CSE 573 FALL 1999 FINAL EXAM *SOLUTIONS*

Question 1

(8 + 4 bonus points)

Recall that the heuristic function in best-first search is $f(n) = g(n) + h(n)$, where $g(n)$ is the exact cost of getting to the current node n, and $h(n)$ is the estimated minimum cost of getting from n to a goal state.

a (4 points) Suppose we run a greedy search algorithm with $h(n) = -g(n)$. What sort of search will the greedy search emulate?

The best nodes (those with the lowest scores), will be those with the longest paths, so this emulates depth first search. A common mistake was to confuse greedy search with best first, or A*. Greedy search uses $f(n) = h(n)$. Since in this case $h(n) = -g(n)$, the deeper a node is in the tree, the better it's f-cost will be.

A common mistake was to use $f(n) = g(n) + h(n)$, which is the formula for best-first, but not greedy search. In this case, $f(n)$ becomes 0, so the search is effectively random (actually, it depends on the details of the queueing function.)

b (4 points) Prove that if the heuristic function h obeys the triangle inequality, then the f-cost along any path in the search tree is nondecreasing. (The triangle inequality says that the sum of the costs from A to B and B to C must not be less than the cost from A to C directly.)

Think of the triangle inequality as meaning that the direct route from A to C is faster than the indirect route through B.

Nondecreasing f-cost along a path means that f of a successor is always at least as large as that of the node:

$f(n) \leq f(n') \text{if} n' \in S(n)$

Substituting $f(n) = g(n) + h(n)$ we get:

$g(n) + h(n) \leq g(n') + h(n') \text{if} n' \in S(n)$

Our goal is to show that this is implied by the triangle inequality. The triangle inequality applied to a heuristic $h(n)$ says that

$h(n) \leq k(n, n') + h(n')$

for any nodes n, n', where $k(n, n')$ is the cost of the shortest path from n to n'. Adding $g(n)$ to both sides we get

$g(n) + h(n) \leq g(n) + k(n, n') + h(n')$

But if n' is a successor of n, then $g(n) + k(n, n')$ is equal to $g(n')$. So

$g(n) + h(n) \leq g(n') + h(n')$

c (BONUS 4 points) Sometimes there is no good evaluation function for a problem, but there is a good comparison method: a way to tell if one node is better than another, without assigning numerical values to either. Show that this is enough to do a best-first search. What properties of best-first search do we give up if we only have a comparison method?

Assuming the comparison function is a total ordering, we can still do best-first search by sorting the queue using the comparison function. However, since we don't have information about how much better a given node is than another, we can't combine the results of the comparison function with other information (such as a $g(n)$ function), so we can't do A*. We also give up the optimality guarantees that come with A* search. (We can be do no better than greedy search.) Also like greedy search, we loose completeness.

Question 2

(8 + 4 bonus points)

Consider the following map coloring problem:

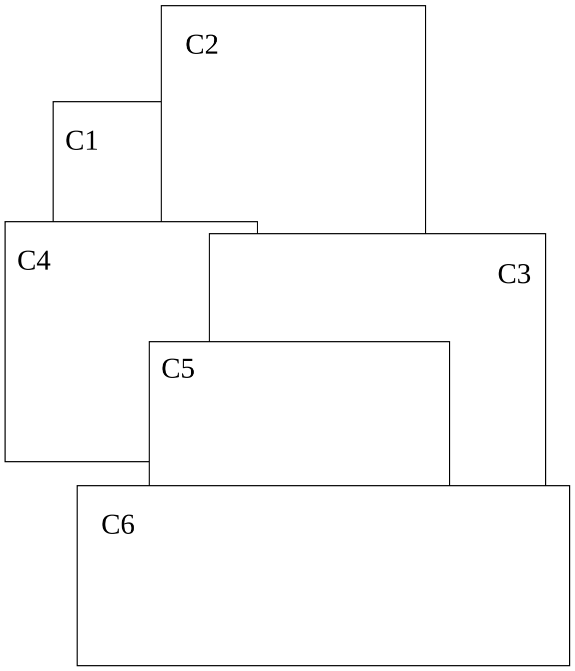


Figure 1: Assign each region one of the three colors (Red, Green or Blue) so that no two adjacent regions have the same colors.

a (8 points) In class, we studied two heuristics for CSPs, *least-constraining-value* and *most-constrained-variable*. Solve the graph coloring problem above using these two heuristics and forward checking. (Show work on back)

b (BONUS 4 points) Describe an additional heuristic that would be useful in solving this problem.

