

Winning colourful Nim with the eeny-meeny rule

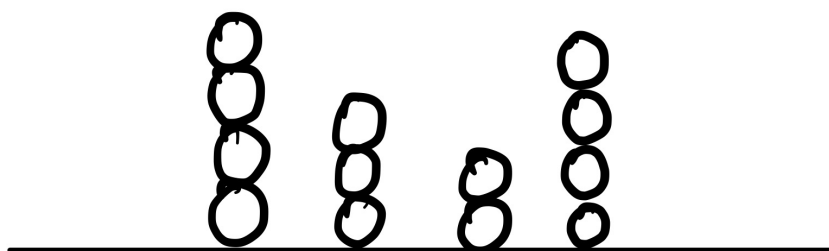
Dou-Dou Liu, Olly Britton

August 2025

1 Introduction

Nim is a game where two players take turns removing stones (“nimming”) from different piles. On each player’s turn, they can remove as many stones as they want, but all have to be from one pile. The player who makes the last move wins, or equivalently, the player who is left with no remaining moves loses.

Take as many as you like from
the top of one pile



This simple game turns out to be surprisingly deep in the field of mathematics and computer science known as *combinatorial game theory* [1], which is concerned for the most part with beating opponents in two-player perfect information games, such as tic-tac-toe, checkers, and Go. Specifically, there is a result called the Sprague-Grundy theorem [5], which states (informally) that every combinatorial game in which each player has access to the same set of moves in each of their turns is *equivalent to a game of Nim*. Combinatorial games like this are called “impartial games”.

This means the following – if you are playing an impartial game, and:

1. You are a perfect Nim player and
2. You know how to convert positions in this game into a game of Nim

then you can always win that impartial game.

Luckily, perfect play at Nim is reasonably easy if you know the trick, a result called Bouton’s theorem [2], which says that you can always guarantee a win if, on your turn, you remove stones to make the *bitwise XOR* (sometimes called the *nim-sum*) of all the pile sizes equal to zero.

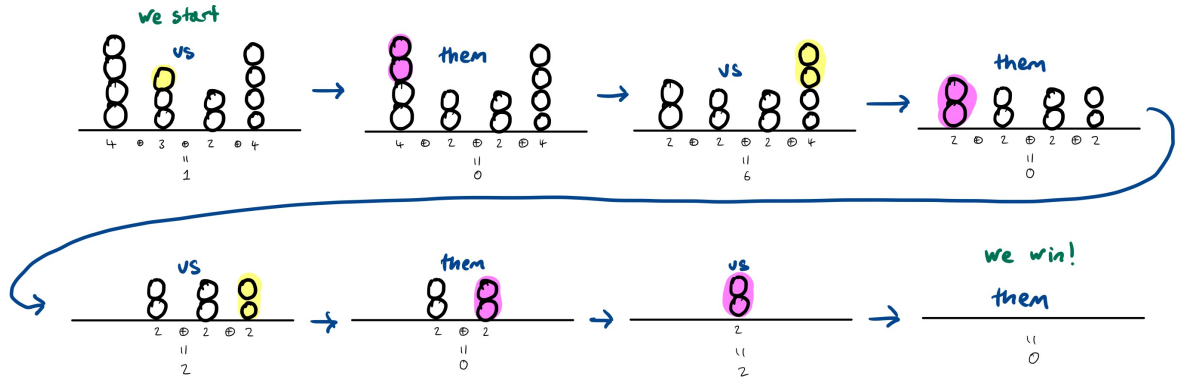


Figure 1: An example game of Nim, annotated with the Grundy value for the overall position.

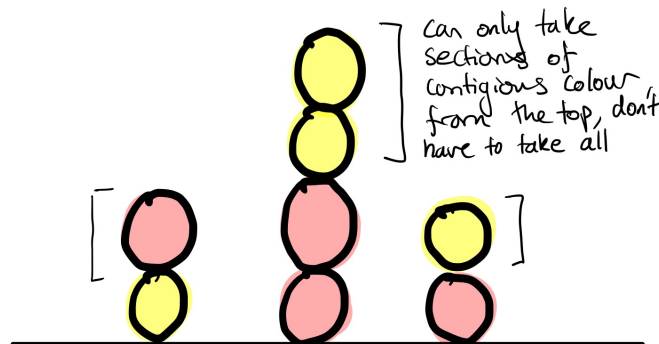
Calculating the bitwise XOR of many numbers sounds difficult, but with enough practice it becomes easy enough that you can even do it in your head while playing.

Hence, the bottleneck in winning at impartial games is determining an efficient procedure for how to convert positions in that game into a game of Nim. This is often a very non-trivial process, and for many impartial games even basic questions about the structure of this process have been open problems for over 40 years [1].

In this article, we present a new variant of Nim called “Colourful Nim” based on the introduction of colours to the individual stones. We then present an efficient rule for converting positions in this game into equivalent games of Nim, initially discovered by computer search, and prove its correctness. Finally, we end with some open questions about further generalisations of this game for which the equivalent Nim games are still unknown.

2 Colourful Nim

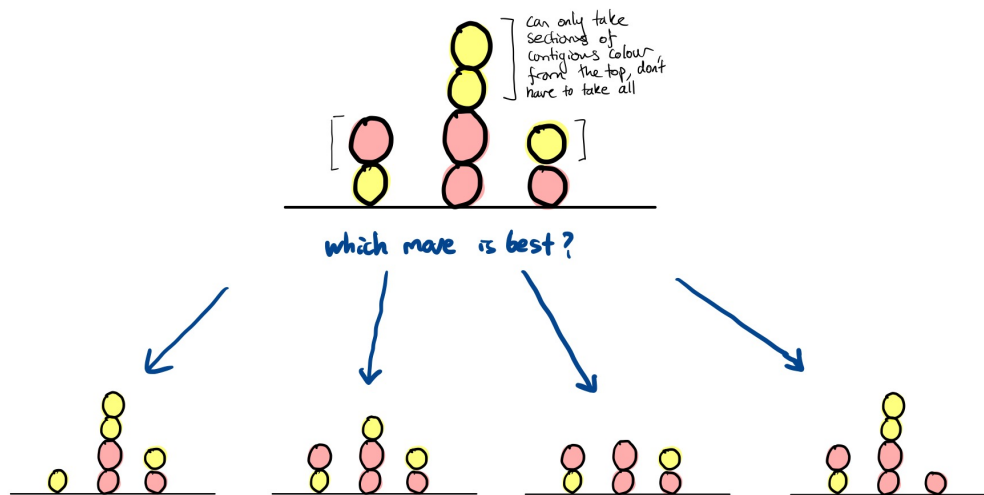
Suppose you’re playing Nim with a set of English pool balls which are red and yellow, and you add the rule that you can only remove sections of contiguous colour from the tops of each pile¹.



You could also play this game with cards facing up and down, knives and forks, bananas and apples, and so on. You just need two distinguishable items you can make piles out of.

¹An interactive version of this game is available online

The winning strategy for just one pile is reasonably straightforward to work out (consider which moves you can force your opponent to take). But when there are multiple piles, things are a lot more interesting and working out the best move is non-trivial.



