CS 662
Artificial Intelligence Programming
Mid-term I Review Sheet

Topics

- Intro
  - What is AI: rationality versus human emulation
  - Domains
  - Agent types and environments
- Search
  - Complete / optimal
  - Time /space complexity
  - State space
  - Path cost
  - Uninformed
    - DFS
    - BFS
    - UCS
    - DLS
    - Iterative deepening
  - Informed
    - Best-first
    - Greedy
    - A*
    - Heuristics – admissibility, developing
    - IDA*
  - Local search
    - Hill climbing
    - Simulated annealing
    - Genetic algorithms
      - Fitness
      - Crossover and mutation
      - Roulette vs Tournament selection of parents
      - Elitism
    - Constraint Satisfaction search
      - What is a constraint?
      - Constraint propagation
      - Forward checking
      - Backtracking and backjumping
  - Adversarial search
    - Min-max
    - Alpha-beta pruning
    - Evaluation functions & their use
  - Propositional logic
    - Definitions - sound, complete, tautology, contradiction, modus ponens, etc

    - Inference
    - Converting into CNF
    - Resolution

Book sections (full chapter except where noted)

- Ch 1
- Ch 2
- Ch 3
- 4.1 and 4.2 of Ch 4
- Ch 5: 5.1-5.4, 5.7, 5.9
- Ch 6 except 6.5; 6.4 only at high-level
- Ch 7: 7.1-7.5

Types of questions

Here are the types of questions I tend to ask; more detailed examples are listed below:

- Define terms in your own words. True/false/explain for given definitions
- Give the setup for a problem. For example, what would the state, successor function, etc. look like for 8-puzzle?
- Sketch the execution of an algorithm. For example, show the queue for A* on a particular problem.
- Work problems (various topics, see below for examples)
- Give pseudocode describing an algorithm.
- Compare different algorithms and their strengths/weaknesses.
- Explain why an algorithm has the characteristics it does. For example, what does the heuristic function do in A*?
- Time and space complexity for different algorithms.

I won’t typically ask:

- Write syntactically correct Python code. (I may ask you to write pseudocode, but I'll be interested in program correctness, not whether every bracket and colon is in the right place)
- Multiple-choice questions.
- "Trick" questions - the point of this exam is to test your knowledge of the topics we've covered. I may ask you hard questions, but I won't ask things that are deliberately misleading or deceptive (unless I make a mistake, for which I will then be lenient in grading or hopefully notice it during the exam itself and correct or drop the question)

Detailed examples

1. Definition Questions. Define in your own words or “True / False / Explain”. If it is true, mark it true. If it is false, correct the statement so that it is true. Note: Adding ``not'' or

otherwise negating the sentence is not acceptable. You must change the facts in the sentence if it is false.

Sample question: The Turing test is a test of whether a computer program is rational

Bad Answer, no credit: The Turing test is **not** a test of whether a computer program is rational

Good Answer: The Turing Test is a test of whether a computer program is indistinguishable from a human.

Other sample questions:

- A complete search algorithm is one that is guaranteed to always find an optimal solution OR define a complete search algorithm
- A stochastic environment is one in which the world does not change when the agent is deciding upon an action OR define a stochastic environment

2. Problem Setup. Given a problem definition, draw a graph of the state space, with initial state and goal states specified, and all operations that connect states to other states.
3. Sketch Execution. Given a partial search space with the heuristic value of each node and cost of each operator labeled:
    1. Determine if the heuristic function is definitely admissible, definitely not admissible, or possibly admissible
    2. Give the order in which nodes are expanded (not generated) under A*
    3. Give the order in which nodes are expanded under IDA* (be sure to include all iterations, so that some nodes will appear in your list several times. Assume that on each iteration, the new depth cutoff is set to the f-value of the largest unexpanded node)
4. Given a 2-player game tree, with values at the leaves:
    1. Back up the values from leaves to the root
    2. Give the sequence of moves that will be made if both players are rational
    3. Assuming the min-max search traverses the tree from left to right, circle the nodes that will not be examined if alpha-beta pruning is used.
5. Work a problem in Genetic Algorithms. Given an encoding of a problem and a fitness function:
    1. For a specific population, give the fitness for each element, and the probability that each element would be chosen by roulette selection
    2. Show the resulting children for a given crossover, given the parents and the crossover point
    3. What makes a good encoding? What makes a bad encoding?
6. Work a problem in Logic
    1. Given a set of propositions, convert English sentences into propositional logic
    2. Convert into CNF
    3. Use resolution by refutation to prove a specific proposition is true

Example: Using the following propositions:

- S: Sarah wins the election
- E: Economic Reform is Passed
- J: John is happy
- M: Mark is happy
- P: Patricia is happy

Convert the following English sentences to Propositional Logic:

- If Sarah wins the election and Economic Reform is passed, then John will be happy
- If Sarah loses the election and if Economic Reform is passed, then Mark will be happy
- Mark is not Happy
- Economic Reform is passed
- It is not true that Patricia is happy if John or Mark is happy

Next, convert these sentences into Conjunctive Normal Form. Then use resolution to prove by refutation that John is happy. Show all your steps.