The complexity of nim

Chris Calabro

February 28, 2006

Abstract

We show that both the decision and search versions of nim are solvable in log space, whether the game is represented in binary or unary. In some games, the losing player has a delaying strategy that can make the game take exponentially long in the binary representation of the game if the winning player plays optimally.

These results all hold no matter whether the first player without a move is declared the winner or loser.

1 Game definition

Nim is a 2 player game. A game state consists of a multiset S of nonnegative integers together with whose turn it is. If S does not contain any positive integers, then whose turn it is loses (we will call this the normal play rule) or wins (misère rule), depending on the version of the game. Otherwise, a move consists of strictly decreasing one of the positive elements S to another nonnegative integer, and then whose turn it is alternates to the other player. Unless said otherwise, we assume the normal play rule.

2 Condition for a forced win

Since players do not own the game pieces in any sense, we can abstract away whose turn it is for the purposes of analysis by representing a game state simply as S and assume that player 1 goes first. Then we can say that S is *losing* iff every legal move on S leads to a winning state; (Notice that the universal quantifier obviates the need to define a base case since if there is no legal move, then certainly every legal move leads to a winning state.) otherwise, S is *winning*.

Define the *nim sum* of a game S as $g =$ the binary XOR of the numbers in S.

Lemma 1. *$g = 0$ iff every move leads to $g \neq 0$.*

Proof. (⇒) Consider some move that changes element $x > 0$ to $x' \neq x$. The nim sum of the resulting game is then $g \oplus x \oplus x' = x \oplus x' \neq 0$.

(⇐) This is the harder direction. Suppose $g \neq 0$. Let i be the position of the most significant bit in the binary representation of g. So $g \oplus 2^i < 2^i$. The number of elements of S that have a 1 bit in position i is odd. Choose one of them arbitrarily, call it x. Then $x \geq 2^i$. So $g \oplus x < x$. So it is legal to change x into $g \oplus x$, which yields a new nim sum of $g \oplus x \oplus (g \oplus x) = 0$. □

Theorem 2. *S is winning iff $g \neq 0$.*

Proof. We use induction on the (ordinary) sum s of the elements of S. If $s = 0$, then $g = 0$ and the game is losing. Now suppose $s > 0$.

If $g \neq 0$, then by lemma 1, some move leads to a 0 nim sum, which, by the inductive hypothesis is losing for the other player and is therefore winning for the current player. If $g = 0$, then by lemma 1, every move leads to a $\neq 0$ nim sum, which, by the inductive hypothesis is winning for the other player and is therefore losing for the current player. □

3 Misère winning condition

Suppose now that the first player without a move wins.

Theorem 3. *S is winning iff ($g \neq 0$ XOR $\forall x \in S\ x \leq 1$).*

Proof. We use induction on the sum s of the elements of S. If $\forall x \in S\ x \leq 1$, then clearly S is winning iff $g = 0$, and we are done. So suppose $\exists x \in S\ x > 1$.

If $g = 0$, then by lemma 1, every move leads to a $\neq 0$ nim sum. Note that since $g = 0$ and there is some number in S of size > 1, there must be at least *two* numbers in S of size > 1. So every move leaves at least one number in S of size > 1. By the induction hypothesis we are done.

If $g \neq 0$, then by lemma 1, there is a move that reduces the nim sum to 0. Let i be the position of the most significant bit in the binary representation of g and let x be some element of S with a 1 in bit position i. Then it is legal to change x to $g \oplus x < x$. If this leaves at least one element of S of size > 1, then by the induction hypothesis we are done. Otherwise x is the only element of S of size > 1 and a winning strategy is to reduce x to 0 if $|S|$ is even, and reduce x to 1 otherwise. □

4 Computational complexity

Suppose the game is represented in binary. Whether the nim sum is nonzero can be decided in log space. Whether there is some number in S of size > 1 can also be decided in log space. An actual winning move can be found in log space as well since this only requires computing the position of the most significant bit in the nim sum, finding an element of S with a 1 in that position, and computing the XOR of the remaining elements.

5 Game length

For some games, every winning strategy takes an exponential amount of time if the losing player wishes to use a delaying strategy. E.g. consider the game where S contains exactly 2 copies of some number n. Then the losing player (player 1) may choose to reduce one number by 1 at each step. The only winning move for the other player is to reduce the other number by 1. (at least until the very end, if using the misère rule) So the game takes $\Omega(n)$ steps, which is exponential in the bit length of n.