2.1 A winning strategy for multiple piles

The following gives the perfect strategy for any number of piles. To describe the winning strategy, we will need some notation.

Notation: Write $(n_1, n_2, \dots, n_k)$ for a pile with n_k balls of one colour at the top, followed by n_{k-1} balls of the next colour, followed by n_{k-2} of the original colour, and so on, all the way until n_1 which gives the number of balls in the bottom section of the pile.

Notation	Game
$()$	<empty>
(3)	
$(3, 1)$	
$(1, 1, 1, 1, 2)$	

Note that for all practical purposes when trying to win, it doesn't matter if the pile starts yellow or starts red, it is really only the sequence of colour changes that matters.

The strategy is similar to ordinary Nim. In ordinary Nim, you win by making the XOR of all pile sizes equal to zero. Now instead of making the XOR of all the pile sizes equal to zero, you make the XOR of the “Grundy values” of each pile equal to zero. Write $G(n_1, n_2, \dots, n_k)$ for the Grundy value of the pile $(n_1, n_2, \dots, n_k)$.

It’s not necessary to know exactly what “Grundy value” means to win, we just need to know how to calculate it. But intuitively, the Grundy value of a pile tells you the size of the equivalent Nim heap as guaranteed to exist by the Sprague-Grundy theorem.

The empty game has Grundy value zero, or in the notation, $G() = 0$. If the pile is just of one colour, then the Grundy value is just the number of balls in that pile. In our notation, this means $G(n_1) = n_1$. This is intuitive since a pile with no colour changes is equivalent to a game of ordinary Nim.

To calculate the Grundy value for piles with colour changes, things get a bit more complicated. To do this, you need the “eeny-meeny rule”.

2.2 The eeny-meeny (E-M) rule

Suppose you are looking at a pile $(n_1, n_2, \dots, n_k)$. The rule is:

$$G(n_1, n_2, \dots, n_k) = \begin{cases} n_k - 1 & \text{if the Eeny-Meeny (E-M) condition is met} \\ n_k & \text{otherwise.} \end{cases} \quad (1)$$

So either $G(n_1, n_2, \dots, n_k)$ is n_k or $n_k - 1$, and the E-M condition tells you if you need to subtract one from n_k . So when is the E-M condition met?

- Start by looking at the top of the pile, n_k , and work your way down each section of colour in steps, saying “eeny” and “meeny” alternately.
 - Call n_k *eeny* compared to n_{k-1} if $n_k < n_{k-1}$ (**e**xiguous, very small in size or amount).
 - Call n_k *meeny* compared to n_{k-1} if $n_k > n_{k-1}$ (**m**assive, large and heavy or solid).
- If you say “eeny” moving from n_k to n_{k-1} and n_k is *eeny* compared to n_{k-1} (i.e., $n_k < n_{k-1}$), then the condition is met! You should subtract one, so $G(n_1, n_2, \dots, n_k) = n_k - 1$ in this case.
- If you say “meeny” moving from n_k to n_{k-1} and n_k is *meeny* compared to n_{k-1} (i.e., $n_k > n_{k-1}$), then the condition is also met! You should subtract one, so $G(n_1, n_2, \dots, n_k) = n_k - 1$ in this case.
- If $n_k = n_{k-1}$, continue working your way down the pile.
- If n_k is meeny (or eeny) compared to n_{k-1} when it is supposed to be eeny (or meeny, respectively), then you can stop checking. The condition is not met, and you can stop looking.
- There is one more special case: if you say *eeny* as you reach the **end** of the pile, then n_2 can be eeny compared to n_1 with **equality**, so $n_2 \leq n_1$ is a valid comparison.

In more mathematical notation, you're applying the formula:

$$G(n_1, n_2, \dots, n_k) = \begin{cases} n_k - 1 & \begin{aligned} &\text{if } n_k < n_{k-1}, \text{ or} \\ &n_k = n_{k-1} > n_{k-2}, \text{ or} \\ &n_k = n_{k-1} = n_{k-2} < n_{k-3}, \text{ or} \\ &\vdots \\ &\begin{cases} n_k = \dots = n_2 > n_1 & \text{if } k \text{ is odd} \\ n_k = \dots = n_2 \leq n_1 & \text{if } k \text{ is even} \end{cases} \end{aligned} \\ n_k & \text{otherwise} \end{cases} \quad (2)$$

for $k \geq 2$.

This looks quite complicated, but it is just a rule that lets you work out whether the Grundy value is the number of balls in the top section, or that number minus one. Some illustrations of this process are included in 2.

This formula looks unwieldy and does not capture the simplicity of the procedure in practice. Hence, we introduce some new notation to simplify the formula and streamline the proofs later on.

The piecewise function G in (2) can be seen as a more complicated version of a function f , given by

$$\begin{aligned} f(x) &= \begin{cases} f_1(x) & \text{if } p(x) \text{ is True} \\ f_2(x) & \text{otherwise,} \end{cases} \\ &= \begin{cases} f_1(x) & p(x) \\ f_2(x) & \neg p(x), \end{cases} \end{aligned}$$

where $p(x)$ is a mathematical proposition (or statement) and $\neg$ is the logical negation operator. We only consider cases where the statement takes the value **True** or **False**. We emphasise the dependence of the statement $p(x)$ on the variable x . In the second equality we suppress the notation to make it tidier, this becomes important for more complicated statements. Note, the $\neg p(x)$ could be replaced with “otherwise”, the two being logically equivalent.

2.2.1 Propositions

We use some basic results from propositional logic [4] to help us simplify our notation. For propositions p and q , write

$$\begin{aligned} (p, q) &= p \wedge q, \\ \binom{p}{q} &= p \vee q, \end{aligned}$$

where $\wedge$ and $\vee$ are the logical “and” and “or” operators, respectively. In this notation, the distributive laws for the propositions p , q and r become

$$\left(p, \binom{q}{r} \right) = \binom{(p, q)}{(p, r)}, \quad (D1)$$

$$\binom{p}{(q, r)} = \left(\binom{p}{q}, \binom{p}{r} \right). \quad (D2)$$

We omit the nested brackets when the structure is unambiguous. For example,

$$\begin{pmatrix} p \\ q, r \end{pmatrix} = \begin{pmatrix} p \\ (q, r) \end{pmatrix}.$$

Additionally, De Morgan's Laws become

$$\neg \begin{pmatrix} p \\ q \end{pmatrix} = (\neg p, \neg q), \quad (\text{DM1})$$

$$\neg (p, q) = \begin{pmatrix} \neg p \\ \neg q \end{pmatrix}. \quad (\text{DM2})$$

We will use square brackets $[\cdot]$ to signify propositions when it may be unclear. For example, given n_1 and n_2 , $[n_1 < n_2]$ is a proposition and so is $\neg[n_1 = n_2]$ which is equivalent to $\begin{pmatrix} [n_1 < n_2] \\ [n_1 > n_2] \end{pmatrix}$. Another noteworthy example is $([n_1 < n_2], [n_1 > n_2]) = \text{False}$ for any n_1 and n_2 . This is all elementary Boolean logic, but one should familiarise themselves with the notation to proceed with confidence.

