

CMSC 421 Homework 3

Due Date: Wednesday, November 5 before 5PM

1. [5 points] Unification

For each of the following pairs of expressions, give the most general unifier (if it exists) or say that the expressions are not unifiable. The lower case letters f and g denote function constants, all other lower case letters denote variables, and upper case letters (except for P , which is a predicate) denote constants.

- (a) $P(x, B, B)$ and $P(A, y, z)$
- (b) $P(g(f(v)), g(u))$ and $P(x, x)$
- (c) $P(x, f(x))$ and $P(y, y)$
- (d) $P(y, y, B)$ and $P(z, x, z)$
- (e) $P(y, g(A, B))$ and $P(g(x, x), y)$

2. [20 points] Resolution Theorem Proving

(Adapted from Problem #5 at <http://www.classroomtools.com/logic1.htm>.)

There are three people in a room: Alice, Bob, and Chris. Each person is either an angel (always tells the truth) or a devil (always lies). Alice says, "All of us are devils." Bob says, "Exactly one of us is an angel."

- (a) Encode this knowledge base in first-order logic. You should only use the predicate $angel(x)$. A devil is just someone who isn't an angel. Statements by an individual can be conditioned on whether (or not) the individual is an angel, since the statement will be true if and only if the speaker is an angel. For example, if Chris said "I am a devil," it could be encoded as:

$$angel(Chris) \Leftrightarrow \neg angel(Chris)$$

which is a contradiction!

Hint: Bob's statement has many different correct encodings. Encoding it as two implications yields a simpler conversion to CNF. If Bob is an angel, what can you infer directly from his statement about Alice and Bob? (How many angels must there be?) If Bob is not an angel, and is lying, then what can you infer about the possible number of angels? How can you represent this information using only the angel predicate?

- (b) Convert the KB to conjunctive normal form.
- (c) Which sentences in the resulting (CNF) KB are subsumed by other sentences in the KB? You should remove these subsumed sentences from the KB before completing question (d).
- (d) Use resolution refutation to prove that the solution you found in a is correct.

3. [20 points] Reasoning about action and change

Frosty the Red-Nosed Reindeer is delivering two new machines (yay!) to AV Williams. (We actually asked for 10, but Frosty had other financial obligations.) Frosty uses a

magical flying sleigh to carry such gifts. Frosty is initially at the North Pole where the two new machines are waiting to be shipped. The first machine, *Machine1*, is already on the sleigh, but the second machine, *Machine2*, is not. Frosty has three actions available to him: loading an object onto the sleigh, flying the sleigh, and unloading an object from the sleigh. In this scenario, Frosty needs only to load the second machine on the sleigh, and fly the sleigh to AV Williams.

The scenario can be partially formalized using the following axioms:

$$\begin{aligned} &\forall x, l. (Action(t) = Fly(x, l) \wedge Flies(x)) \rightarrow At(x, l, t + 1) \\ &\forall x, y. (Action(t) = Load(x, y) \wedge \exists l. (At(x, l, t) \wedge At(y, l, t))) \rightarrow On(x, y, t + 1) \\ &At(Sleigh, NorthPole, 0) \\ &On(Machine1, Sleigh, 0) \\ &At(Machine2, NorthPole, 0) \\ &\neg On(Machine2, Sleigh, 0) \\ &Action(0) = Load(Machine2, Sleigh) \\ &Action(1) = Fly(Sleigh, AVW) \\ &Flies(Sleigh) \\ &\neg Flies(Machine1) \\ &\neg Flies(Machine2) \end{aligned}$$

In our intended interpretation, *At* represents the overall geographical location of any physical object (flying 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 fly.

- (a) **[3 points]** 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)	$Action(0) = Load(\dots)$	$Action(1) = Fly(\dots)$	
$At(Sleigh, NorthPole, t)$			
$At(Sleigh, AVW, t)$			
$At(Machine1, NorthPole, t)$			
$At(Machine1, AVW, t)$			
$On(Machine1, Sleigh, t)$			
$At(Machine2, NorthPole, t)$			
$At(Machine2, AVW, t)$			
$On(Machine2, Sleigh, t)$			

- (b) **[6 points]** 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 $\neg At(Sleigh, AVW, 0)$ will not be accepted). [Hint: you will need two general axioms.]

- (c) **[3 points]** Consider a new predicate $Moving(x, l, t)$, which is true precisely when an object x (whether it flies 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) **[8 points]** 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 *DrinkEggNog*, which does not change anything related to the machines or the sleigh, your frame axioms should remain exactly the same.

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

4. **[10 points] STRIPS planning** (from Nilsson, *AI: A New Synthesis*) Consider the problem of devising a plan for a kitchen-cleaning robot.

Write a set of STRIPS-style operators that might be used. When you describe the operators, take into account the following considerations:

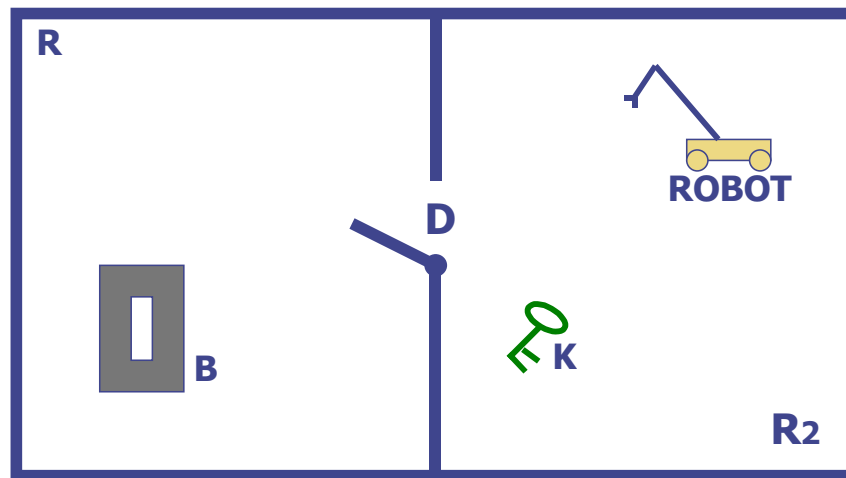
- i. Cleaning the stove or the refrigerator will get the floor dirty.
 - ii. The stove must be clean before covering the drip pans with tin foil.
 - iii. Cleaning the refrigerator generates garbage and messes up the counters
 - iv. Washing the counters or the floor gets the sink dirty
- (a) Write a description of an initial state of a kitchen that has a dirty stove, refrigerator, counters and floor. (The sink is clean, and the garbage has been taken out.) Also write a description of the goal state where everything is clean, there is no trash and the stove drip pans have been covered with tin foil.

5. **[15 points] STRIPS** Russell & Norvig exercise 11.4.

6. **[20 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.

7. [10 points] [GraphPlan] Consider the following (trivial) planning problem. We have a car at 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:

<i>operator</i>	<i>preconditions</i>	<i>add</i>	<i>delete</i>
<i>Drive(P)</i>	<i>At(Car, L)</i> <i>InIgnition(Key)</i>	<i>At(Car, P)</i>	<i>At(Car, L)</i>
<i>Drive(L)</i>	<i>At(Car, P)</i> <i>InIgnition(Key)</i>	<i>At(Car, L)</i>	<i>At(Car, P)</i>
<i>Insert(Key)</i>	<i>Have(Key)</i>	<i>InIgnition(Key)</i>	<i>Have(Key)</i>
<i>Remove(Key)</i>	<i>InIgnition(Key)</i>	<i>Have(Key)</i>	<i>InIgnition(Key)</i>

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.