Graduate AI – Midterm

March 6, 2012.
10:30–11:50am.

Name: ____________________

Andrew ID: ____________________

Read all of the following information before starting the exam:

- Please clearly write your **name** and **Andrew ID** in the spaces above.
- For full credit, please show all work *clearly* and *in order*.
- The test consists of five questions:
  - One multi-part short answer question.
  - Four long-answer questions.
- Good luck!

Problem	Possible	Awarded
1	20	
2	20	
3	20	
4	20	
5	20	
Total	100	

1. (*20 points*)
Please clearly and concisely respond to all of the following short-answer problems. When appropriate, explain your answer or show your work.

a. (*3 pts*) Explain why the number of misplaced tiles heuristic for n-puzzle is dominated by the Manhattan distance heuristic.

b. (*3 pts*) We know that in a planning graph operations monotonically increase, conditions monotonically increase, and mutexes monotonically decrease. Explain in 1-2 lines why it follows that the graph levels off.

c. (*3 pts*) Iterative deepening search never expands more nodes than breadth first search before finding the goal. T/F? Explain.

d. (*3 pts*) Suppose h_1 and h_2 are two admissible heuristics for a search problem such that for all nodes n, $h_1(n) < h_2(n)$. Then there do not exist any starting nodes for which A* using h_1 expands fewer nodes than A* using h_2. T/F? Explain.

e. (*3 pts*) A* uses $O(b^d)$ space while IDA* uses $O(bd)$ space, where b is the branching factor of the search tree and d is the depth of the shallowest goal node. T/F? Explain.

f. (*3 pts*) Depth-first search uses $O(bd)$ space, where b is the branching factor of the search tree and d is the depth of the shallowest goal node. T/F? Explain.

g. (*2 pts*) What does it mean for a search algorithm to be complete? Optimal?

2. (*20 points*)
Consider the following CNF SAT clauses:

$$(x_4 \lor x_5 \lor \neg x_6)$$
$$(x_1 \lor x_4 \lor \neg x_5)$$
$$(x_3 \lor \neg x_4 \lor x_6)$$
$$(x_2 \lor \neg x_3 \lor x_6)$$
$$(x_1 \lor \neg x_2 \lor \neg x_5)$$
$$(x_3 \lor \neg x_4 \lor x_5)$$
$$(\neg x_1 \lor x_7 \lor \neg x_8)$$
$$(\neg x_1 \lor x_8 \lor \neg x_7)$$

A DPLL algorithm tries to make the following assignments sequentially:

$$x_6 \mapsto \mathbf{F}$$
$$x_5 \mapsto \mathbf{T}$$
$$x_1 \mapsto \mathbf{F}$$
$$x_2 \mapsto \mathbf{T}$$
$$x_4 \mapsto \mathbf{T}$$
$$x_3 \mapsto \mathbf{F}$$
$$x_8 \mapsto \mathbf{T}$$
$$x_7 \mapsto \mathbf{T}$$

However, at some point during these assignments, a conflict is discovered.

a. (*7 pts*) Which decision induces the conflict?

b. (*7 pts*) Draw the conflict graph for when the conflict is discovered.

c. (*6 pts*) Draw a conflict cut on the above graph that involves the *fewest* decision variables, and write out what the corresponding learned clause is.

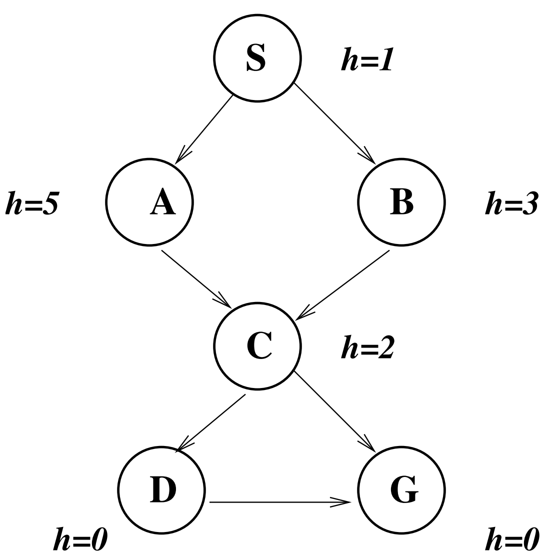
Scrap Page

Please do not remove this page from the test packet.

