CS 371/671 Mid-term Study Guide

To be discussed in class on October 2nd

1 Agents

Question 2.2 Let us example the rationality of various vacuum cleaner agent functions.

a. Show that the simple vaccum-cleaner agent function described in Figure 2.3 is indeed rational under the assumptions in page 38.
b. Describe a rational agent function for the case in which each movement costs one point. Does the corresponding agent program require an internal state?
c. Discuss possible agent designs for the cases in which clean squares can become dirty and the geography of the environment is unknown. Does it make sense to learn from its experience in these cases? If so, what should it learn? If not, why not?

Question 2.3 For each of the following assertions, say whether it is true or false and support your answer with examples or counterexamples.

- An agent that senses only partial information about the state cannot be perfectly rational.
- There exists task environments in which no pure reflex agent can behave rationally.
- There exists task environments in which every agent is rational.
- The input to an agent program is the same as the input to agent function.
- Every agent is implementable by some program/machine combination.
- Suppose an agent selects its actions uniformly at random from a set of possible actions, there exists a deterministic task environment for which this agent is optimal.
- It is possible for an agent to be perfectly rational in two distinct environments.

Question 2.13 Discuss possible agent programs for the following stochastic versions of the vacuum world.

a. Murphy's Law: 25% of the time, the *Suck* action fails to clean floor when dirty and deposits dirt if clean. How will your agent program be affected if the sensor gives the wrong answer 10% of the time?
b. Small Children: At each time step, there is a 10% chance of each clean square becoming dirty. Can you come up with a rational agent in this case?

2 Search and Heuristics

1. Imagine a car-like agent to exit a maze like the one shown below. The agent is directional and at all times faces some direction $d \in (N, S, E, W)$. With a single action, the agent can either move forward at an adjustable velocity v or turn. The turning actions are left and right, which change the agent's direction by 90 degrees. Turning is only permitted when the velocity is zero (and leaves it at zero). The moving actions are $fast$ and $slow$. $Fast$ increments the velocity by 1 and $slow$ decrements the velocity by 1; in both cases the agent then moves a number of squares equal to its NEW adjusted velocity. Any action which would collide with a wall crashes the agent and is illegal. Any action which would reduce v below 0 or above a maximum speed V_{max} is also illegal. The agent's goal is to

   find a plan which parks it (stationary) on the exit square using as few actions (time steps) as possible.

   As an example: if the agent shown were initially stationary, it might

   first turn to the east using (right), then move one square east using fast, then two more squares east using fast again. The agent will of course have to slow to turn.

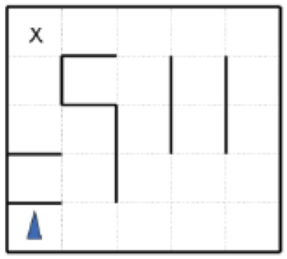

Figure 1:

a. If the grid is M by N, what is the size of the state space? Justify your answer. You should assume that all configurations are reachable from the start state.
b. What is the maximum branching factor of this problem? You may assume that illegal actions are simply not returned by the successor function. Briefly justify your answer.
c. Is the Manhattan distance from the agent's location to the exit's location admissible? Why or why not?
d. State and justify a non-trivial admissible heuristic for this problem which is not the Manhattan distance to the exit.
e. If we used an inadmissible heuristic in A* tree search, could it change the completeness of the search?
f. If we used an inadmissible heuristic in A* tree search, could it change the optimality of the search?
g. Give a general advantage that an inadmissible heuristic might have over an admissible one.

2. Consider the search space below, where S is the start node and G_1, G_2, and G_3 satisfy the goal test. Arcs are labeled with the cost of traversing them and the h functions values are reported beside the graph.

   For each of the following search strategies, indicate which goal state is reached (if any) and list, in order, all the states popped off of the list. When all else is equal (i.e., to break ties), nodes should be placed in the list in alphabetical order.

