Midterm Examination

CSCI 561: Artificial Intelligence

October 10, 2002

Instructions:

1. Date: **10/10/2002 from 11:00am – 12:20 pm**
2. Maximum credits/points for this midterm: 100 points (corresponding to 35% of overall grade).
3. 4 questions.
4. Credits/points for each question is indicated in the brackets [ ] before the question.
5. **No books** (or other material) are allowed.
6. Attach extra sheets (available upon request) if required (write full name on each extra sheet).
7. **Write down name and student ID on every sheet.**
8. No questions during the exam.
9. When finished raise completed exam sheets until approached by proctor.
10. **Adhere to the Academic Integrity code.**

Midterm Examination

CSCI 561: Artificial Intelligence

October 10, 2002

Student ID: ______________________________

Last Name: ______________________________

First Name: ______________________________

Problem	Score	Max score (%)
1.	________	20
2.	________	25
3.	________	25
4.	________	30
Total score		100%

1. [20%] Search Methods.

(a) [12%] The major four criteria for evaluating search methods are: time complexity, space complexity, optimality, and completeness. Using one or more of these criteria, attempt to justify the following statements:

(i) Iterative deepening search is preferred over breadth-first search.

Space complexity

(ii) Bidirectional search is preferred over breadth-first search

time complexity and space complexity

(iii) The A* algorithm is preferred over the hill-climbing method

completeness: (A* search does not fall into local minimum)

(b) [2%] Why worry about the complexity of a search algorithm?

because we have limited amount of time and memory
exponential search problem solve only small size problem

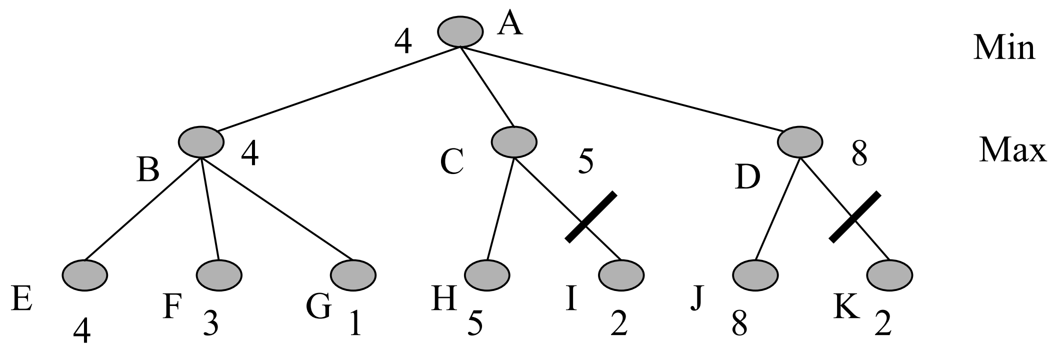
(c) [2%] Give at least two ways by which loop detection can be implemented (they may not need to be optimal).

do not return to the state you just came from
do not create paths with cycle in them
do not generate any state that was ever generated before

(d) [4%] Briefly describe how simulated annealing works. In particular, precise how the algorithm behaves at very high temperatures, and how it behaves at very low temperatures.

In simulated annealing, instead of picking best move, it picks random move. If the move actually improves the situation, it always proceeds. Otherwise, SA algorithm makes the move with some probability. The probability decreases exponentially with the badness of the move. At higher value of T, bad moves are more likely to be allowed. As T tends to zero, they become more and more unlikely until algorithm behaves like hill-climbing

2. [25%] Game Playing.
Consider the following game tree in which the evaluation function values are shown below each leaf node. Assume that the root node corresponds to the minimizing player. Assume that the search always visits children left-to-right.



(a) [5%] Compute the backed-up values computed by the minimax algorithm. Show your answer by writing values at the appropriate nodes in the above tree.

See figure

(b) [10%] Which nodes will not be examined by the alpha-beta pruning algorithm?

See figure

(c) [10%] In general (not just for the above tree), if the search always visits children right-to-left instead of left-to-right,

(i) The minimax value computed at the root will be changed. (just write down yes or no)

No

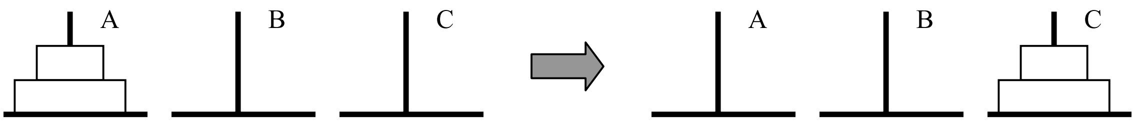
(ii) The number of nodes pruned will be changed. (just write down yes or no)

Yes

3. [30%] Problem Solving.

