Practice Midterm Exam Solutions

Duration: 75 minutes

Question 1: ____ out of ____ points
Question 2: ____ out of ____ points
Question 3: ____ out of ____ points
Question 4: ____ out of ____ points
Question 5: ____ out of ____ points

Total Score:

Grade:
Question 1: Tell the Truth

Tell whether each of the following statements is true or false by checking the appropriate box. Do not check any box if you do not know the right answer, because you will lose points for incorrect answers.

<table>
<thead>
<tr>
<th>Statement</th>
<th>True</th>
<th>False</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) Researchers in AI can be roughly divided into two groups: cognitive</td>
<td>[ ]</td>
<td>[X]</td>
</tr>
<tr>
<td>scientists and philosophers.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b) In genetic programming, the best-performing programs of a generation</td>
<td>[X]</td>
<td>[ ]</td>
</tr>
<tr>
<td>g will reach generation g+1 and have children that can be created through</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mechanisms such as mutation and crossover.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c) In genetic programming, a population of programs always compete</td>
<td>[ ]</td>
<td>[X]</td>
</tr>
<tr>
<td>against each other until only one program – the winner – remains.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>d) In best-first search, we sort the nodes of the OPEN list according to</td>
<td>[X]</td>
<td>[ ]</td>
</tr>
<tr>
<td>a function f'(n) that yields lower values for more “promising” nodes n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(that are assumed to be closer to the goal).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>e) If we had a perfect e(p) function, our program would only have to</td>
<td>[X]</td>
<td>[ ]</td>
</tr>
<tr>
<td>look ahead by one move in order to play perfectly.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>f) In the definition f'(n) = g'(n) + h'(n), we introduced the term g’(n)</td>
<td>[ ]</td>
<td>[X]</td>
</tr>
<tr>
<td>in order to create a greater variety of f'(n) values.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>g) Let us assume that we want to write an algorithm for playing a two-</td>
<td>[X]</td>
<td>[ ]</td>
</tr>
<tr>
<td>player game that is so simple that we can search the entire game tree.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Then we do not need to write an e(p) function but instead just a function</td>
<td></td>
<td></td>
</tr>
<tr>
<td>that detects whether one of the players has won.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>h) In machine evolution, children should be as different from their</td>
<td>[ ]</td>
<td>[X]</td>
</tr>
<tr>
<td>parent(s) as possible.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>i) The A* algorithm is not guaranteed to find the shortest path to a</td>
<td>[ ]</td>
<td>[X]</td>
</tr>
<tr>
<td>solution if we set h'(n) = 0 for all n.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>j) The A* algorithm is not guaranteed to find the shortest path to a</td>
<td>[X]</td>
<td>[ ]</td>
</tr>
<tr>
<td>solution if we use a pessimistic estimator for h'(n).</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Question 2: Knowledge Representation

Are the following arguments valid or not? Use resolution or resolution refutation to find out. Proceed in four steps:

- Extract the propositions from the argument and name them with single letters.
- Describe the hypotheses and the conclusion in propositional calculus.
- Convert these expressions into conjunctive normal form (CNF) suitable for resolution or resolution refutation (whichever method you decide to use).
- List the steps of the resolution (refutation) and state the result, that is, whether the argument is valid or not.

a) Whenever Peter goes downtown, he goes shopping. Whenever Peter goes shopping, he buys useless stuff. Whenever Peter or his sister Doris (or both) buy useless stuff, their mother gets angry. Their mother is not angry. Therefore, Peter did not go downtown.

(Well, I am not using single letters, I do not think that was a helpful rule to come up with)

D: Peter goes downtown
S: Peter goes shopping
PU: Peter buys useless stuff
DU: Doris buys useless stuff
A: Their mother gets angry

Hypotheses:

\[ D \rightarrow S \]
\[ S \rightarrow PU \]
\[ (PU \lor DU) \rightarrow A \]
\[ \neg A \]

Conclusion:

\[ \neg D \]

Appending negated conclusion and converting to CNF:

\[ (\neg D \lor S) \land (\neg S \lor PU) \land (\neg PU \lor A) \land (\neg DU \lor A) \land \neg A \land D \]

Resolution on D:

\[ S \land (\neg S \lor PU) \land (\neg PU \lor A) \land (\neg DU \lor A) \land \neg A \]
Resolution on S:

\[ \text{PU} \land (\neg \text{PU} \lor A) \land (\neg \text{DU} \lor A) \land \neg A \]

Resolution on A:

\[ \text{PU} \land \neg \text{PU} \land (\neg \text{DU} \lor A) \]

False! The argument is valid.

b) On every Sunday and Monday, Christine watches TV. On every Monday and Tuesday, James watches TV. Christine watches TV and James does not. Therefore, today must be Monday.