Using our new notation, the formula for the Grundy value (2) becomes

$$G(n_1, n_2, \dots, n_k) = \begin{cases} n_k - 1 & \mathcal{E}_k(n_1, n_2, \dots, n_k) \\ n_k & \text{otherwise,} \end{cases} \quad (3)$$

where

$$\mathcal{E}_k(n_1, n_2, \dots, n_k) = \begin{pmatrix} [n_k < n_{k-1}] \\ [n_k = n_{k-1} > n_{k-2}] \\ [n_k = n_{k-1} = n_{k-2} < n_{k-3}] \\ \vdots \\ \begin{cases} [n_k = \dots = n_2 > n_1] & \text{if } k \text{ is odd} \\ [n_k = \dots = n_2 \leq n_1] & \text{if } k \text{ is even} \end{cases} \end{pmatrix} \quad (4)$$

for $k \geq 2$.

2.3 Example calculations

Here are some illustrations.



Figure 2: Example Grundy calculations for eight different piles of colourful Nim.

2.4 Winning (or losing)

After you've done this calculation for each pile, you'll have a collection of Grundy values $g_1, g_2, \dots, g_m$, where m is the number of piles. The best move to make is to make the bitwise XOR of all these values equal to zero, and in fact all other moves are losing.

More specifically, writing $g_i \oplus g_j$ for the bitwise XOR of two Grundy values g_i and g_j , there are two cases:

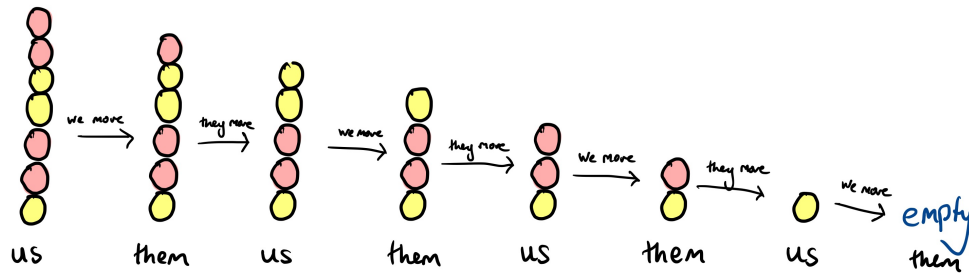
- If you calculate $g_1 \oplus g_2 \oplus \dots \oplus g_m = 0$ at the start of your turn, then you are in trouble

since if your opponent plays perfectly, you have lost. In this case, your best bet is to move into the most complicated position you can spot, hoping they won't be able to analyse it (the so-called "enough-rope" principle [1]).

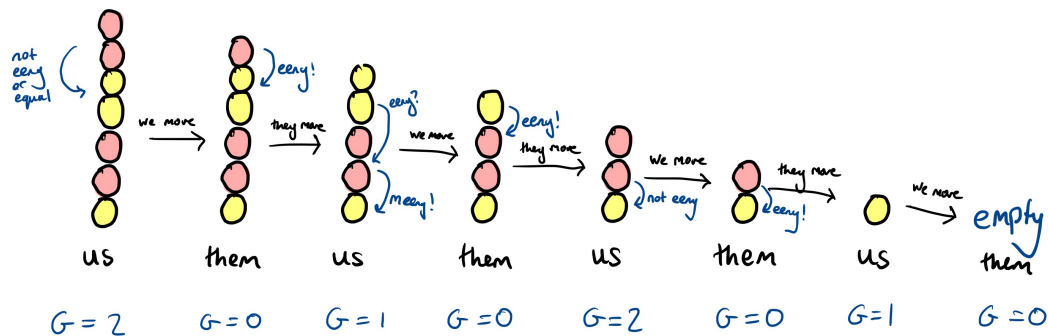
- Otherwise, $g_1 \oplus g_2 \oplus \dots \oplus g_m \neq 0$. In this case, you can force a win. To do this, like in ordinary Nim, you should play a move that changes the bitwise XOR of all the piles to be zero.

2.5 Example games

What's the best move in a game with just a single pile?



(a) An example game starting with the position (1, 2, 2, 2).



(b) The same game annotated with the Grundy value of each intermediate pile.

Figure 3: A game starting with a single pile.

Note that we follow the convention that we play the first move. You don't need the full complication of the E-M rule when there is just one pile, it's reasonably easy to see which move wins. But it is vital when there is more than one pile.

[illegible]

Figure 5: A game starting with more complicated piles.

Figure 5 shows a more complicated game, now the piles are unequal. The move here is to take just one ball from the pile on the right. There's lots of ways the game can unfold from this point, but you win in all of them.

How many winning moves are there in the game shown in Figure 6?

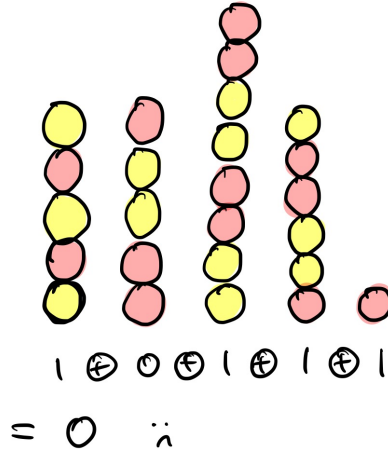


Figure 6: A game starting with five piles.

There are in fact no winning moves. This is because the Grundy values are 1, 0, 1, 1 and 1, and $1 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 0$. Any move you make will move to a position where the XOR is nonzero, and so if your opponent is using the E-M strategy, they will be able to win.

3 Proof of correctness

We now show that the eeny-meeny rule always gives the winning strategy. The Sprague-Grundy theorem [5] states that any position in an impartial game $\mathbf{b}$ is equivalent to a game of Nim of size given by the Grundy value of that position $G(\mathbf{b})$. The Grundy value of a position is defined as follows:

$$\begin{aligned} G(\mathbf{p}) &= \text{mex}\{G(\mathbf{p}') \mid \mathbf{p}' \text{ is reachable from } \mathbf{p}\}, \\ G() &= 0. \end{aligned} \tag{5}$$

If we can calculate the Grundy value for each pile in colourful Nim, say $g_1, g_2, \dots, g_m$, then we can find the Grundy value of the overall game (where the possible moves are considered across every pile) by taking the bitwise XOR of each: $g_1 \oplus g_2 \oplus \dots \oplus g_m$. As a consequence of other results, making sure that $g_1 \oplus g_2 \oplus \dots \oplus g_m = 0$ at the end of your turn is enough to guarantee a win.

Here mex is the minimum excludant [3]: the smallest non-negative integer not appearing in the set. So the function G assigns a natural number to every position. However, it is not injective since many positions can have the same Grundy value. Note also that the recursive nature of this definition makes calculating the Grundy value directly from this definition very difficult (especially without computer assistance) since you have to enumerate the entire game tree.

We will show that for an arbitrary pile $\mathbf{p}$ in colourful Nim, its Grundy value can be calculated by the eeny-meeny rule, which takes time *linear* in the size of the pile and is straightforward to do mentally.