If there is a tie for most-constrained-variable, as in the first choice, use the *most-constraining-variable*, that is, the variable whose assignment will add the most new constraints. For example, using this heuristic it makes sense to select C4 first, since it limits the choices for all other variables except C6. Several people suggested using MOMS, this was OK, if you discuss converting the problem to SAT.

C4 = Red	Red	Green	Blue
C1	X		
C2	X		
C3	X		
C4	$\surd$		
C5	X		
C6			

C5 = Green	Red	Green	Blue
C1	X		
C2	X		
C3	X	X	
C4	$\surd$	X	
C5	X	$\surd$	
C6		X	

C3 = Blue	Red	Green	Blue
C1	X		
C2	X		X
C3	X	X	$\surd$
C4	$\surd$	X	X
C5	X	$\surd$	X
C6		X	X

C2 = Green	Red	Green	Blue
C1	X	X	
C2	X	$\surd$	X
C3	X	X	$\surd$
C4	$\surd$	X	X
C5	X	$\surd$	X
C6		X	X

C1 = Blue	Red	Green	Blue
C1	X	X	$\surd$
C2	X	$\surd$	X
C3	X	X	$\surd$
C4	$\surd$	X	X
C5	X	$\surd$	X
C6		X	X

C6 = Red	Red	Green	Blue
C1	X	X	$\surd$
C2	X	$\surd$	X
C3	X	X	$\surd$
C4	$\surd$	X	X
C5	X	$\surd$	X
C6	$\surd$	X	X

Table 1: CSP solution

Question 3

(6 points)

Let us consider the problem of search in a *three-player* game. (You can assume no alliances are allowed.) We will call the players 0, 1, and 2 for convenience. Assume you have an evaluation function that returns a list of three values, indicating (say) the likelihood of winning for players 0, 1 and 2, respectively. Complete the following game tree by filling in the backed-up values for all remaining nodes including the root.

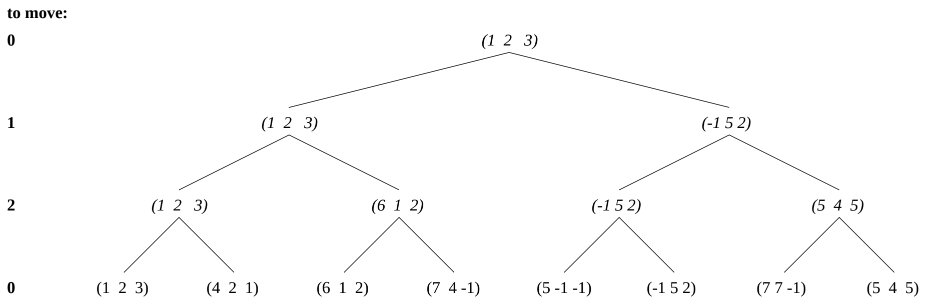


Figure 2: The first three ply of a game tree with three players (0, 1, and 2).

The important thing to understand is that each player will act so as to maximize their score from the choices presented to them, and the the score vectors are moved up the tree as units (no mixing) because they represent the value to each player of a particular game.

Question 4

(4 points)

Given the following, can you prove that the unicorn is mythical? How about magical? Horned?

> If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned.

First a proof by counterexample that you can't prove that the unicorn is mythical:

It is consistent with the above sentences for the unicorn to be a non-mythical, magical, mortal, horned mammal.

