# COMBINATORIAL GAMES AND ERROR-CORRECTING CODES

ANSH TANEJA

Abstract. Combinatorial game theory and coding theory appear to be disparate fields, yet they share a profound connection through structures known as lexicographic codes, or lexicodes. First introduced by Conway and Sloane, lexicodes are generated by applying a greedy algorithm to a vector space, effectively translating the winning strategies of impartial heap games into robust error-correcting codes. We explore this bridge in detail, beginning with the foundations of the Sprague-Grundy theory of combinatorial games and nim-addition, while also exploring alternative ways of representing the position of a game. We demonstrate how lexicodes constructed over bases that are Fermat powers of 2 naturally form linear codes, and introduce the notion of nim-multiplication. Then, we look at how certain combinatorial games' losing codes are error-correcting codes, including the classical Hamming codes and the perfect binary Golay code. Furthermore, we incorporate recent research by Yuki Irie to show that this greedy construction is not limited to bases that are powers of two; by applying a basis modification, we provide a proof that the algorithm successfully generates the perfect ternary Golay code in base 3. Finally, we analyze constant weight lexicodes, presenting rigorous proofs of their connection to Welter's Game and demonstrating how these restricted codes can generate highly symmetric combinatorial designs, specifically the $S(5, 8, 24)$ and $S(5, 6, 12)$ Steiner systems. Ultimately, this paper synthesizes classical results with modern developments to highlight the enduring utility of game theory in understanding optimal error-correcting codes.

## 1. Introduction

Combinatorial games and error-correcting codes are two relatively distinct fields; from an outside perspective, one may never guess that they could be somehow linked. Yet, in 1986, John H. Conway and Neil J. A. Sloane published a paper that managed to connect these two far-flung fields, using an intermediate structure which they referred to as a *lexicode* to connect the two. Essentially, they took a game's winning strategy and turned it into a code. They also proved that preexisting error-correcting codes, such as Hamming codes, were in fact codes that related to certain mathematical games.

This paper will cover their results, and include proofs of certain theorems which were omitted in their 1986 paper. We will start by covering the game theoretic aspect of the codes they discussed, then delve into the theory behind lexicodes and some of their properties. We will then show that certain lexicodes are the same as already-discovered error-correcting codes. Afterwards, we will look at a new type of lexicode with a restriction on the weight of the codewords, and build a theory for certain cases. We will also see how these codes relate to previously-discovered Steiner systems. Finally, we will conclude with a brief discussion of the applications of lexicodes and discuss matters which may be of further interest to the reader.

## 2. Combinatorial 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.

An illustrative example of a heap game is what is known as Grundy's game, explained in [Gru39]. 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 and split it into two heaps of distinct sizes. That is, they can take a heap $P_n$ and split into

$$P_i + P_j,$$

where $i \leq j$, $i \neq j$, and $i+j = n$. The game ends when a player is no longer able to split a pile.

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

$$P_i + P_j + \cdots.$$

Using this notation, the turning sets for Grundy's game would be

$$\{\{3,2,1\},\{4,3,1\},\{5,4,1\},\{5,3,2\},\ldots\}.$$

Heap games and the theory behind them have been well studied. In this paper, we will take the following results from [Gru39] and [Spr35]:

**Theorem 1.** *We define a function $G(P)$, which assigns integer positions to positions $P$. Specifically, $G(P) = mex(G(Q), G(R), \ldots)$, where mex is the "minimum excluded value", or the smallest integer not in the specified set, and $Q, R, \ldots$ are the positions reachable from $P$ in exactly one move. We also refer to $G(P)$ as the G-value of $P$. $G(P)$ has the following properties:*

1. *A player wins by moving to positions where the G-value is 0.*
2. *$G(P_a + P_b + P_c + \cdots) = G(P_a) \oplus G(P_b) \oplus G(P_c) \oplus \cdots$. Here, the operation $\oplus$ is nim-addition, also known as "binary addition without carry" or "exclusive-or".*

*Proof.* We proceed by mathematical induction to prove the properties of the function $G(P)$, utilizing the definitions of $G(P)$ and the minimum excluded value ($mex$).

First, we prove the winning condition. By the rules of normal play, the game must end in a finite number of moves, and the first player unable to move loses. A terminal position has no available moves, so its set of reachable $G$-values is empty. The $mex$ of an empty set

is 0. Therefore, the terminal, losing position always has a $G$-value of 0.