3. (*20 points*)

a. (*8 pts*) Consider the search problem below with start state S and goal state G. The heuristic values are shown. Unfortunately we do not know the transition costs, and we really would like to know them. However we know that the given heuristic is admissible. Furthermore we know what was the priority queue of A^* after each node expansion, namely:

1. { (S, f=1) }
2. { (B, f=5), (A, f=6) }
3. { (A, f=6), (C, f=7) }
4. { (C, f=6) }
5. { (D, f=5), (G, f=7) }
6. { (G, f=6) }



Fill in the transition costs for all the edges.

b. (*6 pts*) Chris claims that: In general, if we are given a search problem, for which we know: (i) the heuristic value of all the nodes; (ii) that the heuristic is admissible; (iii) the solution found by A^* is given; and (iv) all the values of the prioriy queue of A^*'s search performance, then the transition values of all the edges in the search problem can be uniquely determined. Is Chris correct? If yes, then prove it. Otherwise, give an example that shows that Chris is incorrect.

c. (*6 pts*) Would A^* still be guaranteed to find the minimal path to the goal if there are negative transition costs?

Scrap Page

Please do not remove this page from the test packet.

4. (*20 points*)
No-op actions are ones that have one precondition and one effect, and both are the same. In Graphplan, no-op actions are added, at every level, for each proposition that appears in the previous level.

a. (*5 pts*) Explain why no-op actions are needed in Graphplan.

b. (*5 pts*) In Graphplan, no-op actions are included when determining mutex relationships. Why is this done? Would Graphplan still work if it did not find those types of mutexes?

c. (*2 pts*) A CoBot needs to deliver mail between GHC, NSH, and the UC. Let's use planning with GraphPlan to help the CoBot deliver mail to the different mail centers.

$Init(At(M1, GHC) \wedge At(M2, NSH) \wedge At(M3, UC) \wedge Mail(M1) \wedge Mail(M2) \wedge Mail(M3) \wedge MailCenter(GHC) \wedge MailCenter(NSH) \wedge MailCenter(UC) \wedge Robot(CoBot) \wedge At(Cobot, GHC))$
$Goal(At(M1, NSH) \wedge At(M2, UC) \wedge At(M3, GHC) \wedge At(Cobot, GHC))$

Action (Go(b,s,e),
$PRECOND : Robot(b) \wedge At(b, s) \wedge MailCenter(s) \wedge MailCenter(e)$
$EFFECT : \neg At(b, s) \wedge At(b, e))$

Action(PickUp(b,l,m),
$PRECOND : Robot(b) \wedge Mail(m) \wedge MailCenter(l) \wedge At(b, l) \wedge At(m, l) \wedge \neg GripperFull(b)$
$EFFECT : GripperFull(b) \wedge GripperCarries(m) \wedge \neg At(m, l)$

Action(PutDown(b,l,m),

Given ONLY the predicates given in the other two actions (i.e. DON'T introduce any new predicates), write a definition for the PutDown(b,l,m) function that specifies preconditions and effects. Use the predicates Robot(b), MailCenter(l), and Mail(m) in your solution.

d. (*2 pts*) Given the set of actions, formulate a plan that will reach the goal state from the start state.

e. (*6 pts*) Say the initial state is ONLY

$At(M1, GHC) \wedge At(Cobot, GHC) \wedge Robot(CoBot) \wedge Mail(M1) \wedge MailCenter(GHC) \wedge MailCenter(NSH)$

and the goal is ONLY

$Goal(At(M1, NSH) \wedge At(Cobot, NSH))$.

Draw ONLY the first level of the planning graph including the states, actions, and necessary mutexes.

Scrap Page

Please do not remove this page from the test packet.

5. (*20 points*)
This problem deals with linear and integer linear programming. Assume the following linear program (LP) $A\mathbf{x} \leq \mathbf{b}$:

$$A = \begin{bmatrix} 4 & 1 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 9 \\ 2 \\ 2 \\ 0 \\ 0 \end{bmatrix}$$

$$x, y \in \mathbb{R}$$

with linear objective function

$$\max x + y$$

a. (*3 pts*) Draw the polytope representing the set of all feasible solutions to the LP.

b. (*3 pts*) Trace out the Simplex method on this polytope, starting at the origin.

c. (*2 pts*) What is the optimal value (i.e., what maximizes the objective) of the LP?

d. (*6 pts*) Restrain the decision variables x and y to integer points. That is, let $x, y \in \mathbb{Z}$. Draw the branch and bound tree for the problem such that:

- Nodes are expanded in DFS order
- Branch first on x, with the left branch $x \geq 2$ and the right $x \leq 1$
- If necessary, branch on y with the left branch $y \geq 2$ and the right $y \leq 1$

e. (*6 pts*) On the original polytope (but with $x, y \in \mathbb{Z}$), draw *and label* three cuts, one of each type: (A) separating and valid, (B) not separating but still valid, (C) invalid.

Scrap Page

Please do not remove this page from the test packet.