Algorithm Correctness

1 Introduction

In the Nim Game it is known that if both players make no mistakes and the nim-sum at the beginning of the game is not zero, then the player starting is guaranteed to win. If the players make no mistakes and the nim sum at the beginning of the game is zero then the player who does not go first is guaranteed to win.

2 Proof

The proof will show that the algorithm we have created accomplishes the goal of making the nim-sum either zero or non-zero based on the state of the piles on it's turn. We will also prove that the algorithm is capable of changing its move based on the moves the opponent could make.

This code is present in the getZeroSumMoves method in the shortPrediction-Bot class but starts in the isTrap or makeChoice method to find the number of large piles. First the algorithm will count the size of the piles and if there are enough piles with size greater than one then that means there is a zero nim-sum possible. This is to say that in order for there to be a zero nim-sum there needs to be either 0 or 2 or more piles with greater than 1 element in them.

Proof that one pile with two or more elements cannot have a zero nim-sum Suppose there is only one pile with 2 or more objects in it. Then for this pile the binary number which represents it is guaranteed to have a 1 in a position other than the first position. For example: 2 is shown as 00000010 this means that for every other pile of size 1 or less that they will have a binary representation of 00000001 or 00000000 hence the XOR of these numbers can never be equal to zero since there will never be another 1 in the second position.

Proof that any number of piles with 1 or less item in it can have a zero nim-sum Proof by contradiction: suppose that a zero nim-sum is not achievable with any number of piles with 1 or less items in them. Suppose there are to piles with 1 item, then the binary representation would be 1 and 1 and 1 XOR 1 = 0. This means we contradicted the staement "a zero nim-sum is not achievable with any number of piles with 1 or less items in them" hence this statement is false and it is possible to have a zero nim-sum for that condition.

Proof that if there are more than or equal to two piles with two or more objects then a zero nim-sum is achievable Proof by contradiction: Assume that a zero nim-sum is not achievable with two or more piles with two or more objects in each. Suppose there are two piles with two objects in each, then they would have a binary value of 10 and 10 and 10 XOR 10 = 0 so a nim-sum of zero is possible. This contradicts the previous statement and hence it is possible to have a zero nim-sum with two or more piles of two or more objects in them.

Next we will show that the algorithm will appropriately change the nim-sum to zero or not zero based on the current nim-sum. The first for loop in getStandardChoice will always compute the nim-sum for the current piles. The loop invariant is that nim_sum will always be the nim-sum of the all the piles seen so far. In the second loop we need to examine two cases: case 1: nim-sum is zero, and case 2: nim-sum is non zero.

Case 1: Nim-sum is zero

When the nim-sum is already zero then it doesnt matter which move is made because it wont be possible to make the nim-sum zero again. This is guaranteed in the first if statement where the XOR of nim-sum and pile have to be equal to pile since nim-sum is zero, hence any move which doesn't lose the game satisfies the condition.

Case 2: Nim-sum is non zero

For a nim-sum which is non zero then we must change it to zero, and to do this we must find the highest pile and then remove one element from it. The proof that this will happen is that for a non-zero nim sum, when we XOR it with some other number then it can will be less than that number, hence the if condition will be false. Proof: Assume that for a non-zero nim sum, the XOR of it with another number cannot be less than the other number, counterexample: suppose that the nim-sum is 2 and there are two piles with 4,2, and 1 items in them, this has a nim sum of 7 and XOR of 7 and 4 is 3, 3 is less than 4 hence this is a contradiction and it is in fact true that a non zero nim-sum XOR with another number can be less thanthat number and the code will indeed subtract from the largest pile to create a zero nim sum which obeys the algorithm goal.

Proof that the algorithm can react to potential moves the opponent will make

This code is in the shortPredictionBot class with the makeChoice method which uses the isTrap method as well. Beginning with the first if the algorithm first must examine the number of piles with two or more objects in them which will affect the strategy. More than one large piles mean that the game is not in a late stage yet and can then compute potential enemy moves. In the else the bot will use the second nested for loop in order to test potential enemy moves. Proof: first the game state after the current turn is recorded and then in the third nested for loop (line 342) all potential enemy moves are tested to see if they are a trap move where a trap move is defined as a losing state for the bot. The moves are tested by taking the possible next move and taking between 1 and all objects from a pile and doing this for every pile. If a trap move is found then the move is not viable and we must look for a different move.