Let $\mathcal{N}$ be the union of all positive integer sequences of finite length, or

$$\mathcal{N} = \bigcup_{k \in \mathbb{N}} \mathbb{Z}_{>0}^k = \mathbb{Z}_{>0}^0 \cup \mathbb{Z}_{>0}^1 \cup \mathbb{Z}_{>0}^2 \cup \mathbb{Z}_{>0}^3 \cup \dots,$$

with

$$\mathbb{N} = \{0, 1, 2, \dots\},$$

$$\mathbb{Z}_{>0} = \{1, 2, 3, \dots\}.$$

Note that $\mathbb{Z}_{>0}^0$ is not empty; it contains exactly one element: the empty tuple $()$. Then the space of all possible positions in the two-coloured Nim game is exactly $\mathcal{N}$. Now, the empty tuple represents the empty game, $\langle \mathbf{empty} \rangle$.

The possible moves from the position $(n_1, n_2, \dots, n_k)$ are to:

- $(n_1, n_2, \dots, n_{k-1})$, which corresponds to removing all the balls in the top section.
- For each $\ell < n_k$, $(n_1, n_2, \dots, n_k - \ell)$, which corresponds to removing ℓ balls from the top section.

In our case, we have:

$$G(n_1, n_2, \dots, n_k) = \text{mex}\left(\{G(n_1, n_2, \dots, n_{k-1})\} \cup \{G(n_1, n_2, \dots, n_k - \ell) \mid \ell < n_k\}\right). \quad (6)$$

From this it is clear that $G(n_1) = n_1$, since by induction

$$\begin{aligned} G(n_1) &= \text{mex}\{G(), G(1), \dots, G(n_1 - 1)\} \\ &= \text{mex}\{0, 1, \dots, n_1 - 1\} \\ &= n_1. \end{aligned}$$

So justifying the Eeny-Meeny rule is the same as proving the following:

$$\begin{aligned} &\text{mex}\left(\{G(n_1, n_2, \dots, n_{k-1})\} \cup \{G(n_1, n_2, \dots, n_k - \ell) \mid \ell < n_k\}\right) \\ &= \begin{cases} n_k - 1 & \text{if the E-M condition is met} \\ n_k & \text{otherwise.} \end{cases} \end{aligned} \quad (7)$$

Or equivalently:

$$\begin{aligned} &\text{mex}\left(\{G(n_1, n_2, \dots, n_{k-1})\} \cup \{G(n_1, n_2, \dots, n_k - \ell) \mid \ell < n_k\}\right) \\ &= \begin{cases} n_k - 1 & \mathcal{E}_k(n_1, n_2, \dots, n_{k-1}) \\ n_k & \text{otherwise.} \end{cases} \end{aligned} \quad (8)$$

So far, the E-M rule says nothing about piles where $k = 1$, but we can define $\mathcal{E}_1$ to be always False so that (8) is true for $k = 1$. From the recursive definition, we have

$$\begin{aligned} G(n_1) &= \text{mex}\{G(), G(1), \dots, G(n_1 - 1)\} \\ &= \text{mex}\{0, 1, \dots, n_1 - 1\} \\ &= n_1, \end{aligned}$$

and using the E-M condition

$$\begin{aligned} G(n_1) &= \begin{cases} n_1 - 1 & \mathcal{E}_1 \\ n_1 & \text{otherwise} \end{cases} \\ &= \begin{cases} n_1 - 1 & \text{False} \\ n_1 & \text{otherwise} \end{cases} \\ &= n_1, \end{aligned}$$

so the two definitions agree.

3.1 A warmup: true when $k = 2$

The E-M rule for $k = 2$ says that:

$$G(n_1, n_2) = \begin{cases} n_2 - 1 & [n_2 \leq n_1] \\ n_2 & \text{otherwise.} \end{cases}$$

We want to show that this is indeed equivalent to the actual definition:

$$G(n_1, n_2) = \text{mex}\left(\{G(n_1)\} \cup \{G(n_1, n_2 - \ell) \mid \ell < n_2\}\right).$$

To do this, we can induct on n_2 . The base case is $n_2 = 1$:

$$\begin{aligned} G(n_1, 1) &= \text{mex}\{G(n_1)\} \\ &= \text{mex}\{n_1\} \\ &= \begin{cases} 0 & [n_1 \neq 0] \\ 1 & \text{otherwise} \end{cases} \\ &= \begin{cases} 0 & [1 \leq n_1] \\ 1 & \text{otherwise,} \end{cases} \end{aligned}$$

which is indeed the E-M rule for $k = 2$.

Now consider $G(n_1, n_2)$ where $n_2 > 1$. By strong induction, we can assume that for all $G(n_1, n_2 - \ell)$, the E-M rule works, i.e.,

$$G(n_1, n_2 - \ell) = \begin{cases} n_2 - \ell - 1 & [n_2 - \ell \leq n_1] \\ n_2 - \ell & \text{otherwise.} \end{cases}$$

By the definition of G , we have that:

$$\begin{aligned} G(n_1, n_2) &= \text{mex}\left(\{G(n_1)\} \cup \{G(n_1, n_2 - \ell) \mid \ell < n_2\}\right) \\ &= \text{mex}\{G(n_1), G(n_1, 1), G(n_1, 2), \dots, G(n_1, n_2 - 1)\} \\ &= \text{mex}\{n_1, G(n_1, 1), G(n_1, 2), \dots, G(n_1, n_2 - 1)\}. \end{aligned}$$

Now we consider the cases:

Case 1: $n_2 \leq n_1$. If this is true, then it must also be true that $n_2 - \ell \leq n_1$ for all $\ell < n_2$. This means the E-M condition is satisfied for each $G(n_1, n_2 - \ell)$, hence:

$$\begin{aligned} G(n_1, n_2) &= \text{mex}\{n_1, 0, 1, \dots, n_2 - 2\} \\ &= n_2 - 1. \end{aligned}$$

So this case gives the E-M rule as we hope.

Case 2: $n_2 > n_1$. Then $n_1 = n_2 - m$ for some $m > 0$. Then:

$$\begin{aligned} G(n_1, n_2 - \ell) &= \begin{cases} n_2 - \ell - 1 & [n_2 - \ell \leq n_1] \\ n_2 - \ell & \text{otherwise} \end{cases} \\ &= \begin{cases} n_2 - \ell - 1 & [\ell \geq m] \\ n_2 - \ell & \text{otherwise.} \end{cases} \end{aligned}$$

Plugging this into the definition for $G(n_1, n_2)$, we have:

$$\begin{aligned} G(n_1, n_2) &= \text{mex}\{G(n_1), G(n_1, 1), \dots, G(n_1, n_2 - m), G(n_1, n_2 - m + 1), \dots, G(n_1, n_2 - 1)\} \\ &= \text{mex}\{n_1, 0, \dots, n_2 - m - 1, n_2 - m + 1, \dots, n_2 - 1\}. \end{aligned}$$

Since $n_1 = n_2 - m$, this means $G(n_1, n_2) = n_2$ as we hope.

Combining the cases gives us a general formula for $G(n_1, n_2)$:

$$G(n_1, n_2) = \begin{cases} n_2 - 1 & [n_2 \leq n_1] \\ n_2 & \text{otherwise,} \end{cases}$$

which is exactly the E-M rule for $k = 2$.

3.2 A useful lemma

It is possible to adapt the proof for the $k = 2$ case to show a stronger and more general result.