Now for the tedious version, let us define the following propositions:

$MYTHICAL$**: The unicorn is mythical.**
$MORTAL$**: The unicorn is mortal (Which means that** $\neg MORTAL$ **translates as "The unicorn is immortal.")**
$MAMMAL$**: The unicorn is a mammal.**
$HORNED$**: The unicorn is horned.**
$MAGICAL$**: The unicorn is magical.**

We can now translate the statements above as:

$MYTHICAL \rightarrow \neg MORTAL$
$\neg MYTHICAL \rightarrow MORTAL \wedge MAMMAL$
$(\neg MORTAL \vee MAMMAL) \rightarrow HORNED$
$HORNED \rightarrow MAGICAL$
First, we'll look at the last two questions, which are easy. Since either $MYTHICAL$ **or** $\neg MYTHICAL$ **must be true, either** $\neg MORTAL$ **or** $MORTAL \wedge MAMMAL$ **must be true, which means that** $(\neg MORTAL \vee MAMMAL)$ **is true. By the third premise, this means that** $HORNED$ **is true, and by the fourth premise,** $MAGICAL$ **is also true. So we can prove** $MAGICAL$ **and** $HORNED$**.**

To prove $MYTHICAL$**, we would have to to show that** $MORTAL \wedge MAMMAL$ **is false. We could show this by denying either conjunct. If we wanted to show** $MORTAL$ **was false, we'd need to prove** $MYTHICAL$ **and use premise 1, but that would involve a circularity. If we try to deny** $MAMMAL$**, the only way to do it would be to deny** $HORNED$**. The only rule which could achieve that is the last, and we would have to show** $\neg MAGICAL$**. Since there is no way to show that, this also**

fails. This exhausts the possibilities for proving $MYTHICAL$, so it is unprovable from this set of sentences.

Question 5

(6 points)

Consider a world in which there are only four propositions, A, B, C, and D. How many models are there for the following sentences?

a (2 points) $A \wedge B$

4 - A and B must be true, but C and D are unconstrained

A	B	C	D
T	T	T	T
T	T	T	F
T	T	F	T
T	T	F	F

Table 2: Models for $A \wedge B$

b (2 points) $A \vee B$

12 - There are three ways to satisfy $A \vee B$ and for each one, four models corresponding to all possible truth values for C and D.

c (2 points) $A \wedge B \wedge C$

2 - One where D is true and one where D is false.

Question 6

(10 + 2 bonus points)

Here are two sentences in the language of first-order logic:

(A) $\forall x \exists y (x \geq y)$

(B) $\exists y \forall x (x \geq y)$

a (2 point) Assume that the variables range over all natural numbers $0, 1, 2, \ldots, \infty$, and that the "$\geq$" predicate means "greater than or equal to." Under this interpretation, translate the these sentences into English.

The first sentence translates as "For every natural number, there is some (other) natural number that it is greater than or equal to." The second, as "There is a specific natural number that is less than than or equal to every natural number."

b (1 point) Is (A) true under this interpretation?

Yes - for any natural number, you can pick itself as the "other" number.

c (1 points) Is (B) true under this interpretation?

Yes - The number 0 has this property.

d (2 points) Does (A) logically entail (B)?

No, (A) does not logically entail (B). (Counterexample: Consider the integers, A is true, but B is not.)

e (2 points) Does (B) logically entail (A)?

Yes, (B) logically entails (A)

f (2 points) Try to prove that (A) follows from (B) using resolution. Do this even if you think that (B) does not logically entail (A); continue until the proof breaks down and you cannot proceed (if it does break down). Show the unifying substitution for each resolution step. If the proof fails, explain exactly where, how and why it breaks down.

