CISC 481 Homework 1

October 11, 2019

1. (15 pts, 5 ea) From exercise 3.9 in the book: The *missionaries and cannibals problem* is usually stated as follows. Three missionaries and three cannibals are on one side of a river, along with a boat that can hold one or two people. Find a way to get everyone to the other side without ever leaving a group of missionaries in one place outnumbered by the cannibals in that place.
   - Formulate this as a formal search problem. What are the states, actions, goal test, path costs, etc.?
   - Of the uninformed search strategies in Chapter 3, which would you recommend to find a safe plan in the fewest number of bus trips, and why?
   - Trace the search tree produced by the method you named above, assuming a graph search implementation (prune repeated states).
2. (10 pts) Give a complete search space formulation [states, actions, goal test, etc] for the following problem:
   - You have a three gallon and a five gallon measuring device, and a working sink. You wish to measure out four gallons of tap water.
3. Consider the problem of seating 5 guests at a round table. Unfortunately, every guest hates every other guest, although not necessarily to the same degree. You need to seat everyone, but wish to minimize the total misery of your guests. Let the symmetric table in Figure 1 represent the sum total misery that i and j feel when they sit next to one another. Think of this as a search problem (states, operators, path costs). Let a state be a particular arrangement of guests around the table (for example, guest D is in seat 1, guest A in seat 2...etc.) and a list of guests that are unseated. Assume that the initial state is a round table with A seated in seat 1 (since the table is round, this leads to every possible solution by rotation!). You can model the table as a simple array t of length 5.

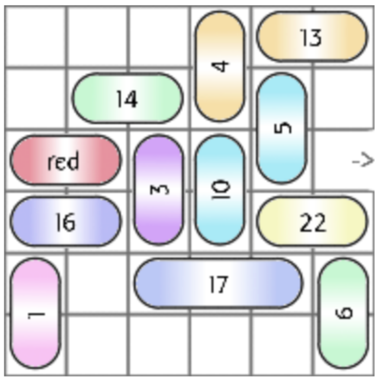
	A	**B**	**C**	**D**	**E**
A	0	4	2	5	1
B	4	0	4	6	4
C	2	4	0	4	3
D	5	6	4	0	2
E	1	4	3	2	0

Figure 1: Misery Matrix

   (a) (5 pts) Describe a successor-state function (expand). **Do not allow repeated states in the search space** (i.e. no node/state can be reached from the initial state by two different paths)—this can be done with a careful successor-state function rather than by costly checking.

   (b) (5 pts) Assume that the path cost will measure total misery. Construct a **reasonable and admissible heuristic function** for this search space (a heuristic of 0 is admissible but not reasonable, a heuristic that computes the optimal placement of guests is not reasonable either!)

4. (5 pts) Develop an admissible heuristic for Rush Hour. Rush Hour is a sort of generalized sliding-block puzzle; the objective is to get the Red car out of the parking lot (at the arrow on the right hand side). Cars can only move forward and backward, never sideways. Cars can move any distance in one move (but cannot leave the board except at the exit).



Reachable Configurations: 1212

Minimum solution: 82 moves

Figure 2: A typical Rush Hour board

5. (15 pts, 3 ea) Give the name of the algorithm that results from (or is most similar to) each of the following special cases. Choose from algorithms in Chapters 3 and 4, e.g. breadth-first search, depth-first search, iterative deepening, A*, hill climbing/gradient descent (with or without first choice, stochastic, random restarts), or just a random search.
   (a) Local beam search with $k = 1$
   (b) Local beam search with one initial state and no limit on the number of states retained
   (c) Simulated annealing with $T = 0$ at all times (and omitting the termination test)
   (d) Simulated annealing with $T = \infty$ at all times
   (e) Genetic algorithm with the population size $N = 1$.

6. (15 pts, 5 ea)Consdider the 2-player game in Figure 3
   (a) Draw the complete game tree, using the following conventions:
      - Write each state as (s_A, s_B) where s_A and s_B are the positions of player A and player B, respectively, and denote the token locations $1, 2, 3, 4$.
      - Put each terminal state in a square box and write its game value in a circle beside it.
      - Put Loop States (states that already appear in the path to the root) in double square boxes. Since their value is unclear, annotate each with a "?" in a circle beside the box.
   (b) Now mark each node with a backed-up minimax value (also in a circle). Explain how you handled the "?" values and why.

