Midterm Examination

COMP4475-Topics in AI (Solution)

NAME: (Given) ____________ (Surname) ____________

Student Number: ____________

Instructions:

- Read all questions carefully;
- York exam rules are in effect;
- No electronic devices allowed;
- There are 12 pages including the cover page;
- Time limit: 120 min.

Marks: ______ + ______ + ______ + ______ + ______ = ______

Section One (10 marks)

An 8-puzzle game consists of 8 sliding tiles, numbered by digits from 1 to 8 and arranged in a 3x3 array of nine cells. A configuration in the puzzle refers to some specific arrangement of the tiles in the array, where each digit is arranged into a different cell. One of the cell is empty (represented by a "*") and any adjacent tile can be moved into the empty cell. An example below

```
  c   | 1 | 2 | 3
  b   | 4 | * | 5
  a   | 6 | 7 | 8
------------------
 Y/X    a   b   c
```

We use a 3-tuple (number, X, Y) to represent the position of a digit, where number is the actual digit, and (X,Y) is the coordinate value of the digit in the current configuration (e.g., in the configuration above, we have (5,c,b), meaning the digit 5 is at the position of (c,b)). Hence the configuration in the example can be represented as: [(1,a,c), (2,b,c), (3,c,c), (4,a,b), (5,c,b), (6,a,a), (7,b,a), (8,c,a)]. Any move of an adjacent tile into the empty cell moves the current configuration into the one adjacent to it.

1. **5 marks** How many different configurations for this game?
   soln: $9! = 1 \times 2 \times \ldots 9$

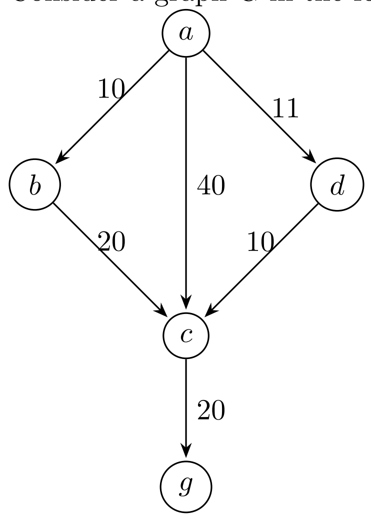
2. **5 marks** List all the adjacent configurations from the example above.

soln:

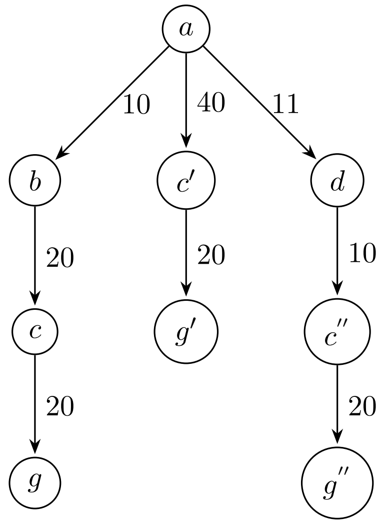
```
move 2 to (b, b) [(1,a,c), (2,b,b), (3,c,c), (4,a,b), (5,c,b), (6,a,a), (7,b,a), (8,c,a)]
move 4 to (b, b) [(1,a,c), (2,b,c), (3,c,c), (4,b,b), (5,c,b), (6,a,a), (7,b,a), (8,c,a)]
move 5 to (b, b) [(1,a,c), (2,b,c), (3,c,c), (4,a,b), (5,b,b), (6,a,a), (7,b,a), (8,c,a)]
move 7 to (b, b) [(1,a,c), (2,b,c), (3,c,c), (4,a,b), (5,c,b), (6,a,a), (7,b,b), (8,c,a)]
```

Section Two (40 marks)

Consider a graph G in the following



If you use a search tree for your answers, you will HAVE TO use the following one, which is rooted at a, and the goal is g.



