CMSC 671 Homework 3 — Fall 2010

Due Date: Monday, November 1, 2010

1 Resolution Theorem Proving (30 points)

1.1 Knowledge Base (8 points)

Represent the following knowledge base in first-order logic. Use the predicates:

- attend(x)
- fail(x,t)
- fair(t)
- pass(x,t)
- prepared(x)
- smart(x)
- study(x)
- umbc-student(x)

where arguments x implicitly have the domain of all people, and arguments t have the domain of all tests. (That is, you don't need to include predicates like person(x).)

1. Everyone who is smart, studies, and attends class will be prepared.
2. Everyone who is prepared will pass a test if it is fair.
3. A student passes a test if and only if they don't fail it.
4. Every UMBC student is smart.
5. If a test isn't fair, everyone will fail the test.
6. Aidan is a UMBC student.
7. Sandy passed the 471 exam.
8. Aidan attends class.

1.2 Conjunctive Normal Form (8 points)

Convert the KB to conjunctive normal form.

1.3 Negated Goal (2 points)

We wish to prove that:

study(Aidan) → pass(Aidan, 471-exam)

Express the negation of this goal in conjunctive normal form.

1.4 Proof (12 points)

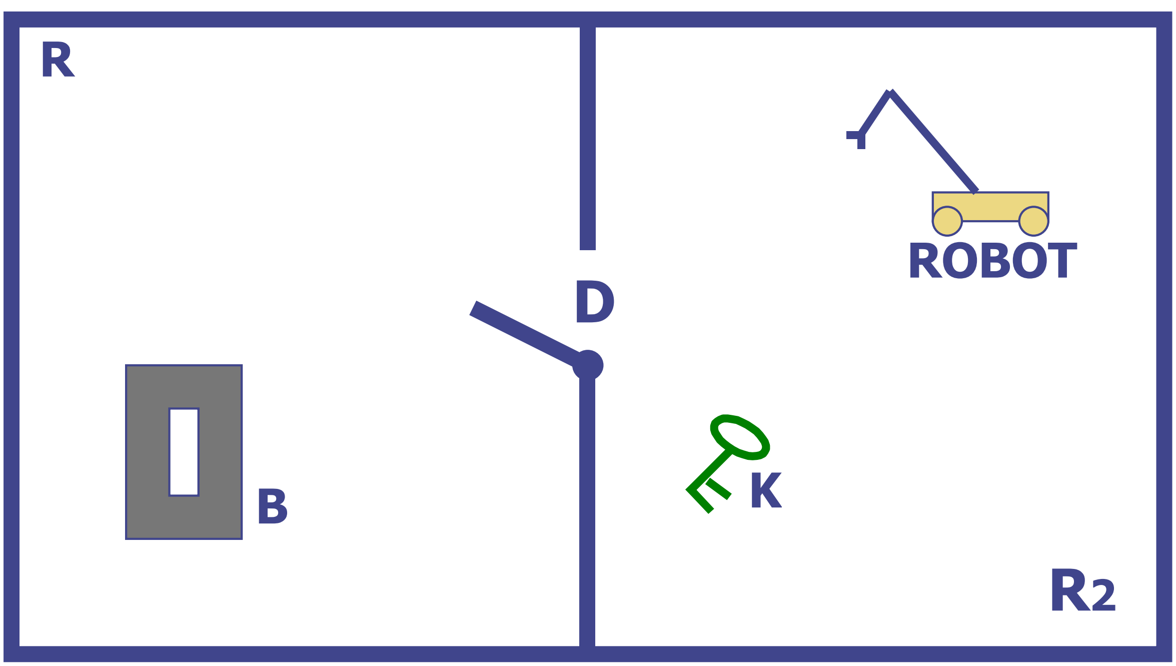
Add the negated goal to the KB, and use resolution refutation to prove that it is true. You may show your proof as a series of sentences to be added to the KB or as a proof tree. In either case, you must clearly show which sentences are resolved to produce each new sentence, and what the unifier is for each resolution step.

2 Partial-Order Planning (35 points)

Adapted from a problem provided by Dr. Lise Getoor.

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, which is 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

and are 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)
```

Show the process that a partial-order planner would follow 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.

3 GraphPlan (35 points)

Consider the following (trivial) planning problem. We have a car in London (L) and we wish to drive it to Paris (P). The car has a key that must be in the ignition in order to drive the car. 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*
Drive(P)	*At(Car, L)* *InIgnition(Key)*	*At(Car, P)*	*At(Car, L)*
Drive(L)	*At(Car, P)* *InIgnition(Key)*	*At(Car, L)*	*At(Car, P)*
Insert(Key)	*Have(Key)*	*InIgnition(Key)*	*Have(Key)*
Remove(Key)	*InIgnition(Key)*	*Have(Key)*	*InIgnition(Key)*

The initial state is $At(Car, L) \wedge Have(Key)$ 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.