Lemma 3.1.

$$G(n_1, n_2, \dots, n_k) = \begin{cases} n_k - 1 & [n_k \leq G(n_1, n_2, \dots, n_{k-1})] \\ n_k & \text{otherwise.} \end{cases} \quad (\star)$$

Proof. We induct on k . The base cases are:

$$\begin{aligned} G(n_1) &= n_1, \\ G(n_1, n_2) &= \begin{cases} n_2 - 1 & [n_2 \leq G(n_1)] \\ n_2 & \text{otherwise,} \end{cases} \end{aligned}$$

which is what we have just verified above.

Now consider $G(n_1, n_2, \dots, n_k)$ where $k > 2$. By strong induction on k , we can assume that $(\star)$ holds for $G(n_1, n_2, \dots, n_{k-1})$.

$$G(n_1, n_2, \dots, n_k) = \text{mex}\left(\{G(n_1, n_2, \dots, n_{k-1})\} \cup \{G(n_1, n_2, \dots, n_k - \ell) \mid \ell < n_k\}\right).$$

We want to verify that $(\star)$ holds for this particular value of k . To do this, we now induct on n_k , like how we inducted on n_2 for the $k = 2$ case.

Now inducting on n_k , the base case we need to verify is that for $G(n_1, n_2, \dots, n_{k-1}, 1)$:

$$\begin{aligned} G(n_1, n_2, \dots, n_{k-1}, 1) &= \text{mex}\{G(n_1, n_2, \dots, n_{k-1})\} \\ &= \begin{cases} 0 & [G(n_1, n_2, \dots, n_{k-1}) > 0] \\ 1 & \text{otherwise} \end{cases} \\ &= \begin{cases} 0 & [1 \leq G(n_1, n_2, \dots, n_{k-1})] \\ 1 & \text{otherwise} \end{cases} \end{aligned}$$

as required. So that establishes the base case for the induction on n_k .

Now assume $n_k > 1$. Like in the $k = 2$ case, we'll break it down by cases again:

Case 1: $n_k \leq G(n_1, n_2, \dots, n_{k-1})$. If this is true, then it must also be true that $n_k - \ell \leq G(n_1, n_2, \dots, n_{k-1})$ for all $\ell < n_k$, and so inductively:

$$\begin{aligned} G(n_1, n_2, \dots, n_k) &= \text{mex}\{G(n_1, n_2, \dots, n_{k-1}), 0, 1, \dots, n_k - 2\} \\ &= n_k - 1 \end{aligned}$$

So $(\star)$ is correct in this case.

Case 2: $n_k > G(n_1, n_2, \dots, n_{k-1})$. Then $G(n_1, n_2, \dots, n_{k-1}) = n_k - m$ for some $m > 0$. Then:

$$\begin{aligned} G(n_1, n_2, \dots, n_k - \ell) &= \begin{cases} n_k - \ell - 1 & [n_k - \ell \leq G(n_1, n_2, \dots, n_{k-1})] \\ n_k - \ell & \text{otherwise} \end{cases} \\ &= \begin{cases} n_k - \ell - 1 & [\ell \geq m] \\ n_k - \ell & \text{otherwise.} \end{cases} \end{aligned}$$

Plugging this into the definition for $G(n_1, n_2, \dots, n_k)$, we have:

$$\begin{aligned} G(n_1, \dots, n_k) &= \text{mex}\left(\{G(n_1, \dots, n_{k-1}), G(n_1, \dots, n_{k-1}, 1), \dots, G(n_1, \dots, n_{k-1}, n_k - m), \right. \\ &\quad \left. G(n_1, \dots, n_{k-1}, n_k - m + 1), \dots, G(n_1, \dots, n_{k-1}, n_k - 1)\}\right) \\ &= \text{mex}\{G(n_1, \dots, n_{k-1}), 0, \dots, n_k - m - 1, n_k - m + 1, \dots, n_k - 1\}. \end{aligned}$$

Since $G(n_1, n_2, \dots, n_{k-1}) = n_k - m$, this means $G(n_1, n_2, \dots, n_k) = n_k$ as we hope. $\square$

3.3 A breather

Here's what we've shown so far: by definition, the Grundy value of a position is defined recursively by:

$$G() = 0, \tag{9}$$

$$G(n_1, n_2, \dots, n_k) = \text{mex}\left(\{G(n_1, n_2, \dots, n_{k-1})\} \cup \{G(n_1, n_2, \dots, n_k - \ell) \mid \ell < n_k\}\right).$$

We now know the following for certain:

$$G(n_1) = n_1 \tag{10}$$

$$G(n_1, n_2) = \begin{cases} n_2 - 1 & [n_2 \leq n_1] \\ n_2 & \text{otherwise,} \end{cases} \tag{11}$$

and we have the more general formula:

$$G(n_1, n_2, \dots, n_k) = \begin{cases} n_k - 1 & [n_k \leq G(n_1, n_2, \dots, n_{k-1})] \\ n_k & \text{otherwise.} \end{cases} \tag{*}$$

This is still a long way from proving the full E-M rule, which says that:

$$G(n_1, n_2, \dots, n_k) = \begin{cases} n_k - 1 & \mathcal{E}_k(n_1, n_2, \dots, n_k) \\ n_k & \text{otherwise,} \end{cases}$$

where

$$\mathcal{E}_k(n_1, n_2, \dots, n_k) = \begin{pmatrix} [n_k < n_{k-1}] \\ [n_k = n_{k-1} > n_{k-2}] \\ [n_k = n_{k-1} = n_{k-2} < n_{k-3}] \\ \vdots \\ \begin{cases} [n_k = \dots = n_2 > n_1] & \text{if } k \text{ is odd} \\ [n_k = \dots = n_2 \leq n_1] & \text{if } k \text{ is even} \end{cases} \end{pmatrix}$$

for $k \geq 2$. In other words, the E-M rule gives a non-recursive formula for the Grundy value of an arbitrary position.

3.4 Full correctness for any k

We have shown that the Grundy value of any position can be calculated using the formula (*). Restating it in terms of the propositions $\Omega_k(n_1, n_2, \dots, n_k)$ as defined in (15), along with the base case (since this is still recursively defined), gives us:

$$G(n_1) = \begin{cases} n_1 - 1 & \Omega_1(n_1) \\ n_1 & \neg\Omega_1(n_1), \end{cases} \tag{12}$$

$$G(n_1, n_2, \dots, n_k) = \begin{cases} n_k - 1 & \Omega_k(n_1, n_2, \dots, n_k) \\ n_k & \neg\Omega_k(n_1, n_2, \dots, n_k), \end{cases} \tag{13}$$

where

$$\Omega_1(n_1) = \text{False}, \quad (14)$$

$$\Omega_k(n_1, n_2, \dots, n_k) = [n_k \leq G(n_1, n_2, \dots, n_{k-1})] \quad (15)$$

for $k \geq 2$.