S: Today is Sunday
M: Today is Monday
T: Today is Tuesday
C: Today Christine watches TV
J: Today James watches TV

Hypotheses:

\[(S \lor M) \supset \text{C}\]
\[(M \lor T) \supset \text{J}\]
\[\text{C}\]
\[\lnot \text{J}\]

Conclusion:

\[\text{M}\]

Appending negated conclusion and converting to CNF:

\[ (\neg S \lor C) \land (\neg M \lor C) \land (\neg M \lor J) \land (\neg T \lor J) \land C \land \neg J \land \neg M \]

Resolution on J:

\[ (\neg S \lor C) \land (\neg M \lor C) \land \lnot M \land (\neg T \lor J) \land C \land \lnot M \]

No further resolution possible! The argument is not valid.
Question 3: Search Strategies

(a) What is the advantage of breadth-first search over depth-first search?

Breadth-first search is guaranteed to find the best (closest to the root) solution, whereas depth-first search may go unnecessarily deep into the tree and find a suboptimal solution.

(b) What is the advantage of depth-first search over breadth-first search?

Depth-first search has a space (memory) complexity of only $O(n)$ for a maximum search depth of $n$, because only the current path needs to be stored at any time. Breadth-first search, on the other hand, actually builds the entire search tree up to the level of the solution, so its space complexity is $O(b^n)$ for branching factor $b$ and solution level $n$.

c) Describe the strategy of iterative deepening and its advantages over breadth-first and depth-first search.

Iterative deepening is an iterated, depth-limited variant of the depth-first search algorithm. In each iteration, depth-first search is carried out with a maximum search depth of 0 (only the root is tested). If no solution is detected, the maximum search depth is increased to 1 and another search is started. With each iteration, the maximum search depth is increased by one, until a solution is detected.

The advantage of iterative deepening over breadth-first search is its low space complexity $O(n)$ for a solution located at level $n$. At the same time, A* is still guaranteed to find the shortest path to a solution. Its time complexity is slightly worse than that of breadth-first search, but for deep trees with a large branching factor, this difference is insignificant. The advantage of iterative deepening over depth-first search is that iterative deepening is guaranteed to find the best solution (i.e., the shortest path) without ever going unnecessarily deep into the tree.
Question 4: Genetic Programming

Draw two trees that represent LISP programs as they might occur in the genetic programming algorithm for the wall-following robot that we discussed in class. These could be any syntactically correct trees with about 8 to 15 nodes each. Draw a third tree that resulted from the first two through crossover mutation. Explain how the crossover operation was applied in your example.

From the first (mother) program, the marked subtree with the root OR was deleted and replaced with the subtree with the marked root AND from the second (father) program.
Question 5 (Bonus): Convincing Arguments

Show that the following two arguments are valid. In (b), use quantifiers.

In each case, proceed in three steps:

- Extract the propositions from the argument and name them with single letters.
- Describe the hypotheses and the conclusion in propositional calculus.
- Use the step-by-step method we discussed in class to show that the conclusion follows from the hypotheses.

a) If Peter attends CS320L, then he will fall asleep. If Peter falls asleep, then he will have sweet dreams. Peter does not have sweet dreams. Therefore, Peter did not attend CS320L.

A: Peter attends CS 320L.
F: Peter falls asleep.
D: Peter has sweet dreams.

Hypotheses:

A \supset F
F \supset D
\neg D

Conclusion:

\neg A

Step 1: A \supset F (Hypothesis)
Step 2: F \supset D (Hypothesis)
Step 3: A \supset D (R.I. Steps 1 + 2)
Step 4: \neg D (Hypothesis)
Step 5: \neg A (R.I. Steps 3 + 4)

The argument is valid.

b) All UMB courses are hard or boring. CS470 is a UMB course, and it is not hard. Therefore, CS470 is boring.

U(x): x is a UMB course
H(x): x is hard
B(x): x is boring
Hypotheses:

\[ \forall x \ (U(x) \supset H(x) \lor B(x)) \]
\[ U(CS470) \]
\[ \neg H(CS470) \]

Conclusion:

B(CS470)

Step 1: \[ \forall x \ (U(x) \supset H(x) \lor B(x)) \] (Hypothesis)
Step 2: \[ U(CS470) \supset H(CS470) \lor B(CS470) \] (R.I. Step 1)
Step 3: \[ U(CS470) \] (Hypothesis)
Step 4: \[ H(CS470) \lor B(CS470) \] (R.I. Steps 2 + 3)
Step 5: \[ \neg H(CS470) \] (Hypothesis)
Step 6: \[ B(CS470) \] (R.I. Steps 4 + 5)

The argument is valid.