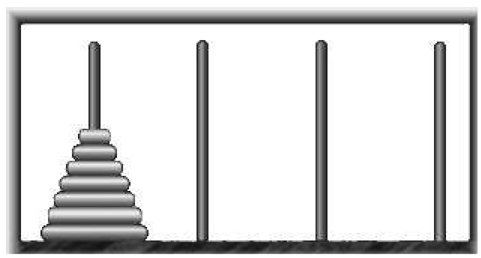
CS 371/671 Mid-term

Answer 4 out of 5 questions. Each question is worth 10 points.

1. Agents

- a. Define the following terms - Performance measure, Environment, Actuators and Sensors. Consider a taxi domain where the taxi picks up passengers and drops them off at their target locations. Give the PEAS description for this problem (4 points).
  Performance measure defines the criterion for success. The environment refers to the external world in which the agent is acting in. The agent perceives the environment through percepts and acts in the environment through its actuators.
  For the taxi domain, the performance measure could be the safety, legal, comfortable trip, the speed and profits. The environment is the set of roads, other traffic, pedestrians, customers, signals etc. The acutators are steering, accelarator, brake, signal, horn etc. The sensors are camera, speedometer, GPS, odometer etc.

- b. The four-peg version of the Tower of Hanoi puzzle consists of four pegs mounted on a board and n disks of various sizes with holes in their centers (see below). If a disk is on a peg, only a disk of smaller diameter d can be placed on top of it. Given all the disks properly stacked on the leftmost peg as presented below, the problem is to transfer the disks to the rightmost peg by moving one disk at a time.

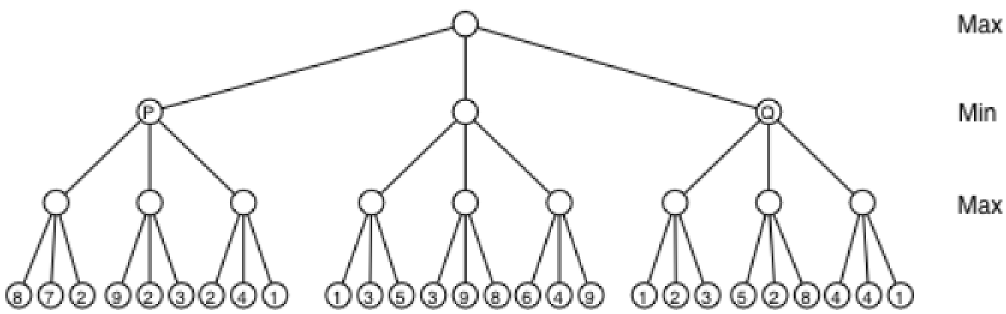


- 1. Define a state representation (multiple representations are possible) (2 points).
  Each disk is represented by D_i, where D_i is smaller than D_i for all $i < j$. The state can be represented by four stacks, $S_1 ... S_4$, representing the pegs numbered left to right. Each stack contains the disks that are on that peg, such that the top-most (front) element of the stack represents the top disk.
- 2. Give the initial and goal states (2 point).
  Initial State: $S_1 : \{D_1, D_2, ...D_n\} S2 : \{\} S3 : \{\} S4 : \{\}$
  Final State: $S1 : \{\} S2 : \{\} S3 : \{\} S4 : \{D_1, D_2, ...D_n\}$
- 3. Define the successor state function for this representation i.e., show how the next state can be computed given the current state in your representation (1 point).Disk D_i can be moved from S_i to S_j if:
  - D_i is the topmost disk on S_i
  - S_j is either empty, or the size of the topmost disk of S_j , D_{jt}, is greater than i
- 4. Define the cost function for the above successor function (1 point).
  Uniform cost

2. Search Answer True or false to the following questions and explain briefly your justification (each is worth 1 point).

