INDUCTIVE PROOF FOR N NIM PILES

2/5/2020

Using XOR to determine if a game is safe

The XOR operator can be used to determine if a game is safely determined or not. This is due to the XOR operation being equivalent to addition modulo 2. This is primarily beneficial due to the following properties of XOR.

$$A \oplus A = 0$$
$$A \oplus B = B \oplus A$$
$$A \oplus 0 = A$$
$$A \oplus (B \oplus C) = (A \oplus B) \oplus C$$

These four properties are useful in determining whether a game is safely determined. By definition a game is safely determined if the XOR of all its piles are equal to some other pile. Due to this a game can be stated to be safely determined if the XOR of all piles is equal to zero, this arises from the self inverse property stated earlier. Another consequence of this is that there can only be one unique solution for a subset of n-1 piles, where n is the number of piles.

The following example from Charles Bouton's paper demonstrates a safe game resulting in the XOR of all piles equaling 0.

$$1001 \oplus 101 \oplus 1100 \ = 0$$

Listing 1: Algorithm to make games safely determined

```
piles = [3, 5, 7]

def MakeSafe(piles):
  #Set x equal to the identity
  x = 0
  #Iterate over all piles, Taking the exclusive-or as we go
  for i in range(len(piles)):
    x ^= piles[i]

  for i in range(len(piles)):
    #Take the exclusive-or again, removing that value
    #This is possible thanks to the commutative
    #and self-inverse properties
    x ^= piles[i]
    #If we found a pile we can set equal to the xor
    #of all the other piles then end
    if(piles[i] > x):
        #Calculate the amount that should be taken away in order
        #for the pile to be equal to x
        remove = piles[i] - x
        piles[i] -= remove
        return
    #Otherwise xor the pile back in and continue
    x ^= piles[i]
```

Algorithm explanation

It can be seen that this algorithm works by looking at the XOR of all piles. if their is a 1 in a column that means there is an odd number of 1s in that column. So in order to find a pile that can be subtracted from to equal the XOR of the rest you just need to grab one of the numbers contributing the largest binary digit of the XOR of all piles.

$$
\begin{array}{r}
1001 \\
101 \\
1100 \\
111
\end{array}
$$

$$111$$

So either the 2nd, 3rd or 4th column can be selected. You can xor the second pile with the xor of all piles to get.

$$10$$

then you must only subtract 3 from the second pile to make the game safe again. A similar process can be applied to the 3rd and 4th pile.

Inductive Proof for n piles

Considering the properties established earlier the inductive proof can be setup easily. For the base case, take a game that is not safely determined or the xor of all its piles is not equal to 0.

$$(x_1 \oplus x_2 \oplus x_3 ...) \oplus x \neq 0$$

The inductive step is then player A's turn where the state of the game is not safely determined. player A then applies the above algorithm making the game safely determined. the k+1th turn, Player B's turn is then forced to make the game unsafe. This is because it is impossible to add or subtract from any pile and still satisfy the requirement for the game being safely determined. This can be seen from the following

$$
\begin{aligned}
(x_1 \oplus x_2 \oplus x_3 ...) \oplus x = 0 \\
=> (x_1 \oplus x_2 \oplus x_3 ...) = x \\
=> x \oplus x = 0
\end{aligned}
$$

Nothing can be subtracted from either pile without changing x, the unique solution, so the game has to be made unsafe.