Now, if a player is in a position where $G(P)$ is greater than 0, the definition of *mex* guarantees that 0 must be in the set of reachable $G$-values. This means the player always has a legal move to a position where $G(Q)$ equals 0. Conversely, if a player is in a position where $G(P)$ equals 0, the definition of *mex* guarantees that 0 is excluded from the set of reachable $G$-values. This ensures that any legal move will hand the opponent a position with a $G$-value greater than 0.

Consequently, a player can guarantee a win by consistently moving to a position with a $G$-value of 0, and this concludes the proof of the first part of the theorem.

Next, we prove the additive property for a game composed of multiple independent subgames. Let us analyze a combined game consisting of two independent subgames, $A$ and $B$. A legal move in the combined game $A + B$ consists of making a valid move in exactly one of the subgames while leaving the other untouched. Therefore, any subsequent position from $A + B$ must take the form of either $A' + B$ (where $A'$ is a valid next position in $A$) or $A + B'$ (where $B'$ is a valid next position in $B$).

Applying the *mex* definition to this combined game, we find that the $G$-value of the total board is the *mex* of the $G$-values of all these possible next positions. We can express this mathematically by stating that $G(A+B)$ is the *mex* of the set containing all $G(A'+B)$ and all $G(A + B')$.

We now apply strong induction, assuming that our additive property already holds true for all game states that precede our current position. Then, we can substitute the $G$-values of the next positions with their nim-sums. This allows us to rewrite our equation so that $G(A + B)$ equals the *mex* of the set containing all $G(A') \oplus G(B)$ and all $G(A) \oplus G(B')$.

Because $G(A)$ is defined as the *mex* of all reachable $G(A')$ states, the set of available $G(A')$ values must contain every integer strictly less than $G(A)$. The same is true for $G(B')$. However, we know the definition of the nim-sum of two numbers $x$ and $y$ is $mex(x \oplus y', x' \oplus y)$ over all nonnegative integer values of $x' < x$ and $y' < y$ (see [Con76].)

Therefore, our equation for the combined game fits the definition of nim-addition, which implies that $G(A + B)$ is equal to $G(A) \oplus G(B)$. Repeatedly applying this identity gives us the desired conclusion.

This concludes the proof of the second part of the theorem, and we are done. ■

This theorem essentially gives us a way to solve any impartial game, and will help us characterize the codes we discover later on.

Recall our expression for the position of a heap game:

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

In [CS86], 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; for example, if $n_2 = 3$, then there are three heaps with size 2.

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.
Conway and Sloane then noticed that since $x \oplus x = 0$ for all $x$, the outcome of any such position only depended on the parities of the $n_i$. Essentially, two piles of the same size cancel each other out.

Using this, they expressed 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 depending on the parity of the number of piles of size $i$. All of the vectors in their list also had the property that

$$\sum \zeta_i G(P_i) = 0,$$

where the sum here is in the nim sense. By Theorem 1, this means that every position in the list is a losing position.
Conway and Sloane referred to this as the *winning code*; however, seeing as this code consists of the positions where the first player loses, we will call this the *losing code* instead.

Now, we can deduce the following theorem:

**Theorem 2.** *The losing code for a game is a linear code over $\mathbb{F}_2$, or in binary.*

*Proof.* For our code to be a linear code, it must include the zero vector, be closed under addition (or in our case, nim-addition), and closed under scalar multiplication.

We see that the codeword $(\cdots 000)$ is in the losing code for any game. This is because when all the piles are 0, there is no legal move for Player 1 to make, and so they lose. Since our code includes all losing positions, it must include the zero vector.

Then, since each vector in the losing code has a nim-sum of 0, the nim-sum of 2 such vectors is also 0, and therefore the losing code is closed under nim-addition.

Finally, multiplying by a scalar does not change the nim-sum whatsoever; therefore, the losing code is closed under scalar multiplication and we are done. ■

With this, we move on to the bridge between combinatorial games and error-correcting codes: lexicographic codes, or lexicodes.

## 3. Lexicodes

We start by considering losing codes in base $B$. In [CS86], Conway and Sloane generalized the definition of a game to base $B$, and we use that definition here.

A game can be defined given a base $B$ and a family of finite turning sets of the form

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

Thinking back to our earlier representation of the losing code of a game, we can represent the position of the typical heap game as

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

where the $\zeta_i$ are 0 or 1. The legal moves for this game are described below in the generalization of this representation.