1. (**5 marks**) Find a path from a to g in the graph G using the search strategy of depth-first search. Is the returned solution path an optimal one? Give your explanation and remarks on "why-optimal" or "why-non-optimal".

   soln: Using a stack according to alphabetical order for push()/pop(), the solution should be $a \to d \to c \to g$, which, by coincidence, is optimal. if answer is $a \to b \to c \to g$, fine. but in this case the path is non-optimal. DFS does not guarantee path optimality, as it ignore the cost of edges.

2. (**5 marks**) Find a path from a to g in the graph G using the search strategy of breadth-first search. Is the returned solution path an optimal one? Give your explanation and remarks on "why-optimal" or "why-non-optimal".

   soln: Using a queue according to alphabetical order for enqueue()/dequeue(), the solution should be $a \to c \to g$, which, is non-optimal. BrFS (Breadth First Search) does not guarantee path optimality, as it ignore the cost of edges.

3. (**6 marks**) Find a path from a to g in the graph G using the search strategy of least-cost first search. Is the returned solution path an optimal one? Give your explanation and remarks on "why-optimal" or "why-non-optimal".

   soln: Using a priority queue according the function g value, for enqueue()/dequeue(), the solution should be $a \to d \to c \to g$, which is optimal with path-cost 41. LCFS is simply Dijkstra's algorithm for shortest path in a weighted graph, so optimality is guaranteed.

4. (**6 marks**) Find a path from a to g in the graph G using the search strategy of best-first search. The heuristics for these nodes are:

   $$h(a, 25); h(b, 43); h(c, 5); h(d, 64); h(g, 0).$$

   Is the returned solution path an optimal one? Give your explanation and remarks on "why-optimal" or "why-non-optimal".

   soln: Using a priority queue according the function h value, for enqueue()/dequeue(), the solution should be $a \to c \to g$, which is non-optimal with path cost 60. BFS is non-optimal, as searching ignore path cost, and is based on heuristics, which can be misleading.

5. (**6 marks**) Suppose AGAIN the heuristics of

   $$h(a, 25); h(b, 43); h(c, 5); h(d, 64); h(g, 0).$$

Find a path from a to g in the graph G using the search strategy of A* search. Is the returned solution path an optimal one? Give your explanation and remarks on "why-optimal" or "why-non-optimal".

soln: Using a priority queue according the function $f = g + h$ value, for enqueue()/dequeue(), the solution should be $a \to b \to c \to g$, which is non-optimal with path cost 50. A* in this case non-optimal, as heuristics values do not satisfies the admissibility requirement (e.g., $h(b, 43)$ is greater than the actual path-cost from b to g.

6. (**6 marks**) Suppose the heuristics is now UPDATED to

$$h(a, 5); h(b, 9); h(c, 5); h(d, 29); h(g, 0).$$

Find a path from a to g in the graph G using the search strategy of A* search again. Is the returned solution path an optimal one? Give your explanation and remarks on "why-optimal" or "why-non-optimal".

soln: Using a priority queue according the function $f = g + h$ value, for enqueue()/dequeue(), the solution should be $a \to d \to c \to g$, which is non-optimal with path cost 41. A* in this case optimal, as heuristics values this time are admissible.

7. (**6 marks**) Suppose the heuristics is FURTHER UPDATED to

$$h(a, 5); h(b, 8); h(c, 5); h(d, 9); h(g, 0).$$

Find a path from a to g in the graph G using the search strategy of A* search again. Is the returned solution path an optimal one? Give your explanation and remarks on "why-optimal" or "why-non-optimal". What would be your additional comments in terms of the solution path finding under this heuristics?

soln: Using a priority queue according the function $f = g + h$ value, for enqueue()/dequeue(), the solution should be $a \to d \to c \to g$, which is non-optimal with path cost 41. A* in this case optimal, as heuristics values this time are admissible. Not only admissible, but the monotone restriction are satisfied. Hence the procedure is also optimized, in the sense that MPC (multiple-path checking) can be applied: Namely, 2nd time dequeued node can be safely ignored, so the searching procedure can actually be speeded up.