Recall that the Eeny-Meeny conditions are statements that are evaluated given a position $(n_1, n_2, \dots, n_k)$; they are defined using our propositional notation as follows:

$$\mathcal{E}_1(n_1) = \text{False} \quad (16)$$

$$\mathcal{E}_k(n_1, n_2, \dots, n_k) = \begin{pmatrix} [n_k < n_{k-1}] \\ [n_k = n_{k-1} > n_{k-2}] \\ [n_k = n_{k-1} = n_{k-2} < n_{k-3}] \\ \vdots \\ \begin{cases} [n_k = \dots = n_2 > n_1] & \text{if } k \text{ is odd} \\ [n_k = \dots = n_2 \leq n_1] & \text{if } k \text{ is even} \end{cases} \end{pmatrix}. \quad (17)$$

Hence, we wish to show the equivalence between Ω_k and $\mathcal{E}_k$. This would establish that the Eeny-Meeny rule (3) is fundamentally the same as $(\star)$ and thus, also the original recursive “mex” definition (5).

To prove this, we notice two useful results.

Lemma 3.2. For $k \geq 1$,

$$\Omega_{k+1}(n_1, n_2, \dots, n_{k+1}) = \left(\begin{array}{l} [n_{k+1} < n_k] \\ [n_{k+1} = n_k] \end{array} , \quad \neg \Omega_k(n_1, n_2, \dots, n_k) \right).$$

Proof. Using the formula for $G(n_1, n_2, \dots, n_k)$ from 13 and basic propositional logic, we have:

$$\begin{aligned} \Omega_{k+1}(n_1, n_2, \dots, n_{k+1}) &= [n_{k+1} \leq G(n_1, n_2, \dots, n_k)] \\ &= \left[n_{k+1} \leq \begin{cases} n_k - 1 & \Omega_k(n_1, n_2, \dots, n_k) \\ n_k & \neg \Omega_k(n_1, n_2, \dots, n_k) \end{cases} \right] \\ &= \left(\begin{array}{l} [n_{k+1} < n_k] \\ [n_{k+1} \leq n_k] \end{array} , \quad \begin{array}{l} \Omega_k(n_1, n_2, \dots, n_k) \\ \neg \Omega_k(n_1, n_2, \dots, n_k) \end{array} \right) \\ &= \left(\begin{array}{l} [n_{k+1} < n_k] \\ [n_{k+1} < n_k] \\ [n_{k+1} = n_k] \end{array} , \quad \begin{array}{l} \Omega_k(n_1, n_2, \dots, n_k) \\ \neg \Omega_k(n_1, n_2, \dots, n_k) \\ \neg \Omega_k(n_1, n_2, \dots, n_k) \end{array} \right) \\ &= \left(\begin{array}{l} [n_{k+1} < n_k] \\ [n_{k+1} = n_k] \end{array} , \quad \neg \Omega_k(n_1, n_2, \dots, n_k) \right). \end{aligned}$$

□

Lemma 3.3. For $k \geq 1$,

$$\neg \Omega_{k+1}(n_1, n_2, \dots, n_{k+1}) = \left(\begin{array}{l} [n_{k+1} > n_k] \\ [n_{k+1} = n_k] \end{array} , \quad \Omega_k(n_1, n_2, \dots, n_k) \right).$$

Proof. Using Lemma 3.2 and distributive laws, we have:

$$\begin{aligned}
\neg\Omega_{k+1}(n_1, n_2, \dots, n_{k+1}) &= \neg \left(\begin{array}{c} [n_{k+1} < n_k] \\ [n_{k+1} = n_k] \end{array} , \quad \neg\Omega_k(n_1, n_2, \dots, n_k) \right) \\
&= \left(\neg[n_{k+1} < n_k] , \quad \left(\begin{array}{c} \neg[n_{k+1} = n_k] \\ \Omega_k(n_1, n_2, \dots, n_k) \end{array} \right) \right) \\
&= \left(\begin{array}{c} [n_{k+1} \geq n_k] \\ [n_{k+1} \geq n_k] \end{array} , \quad \left(\begin{array}{c} [n_{k+1} < n_k] \\ [n_{k+1} > n_k] \\ \Omega_k(n_1, n_2, \dots, n_k) \end{array} \right) \right) \\
&= \left(\begin{array}{c} [n_{k+1} \geq n_k] \\ [n_{k+1} \geq n_k] \\ [n_{k+1} \geq n_k] \end{array} , \quad \begin{array}{c} [n_{k+1} < n_k] \\ [n_{k+1} > n_k] \\ \Omega_k(n_1, n_2, \dots, n_k) \end{array} \right) \\
&= \left(\begin{array}{c} [n_{k+1} > n_k] \\ [n_{k+1} > n_k] \\ [n_{k+1} = n_k] \end{array} , \quad \begin{array}{c} \Omega_k(n_1, n_2, \dots, n_k) \\ \Omega_k(n_1, n_2, \dots, n_k) \\ \Omega_k(n_1, n_2, \dots, n_k) \end{array} \right) \\
&= \left(\begin{array}{c} [n_{k+1} > n_k] \\ [n_{k+1} = n_k] \end{array} , \quad \Omega_k(n_1, n_2, \dots, n_k) \right).
\end{aligned}$$

□

Combining Lemmas 3.2 and 3.3, we can derive a second order recursive formula for Ω_k :

$$\begin{aligned}
\Omega_{k+2}(n_1, n_2, \dots, n_{k+2}) &= \left(\begin{array}{c} [n_{k+2} < n_{k+1}] \\ [n_{k+2} = n_{k+1}] \end{array} , \quad \neg\Omega_{k+1}(n_1, n_2, \dots, n_{k+1}) \right) \\
&= \left(\begin{array}{c} [n_{k+2} < n_{k+1}] \\ [n_{k+2} = n_{k+1}] \end{array} , \quad \left(\begin{array}{c} [n_{k+1} > n_k] \\ [n_{k+1} = n_k] \end{array} , \quad \Omega_k(n_1, n_2, \dots, n_k) \right) \right) \\
&= \left(\begin{array}{c} [n_{k+2} < n_{k+1}] \\ [n_{k+2} = n_{k+1}] \\ [n_{k+2} = n_{k+1}] \end{array} , \quad \begin{array}{c} [n_{k+1} > n_k] \\ [n_{k+1} = n_k] \\ \Omega_k(n_1, n_2, \dots, n_k) \end{array} \right) \\
&= \left(\begin{array}{c} [n_{k+2} < n_{k+1}] \\ [n_{k+2} = n_{k+1} > n_k] \\ [n_{k+2} = n_{k+1} = n_k] \end{array} , \quad \Omega_k(n_1, n_2, \dots, n_k) \right) \tag{18}
\end{aligned}$$