We are now trying to solve the miniature of Tower of Hanoi problem. There are 3 towers (A, B, C), and 2 disks (small one and large one). The purpose of this problem is to move both disks from the tower A to tower C (as illustrated in the figure below), subject to following two conditions:

- You can move only one disk at a time
- You cannot put the large disk on top of the small disk.



The possible state can be denoted as follows:

a:(b c) where a is the state number, b is the tower number for the large disk, and c is the tower number for the small disk. For example 3:(1 3) implies that the large disk is on tower 1 and the small disk is on tower 3 in state 3.

If we use this notation, the following nine states are possible.

1:(1 1), 2:(1:2), 3:(1 3), 4:(2 1), 5:(2 2), 6:(2 3), 7:(3 1), 8:(3 2), 9:(3 3)

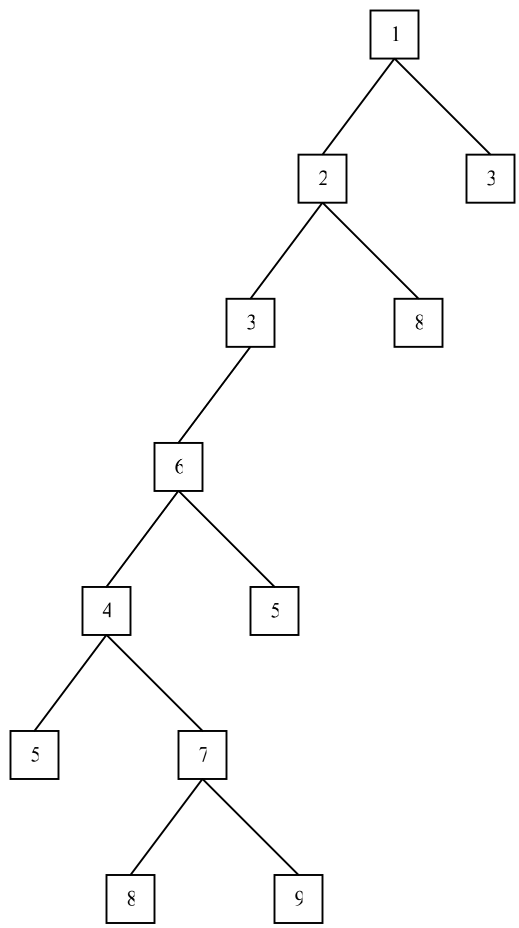
(a) [5%] Which ones are the initial state and goal state?

Initial state: 1
Goal state: 9

(b) [10%] Enumerate all possible moves between states. For example, 1->2, 3 means that it is possible to make a transition from state 1 to state 2 or state 3.

1-> 2,3
2-> 1,3,8
3-> 1,2,6
4-> 5,6,7
5-> 4,6
6-> 3,4,5
7-> 4,8,9
8-> 2,7,9
9-> 7,8

(c) [10%] Find a solution using depth-first search. Show the search tree. Assume that cycles are detected and eliminated by never expanding a node containing a state that is repeated on the path back to the root. In addition, if a search method needs to break a tie, expand the state with the small number first.



4. [30%] Propositional Logic.

(a) [10%] Define the following precisely

(i) Validity of a sentence

A sentence is valid if and only if it is true under all possible interpretation in all possible worlds

(ii) Satisfiability of a sentence

A sentence is satisfiable if and only if there is some interpretation in some world for which it is true

(b) [20%] Assume that the following sentences are in our Knowledge Base (~A denotes the negation of A):

~A
~A => ~B and ~C
~B => ~D and ~E
~C => D or E or F

Using the inference rules that we studied in class for propositional logic, prove that "F is true". When you derive F, specify exactly the sequence of inference rules that you used.

~A 1
~A => ~B and ~C 2
~B => ~D and ~E 3
~C => D or E or F 4

Applying Modus Ponens with 1 and 2, we obtain
~B and ~C 5
Applying And-Elimination to 5, we obtain
~B 6
~C 7
Applying Modus Ponens with 6 and 3, we obtain
~D and ~E 8
Applying And-Elimination to 8, we obtain
~D 9
~E 10
Applying Modus Ponens with 7 and 4, we obtain
D or E or F 11
Applying unit resolution with 9 with 11, we obtain
E or F 12
Applying unit resolution with 10 with 12, we obtain
F 13