Section Three (20 marks)

Suppose propositional sentences and knowledge-base here. We have a knowledge-base called KB. KBPrime is defined as an union of KB and negation of a particular propositional predicate P. Answer true/false to the following questions.

1. (**5 marks**) If $KB \models P$, then $KB \vdash P$. That is, if KB entails P, then KB derives (using resolution) P.
soln: false, as resolution is not complete

2. (**5 marks**) If $KB \vdash P$, then $KB \models P$.
soln: true, as resolution is sound.

3. (**5 marks**) If $KBPrime \vdash [\,]$, then KBPrime is not satisfiable.
soln: true, as resolution is sound.

4. (**5 marks**) If KBPrime is not satisfiable, then $KBPrime \vdash [\,]$.
soln: true, as resolution is refutation complete.

Section Four (20 marks)

A propositional knowledge-base KB is given below:

1. $(\neg P \wedge \neg Q) \rightarrow R$
2. $R \rightarrow S$
3. $\neg S$
4. $P \rightarrow \neg U$
5. U

1. (**15 marks**) Explain in English (in Steps!) that $KB \models Q$.
   soln:

   - From Sentences (4) and (5), we know that P can not be true (i.e., $KB \models \neg P$). Reasoning: If P is true, then U is false (from Sentence (4)), but U is actually true (Sentence (5)).
   - Similarly from (2) and (3), we know that R can not be true (i.e., $KB \models \neg R$). Reasoning: If R is true, then S is true (from Sentence (2)), but S is actually false, (Sentence (3)).
   - So Q must be true. If Q is false, then from (1), we know R must be true, but we now know R is false.

2. (**15 marks**) Show resolution steps that $KB \vdash Q$. You should first convert the sentences in the KB into clauses.
   soln: The KB in clausal form:

   (a) $(\neg P \wedge \neg Q) \rightarrow R$ In clausal form C1: $[P, Q, R]$.
   (b) $R \rightarrow S$ In clausal form C2: $[\neg R, S]$.
   (c) $\neg S$ In clausal form C3: $[\neg S]$.
   (d) $P \rightarrow \neg U$ In clausal form C4: $[\neg P, \neg U]$.
   (e) U In clausal form C5: $[U]$.

   First negate the goal to obtain the clause C6: $\neg Q$. And then:

   (a) resolve C1 with C3 to obtain C7: $[\neg R]$.
   (b) resolve C4 with C5 to obtain C8: $[\neg P]$.
   (c) resolve C8 with C1 to obtain C9: $[Q, R]$.
   (d) resolve C7 with C9 to obtain C10: $[Q]$.
   (e) resolve C6 with C10 to obtain the empty clause: $[\,]$.

Section Five (10 marks)

Consider a First-Order Logic (FOL) language L with two unary predicate symbols P and Q, one unary function f, and two constants a and b. The following KB has been specified in this language

1. $\forall x(Q(x) \rightarrow P(x))$ (which is logically equivalent to $\forall x(\neg Q(x) \vee P(x))$).
2. $P(b)$
3. $Q(b)$
4. $Q(f(a))$
5. $P(f(b))$

Consider an interpretation of $\mathcal{I} = \langle D, I \rangle$, where $D = \{d1, d2, d3\}$, and d1 is denoted by a, d2 is denoted b. For the function f, I(f,d1)=d2, I(f,d2)=d3, I(f,d3)=d1. and $I(P) = \{d1, d2\}$, $I(P) = \{d1, d2\}$. Is this $\mathcal{I}$ a model of L?

soln: $I(P) = \{d1, d2\}$ is repeated, but it was assumed to be $I(Q) = \{d1, d2\}$. Because of this, full mark is given for all. However regardless any assumption to be given for the predicate Q in the interpretation $\mathcal{I}$: We have that $P(f(b))$ is not true. As $I(b) = d2$, thus $I(f, d2) = d3$, but $d3 \notin I(P)$. Hence $\mathcal{I}$ can not be a model of the KB.