Now, for the base $B$ generalization, we describe a position as a number

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

where

$$\zeta_i = 0, 1, 2, \ldots, B - 1.$$

Equivalently, we can take the vector

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

A legal move is to move to the position

$$N' = \sum \zeta_i' B^i$$

provided that

(1) $N' < N$
(2) The set of all $i$ such that $\zeta_i' \neq \zeta_i$ is a turning set.

Note that the conditions do not specify how much a new position must differ from the original position. In addition, the Sprague-Grundy theory still applies to these games, and we can define the losing code for these games in a similar manner to how we defined them earlier.

For each family of turning sets and each base $B$, Conway and Sloane also defined a code known as a lexicographic code, or lexicode. To construct this code, we take our set of possible words $(\cdots \zeta_3 \zeta_2 \zeta_1)$, $0 \leq \zeta_i < B$, and consider them in the lexicographic order determined by the corresponding number $N = \sum \zeta_i B^i$. (This is just numerical order, least to greatest.) 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. Otherwise, the word is included in the code.

We now prove one of the main theorems that Conway and Sloane discovered about lexicodes:

**Theorem 3.** *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 $B$.*

*Proof.* We proceed by induction on the position $N$.

First, if $N$ is not in the lexicode, there must have been some smaller $N'$ for which the move from $N$ to $N'$ was a turning set. By our induction hypothesis, this means that the move from $N$ to $N'$ was a winning move, and thus $N$ is not a losing position.

Second, if $N$ is in the lexicode, and $N$ to $N'$ is a legal move, then $N' < N$ and the list of $i$ for which $\zeta_i' \neq \zeta_i$ is a turning set. Since we must have rejected each of these $N'$, by our induction hypothesis, they were all winning positions and thus $N$ is in the losing code. This completes the proof. ■

By combining Theorems 2 and 3, we can see that for $B = 2$, any lexicode is closed under nim-addition. However, we can generalize this:

**Theorem 4.** *Any lexicode with base $B = 2^n$ is closed under nim-addition.*

*Proof.* We consider the case where $B = 8$, and the proof is very similar for the other cases. We can split an octal digit into 3 binary digits, as follows:

| Octal Digit | Binary Digits |
|---|---|
| 0 | 000 |
| 1 | 001 |
| 2 | 010 |
| 3 | 011 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |

In this way, the original base 8 game becomes a binary game. In this binary game, $T$ is a turning set if

$$\left\{ \left\lfloor \frac{i}{3} \right\rfloor : i \in T \right\}$$

was a turning set in the base 8 game. (Here, $\lfloor x \rfloor$ is the integer part of $x$.) Since we know that this new game's losing code is closed under nim-addition, the original base 8 game's losing code must be as well, and we are done. ■

Now, we have that lexicodes with certain bases are closed under nim-addition. However, in [Con76], Conway introduces the notion of nim-multiplication and defines the operations of nim-addition and nim-multiplication using the logic of *mex*. In addition, he shows that these operations turn the integers into a field with characteristic 2.
In particular, Conway defines nim-addition as

$$a \oplus b = mex(a' \oplus b, a \oplus b')$$

where the *mex* is taken over all $a' < a$ and $b' < b$. He also defines nim-multiplication as

$$a \otimes b = mex(a' \otimes b \oplus a \otimes b' \oplus a' \otimes b').$$

Again, this is taken over all $a' < a$ and $b' < b$. Furthermore, nim-multiplication happens before nim-addition.
Using this definition, we deduce the following result:

**Theorem 5.** *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 $\alpha$ in the range $0 \leq \alpha < B$.*

*Proof.* Since $B = 2^{2^a}$, the set $\{0, 1, \ldots, B-1\}$ is a field under nim-addition $\oplus$ and nim-multiplication $\otimes$ [Con76]; in particular $\otimes$ distributes over $\oplus$, a fact we use below.

Define $f(\zeta, P)$ to be the $G$-value of the position with the single digit $\zeta$ (where $0 \leq \zeta < B$) in coordinate $P$,

$$f(\zeta, P) = G(\ldots, 0, 0, \zeta, 0, \ldots, 0),$$

and write $f(P) = f(1, P)$. Because the lexicode is exactly the set of positions of $G$-value 0, and $G$ is additive across coordinates (Theorem 1), the code is closed under scalar nim-multiplication by $\alpha$ provided

$$f(\alpha, P) = \alpha \otimes f(P)$$

for all $\alpha$ and $P$.
We prove this statement by double induction on $\alpha$ and on $P$.

