Midterm (100 points)

(5 pts) 1. Develop a PEAS description for an intelligent tutoring agent.
   Solution: This is one of many solutions:
   Performance measure: Maximize test scores, improve student understanding
   Environment: Students
   Actuators: Display, Exercises, Progress Indicators
   Sensors: Keyboard, Mouse

(5 pts) 2. Describe the relationship between A* and D* search.
   Solution: Both A* and D* are optimal, complete algorithms based on heuristic search. However, D*
   can handle dynamic or partially known environments by updating and propagating arc cost changes. A
   correct answer should describe both the similarities and differences between the algorithms.

(5 pts) 3. When $h(n) = 0 \forall n$ A* search is equivalent to what other search algorithm?
   Solution: Uniform-cost search

(5 pts) 4. What is the primary advantage of hillclimbing over A*?
   Solution: Hillclimbing uses *much* less storage (just the current state) vs the exponential space required
   for A*.

(5 pts) 5. What is the primary advantage of A* over hillclimbing?
   Solution: A* finds an optimal solution, whereas hillclimbing can get stuck on a locally optimal solution.

(5 pts) 6. Describe two approaches to multi-agent coherence in multi-agent systems.
   Solution: There are many potential solutions to this problem, including but not limited to:
   - Multi-agent planning
   - Recognizing conflicts and negotiation
   - Modularity
   - A control agent that calls the shots
   - Each agent models other agents and reasons about its actions accordingly

(5 pts) 7. Describe two AI techniques used by Poki.
   Solution: There are many potential solutions to this problem, including but not limited to:
   - Neural networks to build opponent models
   - Breadth-based iterative deepening search algorithm
   - Simulation to build rule-based pre-flop betting strategy
8. In the following coordination game, three players must pick integers between 2 and 9 inclusive. If all three players choose the same number, then they receive that number as the payoff. If their numbers are not the same, the payoff for player i that chooses value x is \(-x\) (that is, each player must pay the amount he/she chose). What are the pure strategy Nash equilibria for this game?

Solution: The pure strategy Nash equilibria are all strategies where all players choose the same numbers that is: \{(1, 1, 1), (2, 2, 2), (3, 3, 3), (4, 4, 4), (5, 5, 5), (6, 6, 6), (7, 7, 7), (8, 8, 8), (9, 9, 9)\}

9. Heuristics. Suppose we have two heuristic functions \(h_1\) and \(h_2\), both of which are known to be admissible.

(a) What constraint must hold for \(h_1\) to dominate \(h_2\)?

Solution: \(\forall n, h_1(n) \geq h_2(n)\).

(b) Name and describe the additional property that \(h_1\) must possess to guarantee optimality when it is used with A* search.

Solution: Consistency: for all \(n\) and each successor of \(n\), \(n'\), \(h(n) \leq \text{cost}(n, a, n') + h(n')\).

(c) Which of the following heuristics are admissible?

\[
\begin{align*}
  h_3(n) &= \frac{(h_1(n) + h_2(n))}{2} \\
  h_4(n) &= \min(h_1(n), h_2(n)) \\
  h_5(n) &= \max(h_1(n), h_2(n)) \\
  h_6(n) &= w_1 h_1(n) + w_2 h_2(n), \text{ where } 0 \leq w_1, w_2 \leq 1
\end{align*}
\]

Solution: \(h_3, h_4,\) and \(h_5\).

10. The game of Split-Nim is played with a pile of tokens. To play this game, at each move, the player must divide one of the remaining piles of tokens into two nonempty piles of different sizes. Thus, 6 tokens may be divided into piles of 5/1, 4/2 but not 3/3. The player who can no longer make a move loses the game. Show using a game tree that when the original pile has seven tokens, player 2 can always win.

Solution: Player 1 can make three moves (6,1), (5,2), and (4,3).

- Case (6,1): Player two splits the 6 to create 3 piles, (4,2,1). Player 1 now has only one valid move, splitting the 4 into (3,1) to create (3,1,2,1). Player two splits the 3 (2,1,1,2,1), and there are no available moves for player 1. Player 2 wins.

- Case (5,2): Player two splits the 5 to create 3 piles, (4,1,2). Player one is now in the same predicament as in case (6,1).

- Case (4,3): Player two splits the 3 to create 3 piles, (4,2,1). Player 1 is in the same predicament as the previous two cases.

In all three cases, player two wins.

If there are nine tokens in the initial pile, player 1 can always win. Player 1 makes an initial split of (7,2). Now, the problem is reduced to the previous problem, player 2 must split the 7, and player 1 will always win.

11. Consider the following logical formula:

\[
(x_1 \lor x_2 \lor x_3) \land (\neg x_1 \lor \neg x_2 \lor \neg x_3) \land (\neg x_1 \lor x_2 \lor \neg x_3) \land (x_1 \lor \neg x_2 \lor \neg x_3) \land (\neg x_1 \lor \neg x_2 \lor x_3)
\]

This can be expressed as a constraint satisfaction problems where the variables \(V = \{x_1, x_2, x_3\}\), the domain \(D = \{TRUE, FALSE\}\). The constraints are expressed in the above formula (equation 1).

Solve the constraint reasoning problem by hand using backtracking, forward checking, and the MRV and least-constraining-value heuristics.

Solution: There was some ambiguity to this question. You could either solve the problem five times with the different methods, or combine all five methods as I intended. As a result, I gave credit either way.
To solve them using all methods at once created a table like so (which could be different based on alternate choices).

<table>
<thead>
<tr>
<th></th>
<th>x_1</th>
<th>x_2</th>
<th>x_3</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>TF</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td></td>
</tr>
</tbody>
</table>

All variables are equally constrained to begin with. So, choose $x_1$ without loss of generality. Either choice for $x_1$ leaves both values available for the other values, so choose $x_1 = T$. Then choose $x_2$ (both variables equal by MRV). Choose $x_2$ to be False as $x_2 = T$ leads to 0 remaining values for $x_3$. Assign $x_3 = F$.

(10pts) 12.

From Wikipedia, “Sudoku is a logic-based number-placement puzzle. The objective is to fill a 9 by 9 grid so that each column, each row, and each of the nine 3 by 3 boxes (also called blocks or regions) contains the digits from 1 to 9 only one time each. The puzzle setter provides a partially completed grid. An example puzzle is shown in Figure 1.”

Express sudoku (precisely enough to code up) as a constraint satisfaction problem (CSP).

Solution:
Variables = $X_{ij}$, where $i$ is a row from 1 to 9, and $j$ is a column from 1 to 9
Domain = 1,2,3,4,5,6,7,8,9
Constraints =

Given constraints (some variables will be preassigned by the problem, for instance $X_{11} = 5$ in the example)

Row constraints: For any two variables, $X_{ij}$ and $X_{st}$, if $i = s$ (same row) and $j \neq t$ (different column) then $X_{ij} \neq X_{st}$.

Column constraints: For any two variables, $X_{ij}$ and $X_{st}$, if $j = t$ (same column) and $i \neq s$ (different row) then $X_{ij} \neq X_{st}$.

Box constraints: For any two variables $X_{ij}$ and $X_{st}$, if $i \pmod{3} = s \pmod{3}$ and $j \pmod{3} = t \pmod{3}$ (same box) and ($i \neq s$ or $j \neq t$) (but they’re not the same variable), then $X_{ij} \neq X_{st}$.
Figure 1: Example sudoku puzzle.