We set the knowledge base to the negation of (A) and (B). Again, we convert both sentences to canonical form (which requires introducing a Skolem constant for (A) and a Skolem function for (B)):

($\neg$A): $\neg(F_1 \geq y)$
(B): $x \geq F_2$
Resolving these clauses we use the substitution: $\{x/F_1, y/F_2\}$**.**
This gives us:

$\neg(F_1 \geq F_2)$ **and** $F_1 \geq F_2$

Which does resolve, giving us *False* and proving that (B) entails (A)

g (BONUS 2 points) Now try to prove that (B) follows from (A).

We set the knowledge base to (A) and the negation of (B). First we convert both sentences to canonical form (which requires introducing Skolem functions):

(A): $x \geq F_1(x)$
(¬B): $\neg F_2(y) \geq y$
Now we try to derive a contradiction. There are only two clauses, so we try to unify them. The obvious unification would be: $\{x/F_2(y), y/F_1(x)\}$**, but this is equivalent to** $\{x/F_2(y), y/F_1(F_2(y))\}$**, which fails because an expression containing** y **is being substituted for** y**. The resolution fails, and there are no other clauses or unifications to try, so the proof fails.**

Question 7

(15 + 5 bonus points)

Two astronomers, in different parts of the world, make measurements M_1 and M_2 of the number of stars N in some small region of the sky, using their telescopes. Normally, there is a small possibility of error by up to one star. Each telescope can also (with a slightly smaller probability) be badly out of focus (events F_1 and F_2), in which case, the scientist will undercount by three or more stars. Consider the three networks shown below.

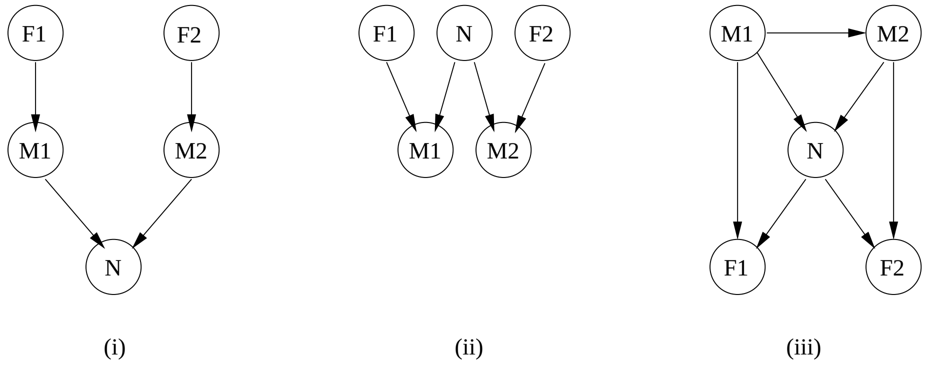


Figure 3: Three possible networks for the telescope problem.

a (5 points) Which of these belief networks correctly (but not necessarily efficiently) represent the above information?

Althrough (i) in some sense depicts the "flow of information" during calculation, it is clearly incorrect as a network, since it says that given the measurements M_1 and M_2, the number of stars is independent of the focus. (ii) correctly represents the causal structure: each measurement is influenced by the actual number of stars and the focus, and the two telescopes are independent of each other. (iii) shows a correct but more complicated network — the one obtained by ordering the nodes M_1, M_2, N, F_1, F_2. If you order M_2 before M_1 you would get the same network except with the arrow from M_1 to M_2 reversed.

Many people confused the notion of correctly representing the causal relationship with correctly representing the conditional independence relation.

b (5 points) Which is the best network?

(ii) requires fewer parameters and is therefore better than (iii).

(Many people wrote that (ii) is better because it is "more causally accurate." While this is true, it's only part of the picture.

c (5 points) Give a reasonable conditional probability table for the values of $\mathbf{P}(M_1|N)$. (For simplicity, consider only the possible values 1, 2, and 3 in this part.)

To compute $\mathbf{P}(M_1|N)$, we will need to condition on F_1 (that is, consider both possible cases for F_1, weighted by their probabilities.)

