CS188
Fall 2000
Solutions to Midterm

Problem 1

a) Two answers accepted: (The problem was more ambiguous than we intended.)

TRUE – without local maxima, hillclimbing will converge on the global maximum solution.

FALSE – it may still encounter a plateau or ridge that may impede it from finding the global maximum.

b) FALSE – IDS and BFS run in asymptotically the same time.

c) FALSE – Resolution is complete for FOPC. –1 point if resolution was not specifically mentioned.

d) FALSE – depends on the definitions of Mother, Parent, and Female in the knowledge base. –1 point if the fact that instantiating the predicated with the smaller number of objects that make it true in the KB (NOT necessarily the number of parameters) will result in more efficient backward chaining.

e) TRUE – without the upward solution property, there may exist a primitive plan that solves the problem that does not have a corresponding abstract plan, so an abstract planner may fail to find a solution.

f) FALSE – heuristic is not admissible because it overestimates the diagonal distance. –2 points if no mention of overestimation is present. Just because the moves are diagonal and the heuristic only uses horizontal and vertical moves does not affect admissibility.

g) TRUE – Resolution can proved any valid sentence that is entailed by a knowledge base. It can not prove that a sentence is satisfiable under some set of models but not all.

Also given full credit: FALSE – resolution cannot handle equality so there are some sentences it cannot prove.

Correct TRUE or FALSE answers without explanations were given one point. Incorrect TRUE or FALSE answers were given partial credit (1-2 points) if the explanation indicated that you understood the problem fully but didn’t put the correct T/F (there were very few cases like this).

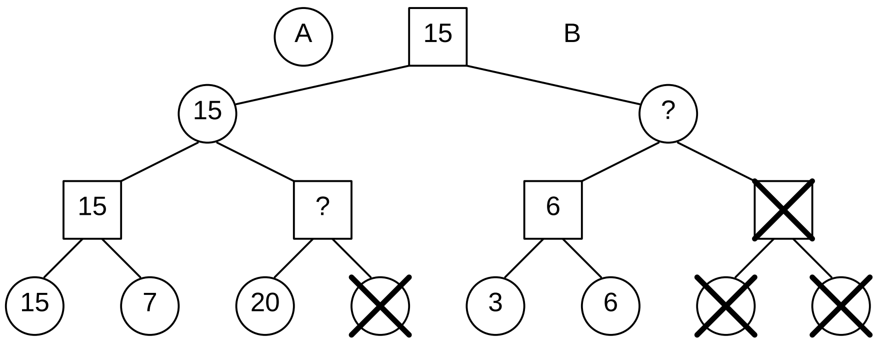
Problem 2

A1) A B C D E F G H I J K – very little partial credit given

A2) A B C D E F J – we must be sure to hillclimb using g() + h(). One point given for either: A B C G H L (hillclimbing just on h()) or A (g()+h() is at it's minimum at the root, so we don't move anywhere). Note that in this problem, hillclimbing does not find a goal.

A3) A B C D E F G H J K M L. –2 if you stopped after K. Unlike in part A1, we A* search can't stop until is visits a goal node, not when it generates it.

B)



Scoring: 1 point for correctly labeling the values in the tree
1 point for each properly placed "x"
1 point for each properly placed "?"
1 point for choosing the correct move

Problem 3. Representation and Inference

A) The difference is that Woman(w) appears in the conjunction in the second sentence but in the conclusion part of the implication. Woman(w) has to be true in the second sentence; but it doesn't have to in the first sentence, if the condition Man(m) is false. Hence in a model where there is no man and no woman, the first sentence is true because Man(m) is always false; the second sentence is false because no w could satisfy Woman(w).

B)

$\exists a\ \forall d$ [Apple(a) $\wedge$ Computer(a) $\wedge$ (Dell(d) $\wedge$ Computer(d) $\Rightarrow$ Greater(Speed(a), Speed(d))]

C)

The goal is: ∃c Computer(c) ∧ (Dell(GX110) ∧ Computer(GX110) ⇒ Greater(Speed(c), Speed(GX110)))

Negation of the goal: ∀c ¬Computer(c) ∨ (Dell(GX110) ∧ Computer(GX110) ∧ ¬Greater(Speed(c), Speed(GX110)))

1) {¬Computer(c),Dell(GX110)}
2) {¬Computer(c),Computer(GX110)}
3) {¬Computer(c), ¬Greater(Speed(c), Speed(GX110))}

From B)

4) {Apple(A0)}
5) {Computer(A0)}
6) {¬Dell(d), ¬Computer(d), Greater(Speed(A0), Speed(d))}

Resolution:

7) {¬Dell(GX110), ¬Computer(GX110), ¬Computer(A0)} 3) and 6) { c/A0, d/GX110 }
8) {¬Computer(GX110), ¬Computer(A0)} 1) and 7)
9) {¬Computer(A0)} 2) and 8)
10) {} 4) and 9)

Problem 4

A) Three possible answers:

Solution1: combined successor state axioms:

$$\begin{aligned}
&\forall a,s,m,n,k\,(\,\mathrm{At}(\mathrm{Robot},n,\mathrm{Result}(a,s)) \wedge \mathrm{At}(k,n,\mathrm{Result}(a,s)))\\
&\qquad \leftrightarrow ((\mathrm{At}(\mathrm{Robot},m,s) \wedge \mathrm{At}(k,m,s) \wedge (a = \mathrm{Push}(k,m,n)))\\
&\qquad\quad \vee (\mathrm{At}(\mathrm{Robot},m,s) \wedge \mathrm{At}(k,n,s) \wedge (a = \mathrm{Move}(m,n)))\\
&\qquad\quad \vee (\mathrm{At}(\mathrm{Robot},n,s) \wedge \mathrm{At}(k,n,s) \wedge (a \neq \mathrm{Push}(k,n,m)) \wedge (a \neq \mathrm{Move}(n,m))))
\end{aligned}$$

