Homework 3

Morgan Ciliv

February 2017

1 Minimax and AB Pruning

1.1 Representing the "game tree" for Connect Four

I would represent the "game tree" for Connect Four by having the initial node's state be an empty grid. I would then draw branches for each of the possible games states which player 1 could choose. Branching off of each of those game states, I would draw all of the possible game states for the grid that player two could choose. I would repeat the above process of drawing out the branching for the exchange of the player's moves till all branches end up as leaves where a player has connected four of their pieces.

1.2 Heuristic function I would use if given infinite look-ahead

As mentioned above, leaf nodes would represent game states where one of the player's has won. Therefore, I would make my heuristic function such that leaf nodes where I connect the four are "+1" and those where the opponent connects four are "-1." The goal of the heuristic function would be to maximize my score as I assigned a positive value to me winning and a negative value to me losing.

1.3 My heuristic subject to a finite number of plies

It is possible with a finite number of plies for the heuristic to search that finite depth and not reach a game state that contains either a three-in-a-row or winning leaf for ones of the players. Therefore, the heuristic could go in any direction if none of the game states reach a three-in-a-row or leaf state.

My heuristic:

- $+2$ for each of your discs that are in a two-in-a-row
- -2 for each of your opponent's discs that are in a two-in-a-row
- $+8$ for each of your discs that are in a three-in-a-row
- -8 for each of your discs that are in a three-in-a-row

- +512 (or infinity) for a winning leaf
- –512 (or negative infinity) for a losing leaf

In this heuristic, I have allocated points to two-in-a-row, so that the finite depth search will have more of a direction if it sees only two-in-a-row options. I have allocated very few points to the non-four-in-a-row states because they are not the goal. The goal has a much larger point value because that is the whole purpose of the game. The heuristic will take advantage of it when it reaches that depth. Also, a three-in-a-row is not required before a four-in-a-row because a player can fill in to make four in a row instead of just adding at the end of a three-in-a-row. That is another advantage of this heuristic over the example one.

2 Stochastic Games

Instead of playing against itself. The monte carlo simulation could simply repeatedly play randomly against this specific probabilistic strategy. The monte carlo simulation's resulting win percentages will indicate the value of certain positions. From that information, we can identify a strategy that is likely to win.

3 DPLL

The statements given are equivalent to the following...
$\neg A \vee B$
$\neg C \vee B \vee E$
$\neg B \vee C \vee \neg D$
$A \vee C \vee \neg D$

Using DPLL, we set D = $false$ then remove the $\neg D$ to reduce to the following equivalent expressions:
$\neg A \vee B$
$\neg C \vee B \vee E$
$\neg B \vee C$
$A \vee C$

Next, we set E = $true$ then remove the E to reduce to the following equivalent expressions:
$\neg A \vee B$
$\neg C \vee B$
$\neg B \vee C$
$A \vee C$

There is no literal that is consistent through each of the four statements. However, there is a truth-value assignment that makes each of the clauses true. We can set B = $true$ and C = $true$.

Therefore, there is indeed no inconsistency in this knowledge base.

The final model is V = {B = $true$, C= $true$, D= $false$, E= $true$}

Now, to see if B ∨ E is entailed by this knowledge we include the negation of that sentence into the list of sentences to test using DPLL. We will use DPLL in the same way.

First we must convert $\neg(B \vee E)$ to CNF.

This becomes $\neg B \wedge \neg E$ Therefore, we B and E are just separate clauses, and we begin DPLL below.

$\neg B$
$\neg E$
$\neg A \vee B$
$\neg C \vee B \vee E$
$\neg B \vee C \vee \neg D$
$A \vee C \vee \neg D$

Then, D can be set equal to false, so $\neg D$ can be removed.
$\neg B$
$\neg E$
$\neg A \vee B$
$\neg C \vee B \vee E$
$\neg B \vee C$
$A \vee C$

There are no more literals that are consistent all the way through. There is no truth-value assignment that makes all the others true. By following the rest of the DPLL algorithm, no truth-value assignment can be made. Therefore, $B \vee E$ is not entailed.