Artificial Intelligence
School of Computer Science, University of Adelaide

Assignment 2: The Three Piles Problem

Due 5PM, Wednesday 30$^{\text{th}}$ May, 2007

1 Overview

In this assignment, the aim is to develop a program that plays an adversarial game with optimal performance. The game is as follows. Suppose that there are three piles of tokens. Each pile may contain an arbitrary number of tokens. When making a move, a player must remove one or more tokens from just one of the piles. Players move alternately in this way, and the winner is the player who forces the opponent to remove the last token.

2 Example Sequence

Suppose that the three piles contain 3, 2, and 1 tokens, respectively, represented as 321. Here are a couple of legal (but not necessarily optimal!) games, as described by successive pile states:

Move	Pile State
Initial state:	321
P1 takes one stick from P_1:	221
P2 takes two sticks from P_2:	201
P1 takes one stick from P_1:	101
P2 takes one stick from P_1:	001
P1 must take one stick from the P_3:	000

Second player wins

Move	Pile State
Initial state:	321
P1 takes three stick from P_1:	021
P2 takes one sticks from P_2:	011
P1 takes one stick from P_3:	010
P2 must takes one stick from P_2:	000

First player wins

3 Task

Your task is to write a class in Java called `MyPlayer` that plays optimally. (You may assume that an individual pile will never contain more than 5 tokens, and the total number of tokens will never exceed 12.) It should do this by generating the complete game tree, and employing a minimax-like technique to determine the optimal move. Note, therefore, that the lookahead in this game is complete, unlike that in chess. Effectively, your game tree assesses all possible game sequences. Observe, therefore, that it is necessary to generate the game tree only once, at the first move, again unlike chess. Note that your board evaluation function will only apply to the leaf-node situation 000, and should return a value of 0 or 1 according to who made the last move. Values 0 or 1 should then be percolated up the tree as appropriate. You will be given an interface that you must use, and you will be able to run three of our versions of the program that play randomly, optimally, and with human input. It will be necessary for you to integrate your program into our framework. More instructions and code are provided on the web-site.

4 Hints

Note that the game tree can become very large for non-trivial pile sizes. At each node in your tree, you need to represent both the state of the piles and the percolated minimax value. You should do this compactly

using bytes to store the pile sizes and minimax value. Having got your program working on very small pile sizes, you should consider how it might run more efficiently. One critical improvement is to avoid generating multiple instances of a similar position when generating children in a sub-tree. Thus, given, say, 222, the exhaustive set of possible moves leads to a sub-tree containing 122,022,212,202,221,220. But given that the ordering of piles does not matter in the analysis, these can each be sorted, and redundancies can be eliminated, to yield 122,022, a saving of four nodes as well as their subsequent sub- trees. This dramatically improves program efficiency.

5 Marking

Your solution will be marked on the basis of the quality of your code and its performance. Note that it is essential that you follow the minimax approach outlined above, an alternate solution method being unacceptable. Performance of your method will be tested against our optimal performer. Marks will be awarded for an optimal solution, even if it is slow; more marks will be awarded for an efficient solution that deletes redundant nodes as indicated earlier.

6 Help/Consultant

Please use the bulletin board if you are stuck on something. You should not seek a solution via the bulletin board, but rather should ask specific details. Note that other students should not make unreasonable amounts of information available via the bulletin board. Tutorials will be provided if necessary, if there is enough interest.

Rhys Hill
April, 2007