From the position with $\alpha$ in coordinate $P$, a legal move changes a turning set $\{P, Q_1, Q_2, \ldots\}$ of coordinates: it lowers $\alpha$ to some $\zeta'$ with $0 \leq \zeta' < \alpha$ and raises each $Q_i$ from 0 to some $n_i$ with $0 < n_i < B$. By the additivity of the function $G(P)$,

$$f(\alpha, P) = \operatorname{mex}\Big(f(\zeta', P) \oplus \sum_i f(n_i, Q_i)\Big),$$

where the mex is taken over all $\zeta' < \alpha$ and all turning sets $\{P, Q_1, \ldots\}$ with nonzero entries $n_i$.

Applying the induction hypothesis to our above equation - namely $f(\zeta', P) = \zeta' \otimes f(P)$ since $\zeta' < \alpha$, and $f(n_i, Q_i) = n_i \otimes f(Q_i)$ since $Q_i < P$ - gives

$$f(\alpha, P) = \operatorname{mex}\Big(\zeta' \otimes f(P) \oplus \sum_i n_i \otimes f(Q_i)\Big).$$

If we change the parameters of our earlier equation to $\alpha = 1$, $\zeta' = 0$ (so the first term vanishes), we can see that

$$f(P) = \operatorname{mex}\Big(\sum_i n_i \otimes f(Q_i)\Big),$$

so the values $\sum_i n_i \otimes f(Q_i)$ realized by the moves are exactly $\{0, 1, \ldots, f(P) - 1\}$. Hence every $\lambda < f(P)$ can be written as $\lambda = \sum_i n_i \otimes f(Q_i)$ for some legal move.

We now expand the target $\alpha \otimes f(P)$ independently. By the definition of nim-multiplication

and the distributive law,

$$\begin{aligned}\alpha \otimes f(P) &= \operatorname{mex}\big(\zeta' \otimes f(P) \oplus \alpha \otimes \mu \oplus \zeta' \otimes \mu\big) \\ &= \operatorname{mex}\big(\zeta' \otimes f(P) \oplus (\alpha \oplus \zeta') \otimes \mu\big),\end{aligned}$$

the mex taken over all $\zeta' < \alpha$ and all $\mu < f(P)$.

Now, we need to check that our expressions for $f(\alpha, P)$ and $\alpha \otimes f(P)$ have the same set of arguments.

Fix $\zeta' < \alpha$; then $\alpha \oplus \zeta' \neq 0$, since $\zeta' < \alpha$ forces $\zeta' \neq \alpha$. As each $n_i$ ranges over all nonzero field elements, so does $(\alpha \oplus \zeta') \otimes n_i$, because multiplication by the nonzero constant $\alpha \oplus \zeta'$ permutes the nonzero elements of the field. Therefore, using our earlier finding about $\lambda$ and distributivity,

$$\begin{aligned}\big\{(\alpha \oplus \zeta') \otimes \mu : \mu < f(P)\big\} &= \Big\{(\alpha \oplus \zeta') \otimes \textstyle\sum_i n_i \otimes f(Q_i)\Big\} \\ &= \Big\{\textstyle\sum_i (\alpha \oplus \zeta') \otimes n_i \otimes f(Q_i)\Big\} = \Big\{\textstyle\sum_i n_i \otimes f(Q_i)\Big\} = \big\{\lambda : \lambda < f(P)\big\}.\end{aligned}$$

So for each $\zeta'$ the inner argument-set of $\alpha \otimes f(P)$ coincides with that of $f(\alpha, P)$, which means that the two mex expressions agree:

$$f(\alpha, P) = \alpha \otimes f(P).$$

This completes the induction and establishes our desired result, so we are done. ■

Remarkably, we now have a general structure for lexicodes with bases that are powers of 2 or Fermat powers of 2. However, other bases are not so yielding. For example, take the following lexicode with base 3, where the turning sets are all of the sets of size $\leq 1$:

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

and so on. However, the sum of the third and fourth words is not in the code, and this code is not linear.
Progress is still being made when it comes to the structure of lexicodes in other bases, but for now, the structure we have discovered can lead us to some interesting results.

