Midterm Examination

CS540: Introduction to Artificial Intelligence

October 27, 2010

LAST NAME: ___________SOLUTION_________________

FIRST NAME: ____

SECTION (1=Zhu, 2=Dyer): ______________________________

Problem	Score	Max Score
1	____________	15
2	____________	16
3	____________	7
4	____________	15
5	____________	12
6	____________	10
7	____________	15
8	____________	10
Total	____________	100

Question 1. [15] State Space Search

You have two jugs, measuring 3 gallons and 4 gallons, and a water faucet. You can fill the jugs up or empty them out from one to another or onto the ground. You need to measure out exactly 2 gallons.

(a) [5] Let state (*a*, *b*) mean the amount of water in the 3 and 4 gallon jugs, respectively. Define all successors of this state.

(3,b); if a<3

(a,4); if b<4

(0,b); if a>0

(a,0); if b>0

(3, b-(3-a)=a+b-3); if a<3 and a+b>3

(a+b, 0); if a<3 and b>0 and a+b<=3

The two above can be combined into (min(a+b,3), max(a+b-3,0))

(a-(4-b)=a+b-4, 4); if b<4 and a+b>4

(0, a+b); if b<4 and a>0 and a+b<=4

The two above can be combined into (max(a+b-4,0), min(a+b,4))

(b) [5] Write down a solution path from (3, 4) to a goal state. You do not need to show how you find this path.

(3,4)-(3,0)-(0,3)-(3,3)-(2,4) or

(3,4)-(0,4)-(3,1)-(0,1)-(1,0)-(1,4)-(3,2)

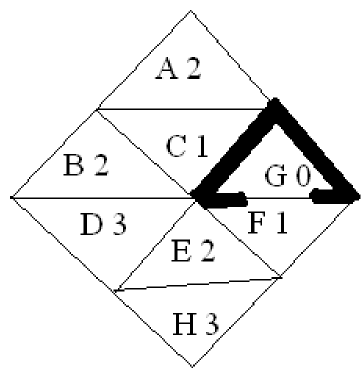
(c) [5] Let the cost of every edge be 1. Consider the function *h*(*a*, *b*) = |*a* - *b*|. Is *h* an admissible heuristic? Justify your answer.

No. On many goal states this h is not zero. On the first solution path, (0,3) has actual cost 2 (i.e. two steps away from goal) but h(0,3)=3>2. Similarly, on the 2nd solution path, (1,4) has actual cost 1 but h(1,4)=3>1.

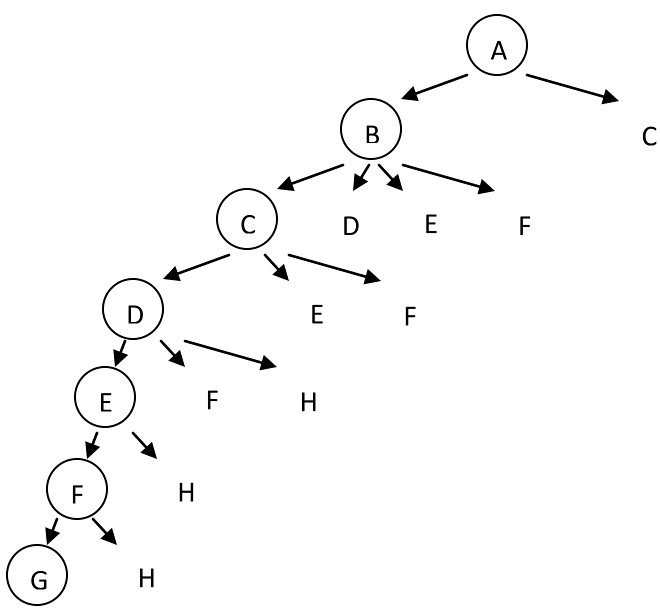
Question 2. [16] Uninformed and Informed Search

Consider the following path-finding problem. One can move from one small triangle to another if they share a vertex (e.g., *A* can go to *B* and *C*). However, the goal *G* can only be accessed from *F*. The number after the letter is the heuristic function value for that state. The actual cost of each move is as follows:

- A move down one level (e.g. *A* → *C or B* → *E*) costs 1
- A move sideways on the same level (e.g. *C* → *B* or *E* → *F*) costs 2
- A move up one level (e.g. *B* → *A* or *C* → *A*) costs 3



(a) [8] Perform **Depth-First Search**, starting from *A*, using <u>path-checking</u> to avoid repeated states if they occur on the path back to the root in the search tree. Expand successors in alphabetical order. Show your search tree, and *circle* states that are expanded. What is the *cost* of your solution path?



