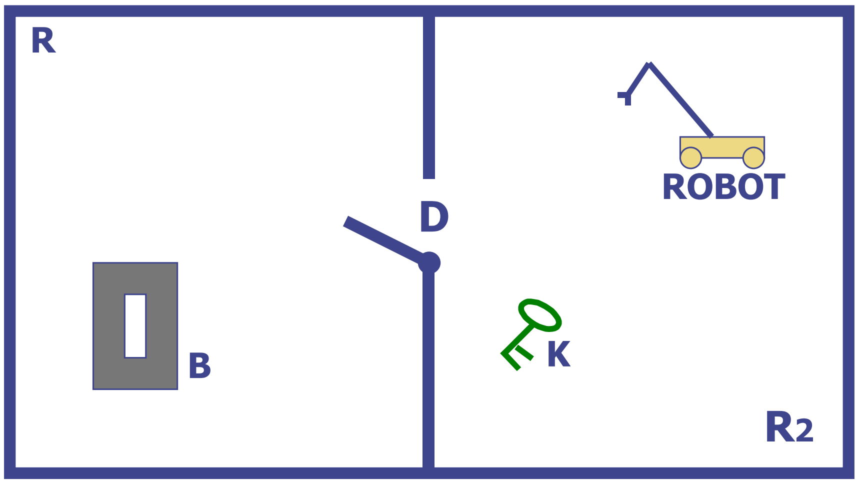
Hint: One of the axioms is *much* easier if you use the *Moving* predicate.

3. **[15 points] STRIPS** Russell & Norvig exercise 11.4.

4. **[15 points] POP**

A robot ROBOT operates in an environment made of two rooms R1 and R2 connected by a door D. A box B is located in R2 and the door's key is initially in R2. The door can be open or closed (and locked). the figure illustrates the initial state described by:

IN(ROBOT,R2)
IN(K,R2)
OPEN(D)



The actions are:

Grasp-Key-In-R2
Lock-Door
Go-From-R2-To-R1-With-Key
Put-Key-In-Box

defined as follows:

```
Grasp-Key-In-R2
P: IN(ROBOT,R2), IN(K,R2)
E: HOLDING(ROBOT,K)

Lock-Door
P: HOLDING(ROBOT,K), OPEN(D)
E: ~OPEN(D), LOCKED(D)

Go-From-R2-To-R1-With-Key
P: IN(ROBOT,R2), HOLDING(ROBOT,K), OPEN(D)
E: ~IN(ROBOT,R2), ~IN(K,R2), IN(ROBOT,R1), IN(K,R1)

Put-Key-In-Box
P: IN(ROBOT,R1), HOLDING(ROBOT,K)
E: ~HOLDING(ROBOT,K), ~IN(K,R1), IN(K,B)

The goal is:
IN(K,BOX), LOCKED(D)
```

Construct a partial order plan to solve this problem. **Clearly** indicate at each step the modifications made to the plan: the action added, the causal links added and/or the ordering constraints added. Indicate any threats at each step.

5. [**15 points**] [**GraphPlan**] Consider the following (trivial) planning problem. We have a plane in London (L) and we wish to fly it to Paris (P). The plane has a key that must be in the ignition in order to fly the plane. Initially we have the key in our possession, and we wish to have the key at the end of the plan. We have the following grounded operators:

operator	*preconditions*	*add*	*delete*
Fly(P)	*At(Plane, L)* *InIgnition(Key)*	*At(Plane, P)*	*At(Plane, L)*
Fly(L)	*At(Plane, P)* *InIgnition(Key)*	*At(Plane, L)*	*At(Plane, P)*
Insert(Key)	*Have(Key)*	*InIgnition(Key)*	*Have(Key)*
Remove(Key)	*InIgnition(Key)*	*Have(Key)*	*InIgnition(Key)*

The initial state is $At(Plane, L) \wedge Have(Key)$ and the goal state is $At(Car, P) \wedge Have(Key)$. Show how GraphPlan would solve this problem. You must show the propositions and actions at every time slice. For each time slice show the mutual exclusions. For the actions show which mutual exclusions are implied directly from the definition of the operators and which were propagated by GraphPlan.

6. [**25 points**] **Design Your Own Planning Problem** Define your own planning problem.

   (a) Define at least 4-5 operators.

   (b) Define an initial state and a goal state.

   (c) Come up with a plan that achieves the goal state from the initial state.

   Your planning problem should be non-trivial in that there may be interactions among subgoals, or alternate solutions, etc.