## 4. Lexicodes and Error Correcting Codes

Here, we will explore how lexicodes map to certain types of 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 $< 2$. We also define a length $n$ so that the lexicode consists of all appropriate $n$-dimensional vectors.

We start with the following theorem by Conway and Sloane:

**Theorem 6.** *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.*

*Proof.* We know that a lexicode of this form is a linear code in base $B$ because it is closed under nim-addition and nim-multiplication. We call this linear code $C$.

Since $C$ is a linear code, it can be characterized by an $m \times n$ parity-check matrix $H$ over $\mathbb{F}_B$ [MS77], such that

$$C = \{x \in \mathbb{F}_B^n \mid Hx^T = 0\}.$$

Let the columns of $H$ be $h_1, h_2, h_3, \ldots, h_n$. Because of how our lexicode is made, we know the following about our columns:

(1) No column consists of all 0s.
(2) $h_i \neq \alpha \cdot h_j$ for any scalar $\alpha \in \{0, 1, 2, \ldots, B-1\}$.

This implies that every column $h_i$ is essentially a representative of a 1-dimensional vector subspace in $\mathbb{F}_B^m$. Each such subspace contains exactly $B - 1$ non-zero vectors.

The parity-check matrix $H$ has $m$ rows. Therefore, there are $B^m$ possible vectors that can fill the columns. However, we also have that the columns are non-zero, so our total count for how many possible columns we can have is $B^m - 1$.

We know that no two of our columns are linearly dependent because of our value for $d$. This implies that the maximum length for our code is

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

as we can only select one vector from each subspace. When this is the length of our lexicode, it is the same as a Hamming code.

Furthermore, if the length is less than our maximum, then we can see that the resulting code is a shortened Hamming code and we are done. ■

Using just our theory of lexicodes from earlier and some basic facts about error-correcting codes, we now have the result that one of the most well-known error-correcting codes and one of the three nontrivial perfect codes is a lexicode. Furthermore, when $a = 1$ (and hence $B = 2$), the game that produces the binary Hamming code is Nim, one of the most well-known heap games [CS86].

We now move on to a slightly more complex code, known as the Golay code.

**Theorem 7.** *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.*

*Proof.* We will show in turn that $L$ is linear, that its minimal distance is 8, and that its dimension is 12.

The fact that $L$ is linear follows directly from Theorem 4, so $L$ is a binary linear code of length 24.

We now pin down the minimal distance. Because $L$ is linear, the distance between any two codewords is the weight of their nim-sum. We also know that their nim-sum is a codeword. The algorithm that generates our lexicodes accepts a word only when it lies at Hamming distance at least 8 from every previously accepted word. Now, if we assume that one of the two words in question is the zero codeword, we see that every nonzero codeword has weight at least 8. Conversely, the first nonzero word the algorithm accepts is the numerically smallest word at distance 8 from the zero codeword, which has weight exactly 8. Therefore, the minimal distance of $L$ is exactly 8.

We now find the dimension by using a bounding argument.
We can bound the dimension from above using the sphere-packing bound. Since the minimal distance is 8, the code corrects 3 errors, so the Hamming balls of radius 3 centered at the codewords are pairwise disjoint. Each such ball contains

$$\binom{24}{0} + \binom{24}{1} + \binom{24}{2} + \binom{24}{3} = 1 + 24 + 276 + 2024 = 2325$$

words, so

$$|L| \cdot 2325 \leq 2^{24},$$

This means that

$$|L| \leq 7216 < 2^{13}.$$

Since $L$ is linear, $|L| = 2^{\dim L}$, and therefore $\dim L \leq 12$.

For the lower bound, the construction of our lexicode produces 12 linearly independent codewords. By the definition of the dimension, $\dim L \geq 12$.

Combining the two inequalities we have for the dimension gives us that $\dim L = 12$, so the dimension of our lexicode is 12.

Now, we have that $L$ is a binary linear code with parameters $[24, 12, 8]$. By [Ple68], the extended Golay code is the unique code with these parameters up to equivalence, so $L$ is the extended $[24, 12, 8]$ Golay code, and we are done. ■

The corresponding game for the above code is called Mogul; more details about it can be found in [BCG82].

If we remove the last digit from the code we just looked at, we get the perfect binary Golay code, one of the three perfect error-correcting codes.