Theorem 3.4. For $k \geq 1$,

$$\Omega_k = \mathcal{E}_k.$$

Proof. We induct on k , starting with the base cases. We have for any $n_1 \in \mathbb{N}$, $\Omega_1(n_1) = \mathcal{E}_1(n_1) = \text{False}$ by definition and it can be immediately shown that $\Omega_2(n_1, n_2) = \mathcal{E}_2(n_1, n_2) = [n_2 \leq n_1]$

for any pair $(n_1, n_2) \in \mathbb{N}^2$. Now assume that for some $k \geq 1$, $\Omega_k = \mathcal{E}_k$. Then

$$\begin{aligned}
\Omega_{k+2}(n_1, n_2, \dots, n_{k+2}) &= \left(\begin{array}{l} [n_{k+2} < n_{k+1}] \\ [n_{k+2} = n_{k+1} > n_k] \\ [n_{k+2} = n_{k+1} = n_k] \end{array} \right. , \quad \Omega_k(n_1, n_2, \dots, n_k) \left. \right) \\
&= \left(\begin{array}{l} [n_{k+2} < n_{k+1}] \\ [n_{k+2} = n_{k+1} > n_k] \\ [n_{k+2} = n_{k+1} = n_k] \end{array} \right. , \quad \mathcal{E}_k(n_1, n_2, \dots, n_k) \left. \right) \\
&= \left(\begin{array}{l} [n_{k+2} < n_{k+1}] \\ [n_{k+2} = n_{k+1} > n_k] \\ [n_{k+2} = n_{k+1} = n_k] \end{array} \right. , \quad \left(\begin{array}{l} [n_k < n_{k-1}] \\ [n_k = n_{k-1} > n_{k-2}] \\ [n_k = n_{k-1} = n_{k-2} < n_{k-3}] \\ \vdots \\ \begin{cases} [n_k = \dots = n_2 > n_1] & \text{if } k \text{ is odd} \\ [n_k = \dots = n_2 \leq n_1] & \text{if } k \text{ is even} \end{cases} \end{array} \right) \left. \right) \\
&= \left(\begin{array}{l} [n_{k+2} < n_{k+1}] \\ [n_{k+2} = n_{k+1} > n_k] \\ [n_{k+2} = n_{k+1} = n_k < n_{k-1}] \\ [n_{k+2} = n_{k+1} = n_k = n_{k-1} > n_{k-2}] \\ [n_{k+2} = n_{k+1} = n_k = n_{k-1} = n_{k-2} < n_{k-3}] \\ \vdots \\ \begin{cases} [n_{k+2} = \dots = n_2 > n_1] & \text{if } k \text{ is odd} \\ [n_{k+2} = \dots = n_2 \leq n_1] & \text{if } k \text{ is even} \end{cases} \end{array} \right) \\
&= \mathcal{E}_{k+2}(n_1, n_2, \dots, n_{k+2}).
\end{aligned}$$

□

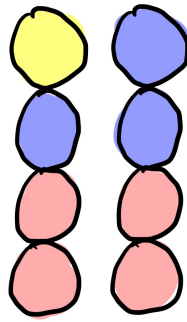
With this equivalence, we have shown that the recursive definition of the Grundy value and the eeny-meeny rule agree.

4 Further generalisations

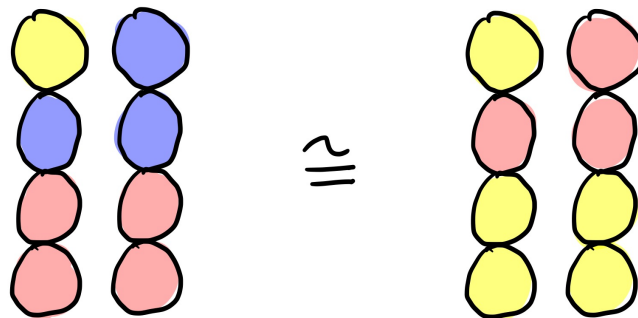
Colourful Nim is a generalisation of ordinary Nim – ordinary Nim is just the special case where there are no colour changes. But what are some ways you could generalise further?

4.1 3-colour Nim?

What about instead of having two colours, you have three?



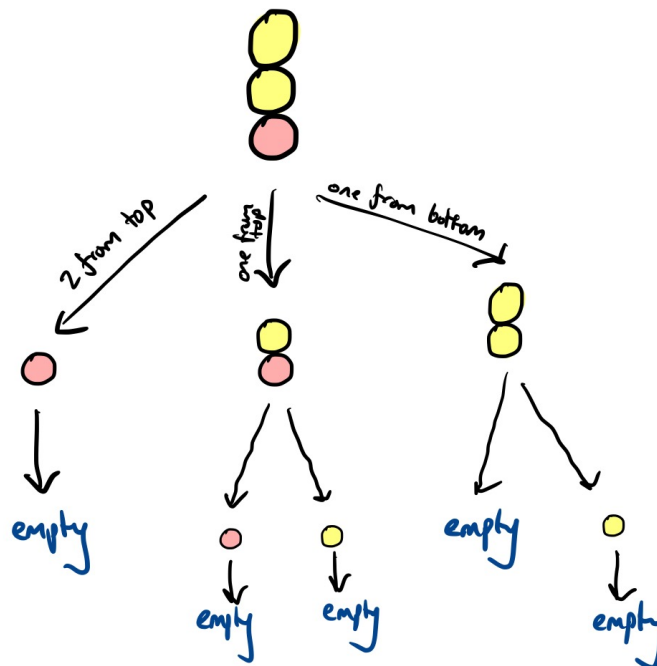
Unfortunately, this is no more interesting or difficult than having 2 colours like above. The reason for this is that the colours don't actually matter, only the colour changes. So the above game is equivalent to the 2-colour game:



So to play 3-colour Nim, just translate it into 2-colour Nim and play with the E-M rule as before.

4.2 Taking balls from both ends

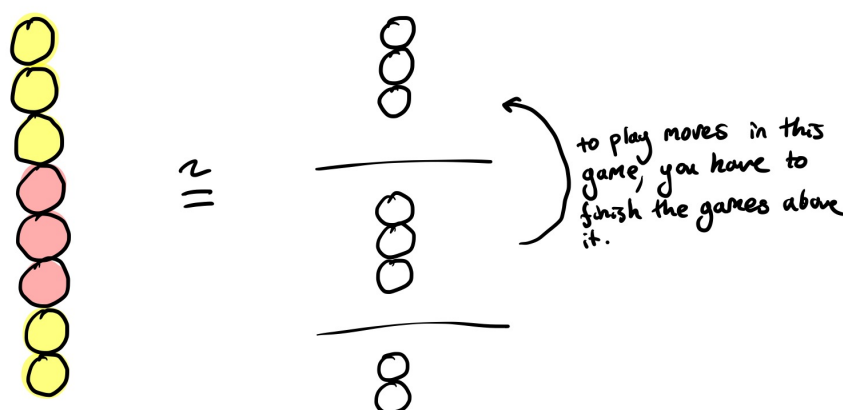
What about if you can take balls from both ends of each pile?



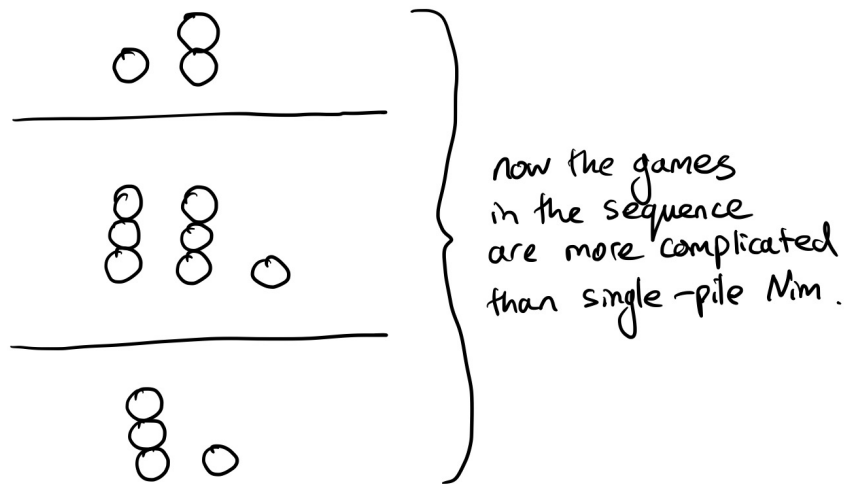
This is a bit more complicated. The general strategy would still stand, namely calculating the Grundy value of each pile and then making the XOR equal to zero. But the problem once again becomes how to efficiently calculate the Grundy value. We are still unsure of an efficient procedure for doing this.