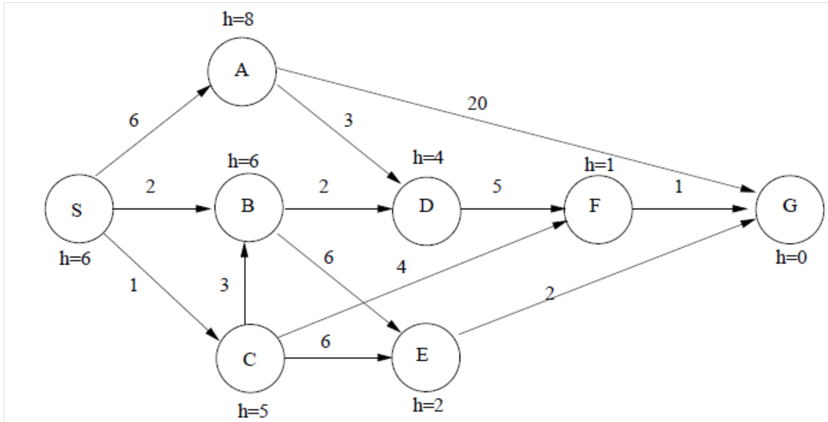
- Depth first search will find an optimal path with respect to the cost of the path.
  False
- Depth first search will find an optimal path with respect to the number of steps in the path.
  False
- Breadth first search will find an optimal path with respect to the cost of the path.
  Both answers were accepted. BFS is not guaranteed to find the optimal-cost solution because it does not take cost into account. However, in this particular problem, the BFS solution happens to be the optimal-cost path.
- Breadth first search will find an optimal path with respect to the number of steps in the path.True
- An inadmissible heuristic in A^* search can change the optimality of the search but not the completeness.
  True

2.1 Adversarial Search The figure below is the game tree of a two player game. The first player is the maximizer and the second player is the minimizer. Use the tree to answer the following questions.



- a. What is the final value of this game? (1 point)
  5
- b. What is the (beta) value at P? (1 point)
  4
- c. Will any of the nodes be pruned in the sub-tree below Q? If so, what are they? (2 points)
  Yes. If we work out the algorithm on the leftmost subtree of Q, we see that it returns a value of 3. Beta at Q will then become 3, and 3 is less than the alpha value at Q, so Q will immediately return a value to its parent. This means that the other two subtrees of Q are pruned.
- d. What is the value at Q? (1 point)
  3

3. Search Consider the map shown below with start state S and goal state G. The transition costs are next to the edges, and the heuristic values are next to the states.



- a. What is the final path of an uniform-cost search? (2 points)
  $S \rightarrow C \rightarrow F \rightarrow G$
- b. If we use Depth First Search, and it terminates as soon as it reaches the goal state, what is the final path for this DFS search? If a node has multiple successors, then we always expand the successors in increasing alphabetical order.(2 points)
  $S \rightarrow A \rightarrow D \rightarrow F \rightarrow G$
- c. If we use A* search, what is the final path for this A* search? (2 points)
  $S \rightarrow C \rightarrow F \rightarrow G$
- d. Is the heuristic function in this example admissible? (2 points)
  Yes, since $h(s) \leq h^*(s)$ for all s
- e. What is the final path of a best-first search? (2 points)
  $S \rightarrow C \rightarrow B \rightarrow D \rightarrow F \rightarrow G$

4. Propositional Logic State whether the following are true or False (5 points)

- $False \models True$ – True, because False has no models and hence entails every sentence and because True is true in all models and hence is entailed by every sentence.
- $(A \wedge B) \models (A \leftrightarrow B)$ – True because the left-hand side has exactly one model that is one of the two models of the right-hand side.
- $(A \vee B) \wedge \neg(A \rightarrow B)$ is satisfiable – True
- $(P \rightarrow Q) \wedge (P \rightarrow \neg Q)$ is valid – False
- $(A \leftrightarrow B) \wedge (\neg A \vee B)$ is satisfiable – True

4.1 Consider the following sentence,

$$[(Food \rightarrow Party) \vee (Drinks \rightarrow Party)] \rightarrow [(Food \wedge Drinks) \rightarrow Party]$$

- Determine, using enumeration whether this sentence is valid, satisfiable or invalid. **Ans** A simple truth table has eight rows, and shows that the sentence is true for all models and hence valid.

- Convert the left hand and right hand sides of the implication into CNF, showing each step and explain how the results confirm to the previous answer. **Ans** For the left hand side we have

$$\begin{aligned}(Food \to Party) \vee (Drinks \to Party)\\(\neg Food \vee Party) \vee (\neg Drinks \vee Party)\\(\neg Food \vee Party \vee \neg Drinks \vee Party)\\(\neg Food \vee \neg Drinks \vee Party)\end{aligned}$$

For the right hand side, we have,

$$\begin{aligned}(Food \wedge Drinks) \to Party\\\neg(Food \wedge Drinks) \vee Party\\(\neg Food \vee \neg Drinks) \vee Party\\(\neg Food \vee \neg Drinks \vee Party)\end{aligned}$$

The two sides are identical in a CNF and is of the form $P \to P$, which is valid for any P.

