Combinatorial Games and Error-Correcting Codes

Ansh Taneja

July 7, 2026

Introduction

Introduction

By playing a game of Nim, you are connecting to one of the most fundamental methods of error correction today. By playing a game of Mogul, you are doing the same thing that prevented Voyager 1's photos from being distorted. This sounds unbelievable at first, but these seemingly different actions are fundamentally related.

Combinatorial Games

Heap Games

We can look at the game theory aspect of these codes through what are known as heap games. These games involve heaps of objects, and there are two players. On a player's turn, they have a certain set of moves they can make in relation to the heaps. If a player cannot make a valid move, they lose.

Grundy's Game

An illustrative example of a heap game is what is known as Grundy's game. At any given moment, the position of the game can be represented as the sum

$$P_a + P_b + P_c + \cdots$$

where P_k represents a heap containing k objects. On a player's turn, they can take a heap P_n and split it into $P_i + P_j$, where i and j are distinct.

Game Positions

A general heap game may be taken to have atomic positions P_i and overall position

$$P_a + P_b + P_c + \cdots ,$$

where each atomic position here represents an independent heap.

We can express the legal moves a player can make as a family of sets known as *turning sets*.

Game Positions

A general heap game may be taken to have atomic positions P_i and overall position

$$P_a + P_b + P_c + \cdots ,$$

where each atomic position here represents an independent heap.

We can express the legal moves a player can make as a family of sets known as *turning sets*.

A turning set is of the form

$$\{h, i, j, \ldots\}, h > i > j > \cdots ,$$

and it represents the move of splitting P_h into

$$P_i + P_j + \cdots .$$

In their 1986 paper, Conway and Sloane expressed this same sum in the form

$$\sum n_i P_i,$$

where the n_i account for how many times a heap with a given size appears.

In their 1986 paper, Conway and Sloane expressed this same sum in the form

$$\sum n_i P_i,$$

where the n_i account for how many times a heap with a given size appears.

The power of this alternate form is that it gives us an alternative way to represent the game's position. Namely, we can take the vector

$$(\cdots n_3 n_2 n_1)$$

as an alternative representation of the game position.

In their 1986 paper, Conway and Sloane expressed this same sum in the form

$$\sum n_i P_i,$$

where the n_i account for how many times a heap with a given size appears.

The power of this alternate form is that it gives us an alternative way to represent the game's position. Namely, we can take the vector

$$(\cdots n_3 n_2 n_1)$$

as an alternative representation of the game position.

In addition, since $x \oplus x = 0$ for all x, the outcome of any such position only depends on the parities of the n_i. This means that we can look at this code as a binary code.

The Losing Code

This means we can express the winning strategy of any game as the list of all binary vectors

$$(\cdots \zeta_3 \zeta_2 \zeta_1),$$

where each $\zeta_i = 0$ or 1, and the position represented by a vector in the list is a losing position.

The Losing Code

This means we can express the winning strategy of any game as the list of all binary vectors

$$(\cdots \zeta_3 \zeta_2 \zeta_1),$$

where each $\zeta_i = 0$ or 1, and the position represented by a vector in the list is a losing position.

An interesting result we get is the following:

Theorem

The losing code for a game is a linear code in binary.

Lexicodes

Losing Codes in Other Bases

We can represent the position of the typical heap game as

$$N = \sum \zeta_i 2^i,$$

where the ζ_i are 0 or 1.

To generalize, we just replace 2 by B.

Lexicodes

We now define a lexicographic code, or lexicode. To construct this code, we take our set of possible words and consider them in lexicographic order.

Words are rejected from the code if we have already considered some earlier word

$$N' = (\cdots \zeta_3' \zeta_2' \zeta_1')$$

for which the set of i such that $\zeta_i' \neq \zeta_i$ is a turning set.

Example

We construct an example lexicode. Here, our base is $B = 4$ and our turning sets are all of the sets of size ≤ 1:

$$\ldots 000$$

Example

We construct an example lexicode. Here, our base is $B = 4$ and our turning sets are all of the sets of size ≤ 1:

$$\ldots 000$$

$$\ldots 011$$

Example

We construct an example lexicode. Here, our base is $B = 4$ and our turning sets are all of the sets of size ≤ 1:

$$\ldots 000$$
$$\ldots 011$$
$$\ldots 022$$

Example

We construct an example lexicode. Here, our base is $B = 4$ and our turning sets are all of the sets of size ≤ 1:

$$\ldots 000$$
$$\ldots 011$$
$$\ldots 022$$
$$\ldots 033$$

Example

We construct an example lexicode. Here, our base is $B = 4$ and our turning sets are all of the sets of size ≤ 1:

$$\ldots 000$$

$$\ldots 011$$

$$\ldots 022$$

$$\ldots 033$$

$$\ldots 101$$

Example

We construct an example lexicode. Here, our base is $B = 4$ and our turning sets are all of the sets of size ≤ 1:

$$\ldots 000$$
$$\ldots 011$$
$$\ldots 022$$
$$\ldots 033$$
$$\ldots 101$$
$$\ldots 110$$

Example

We construct an example lexicode. Here, our base is $B = 4$ and our turning sets are all of the sets of size ≤ 1:

$$\ldots 000$$
$$\ldots 011$$
$$\ldots 022$$
$$\ldots 033$$
$$\ldots 101$$
$$\ldots 110$$
$$\ldots 132,$$

Example

We construct an example lexicode. Here, our base is $B = 4$ and our turning sets are all of the sets of size ≤ 1:

$$\ldots 000$$
$$\ldots 011$$
$$\ldots 022$$
$$\ldots 033$$
$$\ldots 101$$
$$\ldots 110$$
$$\ldots 132,$$

and so on.