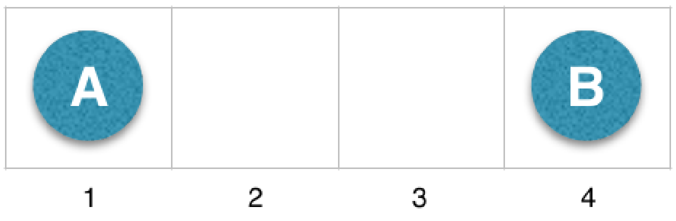
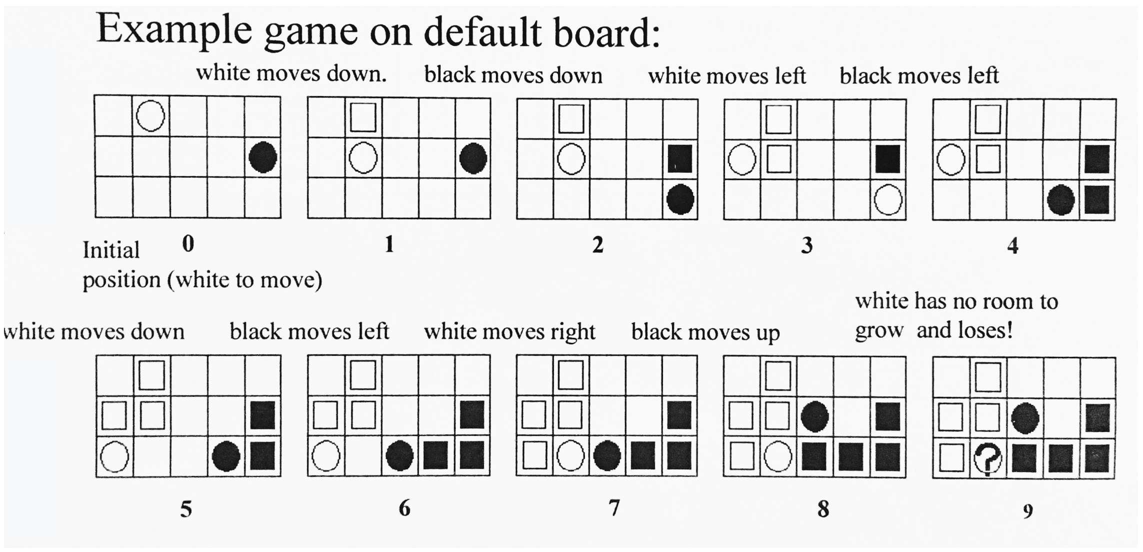


Figure 3: Player A moves first. The two players take turns moving, and each player must move their token to an open adjacent space in either direction. If the opponent occupies adjacent space, then the player may jump over the opponent to the next open space if any. (For example, if A is on 3 and B is on 2, then A may move back to 1.) The game ends when one player reaches the opposite end of the board. If Player A reaches space 4 first, then the value of the game to A is 1; if Player B reaches space 1 first, then the value of the game to player A is -1.

(c) Explain how the standard minimax algorithm would fail on this game tree, and briefly sketch how to fix it, using the idea(s) in your answer to part b. Does your modified algorithm give optimal decisions for all games with loops??

7. (20 pts, 4 ea) The game Surround is essentially a strategy version of two-player Snake (or the uni-level light cycle game from the original Tron movie). Played on an $n \times m$ rectangular board, each player, in turn, moves the "head" of their snake to an open spot one space away. The "body" of the snake grows to the previous head location, and the body never goes away (i.e. that space is forever closed to both players). The goal is to block your opponent from moving (i.e. opponent has no legal moves so you win). Movement is up, down, left, right but not diagonally. You may not move to a square ever touched by you or your opponent (the body), or the current heads, or grow off the board, or "pass" your turn. For this particular



problem, use the board configuration in Figure 4

(a) **Approximately** how many possible games of surround are there, assuming $N \times M$ squares (9 here in our example, but other configurations are possible)?

(b) Show the entire game tree starting from the start board down to depth 2 (i.e. one White, and one Black move, White moves first).

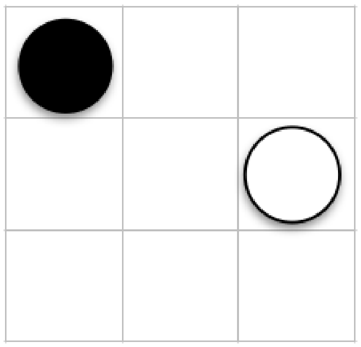


Figure 4: 3×3 board with Start positions for both players (White moves first)

(c) Assume the following evaluation function for White: *Utility = (Number of spaces White can move to next turn) - (Number of spaces Black can move to next turn).* Mark the evaluation of all states at level 2 of the tree. Example: Starting state is 3 white moves - 2 black moves = 1.

(d) Using minimax, mark on the tree the backed-up values for the positions at depths 1 and 0, and use those values to choose the best starting move.

(e) Circle the nodes at depth 2 that would *NOT* be evaluated if alpha-beta pruning were applied, assuming the nodes are generated in the optimal order for alpha-beta pruning.

8. (15 pts, 5 ea) Translate the following into first order logic. **Be consistent** with your predicates. For example, if you translate the sentence "All men must die" as $\forall x(Man(x) \implies Dies(x))$ then the response "All men must serve" should be something like $\forall x(Man(x) \implies Serves(x))$.

(a)
```
No homework is fun
This assignment is homework
Therefore, this assignment is not fun
```

(b)
```
Some guys have all the luck
Some guys have all the pain
Some guys get all the breaks
Some guys do nothing but complain
```

(c)
```
All dogs have a breed
A dog is a mutt only if it is not purebred
A dog is purebred if both of its parents are purebred and are the same breed
A Yellow Labrador is a purebred
```

9. Given a knowledge base consisting of the sentences from problem 8 part c, as well as the following:

```
Brandi was a dog
Brandi's mother was Tabatha, and her father was Moondog Moses
Moondog Moses was a Yellow Labrador
Tabatha was a Yellow Labrador
```

- (10 pts) Convert all sentences in the KB to conjunctive normal form
- (10 pts) Prove that `Brandi was not a mutt` by resolution.