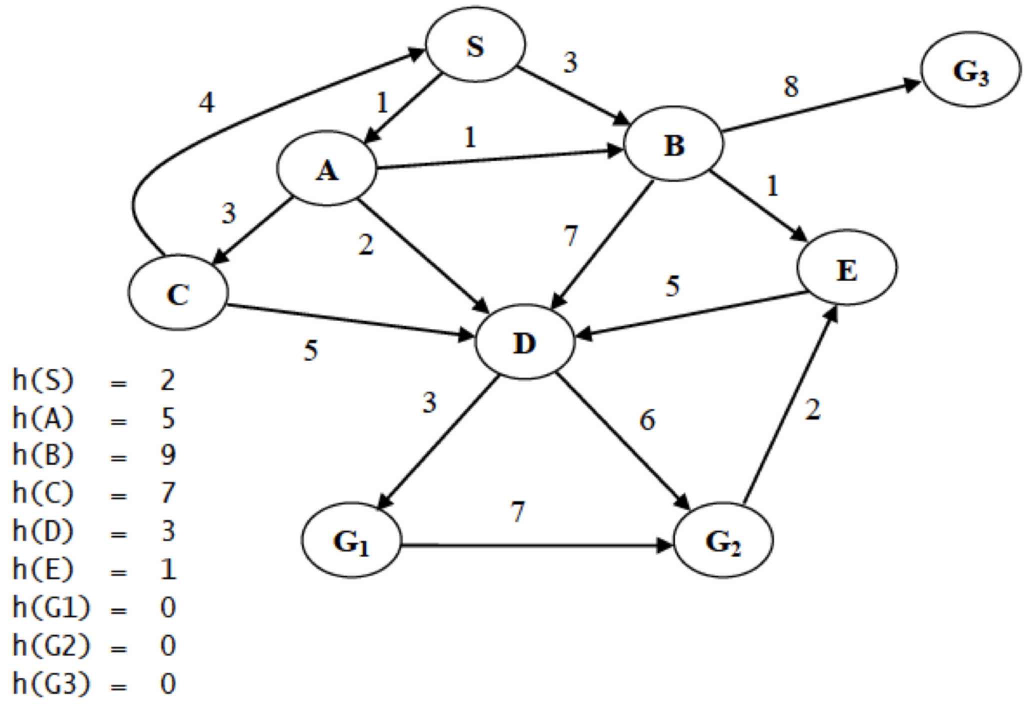

Figure 2:

a. Breadth First b. Depth First c. Iterative Deepening d. Uniform Cost h. e. Best-first (using f = h) f. Best-first (using f = g + h) g. Beam Search (with beam width = 2 and f = h) h. Hill Climbing (using the h function only)

Is this h function admissible? Explain you answer. What impact did this have on each of parts e-h above? Verbal answers are fine; you do not need to fix h and redo the searches in these parts.

3. Question 3.6 Check every part. We will discuss (b) in class – Give a complete formulation for each of the following. Choose a formulation that is precise enough to be implemented. (b) A 3-foot tall monkey is in a room where some bananas are suspended from the 8-foot ceiling. He would like to get the bananas. The room contains two stackable, movable, climbable, 3 foot high crates.

4. Question 3.11 What is the difference between a world state, a state description and a search node? Why is this distinction useful?

5. Question 4.1 Give the name of the algorithm that results from each of the following special cases:

- Local beam search with k =1.
- Local beam search with one initial state and no limit on the number of states retained.
- Simulated Annealing with T = 0 at all times (and omitting the termination test).
- Simulated Annealing with $T = \infty$ at all times.
- Genetic algorithm with population size, N = 1.

3 Adversarial Search

1. Question 5.7 Prove the following assertion: For every game tree, the utility obtained by MAX using minimax discussions against a suboptimal MIN will never be lower than the utility obtained playing against an optimal MIN. Can you come up with a game tree in which MAX can do still better using a *suboptimal* strategy against a suboptimal MIN?

2. Question 5.12 Describe how the minimax and alpha-beta algorithms change for two-player, non-zero sum games in which player has a distinct utility function and both utility functions are known to both players. If there are no constraints on the two terminal utilities, is it possible for any node to be pruned by alpha-beta? What if the player's utility functions on any state differ by at most a constant k, making the game almost cooperative?