A natural question now is whether we can construct the third perfect code, namely the ternary Golay code. However, one major problem we face is that in ternary, lexicodes are not nearly as nice - at least, not at first.

We noted earlier that lexicodes constructed over bases that are not powers of 2 do not

inherently guarantee linearity. However, this is only what we get when we naïvely apply the standard greedy algorithm. Recent work by Yuki Irie [Iri26] demonstrates that this issue can be bypassed by applying a slight modification to the order in which we look at vectors.

To understand how this works, we first define the notion of a *basis*. This is a collection of vectors that defines a vector space. These basis vectors are linearly independent. Furthermore, we can reach any point in our vector space through a linear combination of our basis vectors.

Now, we define the standard lexicographical ordering for vectors in $\mathbb{F}_3^n$ slightly more rigorously, using our new idea of a basis.
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. We fix a specific column index $\xi \in \{0, 1, \dots, 8\}$ (Irie proves that any of these columns will yield the same result; if this notion feels unnatural, just select a number from 0 to 8 and use that column index.) Here, "column index" means that our first column has index 0, our second has index 1, and so on. The modified basis is constructed by replacing the $\xi$-th standard basis vector with the sum of itself and the 10th standard basis vector (index 9).

Specifically, the basis $F$ is defined as:

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

Candidate words are now evaluated in the lexicographical order induced by this new basis $F$, sorting by $N = \sum \zeta_i f_i$.

Using this new basis, we have the following theorem:

**Theorem 8.** *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.*

*Proof.* We apply the greedy algorithm to $\mathbb{F}_3^{12}$, evaluating candidate words in the lexicographic order induced by the modified basis $F$ and accepting a word whenever it lies at Hamming distance at least $d = 6$ from every previously accepted word.

We first establish linearity. Because the ordering is induced by the modified basis $F$, Theorem 4 does not apply directly, so we instead verify by computer (Appendix A) that the accepted set is closed under vector addition and under scalar multiplication mod 3. This means that $L$ is a linear subspace of $\mathbb{F}_3^{12}$. (Irie has a synthetic proof of this in [Iri26].)

The minimal distance is 6: as in the binary case, linearity makes the distance between two codewords equal to the weight of their difference, which is again a codeword, and the acceptance rule forces every nonzero codeword to have weight at least 6, while the first nonzero accepted word has weight exactly 6.

We apply a a bounding argument similar to the one in the previous proof to determine the dimension.
We can bound the dimension from above using the sphere-packing bound over $\mathbb{F}_3$. Since the minimal distance is 6, the code corrects 2 errors, so the Hamming balls of radius 2 centered at the codewords are pairwise disjoint. Each such ball contains

$$\binom{12}{0}2^0 + \binom{12}{1}2^1 + \binom{12}{2}2^2 = 1 + 24 + 264 = 289$$

words, so

$$|L| \cdot 289 \leq 3^{12},$$

which means that

$$|L| \leq 1838 < 3^7.$$

Since $L$ is linear over $\mathbb{F}_3^{12}$, $|L| = 3^{\dim L}$, and therefore $\dim L \leq 6$.

For the lower bound, the greedy construction itself produces 6 linearly independent codewords, so $\dim L \geq 6$.

Combining our two bounds, we get that the dimension of $L$ is 6.

Now, we have that $L$ is a linear code over $\mathbb{F}_3$ with parameters $[12, 6, 6]$. By [Ple68], the extended ternary Golay code is the unique code with these parameters up to equivalence, so $L$ is the extended ternary Golay code. ■

We see that the punctured code obtained by deleting the last coordinate of this code is the perfect ternary Golay code. This mean that we have now constructed all three nontrivial perfect codes using our simple greedy algorithm approach (and a few more tricks.)

Now, by adding additional constraints to how we construct our lexicodes, we can construct more systems essential to coding theory.

## 5. Constant Weight Lexicodes

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. Here, the distance $d$ is necessarily even.

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. Because of the distinctness condition, we can no longer easily describe these games using turning sets.

We now have the following theorem:

**Theorem 9.** *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 $w$.*

The proof of this theorem is analogous to the proof of Theorem 3, so we omit it.

For $d = 2$, the distance condition is automatically satisfied by the weight condition, so the constant weight lexicode with length $n$, minimal distance 2, and weight $w$ consists of all length-$n$ words with weight $w$.