$\mathbf{P}(M_1|N) = \mathbf{P}(M_1|N, F_1)\mathbf{P}(F_1|N) + \mathbf{P}(M_1|N, \neg F_1)\mathbf{P}(\neg F_1|N)$
$\mathbf{P}(M_1|N) = \mathbf{P}(M_1|N, F_1)\mathbf{P}(F_1) + \mathbf{P}(M_1|N, \neg F_1)\mathbf{P}(\neg F_1)$
Let f be the probability that the telescope is out of focus. The problem states that this will cause an "undercount of three or more starts." For N=3 or less stars, we assume this means the count will be 0 if the telescope is out of focus. If it is in focus, then we will assume there is a probability of e of counting one too few, and e of counting one too many. The rest of the time $(1-2e)$, the count will be accurate. Then the table is as follows:

	$N=1$	$N=2$	$N=3$
$M_1=0$	$f+e(1-f)$	f	f
$M_1=1$	$(1-2e)(1-f)$	$e(1-f)$	0.0
$M_1=2$	$e(1-f)$	$(1-2e)(1-f)$	$e(1-f)$
$M_1=3$	0.0	$e(1-f)$	$(1-2e)(1-f)$
$M_1=4$	0.0	0.0	$e(1-f)$

Table 3: Conditional probabilities of $M_1|N$

Notice that each column has to add up to 1. Reasonable values for e and f might be 0.05 and 0.02.

d (BONUS 5 points) Suppose $M_1 = 1$ and $M_2 = 3$. What are the possible numbers of stars?

Consider all the possible values of the focus and off-by-one variables, and the implications each has on the resulting possible values of N.

- **If neither F_1 nor F_2 are true, then the only possible value for N is 2 (astronomer 1 undercounts by 1 and astronomer 2 overcounts by 1).**
- **If F_1 is true and F_2 is false, then the only possible value for N is 4 (astronomer 1 undercounts by 3, astronomer 2 undercounts by 1).**

- **If F_1 if false, and F_2 is true, then there is no consistent value for N, so this can't be the case.**
- **Finally, if both F_1 and F_2 are true, then the possible values are $N \geq 6$, with astronomer 2 undercounting by $i \geq 3$ and astronomer 1 undercounting by $i + 2$.**

Question 8

(10 + 3 bonus points)

Let the instance space be $X = \{0,1\}^4$, the training set be $D = \{(< 0,0,0,0 >, 1)\}$, and the hypothesis space H be the set of conjunctions over X.

Notation: Let x_i be the ith attribute, and $\neg x$ denote "not x".

a (4 points) Compute the cardinality of the version space of H over D, $|VS_{H,D}|$.

$|VS| = 2^4 = 16$ **(VS = the set of conjunctions with all negated literals**

b (3 points) Derive the S and G frontiers using the candidate elimination algorithm.

$S = \{\neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge \neg x_4\}$ **(singleton set),** $G = \{True\}$ **(the null conjunction)**

c (3 points) Suppose you see the additional example $(< 1,1,1,1 >, 0)$. Derive the new S and G frontiers.

$S = \{\neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge \neg x_4\}$**,** $G = \{\neg x_1, \neg x_2, \neg x_3, \neg x_4\}$

d (BONUS 3 points) Suppose you see one more example, $(< 0,1,1,1 >, 1)$. Derive the new S and G frontiers.

$S = \{\neg x_1\}$**,** $G = \{\neg x_1\}$

Question 9

(4 points)

Suppose a training set is made up of 16 examples of class A, 8 examples of class B, 32 examples of class C, and 8 examples of class D. When growing a decision tree from this training set, what is the maximum information gain that any attribute can have?

Consider a single attribute x that is perfectly correlated with the class. I.e. $x = 1 \equiv C = A$, $x = 2 \equiv C = B$, $x = 3 \equiv C = C$, and $x = 4 \equiv C = D$. In this case, the entropy after splitting on this attribute will be 0 (all subsets are pure.) So the maximum information gain is the entropy of the training set - 0.

