

Nim

Spring 2017

Math-M330

Steven Myers

April 7, 2017

Rules of Nim

Nim is a two player game. It can be played with several different collections of items, such as dice, coins, sticks or cards. We will assume that our version of Nim uses a traditional deck of cards. To play the game, rows of cards must be laid out. The number of cards in each row can change from game to game. For example, one possible initial board set-up could be three rows of cards, each row containing 3 cards. Whenever it is a player's turn, they must take *at least* one card from the board. The player may take as many cards from the same row as they like. Players take turns until there are no cards left on the table. In our version of Nim, the player who takes the last card(s) from the board wins the game. This is called the *normal* form of the game. Alternatively, there is a *misère* form of the game in which the last player who takes a card *loses*.

Turns	0	1	2	3
	xxx	xxx	xxx	
	xxx	xxx		
	xxx			

Figure 1: Sample Nim Game, where x is a card

We can distinguish the two players of the game by whoever goes first. We'll call the person to make the first move player M , and the second player N . In the example game above, we would say that M is the winner, since M takes the last row of cards in turn 2. Since each player *must* take a card each turn, the game is deterministic and there is always exactly one winner who takes the last card. We will find that not only is Nim deterministic, but a winner can be predicted for every single initial board set-up assuming players play optimally. In other words, Nim is a *solved game*.

M and N Positions

In order to show that Nim is a solved game, we can look at some trivial board set-ups where the winner is immediately obvious, then we can move on to more complex board set ups. First, let us come up with a notation for boards. We can represent them as a vector or list of numbers, where each element represents a row and number of cards in that row.

$$\begin{array}{ccc}
 \text{xxx} & & \text{x} \\
 \text{xxx} & & \text{xx} \\
 \text{xxx} & & \text{xxx} \\
 \\
 \text{Board} = (3, 3, 3) & & \text{Board} = (1, 2, 3)
 \end{array}$$

Figure 2: Some example vector representations of the game boards

The simplest board set up we can think of is $\text{Board} = \text{NULL}$ or alternatively $\text{Board} = (0)$, where there are no initial cards on the board. In this board set up, the second player, N , is actually the winner. We can revise our rules of Nim to clarify that a player *loses* if they are unable to take a card when it is their turn. This makes it more clear that M loses the game when there are no initial cards.

In the game where $\text{Board} = (1)$, M always wins because M always takes the last and only card. In fact, a board in which there is only one row and at least one card is *always* a win for M , since player M can always take all of the cards in the single row. Thus, we call such board set-ups or games M or N positions to indicate that either M or N will always win that given position.

A crucial aspect of analyzing Nim games is understanding that each turn of a Nim game is effectively a new game board, which is in and of itself either an M or N position. Once again, looking at a trivial case, an initial game with $\text{Board} = (1)$ becomes $\text{Board} = (0)$ after M takes the first move. We know that $\text{Board} = (1)$ is a win for M , and $\text{Board} = (0)$ is a win for N . This makes sense because M effectively becomes N in the second turn of the game. We can think of M as the *player on move* and N as the *player on deck*, and they swap each turn.

A Strategy for Nim

Even in the trivial examples, a strategy for winning a game of Nim becomes obvious early on. If we are player M , we want to make a move that will make the next board state become a board which guarantees that N will win. So, if we have some game board position P , we can win the game if we are able to make a move such that the resulting game P' is an N winning position.

Using this strategy and our trivial cases with Nim games consisting of single rows or no cards at all, we can begin to reason about Nim games with two rows of cards. Once again, we can begin with the trivial example, where we have two rows with a single card each.

Turn	Board	Move
0	(1, 1)	Player 1 takes 1 card from row 1
1	(1)	Player 2 takes 1 card from the last row
2	(0)	Player 1 loses

Figure 3: Full Nim game for trivial 2-row game

So, in the trivial case for two rows containing 1 card each, we find that no matter what, Player 2 wins. This means that $Board = (1, 1)$ is an N position. No matter what move M makes, M always loses. Looking back to our strategy, if we *cannot* come up with a move that puts a our board into some position P' that is an N position, M cannot win the game.

Board	Winner
(0)	N
(1)	M
(c)	M where $c \geq 1$
(1, 1)	N
(1, 2)	M
(1, c)	M where $c > 1$
(c , c)	N for all c
(c , d)	M where c and $d \geq 1$, $c \neq d$
(c , c , c)	M for all c
(c , c , d)	M where $c \neq d$

Figure 4: Trivial M and N Positions

We can exhaustively search for all trivial games and determine whether or not they are M or N positions, then iterate through possible P' positions from our current board that is a triovare *reachable* from our current board. However, this becomes burdensome (computationally expensive) as board sizes become larger and possible positions become numerous. Fortunately, we can determine whether or not some position is an N position by some game algebra and bitwise operations.

Bitwise XOR and Solving for N Positions

In order to solve for the winner of a particular Nim game without searching through all the possible moves in a game and its sub-games, we need to convert our game board representation into a format that can easily determine whether or not it can be reduced to (0) game, where N wins. Semantically, we want to show that an N -position can always be reduced to the (0) position of the game, and moreover that all N positions are *congruent* with respect to the winner of the game. By extension, a position P is congruent to a position Q if $P + Q$ is an N -position. We will call adding two Nim game positions *Nim addition*.

All game positions can be converted to binary representations, which allow us to easily perform *bitwise XOR* over the the positions to get from one position to the next. This bitwise XOR operation is our Nim addition concept, introduced in the previous paragraph. So, we can add two positions together by performing bitwise XOR operations on the two positions converted to a binary representation.

Binary	Decmial
011 ₂	3 ₁₀
011 ₂	3 ₁₀

Figure 5: Binary Representation for $P = (3, 3)$

Let's consider a game position we know to be an N position and show that when we perform bitwise xor, we resolve to (0) game. One N position is $P = (3, 3)$. We can equivalently write P as $(1, 2, 1, 2)$ by expanding each row of the game position into factors of 2. We call these factors of 2 *generators*. Once we've expanded a Nim position into its generators, we can easily convert it a binary expression.

Binary XOR
011 ₂
011 ₂
000 ₂

Figure 6: Binary XOR for $P = (3, 3)$