The only other case for which we have a complete theory is $d = 4$. In this case, the game that this code corresponds to is Welter's Game.

Welter's game is just the $t = 2$ case of the game discussed above; here, a legal move is to take one of the piles and decrease the number of objects in it while ensuring that all piles have a distinct number of objects in them.

The solution to this game uses Welter's function [Wel54], which can be defined recursively as

$$\begin{aligned} W(a_1) &= a_1, \\ W(a_1, a_2) &= (a_2 \oplus a_1) - 1, \\ &\text{and} \\ W(a_1, a_2, \ldots, a_{k+1}) &= W(a_2, a_3, \ldots, a_k) \oplus [W(a_1, a_2, \ldots, a_k) \oplus W(a_2, a_3, \ldots, a_{k+1})] - 1. \end{aligned}$$

Note that the $-1$ in the formula for Welter's function is regular subtraction, not nim-subtraction. Therefore, the definition of Welter's function uses both nim-addition and regular operations.

Before we prove that the $d = 4$ code corresponds to Welter's game, we first prove the following:

**Lemma 10.** *If $W(a_1, a_2, \ldots, a_w) = n$, and $n' \neq n$, there are unique nonnegative integers $a'_1, a'_2, \ldots, a'_w$ such that*

$$a_1, a_2, \ldots, a_w, a'_1, a'_2, \ldots, a'_w$$

*are all distinct and that*

$$W(a_1, a_2, \ldots, a_w) = n$$

*remains true if any even number of the letters $a_1, a_2, \ldots, a_w, n$ are replaced by the corresponding primed letters. Additionally, an even number of the inequalities*

$$a_1' < a_1$$
$$a_2' < a_2$$
$$a_3' < a_3$$
$$\ldots$$
$$a_w' < a_w$$
$$n' < n$$

*are true.*

*Proof.* We will prove this lemma by strong induction on $w$, the number of variables.

Before we begin, we state a fundamental property of nim-addition:

If $x \oplus y = S$, and we are given a new value $S' \neq S$, then setting $x' = S' \oplus y$ and $y' = S' \oplus x$ yields the unique nonnegative integers for which

$$x' \oplus y = S'$$
$$x \oplus y' = S'$$
$$x' \oplus y' = S.$$

Additionally, an even number of the following inequalities are true:

$$x' < x$$
$$y' < y$$
$$S' < S.$$

To see this, observe that $x \oplus x' = y \oplus y' = S \oplus S'$, so all three pairs differ in exactly the same bit positions; let $b$ be the highest such bit. For any pair $(u, u')$ differing at top bit $b$, we have $u' < u$ if and only if bit $b$ of $u$ equals 1. Since $S = x \oplus y$ gives $S_b = x_b \oplus y_b$, the number of true statements among $x' < x$, $y' < y$, $S' < S$ equals $[x_b] + [y_b] + [x_b \oplus y_b]$, which is 0 when $x_b = y_b = 0$ and 2 in each of the other three cases. In every case this count is even.

We now proceed with the induction.

We start with the case $w = 1$.
By the definition of Welter's function, $W(a_1) = a_1$. We are given that $W(a_1) = n$, which simply means $a_1 = n$. We are also given an $n' \neq n$. Now, we need to find a unique $a_1'$ for which $W(a_1') = n'$, which implies that $a_1' = n'$.

Now, we have the equation $W(a_1) = n$.
If we replace 0 variables, we get $W(a_1) = n$, which is true.
If we replace 2 variables, we get $W(a_1') = n'$. Since $a_1' = n'$, this evaluates to $n' = n'$, which is true.

Now, we must check the inequalities $a_1' < a_1$ and $n' < n$. Because $a_1 = n$ and $a_1' = n'$,

these two inequalities are the same. Therefore, they are either both true, or both false, so the number of true inequalities is 0 or 2 — in either case an even number. This proves the base case for $w = 1$.

Now, we move to the case $w = 2$. By definition, $W(a_1, a_2) = (a_1 \oplus a_2) - 1$.
We are given $W(a_1, a_2) = n$, which implies $a_1 \oplus a_2 = n + 1$.