5. Propositional Logic - CNF and Resolution

- a. Convert the following set of sentences to clausal form. (3 points)

  1. $A \leftrightarrow (B \vee E)$
  2. $E \to D$
  3. $C \wedge F \to \neg B$
  4. $E \to B$
  5. $B \to F$
  6. $B \to C$

  Ans:

  1. $(\neg A \vee B \vee E) \wedge (\neg B \vee A) \wedge (\neg E \vee A)$
  2. $\neg E \vee D$
  3. $\neg C \vee F \vee \neg B$
  4. $\neg E \vee B$
  5. $\neg B \vee F$
  6. $\neg B \vee C$

- b. Use resolution to verify if $\neg A \wedge \neg B$ can be proved from the above sentences. (3 points) (Hint: Prove each part separately)

  Ans: To prove the conjunction, it suffices to prove each literal separately. To prove $\neg B$, add the negated goal $S7 : B$.

  - Resolve $S7$ with $S5$, giving $S8 : F$.
  - Resolve $S7$ with $S6$, giving $S9 : C$.
  - Resolve $S8$ with $S3$, giving $S10 : (\neg C \vee \neg B)$.
  - Resolve $S9$ with $S10$, giving $S11 : \neg B$.
  - Resolve $S7$ with $S11$ giving the empty clause.

  To prove $\neg A$, add the negated goal $S7 : A$.

  - Resolve $S7$ with the first clause of $S1$, giving $S8 : (B \vee E)$.
  - Resolve $S8$ with $S4$, giving $S9 : B$.

- Proceed as above to derive the empty clause.

- c. Put each of the following in CNF form (4 points)

  1. $(P \wedge Q \rightarrow (X \vee Y)) \vee (R \wedge S)$
     Note that $\wedge$ has a higher precedence than $\rightarrow$. So you should assume $P \wedge Q$ as the precondition of the implication. So the above must be rewritten as follows:

$$
\begin{aligned}
&((P \wedge Q) \rightarrow (X \vee Y)) \vee (R \wedge S) \\
&= (\neg(P \wedge Q) \vee (X \vee Y)) \vee (R \wedge S) \\
&= (\neg P \vee \neg Q \vee X \vee Y) \vee (R \wedge S) \\
&= (\neg P \vee \neg Q \vee X \vee Y \vee R) \wedge (\neg P \vee \neg Q \vee X \vee Y \vee S)
\end{aligned} \tag{1}
$$

  2. $((P \rightarrow Q) \rightarrow \neg X) \leftrightarrow (A \wedge B)$

$$
\begin{aligned}
&((P \rightarrow Q) \rightarrow \neg X) \leftrightarrow (A \wedge B) \\
&= [((P \rightarrow Q) \rightarrow \neg X) \leftarrow (A \wedge B)] \wedge [(A \wedge B) \leftrightarrow ((P \rightarrow Q) \rightarrow \neg X)] \\
&= [((\neg P \vee Q) \rightarrow \neg X) \leftarrow (A \wedge B)] \wedge [(A \wedge B) \leftrightarrow ((\neg P \vee Q) \rightarrow \neg X)] \\
&= [(\neg(\neg P \vee Q) \vee \neg X) \leftarrow (A \wedge B)] \wedge [(A \wedge B) \leftrightarrow (\neg(\neg P \vee Q) \vee \neg X)] \\
&= [\neg(\neg(\neg P \vee Q) \vee \neg X) \vee (A \wedge B)] \wedge [\neg(A \wedge B) \vee (\neg(\neg P \vee Q) \vee \neg X)] \\
&= [((\neg P \vee Q) \wedge X) \vee (A \wedge B)] \wedge [\neg A \vee \neg B \vee (P \wedge \neg Q) \vee \neg X] \qquad \text{DeMorgan's Law} \\
&= (((\neg P \vee Q) \wedge X) \vee A) \wedge (((\neg P \vee Q) \wedge X) \vee B) \wedge [\neg A \vee \neg B \vee (P \wedge \neg Q) \vee \neg X] \qquad \text{distribution of } \vee \text{ over } \wedge \\
&= (\neg P \vee Q \vee A) \wedge (X \vee A) \wedge (\neg P \vee Q \vee B) \wedge (X \vee B) \wedge [\neg A \vee \neg B \vee (P \wedge \neg Q) \vee \neg X] \qquad \text{distribution of } \vee \text{ over } \wedge \\
&= (\neg P \vee Q \vee A) \wedge (X \vee A) \wedge (\neg P \vee Q \vee B) \wedge (X \vee B) \wedge (\neg A \vee \neg B \vee \neg X \vee P) \wedge (\neg A \vee \neg B \vee \neg X \vee \neg Q)
\end{aligned}
$$