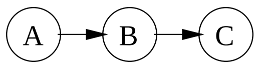
The entropy of the training set is:

$$\begin{aligned} H(D) &= -\frac{16}{64}\log_2(\frac{16}{64}) - \frac{8}{64}\log_2(\frac{8}{64}) - \frac{32}{64}\log_2(\frac{32}{64}) - \frac{8}{64}\log_2(\frac{8}{64}) \\ &= \frac{1}{4}\cdot 2 + \frac{1}{8}\cdot 3 + \frac{1}{2}\cdot 1 + \frac{1}{8}\cdot 3 \\ &= \frac{4}{8} + \frac{3}{8} + \frac{4}{8} + \frac{3}{8} \\ &= \frac{14}{8} = 1.75 \end{aligned}$$

Question 10

(5 points)

Consider the following Bayesian network, in which variables A, B and C are Boolean:



Suppose you want to learn the parameters for this network using the training set $\{< 0, 1, 1 >, < 1, 0, 0 >, < 1, 1, 1 >, < 1, ?, 0 >\}$, where examples are in the form $< A, B, C >$, and "?" indicates a missing value. Show the sequence of filled-in values and parameters produced by the EM algorithm, assuming the parameters are initialized by ignoring missing values. (Hint: EM converges very quickly on this problem.)

Initialization:
$P(A) = 0.75$
$P(B|A) = 0.5, P(B|-A) = 1$
$P(C|B) = 1, P(C|-B) = 0$

First iteration:
E step:

$$\begin{aligned} P(?=1) &= P(B|A,-C) = \frac{P(A,B,-C)}{P(A,-C)} \\ &= \frac{P(A)P(B|A)P(-C|B)}{P(A)P(B|A)P(-C|B) + P(A)P(-B|A)P(-C|-B)} \\ &= \frac{(0.75 \cdot 0.5 \cdot 0)}{(0 + 0.75 \cdot 0.5 \cdot 1)} \\ &= 0 \end{aligned}$$

So ? = 0 with probability 1. Compute conditional probabilities with this substitution.
M step:
$P(A) = 0.75$
$P(B|A) = 0.333\ldots, P(B|-A) = 1$
$P(C|B) = 1, P(C|-B) = 0$

Second iteration:

E step:

$$\begin{aligned}
P(? = 1) &= P(B|A,-C) = \frac{P(A,B,-C)}{P(A,-C)} \\
&= \frac{P(A)P(B|A)P(-C|B)}{P(A)P(B|A)P(-C|B) + P(A)P(-B|A)P(-C|-B)} \\
&= \frac{(0.75 \cdot 0.333\ldots \cdot 0)}{(0 + 0.75 \cdot 0.666\ldots \cdot 1)} \\
&= 0
\end{aligned}$$

M step: Same result as first iteration (converged).

Question 11

(3 points)

Let H be the set of hypotheses of the form $x = x_0 \vee x = x_1$, where $x_0, x_1 \in X$ (i.e., x_0 and x_1 are arbitrary elements of the instance space). What is the VC dimension of H?

VC-dim(H) = 2, because:

- **A set of cardinality 2 can be shattered by H. Let this set be $\{y_1, y_2\}$. All we have to do is pick y_1 and y_2 so that $y_1 \neq y_2$.**

Dichotomy	Realizable by making
y1 -, y2 -	x1 != y1, x2 != y1, x1 != y2, x2 != y2,
y1 -, y2 +	x1 != y1, x2 = y2
y1 +, y2 -	x1 = y1, x2 != y2
y1 +, y2 +	x1 = y1, x2 = y2

Table 4: Assignments to shatter set of 2 examples.