4.3 Sequences of impartial games in general

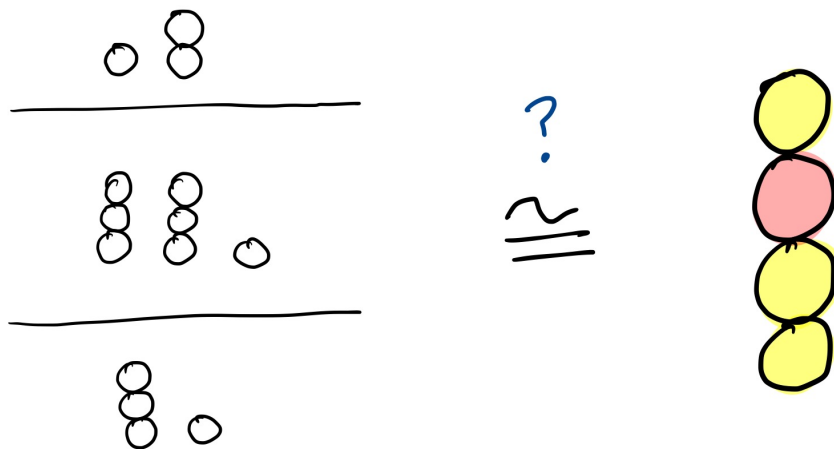
A one-pile colourful Nim game could also be seen as a sequence of one-pile Nim games, where the loser of each sub-game goes first in the next-sub game.



But what if you make each game in the sequence more complicated? For example, you could instead play a sequence of three multiple-pile Nim games:



This is also much more complicated. One thing that definitely won't work is calculating the Grundy value of each game in the sequence, and then considering the corresponding colourful Nim game, like so:



It is a reasonable guess that you can just omit these zero games and “compress” the stack before applying the rule, but this is not the case – consider the game $(1, [2, 2])$ where $[2, 2]$ denotes a game with two Nim piles of two stones each. Since $[2, 2]$ has Grundy value 0 by the XOR trick, this guess would say that the overall Grundy value is 1, and hence winnable for us. But by trying to win this game yourself, you can see that this is impossible.

But it gets worse: in fact, the Grundy value of the overall game cannot be any function of the Grundy values of each intermediate game. To see this, consider this game:

$$\left([n_1^{(1)}, n_2^{(1)}, \dots, n_{k_1}^{(1)}], [n_1^{(2)}, n_2^{(2)}, \dots, n_{k_2}^{(2)}], \dots, [n_1^{(\ell)}, n_2^{(\ell)}, \dots, n_{k_\ell}^{(\ell)}] \right),$$

i.e., a sequence of ℓ games of Nim, at the top a k_ℓ -pile Nim game, followed by a $k_{\ell-1}$ -pile game, and so on, until a k_1 -pile game. The Grundy value of this game cannot be a function of the

Grundy values of the intermediate games;

$$G\left([n_1^{(1)}, \dots, n_{k_1}^{(1)}], \dots, [n_1^{(\ell)}, \dots, n_{k_\ell}^{(\ell)}]\right) \neq f\left(n_1^{(1)} \oplus \dots \oplus n_{k_1}^{(1)}, \dots, n_1^{(\ell)} \oplus \dots \oplus n_{k_\ell}^{(\ell)}\right)$$

for any f . If this were the case, $(1, [1, 1])$ and $(1, [2, 2])$ would have the same Grundy value, but this can't be the case since $(1, [1, 1])$ is a win for the first player, while $(1, [2, 2])$ is a loss. This means, among other things, that it's not possible to come up with a uniform strategy for all sequences of impartial games just by considering the Grundy values of each impartial game in the sequence, and converting it to a game of Nim.

Another reason to expect that it is much more complicated is because a general strategy for playing sequences of games where you have the perfect strategies for the intermediate games would give you a general technique for **misère** play of any impartial game. The misère version of an impartial game is where you change the win condition so that whoever makes the last move loses.

Misère play is surprisingly often much more complicated than normal play – here's the winning strategy for misère Nim, paraphrasing from *On Numbers and Games*, Conway [3]:

“Play as you would in normal Nim, making the XOR of the heap-sizes zero, unless your move would leave only heaps of size one, (discounting empty heaps). In this case, move so as to leave either one more or one fewer one-heaps than the normal play move.”

If you had a general rule for playing sequences of impartial games, you could convert any game into its misère variant by considering the that game followed by a game of Nim with a single stone. A general procedure for converting perfect play at the normal version of an impartial game to perfect play at the misère version would be remarkable, so it's quite unlikely the generalisation to sequences of impartial games is straightforward.

5 Conclusion

In this article, we have analysed *Colourful Nim*, a new variant of Nim. We have derived an efficient procedure for determining its Sprague-Grundy values so that optimal play on any position with total height N can be computed in $O(N)$ time.

This project illustrates the tight coupling between computer search and human proof. An exhaustive search for the Grundy values of this game first highlighted the n_k or $n_k - 1$ pattern and then we converted this into a simple formula with a proof based on a nested induction on n and k .

We have also considered further generalisations with partial results: we have shown that any fixed number of colours reduces to the two-colour case and demonstrated that there is much more complexity when you allow removing stones from both ends or stacking sequences of impartial games in general.

Summary of work

- A complete, closed-form characterisation of the Grundy values of colourful Nim via the eeny-meeny rule (2).
- A short proof of its correctness that may transfer to other impartial games (3.4).
- An interactive implementation for experimentation and teaching².

²<https://projects.ollybritton.com/games/colourful-nim>

Potential for further work

- **Two-ended play.** Find an efficient formula for G when one may remove stones from both ends.
- **Sequences of impartial games.** Find an efficient formula for G given a general sequence of impartial games.

References

- [1] Elwyn R. Berlekamp, John H. Conway, and Richard K. Guy. *Winning Ways for Your Mathematical Plays*. Academic Press, 1982.
- [2] Charles L. Bouton. Nim, a game with a complete mathematical theory. *Annals of Mathematics*, 3(1/4):35–39, 1901.
- [3] J.H. Conway. *On Numbers and Games*. Ak Peters Series. Taylor & Francis, 2000.
- [4] Kevin C. Klement. Propositional logic. In *Internet Encyclopedia of Philosophy*. 2004.
- [5] Gabriel Nivasch. Sprague-grundy theory. <https://www.gabrielnivasch.org/fun/combinatorial-games/sprague-grundy>, November 2005. Migrated August 2021.