Lexicodes and Losing Codes

If we look at a few losing codes and lexicodes, we can notice the following:

Theorem

The lexicode with respect to a certain family of turning sets and a base B is the same as the losing code for the same family of turning sets and base B.

The proof of this is by induction.

Linearity of Lexicodes

Lexicodes have remarkable structures that we can explore. We have the following two theorems:

Theorem

Any lexicode with base $B = 2^n$ is closed under nim-addition.

Theorem

If B is a Fermat power of 2, or 2^{2^a}, then the lexicode defined by any family of turning sets is closed under scalar nim-multiplication by numbers α in the range $0 \leq \alpha < B$.

We now have a general structure for lexicodes with bases that are powers of 2. However, other bases are not so organized. For example, take the following lexicode with base 3, where the turning sets are all of the sets of size ≤ 1:

We now have a general structure for lexicodes with bases that are powers of 2. However, other bases are not so organized. For example, take the following lexicode with base 3, where the turning sets are all of the sets of size ≤ 1:

$$\begin{gathered} \ldots 000 \\ \ldots 011 \\ \ldots 022 \\ \ldots 101 \\ \ldots 110 \\ \ldots 202, \end{gathered}$$

and so on.

We can see here that the sum of the third and fourth words is not in the code.

Lexicodes and Error-Correcting Codes

Turning Sets II

Here, we will explore how lexicodes map to error-correcting codes.

We first define the turning sets of a lexicode using an integer d. When $d = 2$, for example, the family of turning sets consists of all sets with cardinality < 2.

Hamming Codes

We start with the following theorem by Conway and Sloane:

Theorem

When $B = 2^{2^a}$ and $d = 3$, lexicodes of length

$$n = 1 + B + B^2 + \cdots + B^{m-1} = \frac{B^m - 1}{B - 1}$$

are Hamming codes, and those of other lengths are shortened Hamming codes.

Binary Golay Codes

We also have the following theorem:

Theorem

When $B = 2$, $d = 8$ *and* $n = 24$, *the code produced is the* $[24, 12, 8]$ *Golay code, where* $[24, 12, 8]$ *means that the codewords are of length* 24, *have dimension* 12, *and have minimal distance* 8.

The Third Perfect Code

We have now constructed two of the three nontrivial perfect codes using just the basic definition of lexicodes. A natural question to ask is whether we can produce the third of these codes; namely, the ternary $[11, 6, 5]$ Golay code.

The Third Perfect Code

We have now constructed two of the three nontrivial perfect codes using just the basic definition of lexicodes. A natural question to ask is whether we can produce the third of these codes; namely, the ternary $[11, 6, 5]$ Golay code.

However, this is made difficult by the fact that the structure of lexicodes is not immediately clear in ternary. However, recent research made by Yuki Irie has shown that the structure of these lexicodes still has potential.

The Basis Modification

Under the standard basis $E = \{e_0, e_1, e_2, \dots\}$, a vector is evaluated based on the numerical value $N = \sum \zeta_i B^i$, where $\zeta_i \in \{0, 1, 2, \dots, B-1\}$.

The Basis Modification

Under the standard basis $E = \{e_0, e_1, e_2, \dots\}$, a vector is evaluated based on the numerical value $N = \sum \zeta_i B^i$, where $\zeta_i \in \{0, 1, 2, \dots, B-1\}$.

Irie introduces a modified basis $F = \{f_0, f_1, f_2, \dots\}$ for the vector space we look at. Specifically, the basis F is defined as:

$$f_\xi = e_\xi + e_9,$$
$$f_i = e_i \text{ for all } i \neq \xi.$$

The Ternary Golay Code

Using this new basis, we have the following theorem:

Theorem

For base $B = 3$, minimal distance $d = 6$, and length 12, the lexicode generated using the modified basis F is the $[12, 6, 6]$ extended ternary Golay code.

Lexicodes of Constant Weight

The Construction

We now explore lexicodes with an additional restriction; instead of greedily selecting codewords from the full space $\mathbb{F}_B^n$, they instead select from all words in $\mathbb{F}_B^n$ with weight w. We refer to these codes as *constant weight lexicodes*. Here, we will only consider the binary case, so B is 2.

To construct a constant weight lexicode with weight w, length n, and Hamming distance d, we look at all words of weight w with length n in order, and accept a word if it is has a Hamming distance of at least d from all previously accepted words in the code.

The Game

The game we consider here has parameters w and $d = 2t \geq 4$. The typical position of this game can be represented as a set

$$\{a_1, a_2, \ldots, a_w\}$$

of distinct nonnegative integers. A legal move is to decrease $1, 2, \ldots,$ or $t - 1$ of these integers while keeping all integers in the set distinct.

We now have the following theorem:

Theorem

For any $d = 2t \geq 4$ and any w, the losing code for the above game is the same as the constant weight lexicode with the same d and w.

If we look at the constant weight lexicode with $w = d = 8$ and $n = 24$, a verification by computer yields the following:

Theorem

The words of the lexicode with $w = d = 8$ and $n = 24$ are the blocks, or octads, of a Steiner system $S(5, 8, 24)$.

In addition, by imposing another restriction on the lexicode, we obtain the following:

Theorem

The words of the constant weight lexicode with $w = 6$, $d = 4$, $n = 12$ with minimum sum $s = 21$ are the blocks, or hexads, of a Steiner system $S(5, 6, 12)$.

Further Discoveries

Additional Progress

There is still much more to be discovered about lexicodes. We have covered the theory of lexicodes for bases that are powers of 2, but for other bases, the structure is less clear. If we can generalize this appropriately, there are many more connections between game theory and error correction that are waiting to be discovered.

Thank You!

Questions?