- **A set of cardinality 3 cannot be shattered: there is no way to realize the dichotomy y1 +, y2 +, y3 +, because H only allows at most two points to be positive.**

Some people thought the VC dimension was greater than 2. For example, the claim is that you can shatter a sets of size 3, $\{a, b, c\}$ by selecting, for the dichotomy $\{a, b, c\}, \{\}$ the hypothesis $x = d \vee x = d$ (i.e. by giving a hypothesis that selects the null set w.r.t. the 3 elements in question, and thereby shattering out the rest of the elements by elimination.) However, you need to be able to distinguish between the dichotomy above and $\{\}, \{a, b, c\}$, and now you're forced to list the elements explicitly, which you can't do with only 2 elements in the disjunction. Alternatively you can think of this as saying that you need to be able to shatter the set into all possible class assignments. So you can cover $< a, + >, < b, + >, < c, + >$, but not $< a, - >, < b, - >, < c, - >$.

Question 12

(5 points)

Suppose you want to learn to recognize digits in a 7-segment LED display from noisy examples (i.e., each segment has been flipped with 10% probability). Which of the learning algorithms you studied would you use?

The best choice is naive Bayes, because the attributes (i.e., whether each segment is on or off) are independent given the class (i.e., the digit), so naive Bayes is the optimal classifier for this problem. Nearest-neighbor with overlap distance is also a reasonable solution, and can be given half-credit. A general Bayes net can also be given half-credit (it will have zero bias, like naive Bayes, but more variance, and will also be slower). Likewise for a neural net with one output per class and a "max" function. Decision trees and rules are the least appropriate.

Question 13

(8 points)

How would you modify Graphplan to handle uncertainty (i.e., the initial state is a set of worlds) assuming that actions are still STRIPS (i.e., no sensing actions are available)?

The problem asks to refine Graphplan to handle a limited level of uncertainty, namely uncertainty as to the initial state of the world (but not about the outcomes of actions). Furthermore, this uncertainty is only represented as a set of possible initial states, rather than probabilities over initial states or any other representation.

First a very simple, and unsatisfying solution, but one that might start to expose the issues. Using the standard STRIPS representation, a plan (i.e. specification of which actions to execute at each level of the planning graph) can only be correct if it succeeds in all possible worlds. Any proposition which is required in the initial state for success, and which does not appear in all the initial states cannot contribute to a successful plan. So simply take the intersection of the initial states (i.e. the set of propositions that are true in all possible initial states) and use that as the initial state to run graphplan from. This will handle situations where you don't know if a precondition is going to be true, but you can easily make it true. So if the work surface may be clean or not, but you're not sure, and you need it clean to work, then you might as well add a clean action. It doesn't handle many other situations where uncertainty is present.

To motivate a better solution, consider the "bomb in the toilet" problem. You have two packages, one of which contains a bomb. You can disarm the bomb by putting it into the toilet. This isn't handled by the solution described above because the uncertainty here is about which package the bomb is in. If we just take the intersection of BOMB-IN-A and BOMB-IN-B, we lose all information about the bomb. The basic idea for the solution to this sort of problem is to run Graphplan in each of the possible worlds in parallel. I give a sketch here, and a reference to a paper with complete details below.

First, expand a separate planning graph out for each possible initial world state. Expand them out until all "worlds" have the goal conditions in the final proposition level, and there are no mutexes between them. Now, the instantiations of the actions will actually have to vary. In the world with BOMB-IN-A, you want to execute DUNK-A, and in the world with BOMB-IN-B, you want to execute DUNK-B. So as a second cut, let's try just taking the union of the

plans in each world. We end up with the plan DUNK-A and DUNK-B both at time step 1.

So we're almost there, but not quite. Since there are no sensing actions, we can't just execute the actions that are called for in each world, because we don't know which world we'll be in. An action that's required in world u might be deadly in world v. To exemplify, lets add a new condition, you can only stick one package into the toilet at a time, and once you've done so, the package gets stuck and you can never dunk another one. We can see that in this world, there is no successful plan, but our algorithm doesn't know it. To handle this situation, we must check each action in world u against all the other worlds v, w, etc. the same time step. By looking for mutexes between an action in world u and the actions and propositions in the other world, we can determine if that action is indeed legal in <u>all</u> worlds. If not, we add mutexes as in regular graphplan and force additional expansion of the planning graph.