Cost=11

(b) [8] Perform **A* Search**, starting from *A*. Break ties alphabetically. Show the *expanded states* and the *priority queue* contents at each step. What is the *cost* of your solution path?

```
Expanded states              priority queue
-                            A(0+2)
A(0+2)                       B(1+2), C(1+1)
C(1+1)                       B(1+2), D(2+3), E(2+2), F(2+1)
B(1+2)                       D(2+3), E(2+2), F(2+1)
F(2+1)                       D(2+3), E(2+2), G(5+0), H(3+3)
E(2+2)                       D(2+3), G(5+0), H(3+3)
D(2+3)                       G(5+0), H(3+3)
G(5+0)                       H(3+3)

Path: A-C-F-G, cost=5
```

Question 3. [7] Simulated Annealing

Assume that simulated annealing starts from a state *s* at the center of a large plateau. That is, the values of all states on the plateau are exactly the same.

(a) [4] Say in the first step the random neighbor we picked is *t*, which of course has the same value as *s*. What will simulated annealing do? Justify your answer.

```
Since P(s->t) = exp(-|f(s)-f(t)|/T) and f(s)=f(t), we see that with
probability 1 we will move to t.
```

(b) [3] To encourage the search to escape this plateau as quickly as possible, what should we do with the temperature *T*? Justify your answer.

```
Since the above probability P(s->t) does not depend on T (because
it is exp(0)), there is nothing you can do about temperature T to
encourage an escape.
```

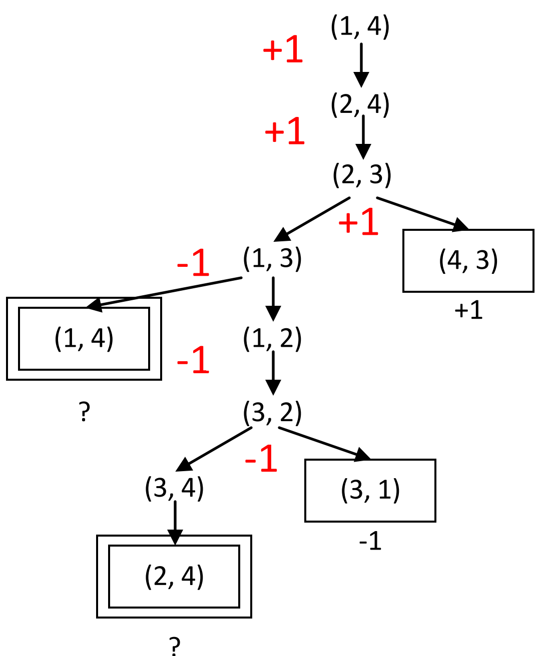
Question 4. [15] Game Playing

Consider the following game where *A* moves first. The two players take turns. Each player must move to an open adjacent space in either direction. If the opponent occupies an adjacent space, then a player may jump over the opponent to the next open space, if there is one (for example, if *A* is on 1 and *B* is on 2, then *A* may jump to 3). The game ends when *A* reaches 4 (the value of this game to *A* is +1), or *B* reaches 1 (the value of this game to *A* is -1).

A			B
1	2	3	4

(a) [8] Draw the complete game tree, using the following conventions:

- Write each state as (*a*, *b*) where *a* and *b* denote the locations of *A* and *B*, respectively.
- Put each *terminal state* in a square box, and write its game value below the box.
- Put *loop states* (i.e., states that already appear on the path back to the root) in double square boxes. Since it is not clear how to assign values to loop states, write a "?" below these double boxes.



(b) [4] Compute the *minimax value* at every internal node, showing them in the search tree as well. Explain how you handled the "?" values and why.

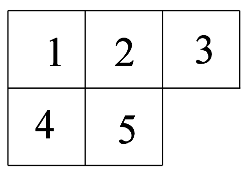
See red numbers above. The ? value can be obtained by copying the value of the (2,4) node next to the root. Or: it doesn't matter because B will select the right branch at (3,2).

(c) [3] Explain why the standard minimax algorithm would fail on this game tree.

```
The game tree is infinite.  Because minimax is depth first search, it
will not terminate.
```

Question 5. [12] Constraint Satisfaction

Consider the problem of assigning colors to the five squares on board below such that horizontally adjacent and vertically adjacent squares do not have the same color. Assume there are possible two colors, red (*R*) and black (*B*). Formulated as a constraint satisfaction problem, there are five variables (the squares) and two possible values (*R*, *B*) for each variable.



