CMPSCI 683 Fall 2008
Prof. Victor Lesser October 27, 2008

# MID-TERM EXAM

The answers to these questions should be specific and to the point; we are not looking for essays! There are two types of questions 5 short essay like questions (8,8,12,8,8 points) and 3 long questions (32, 12, 12 points). Please be careful about timing.

1. Short questions (44 points)

A. (8 points) Explain why the heuristic search control technique of mini-conflict cannot be used in a normal best-first search strategy.

B. (8 points) A variant of IDA* involves increasing the f contour by a fixed amount each time instead of replacing it by the smallest f found that is larger than current f.

B-1) Explain why this may result in not finding an optimal answer.

B-2) Suppose you wanted to guarantee that you were within 5% of the optimal answer. How would you decide how much to increase the f contour at each stage?

C. (12 points) RTA* was a modification of A* for real-time decision making.

C-1) Explain the key ideas behind this approach.

C-2) How would you do something similar for an MDP type decision making. Give a brief sketch of your newly developed RTMDP*

D. (8 points) Draw the performance profile of Anytime-A* in the following situations:

D-1) There is only one solution

D-2) There are very few solution nodes

D-3) There are a large number of solution nodes.

E. (8 points) The HEARSAY-II speech understanding system when searching for words to extend a phrase (partial sentence) re-analyzed acoustic data at the lowest level to determine the likelihood of the desired words. Explain the difference between words generated bottom-up and those coming from this top-down process.

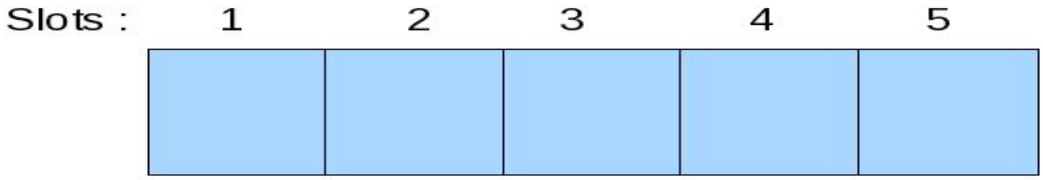
**********

Long Questions (56 points)

1 (32 points)
Consider the following task scheduling problem. You are provided with the following 5 tasks:

| Task | Deadline | Time | Utility |
|---|---|---|---|
| A | Slot 3 | 1 Slot | 10 |
| B | Slot 3 | 2 Slots | 15 |
| C | Slot 5 | 1 Slot | 5 |
| D | Slot 2 | 2 Slots | 20 |
| E | Slot 3 | 1 Slot | 5 |

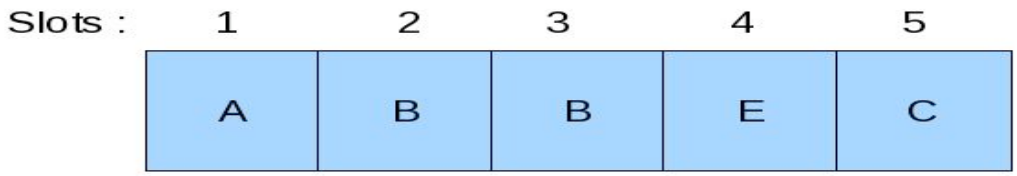
As many of the above tasks as possible must fit into the following 5 time slots:




Each slot can have exactly 1 task. A task that requires 2 slots must occupy consecutive slots. If a task is completed by the deadline slot, the utility from the task is as shown. If the task is not completed by its deadline, the utility is 0. For example, if Task A is placed in slot 4 or 5, utility is 0. For slots 1, 2 and 3 the utility is 10. Similarly if Task B is

placed in slot 1 or 2, the utility gained is 15. If Task B is placed in slot 3, it also has to occupy slot 4, which is past its deadline. Therefore the utility received is 0. There is no partial utility for providing 1 slot for a 2 slot task.

Therefore an example schedule would be:




The utility for this schedule would be 10(A) + 15(B) + 0 (E) + 5 (C). Note there is no utility for completing E since the task was performed after its deadline.

1a (16 out of 32 points)
Provide a heuristic for A* search. Argue that your heuristic is both an overestimate and leads to a monotone search.

1b (16 out of 32 points}
Show the first three steps of the A* search for this problem. In the three steps, you will expand your root node plus two other nodes depending on your f-value. Show the h and g values for all the nodes in your tree so far.

**********

2 (12 points)}

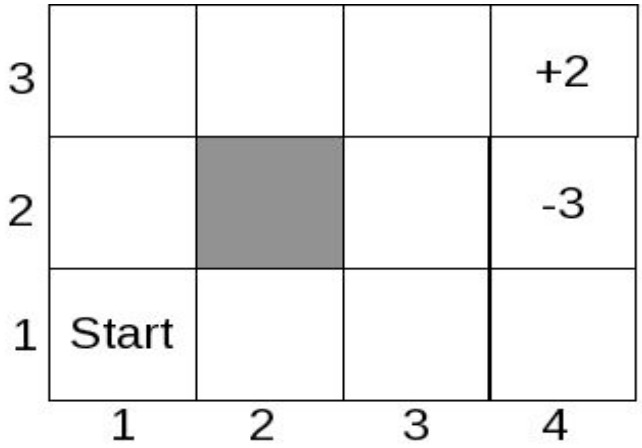
Consider the following 3-SAT problem:

(A v B v not C) and (A v not B v C) and (not A v B V C) and (not A v not B V C) and (not A v not B v not C) and (A v B v C)

Try to solve this 3-SAT problem using the GSAT algorithm. For the first try, assume the following truth assignment --- A = F; B = F; C = F. Do it for a maximum of three steps. At each step, explain your decision process.

**********

3 (12 points) Recall the grid environment from class.




Also recall the transition model of the environment: the ``intended'' outcome occurs with probability 0.8, but with probability 0.2 the agent moves at right angles to the intended direction. Assume the following: 1) The cost of an action is -0.2. and 2) The discount factor is set to 0.9.

Use the value-iteration algorithm to calculate the value at the end of the next iteration for state 1,1 (i.e. the start state). To avoid some unnecessary calculation assume from state 1,1 that you can only move right or up, and 1,1 is the first state that is re-evaluated in the value-iteration.