Two additional notes. First, when we get to the solution extraction phase (in which we search backwards through the planning graph for a solution), it's much more efficient to do this one level at a time in each possible world than all the way through one world's planning graph and then the next one. Searching the planning graphs for the different worlds in an interleaved fashion allows you to find interactions more quickly. Also, there's one additional tweak that is required to make the algorithm complete. If an action that is necessary in one world causes problems in another, there may be an additional way to address the problem besides expanding the graph out more. This is called *confrontation*. The basic idea is that you might be able to avoid the unfortunate side effects of the action by forcing the appropriate preconditions to be true. For example, suppose your goal is to get to school. There are two worlds, in one it's RAINING, and in the other you have ¬CASH. You can BUS or WALK (but not both). In world u you don't have money(¬CASH), so you have WALK, but in world v, it's RAINING, so if you WALK (which you've decided to do because of world u), you'll get WET. You can confront this undesired outcome by first executing WEAR-RAINCOAT, even though it isn't raining in world u.

For a complete discussion of conformant graphplan addressing both uncertain initial states and unceratin actions, see ftp://ftp.cs.washington.edu/pub/ai/cgp-aaai98.ps, particularly the section called "Conformant Graphplan", up to the section entitled "Actions with uncertain outcomes."

Question 14

(8 points)

Describe how you would design a probabilistic part-of-speech tagger. You may assume you have a tagged corpus (that is, a large body of text in which each word is already labelled with its correct part of speech.) Describe the objects you will compute probabilities over, and how you will use those probabilities to predict the tags for unseen text. (Your scheme can be as simple as you want, but describe its shortcomings.)

There are many reasonable answers to this question. Two common techniques used in statistical natural language parsing are Bayesian networks and Hidden Markov Models. First let's consider a Bayesian network. The objects over which you'll compute probabilities are the parts of speech of the words in the tagged corpus. You could compute the probability of a tag given the word being tagged (e.g. $p(VERB|``fly'') = .8$), or conditional on the word and the tag of the previous word, ($p(VERB|ADJECTIVE, ``fly'') = .1$, as in "The big black fly buzzed around my head.") or conditional on the word and the previous word ($p(VERB|``black'', ``fly)$). Or you could consider more context. The tradeoff is between more discriminatory power (by including more information in the conditions) and more training data required to get good probabilities for all things being conditioned on. (How many times are you going to see "black fly" vs. "ADJECTIVE fly".)

A special case of Bayesian models is the Hidden Markov Model (HMM). This is a probabilistic finite state machine that models a process using the following two assumptions. (1) the probabilities for transitioning between two states depend only on the previous k states for some constant k, and (2) these probabilities don't change over time. At each transition (or equivalently for each state), an output symbol is emitted, also probabilistically. Consider a vending machine which can either be in Coke mode or Pepsi mode. When you stick in a coin, it emits a soda, which is more likely to be Coke in Coke mode or Pepsi in Pepsi mode, and it also transitions to one of the two states, based on some probability table which is conditioned only on the previous state. The kinds of questions you can ask with a HMM are: 1) From a model and a sequence of symbols, compute how likely the sequence is, given the model, 2) Given a sequence of symbols and a model, compute the sequence of states that best explains the sequence of symbols, and 3) Given a sequence of symbols and a space of models, find the model that best explains the sequence. Efficient algorithms exist for each of these questions. (Hint, the last one is a variation on EM.) So how does this help us with the task at hand. Consider the

sequence of states corresponding to parts of speech. With each state, a symbol (word) is emitted with some probabilitity. Using the algorithms above, we can train a hidden markov model on the labelled data and then use that to predict parts of speech on new sentences. For more information on these techniques, see the resources pointed at in the course web. In particular, there is a nice sample chapter on HMMs on the webpage for Foundations of Statistical Natural Language Processing. The subsequent chapter (which isn't on the web) talks about how to apply this to various problems like POS tagging.