(a) [3] If initially every variable has both possible values and we then assign variable 1 to have value *R*, what is the result of the *Forward Checking algorithm*?

```
Forward checking propagates constraints from the current
assigned variable to all adjacent unassigned variables,
which in this case are variables 2 and 4.  This results in
the following domains for the variables:  1 = {R}, 2 = {B},
3 = {R, B}, 4 = {B}, 5 = {R, B}
```

(b) [4] If initially every variable has both possible values and the *Arc Consistency algorithm* is run, what are the resulting domains for each of the variables?

```
None of the domains change, with each variable still have
both possible values, {R, B}.
```

(c) [5] If initially every variable has both possible values except variable 5 has only value *B*, what is the result of the *Arc Consistency algorithm*?

```
1 = {B}, 2 = {R}, 3 = {B}, 4 = {R}, 5 = {B}
```

Question 6. [10] Propositional Logic

(a) [2] True or false: $(A \Leftrightarrow B) \models (\neg A \vee B)$

```
True
```

(b) [4] Is the PL sentence $((P \Rightarrow Q) \wedge Q) \Rightarrow P$ valid, unsatisfiable or satisfiable? Justify your answer.

```
Satisfiable.  When P=T and Q=T the sentence is true, but
when P=F and Q=T the sentence is false.
```

(c) [4] Given two arbitrary sentences α and β in PL, $\alpha \models \beta$ if and only if

i. [2] the sentence $\alpha \Rightarrow \beta$ is ________valid________________________
(one word answer)

ii. [2] the sentence $\alpha \wedge \neg\beta$ is ________unsatisfiable_______________
(one word answer)

Question 7. [15] Deductive Inference in Propositional Logic

Consider the following *KB* containing 4 sentences in PL:

$P \Rightarrow (R \vee S)$, $\neg P \Rightarrow (R \vee S)$, $\neg S$, $(R \vee U) \Rightarrow Q$

(a) [3] Can these 4 sentences be expressed as a set of Horn clauses? If yes, give them; if no, explain why not.

> No because the first sentence has two literals on the right hand side, meaning that the CNF has two positive literals in it.

(b) [4] Convert the given sentences into conjunctive normal form (CNF) and show the result as a set of clauses.

> $\neg P \vee R \vee S$, $P \vee R \vee S$, $\neg S$, $\neg R \vee Q$, $\neg U \vee Q$

(c) [8] Prove the query sentence *Q* is entailed by *KB* using the Resolution algorithm.

1. $\neg Q$	negation of query
2. $\neg P \vee R \vee S$	KB
3. $P \vee R \vee S$	KB
4. $\neg S$	KB
5. $\neg R \vee Q$	KB
6. $\neg U \vee Q$	KB
7. $\neg R$	RR(1,5)
8. $\neg P \vee S$	RR(2,7)
9. $\neg P$	RR(4,8)
10. $R \vee S$	RR(3,9)
11. R	RR(4,10)
12. False	RR(7,11)

Question 8. [10] Representation in First-Order Logic

For each of the following sentences in English, is the accompanying sentence in first-order logic a good translation? If yes, answer yes. If no, explain why not.

(a) [5] No two UW students have the same ID number.

$\neg\exists p,q,n\ (UWStudent(p) \wedge UWStudent(q) \wedge \neg(\mathrm{p} = \mathrm{q})) \Rightarrow (IDNum(p, n) \wedge IDNum(q, n))$

This is NOT correct because it uses $\Rightarrow$ instead of $\wedge$ when quantified by $\exists$. It should be:

$\neg\exists p,q,n\ UWStudent(p) \wedge UWStudent(q) \wedge \neg(\mathrm{p} = \mathrm{q}) \wedge IDNum(p, n) \wedge IDNum(q, n)$

(b) [5] All UW students except math majors like CS majors.

$\forall p,q\ (UWStudent(p) \wedge \neg MathMajor(p)) \Rightarrow (UWStudent(q) \wedge CSMajor(q) \wedge likes(p, q))$

This is NOT correct because it says that if there is at least one UW student who is not a math major, then everyone has to be a UW CS major. A corrected version is:

$\forall p,q\ (UWStudent(p) \wedge \neg MathMajor(p) \wedge UWStudent(q) \wedge CSMajor(q)) \Rightarrow likes(p, q)$