CPS 570: Artificial Intelligence - Practice Midterm

Your name:

Please read instructions carefully. Do not write down disorganized answers in the hope of getting partial credit; it's better to do a few questions completely right. Please write your answers down clearly. You can use extra pages. Good luck!
–Vince

Problem 1: Using search to solve maximum weighted tripartitie matching.

In *tripartite matching*, we have three sets, X, Y, and Z—women, men, and pets. We want to create families that each consist of a woman, a man, and a pet. Only some combinations of a woman, a man, and a pet are feasible. Formally, there is a set $S \subseteq X \times Y \times Z$, where $(x, y, z) \in S$ indicates that the family (x, y, z) is feasible. We have to choose which families we want to create, but we can only create feasible families, and we can use each woman, each man, and each pet at most once. (Not everyone has to be used.) A feasible solution is called a *matching*.

In *maximum weighted tripartite matching*, each feasible family $(x, y, z) \in S$ also has a weight $w(x, y, z) \in \mathbb{R}$, and our goal is to maximize the total weight of the families we create. Maximum weighted tripartite matching is a constraint optimization problem.

For example, suppose that the women are A(lice), B(renda), and C(arol); the men are D(irk), E(lliot), and F(rank); and the pets are G(rouchy), H(airy), and I(cy). Furthermore, suppose that we have the following feasible families:

- (A, E, I), with weight 4.
- (A, D, H), with weight 5.
- (B, F, G), with weight 3.
- (C, E, G), with weight 5.
- (C, D, H), with weight 2.

One matching consists of (A, E, I), (B, F, G), and (C, D, H), for a total weight of $4 + 3 + 2 = 9$. A better solution is to use (A, D, H) and (C, E, G), for a total weight of $5 + 5 = 10$.

Maximum weighted tripartite matching is NP-hard, but we can try to find an optimal solution using search. This is what you will do next.

Part a. Give a good search-based approach for finding an optimal solution for the (general) weighted tripartite matching problem. Your description should answer questions such as the following. What is a search state in your approach? Do you use variables, and if so, what are they? How are successors defined? Which states correspond to solutions? Are repeated states a problem? What is a good admissible heuristic (and why is it admissible)? (Note that this is a maximization problem, so an admissible heuristic should never *under*estimate.)

If your solution is based on an integer programming formulation, that is allowed, but you still need to answer the above questions.

Part b. Use your search-based approach from part a, together with the A* algorithm, to solve the tiny example problem instance from above (with Alice, Brenda, ...). Show the algorithm's search tree, and at each node, show the current state, and the value obtained so far + the heuristic value. If you decide to change something from your answer in part a (e.g., you decide to use a different heuristic), please explain this.

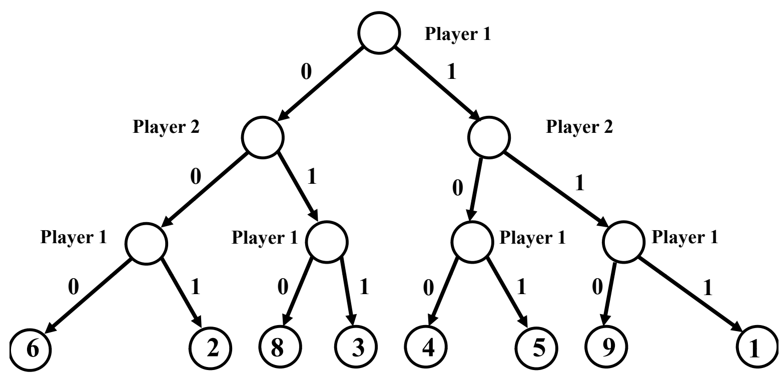

Figure 1: A two-player zero-sum game of perfect information.

Problem 2: Alpha-Beta.

Consider the game in Figure 1. Suppose you use alpha-beta to solve this game, and you explore the nodes from left to right. For each leaf node (= a node at the bottom of the tree with a utility for player 1), mark it as "yes" if alpha-beta explores it and "no" otherwise (i.e., if it is pruned). For each leaf node that you mark "no," state whether it is pruned based on alpha or based on beta.

Problem 3: Using resolution to derive an entailment.
Consider the following sentences:

- If it is raining and the sun is shining, then there is a rainbow.
- If the sun is not shining, then we do not wear sunscreen.
- If it is not raining, then we do not carry an umbrella.

We wish to show that they entail:

- If we carry an umbrella and wear sunscreen, then there is a rainbow.

You should show this by first converting the first three sentences and the negation of the fourth sentence to conjunctive normal form (an “and” of “or”s – of course, you can interpret an “and” as multiple sentences), and then using (only) resolution to derive *false* (i.e., the empty clause) in a few steps. You may just use plain propositional logic, not first-order logic; e.g., just use “Sunny” rather than “IsShining(Sun)”. You may wish to first convert the sentences to formal logic using $\Rightarrow$ before then further converting them to conjunctive normal form.

Problem 4: Using first-order logic to create a family.

We will consider a similar domain as in problem 1, but now with a first-order logic flavor. Your goal is to show that there exist some x, y, z that, together, are a feasible family.

Part a. Convert the following facts about this domain into first-order logic:

1. For all x, y, z, if x is a woman, and y is a man, and z is a pet, and x thinks the family (x, y, z) is acceptable, and y thinks the family (x, y, z) is acceptable, and z thinks the family (x, y, z) is acceptable,[1] then (x, y, z) is a feasible family.

2. Alice is a woman.

3. If z is hairy, and y vacuums, then Alice thinks the family (Alice, y, z) is acceptable.

4. The pet Hairy is hairy.

5. For all x, y, z, if x gives treats or y gives treats (or both), and z is a pet, then z thinks the family (x, y, z) is acceptable.

6. Everyone who gives treats also vacuums.

7. There exists a man y who thinks that for any x, z, the family (x, y, z) is acceptable; also, this man gives treats.

Finally, convert the goal into first-order logic: there exist some x, y, z that, together, are a feasible family.

[1] While pets have no legal recourse (or at least are cognitively unable to seek it), in practice they can nevertheless succeed in making a family infeasible...

Part b. Using standard reasoning patterns such as modus ponens, use the given facts to formally show that there exists a feasible family. Whenever you derive a new fact, state how you obtained it (with which reasoning pattern, and from which other facts).