3. The standard Minimax algorithm calculates worst-case values in a zero-sum two player game, i.e. a game in which for all terminal states s, the utilities for players A (MAX) and B (MIN) obey $U_A(s) + U_B(s) = 0$. In the zero sum case, we know that $U_A(s) = -U_B(s)$ and so we can think of player B as simply minimizing $U_A(s)$. In this problem, you will consider the non zero-sum generalization in which the sum of the two players' utilities are not necessarily zero. Because player A's utility no longer determines player B's utility exactly, the leaf utilities are written as pairs (U_A, U_B), with the

   first and second component indicating the utility of that leaf to A and B respectively. In this generalized setting, A seeks to maximize U_A, the

   first component, while B seeks to maximize U_B, the second component.

- Propagate the terminal utility pairs up the tree presented in Figure 3 using the appropriate generalization of the minimax algorithm on this game tree. Fill in the values (as pairs) at each of the internal node. Assume that each player maximizes their own utility.
- Briefly explain why no alpha-beta style pruning is possible in the general non-zero sum case. Hint: think first about the case where $U_A(s) = U_B(s)$ for all nodes.

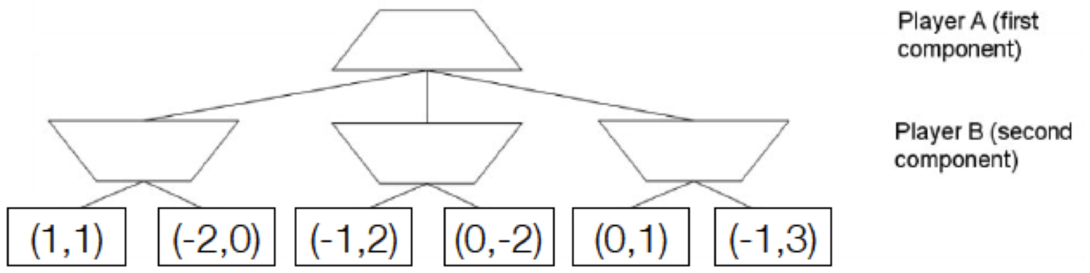

Figure 3:

4 Propositional Logic

1. Represent each of these sentences in propositional logic.

- If one has a fever, one should not go to work.
- It is hot but not sunny.
- One should not bike on highways.
- Do not drink or text when driving.

2. For each of the formulae, use truth tables to show whether each of them is valid, satisfiable or unsatisfiable

a. $(P \rightarrow Q) \land (P \rightarrow \neg Q)$

b. $(\neg P \lor \neg Q) \land (\neg Q \lor \neg R) \land (\neg R \lor \neg P) \land P \land Q$

c. $((P \rightarrow Q) \land (Q \rightarrow R)) \leftrightarrow (P \rightarrow R)$

d. $((P \rightarrow Q) \rightarrow (Q \rightarrow R)) \leftrightarrow (P \rightarrow R)$

3. Put each of the following in CNF form

- $(P \rightarrow Q) \land (R \lor S)$
- $(P \land Q \rightarrow (X \lor Y)) \lor (R \land S)$
- $((P \rightarrow Q) \rightarrow \neg X) \leftrightarrow (A \land B)$

4. Given,

   1. $P \land Q \rightarrow R$
   2. $A \land R \rightarrow B$
   3. $Q \land \neg C \rightarrow A$
   4. Q
   5. $\neg(\neg P)$
   6. $\neg C$

   Show: $R \land A$

   Prove the above concept in two ways: first a normal deduction – let us call it "natural deduction" and second method is using *only* resolution rule.

5. Consider the following sentence,

$$[(Food \rightarrow Party) \land (Drinks \rightarrow Party)] \rightarrow [(Food \land Drinks) \rightarrow Party]$$

- Determine, using enumerate whether this sentence is valid, satisfiable or invalid.

- Convert the left hand and right hand sides of the implication into CNF, showing each step and explain how the results confirm to the previous answer.
- Prove your answer to the first part using resolution.

6. Question 7.20. Convert the following set of sentences to clausal form.

    1. $A \leftrightarrow (B \vee E)$
    2. $E \rightarrow D$
    3. $C \wedge D \rightarrow \neg B$
    4. $E \rightarrow B$
    5. $B \rightarrow F$
    6. $B \rightarrow C$

    Give a trace of the execution of DPLL on the conjunction of these clauses.