Solution 2: Separate successor state axioms:

$$\begin{aligned}
&\forall a,s,m,n,k\ \mathrm{At}(\mathrm{Robot},n,\mathrm{Result}(a,s))\\
&\qquad\qquad \leftrightarrow ((\mathrm{At}(\mathrm{Robot},m,s)\wedge \mathrm{At}(k,m,s)\wedge (a=\mathrm{Push}(k,m,n)))\\
&\qquad\qquad\quad \vee (\mathrm{At}(\mathrm{Robot},m,s)\wedge (a=\mathrm{Move}(m,n)))\\
&\qquad\qquad\quad \vee (\mathrm{At}(\mathrm{Robot},n,s)\wedge (a\neq \mathrm{Move}(n,m))\wedge (a\neq \mathrm{Push}(k,n,m))))\\
&\forall a,s,m,n,k\ \mathrm{At}(k,n,\mathrm{Result}(a,s))\\
&\qquad\qquad \leftrightarrow ((\mathrm{At}(\mathrm{Robot},m,s)\wedge \mathrm{At}(k,m,s)\wedge (a=\mathrm{Push}(k,m,n)))\\
&\qquad\qquad\quad \vee (\mathrm{At}(k,n,s)\wedge (a\neq \mathrm{Push}(k,n,m))))
\end{aligned}$$

Solution 3: effect axioms and frame axioms:

$$\begin{aligned}
&\forall a,s,m,n,k\ \mathrm{At}(\mathrm{Robot},m,s)\wedge \mathrm{At}(k,m,s)\\
&\qquad \rightarrow (\mathrm{At}(\mathrm{Robot},n,\mathrm{Result}(\mathrm{Push}(k,m,n),s))\\
&\qquad\quad \wedge \mathrm{At}(k,n,\mathrm{Result}(\mathrm{Push}(k,m,n),s)))\\
&\forall a,s,m,n,k\ \mathrm{At}(\mathrm{Robot},n,s)\wedge \mathrm{At}(k,n,s)\wedge (a\neq \mathrm{Push}(k,n,m))\wedge (a\neq \mathrm{Move}(n,m))\\
&\qquad \rightarrow \mathrm{At}(\mathrm{Robot},n,\mathrm{Result}(a,s))\wedge \mathrm{At}(k,n,\mathrm{Result}(a,s))
\end{aligned}$$

Any of these three approaches could earn full credit. Points were given on the basis of how close we thought you were to the solution. A few things definitely cost points: -2 points if you missed the fact that Move and Push are intimately intertwined, -2 points if you didn't add the situation argument to your predicates, -1 point if you didn't properly use Result (or Do or whatever you want to call it) to transition from one situation to the next, -2 to –4 points for missing clauses. We only gave a small bit of partial credit if most of the axioms were missing or wrong, or if you used STRIPS instead of situation calculus.

B) The foolish thing that is done is that once we choose the action Move-to-table with from=B in order to get B clear, the planner is free to put any old rot in obj and to variables. For example, it may set obj=table or to=A, which would cause the preconditions to go unsatisfied.

The solution is to add some sort of constraints on the variables, such as when/needs lists or even just simple type constraints.

-1 to -3 points if your explanation was missing or incorrect, depending on how off it was. –1 to –2 points if your solution was missing or incorrect.

We also gave full or mostly full credit if you observed that the fact Clear(A) is not in the initial KB and therefore the planner could essentially do nothing. The solution, of course, is to add Clear(A) to the KB.

Problem 5. Partial-Order Planning

A)

Action: Marry-Daughter(h, l, b)
    Precondition: Father(l,b), Single(h), Single(b)
    Add: Son-in-law(h,l), Married(h,b)
    Delete: Single(h), Single(b)

Action: Assassinate-Father-in-law(h, l)
    Precondition: King(l), Son-in-law(h,l)
    Add: King(h)
    Delete: King(l)

Action: Divorce-Daughter(h,l,b)
    Precondition: Son-in-law(h,l), Married(h, b)
    Add: Single(h), Single(b)
    Delete: Son-in-law(h,l), Married(h,b)

B)

Yes, there is a conflict between Assassinate-Father-in-law and Divorce-Daughter. Son-in-law(h,l) is a precondition of Assassinate-Father-in-law but Divorce-Daughter removes it. This conflict can be resolved by applying promotion on Divorce-Daughter, such that it has to be executed after Assassinate-Father-in-law in the plan.

C)

If we use a general operator of Marry instead of Marry-Daughter, in order to make the planner work in this problem, we would need inference rules that relates Son-in-law to Father and Marry. Son-in-law needs to be updated every time the relations Father and Marry change. This is a truth-maintenance problem in general. In particular, planning becomes more difficult because at each step, instead of matching the preconditions directly to the facts, the planner is now required to do inference to close the gap. For example, it needs to infer from the rules above that Married(h,b) and Father(l,b) imply Son-in-law(h,l), so a Marry-Daughter action can achieve Son-in-law in the plan. Inference is hard so this would make the planner much slower than it is currently.