Comments on Final Exam (CSE 573)

This exam (problem set) contains difficult problems. If your score is not as good as those of your previous problem sets, do not worry too much, since most people did not do as good as on the previous ones.

1. Graphplan and CSP

While many people answered correctly to (a), the perfect answer to (b) were rare. First read Dan's comment on (b):

The key point on this problem was that there is NO value of k such that k consistency does exactly what mutex reasoning does.

Some people talked about k=2 and 3, but to get the pruning of mutex reasoning you have to consider very large k. Remember that P and Q are mutex if every pair of supporting actions is pairwise mutex, so testing this requires MN constraints when M and N are the number of supporting actions respectively. However, note that the value of M and N may change over time. Furthermore, if one did MN-consistency the check would potentially cross time layers and hence prune much more (and be much more expensive than) mutex reasoning.

(a)

[Solution by Daniel Lowd]

No: if one action deletes a proposition that the other makes true in a single time-step, then at most one will be executed because their conflicting results (p and $\neg p$) will never permit a satisfying truth assignment.

(b)

[Solution by Tyler Robison]

Wrong: for most graphs there is no m value that makes the two equal. Mutual exclusion examines groups of 2 actions at a given time-step, and the propositions immediately they effect & depend on immediately before and after them. So if there are pb involved propositions on the time-step before, and pa involved propositions on the time-step after, mutual exclusion here involves looking at relationships between these 2+pb+pa variables. If we set m to this value, it might be able to prune similar to mutual exclusion, but m-consistency will also consider other groups of variables that mutual exclusion does not, for instance it would also look at a proposition and an adjacent action from time 0 to time m/2 (a total of m variables). Mutual exclusion does not consider these kinds of relationships, and yet if we reduce the m value then mutual exclusion will consider certain relationships that consistency does not.

2. MDP policy iteration

[Solution by Daniel Lowd]

(a) Step 1, policy evaluation:

$$U_0(X) = -1 + (0.9)U_0(Z) + 0.1U_0(X)$$

$$U_0(Y) = -2 + (0.9)U_0(Z) + 0.1U_0(Y)$$

$$U_0(Z) = 0 + U_0(Z)$$

Solving, we find that $U_0(Z) = 0$, $U_0(X) = -1.111$, and $U_0(Y) = -2.222$ is a valid solution to this system of equations.

Step 2, policy improvement:

i. State X

A. $\sum_{s'} T(X,a,s')U[s'] = T(X,a,X)U[X] + T(X,a,Y)U[Y] + T(X,a,Z)U[Z]$, which evaluates to $0.2 \times -1.111 + 0.8 \times -2.222 + 0 \times 0 = -2$.

B. $\sum_{s'} T(X,b,s')U[s'] = T(X,b,X)U[X] + T(X,b,Y)U[Y] + T(X,b,Z)U[Z]$, which evaluates to $0.1 \times -1.111 + 0 \times -2.222 + 0.9 \times 0 = -0.1111$.

C. Therefore, action b is still the best choice from state X; no change.

ii. State Y

A. $\sum_{s'} T(Y,a,s')U[s'] = T(Y,a,X)U[X] + T(Y,a,Y)U[Y] + T(Y,a,Z)U[Z]$, which evaluates to $0.8 \times -1.111 + 0.2 \times -2.222 + 0 \times 0 = -1.333$.

B. $\sum_{s'} T(Y,b,s')U[s'] = T(Y,b,X)U[X] + T(Y,b,Y)U[Y] + T(Y,b,Z)U[Z]$, which evaluates to $0 \times -1.111 + 0.1 \times -2.222 + 0.9 \times 0 = -0.2222$.

C. Therefore, action b is still the best choice from state Y; no change.

Termination: Since no change was made to the policy for any state, we're done.

(b) If the initial policy has a in both states, then policy evaluation will find a U-value of $-\infty$ for either state and the algorithm will be unable to proceed.

(c) Discounting by some γ $(0 < \gamma < 1)$ has the effect of making the utility of each state finite so that the utility of different actions can meaningfully be compared.

(d) As long as the discount factor is strictly between 0 and 1, it should have no effect on the final policy.

3. Bellman Equation

(a)

$$U(s) = \max \sum_{s'} (T(s,a,s')(R(s,a,s') + \gamma U(s')))$$

(b)

[Solution by Alan Liu]

Instead of operating in state space (which I will denote in lower-case s), operate in source-action-outcome space (denoted by upper-case S). That is, states in the new MDP actually represent source-action-outcome tuples.

For each transition $T(s,a,s')$, create S (equivalent to (s,a,s')) and $R(S)$ (equal in value to $R(s,a,s')$). In English: the state S in the new MDP corresponds to the state of being at s' after taking action a from state s in the original MDP. The reward function exactly corresponds to the one in the original MDP by construction.

For each pair of transitions $T(s,a,s')$ and $T(s',a',s'')$, create $T(S,a',S') = T(s',a',s'')$. In English: a transition in the original MDP took the agent from a source state (s'), via action (a'), to an outcome state (s''). In the new MDP, the same action would take the agent from its current outcome state to a new source-action-outcome state, for all source states and previous actions.

We create the initial state S_0 with only one action a_0 such that taking it gets to S_1 where S_1 corresponds to being at state s_0 in the original MDP. The reward $R(S_0)$ would be whatever reward we get right at the beginning of the original MDP.

We are searching in fully-observable source-action-outcome state-space and we know value iteration will give us the optimal policy that maximizes the expected discounted utility of source-action-outcome states. To get the optimal policy in the original MDP, just drop the "source" and "action" parts of the state and use the policy, because optimal actions only depend on the current "outcome" state that the agent is in. So if we were in S and $\pi[S] = a$, we know S corresponds to (s,a,s') in the original MDP, and dropping (s,a) we get $\pi[s'] = a$.

4. Tic-Tac-Toe

[(b) - (e) Solutions by Daniel Lowd]

(a) Most people answered well to this problem.

(b) Yes; the opponent's behavior affects the transition function probabilities, but this shouldn't stop us since we don't model the transition function at all, only the effects of actions.

(c) My agent would eventually adapt to the new strategy provided it affected the optimal behavior.

(d) I would consider board states that are equivalent by reflection or rotation to be the same state. By reducing the number of states, the learner can pool its knowledge together more effectively and learn faster.

(e) If the opponent employed an asymmetrical strategy (i.e. if agent plays in the lower right corner, then lose), then our learner could do worse because it would lose the ability to exploit weaknesses in that strategy.

5. (R&N 21.8) Updating equations

$$\theta_0 \leftarrow \theta_0 + \alpha[R(x,y) + \gamma\hat{U}_\theta(x',y') - \hat{U}_\theta(x,y)]$$
$$\theta_1 \leftarrow \theta_1 + \alpha[R(x,y) + \gamma\hat{U}_\theta(x',y') - \hat{U}_\theta(x,y)]x$$
$$\theta_2 \leftarrow \theta_2 + \alpha[R(x,y) + \gamma\hat{U}_\theta(x',y') - \hat{U}_\theta(x,y)]y$$
$$\theta_3 \leftarrow \theta_3 + \alpha[R(x,y) + \gamma\hat{U}_\theta(x',y') - \hat{U}_\theta(x,y)]\sqrt{(x-x_g)^2 + (y-y_g)^2}$$