Solution to the Game of Nim

Aditya Neeraje

2021-07-01

1 Normal Play

Nim has long been a solved combinatorial game. I claim that the function G given by the xor of the lengths of all the lengths of the piles of Nim is the Grundy value of a game of Nim. In Normal play, I will show that a player ending at a position with Grundy value 0 can always win the game. For that, I will show two things: if the player is at a position with Grundy value 0, he can only move to a position with non-zero Grundy value, and if a player is in a position with non-zero Grundy value, he can always end at a Grundy value of zero.
Claim 1: If the player is at a position with Grundy value 0, he can only move to a position with non-zero Grundy value.
Proof: Let the xor of all lengths initially be 0. If the player chooses a pile of length a_i and reduces it to a pile of length a_f, the new xor is equal to $a_i \hat{} a_f$. This new value cannot be zero unless $a_i = a_f$. However, any valid move in Nim must reduce the length of the pile, hence the new xor cannot be zero.
Claim 2: If a player is in a position with non-zero Grundy value, he can always end at a Grundy value of zero.
Proof: Let the xor of all lengths be a non-zero value ω. The player can always find a pile of length a_i such that $a_i \hat{} \omega$ is less than a_i. This is because $a_i \hat{} \omega = a_i + \omega - 2 \cdot a_i \& \omega$. If x is the greatest bit of ω set to 1, x must be set to 1 in at least one a_i. Two times the bitwise and of this value of a_i with ω is greater than omega, since the bit one higher than the largest set bit in ω is set in $2 \cdot a_i \& \omega$.

Having concluded the above, we note that Nim is a finite game (since every move strictly decreases the number of matchsticks remaining), and that a position with Grundy value 0 at the end is a losing position. Hence, the first player to get a position with non-zero Grundy value can always ensure his opponent gets positions with zero Grundy value until the game ends with him winning.

2 Misère Play

In Misère Play, the player who takes the last matchstick loses. The only truly forced moves occur when all piles are of size 1, and in such a case it is obvious the game is a $\mathscr{P}$ win only if there are an odd number of ones left. The only way to move to a position with only ones from a different position is for there to have only been one heap of size n $\geq$ 2 which was taken down to either 0 or 1, according to which will help that player win. In that case, however, the xor of all the piles is non-zero (n $\oplus$ 1 and n $\oplus$ 0 are at least n-1, and n $\geq$ 2). Going by the earlier proofs, since either player can preserve the property of having a position with non-zero Grundy value, Player 1 wins if the xor of all piles is initially non-zero, and loses otherwise. If the initial xor is non-zero, he can win by maintaining a non-zero xor until the obvious eventuality where at most one pile has more than one elements, and taking that pile to 0 if there are an odd number of piles left, or to 1 if there are an even number of piles left.

3 Understanding the Code

Using tkinter, a rough GUI has been built. The number of heaps varies from 5 to 14 and the number of matchsticks per heap varies from 1 to 14. The player can choose to play first or let the computer go first. At the end of the game, simply close the window to exit.