Let $S = n + 1$, so $a_1 \oplus a_2 = S$. We are given an $n' \neq n$. Let us define $S' = n' + 1$.
By the property of nim-addition stated above, there exist unique $a_1'$ and $a_2'$ such that:

$$a_1' \oplus a_2 = S',$$
$$a_1 \oplus a_2' = S',$$
$$\text{and}$$
$$a_1' \oplus a_2' = S.$$

In every valid scenario, exactly two variables were replaced by their primed versions, satisfying the "even number of replacements" rule.

Now, the property of nim-addition guarantees that an even number of the inequalities $a_1' < a_1$, $a_2' < a_2$, and $S' < S$ are true. Because subtracting 1 from both sides does not change the truth of an inequality, $S' < S$ is identical to $n' < n$. Therefore, an even number of the required inequalities are true and our lemma holds for $w = 2$.

Now, we proceed with the inductive step. Assume the lemma holds for all sizes up to $w - 1$. We will now prove it holds for $w$.
Using the recursive definition of Welter's function, we let:

$$x = W(a_1, a_2, \ldots, a_{w-1})$$
$$y = W(a_2, a_3, \ldots, a_w)$$
$$z = W(a_2, a_3, \ldots, a_{w-1})$$

The recurrence relation can now be written as

$$n = z \oplus ((x \oplus y) - 1).$$

Let $S = x \oplus y$. We can substitute this into our equation:

$$n = z \oplus (S - 1).$$

Adding $z$ to both sides (where the addition here is nim-addition), we can isolate $S$:

$$S = (n \oplus z) + 1.$$

Now, suppose we change $n$ to $n'$. Because $z$ depends only on the inner variables (which we can consider constant for now), changing $n$ to $n'$ forces a change in $S$ to a new value

$$S' = (n' \oplus z) + 1,$$

and $S' \neq S$ because $n' \neq n$ implies $n' \oplus z \neq n \oplus z$.

Since $x = W(a_1, \ldots, a_{w-1})$ and $y = W(a_2, \ldots, a_w)$ are Welter functions of size $w - 1$, the

fundamental nim-addition property applied to $x \oplus y = S$ with the new value $S'$ produces unique $x', y'$, and the inductive hypothesis converts each of the changes $x \to x'$ and $y \to y'$ into a corresponding, unique change of the underlying variables $a_i \to a_i'$. These determine the unique primed variables $a_1', \ldots, a_w'$ asserted by the lemma, and tracking which letters are replaced through the recursion shows that any valid substitution replaces an even number of the letters $a_1, \ldots, a_w, n$.

Finally, the parity count is inherited through the recursion: the base nim-addition property contributes an even number of true inequalities at the $S' < S$ level, each size-$(w-1)$ block contributes an even number by the inductive hypothesis, and since $S' < S$ is equivalent to $n' < n$, combining these even contributions across the overlapping blocks (removing the shared inner comparisons, which cancel in pairs) leaves an even number of the inequalities $a_1' < a_1, \ldots, a_w' < a_w$, $n' < n$ true.

Therefore, by strong induction, the unique $a_i'$ variables exist, any valid substitution involves an even number of primed letters, and an even number of the inequalities are true for all $w$. ■

We can now use this to prove the following theorem:

**Theorem 11.** *$(a_1, a_2, \ldots, a_w)$ is a losing position for Welter's game, or equivalently the vector with 1's in positions $(a_1, a_2, \ldots, a_w)$ is in the lexicode with constant weight $w$ and minimal distance 4, if and only if*

$$W(a_1, a_2, \ldots, a_w) = 0.$$

*Proof.* When the $W$-value of your position is some $n \neq 0$, we take $n' = 0$. Since $n' = 0 < n$, the inequality $n' < n$ is true. By Lemma 10, an even number of the inequalities $a_1' < a_1, \ldots, a_w' < a_w$, $n' < n$ are true; as $n' < n$ is one true inequality, an odd (hence nonzero) number of the coordinate inequalities $a_i' < a_i$ must also be true. Thus there is at least one $i$ with $a_i' < a_i$, and decreasing coordinate $i$ to $a_i'$ (adjusting an even number of the accompanying letters to their primed values so that the $W$-value becomes $n' = 0$) is a legal move from a position with $W \neq 0$ to a position with $W = 0$. Conversely, from a position with $W = 0$ no single decrease can reach another position of $W$-value 0, so every legal move leads to $W \neq 0$.

For the converse, we show that a single legal move can never preserve the $W$-value; equivalently, that Welter's function is injective in each coordinate. Because a position is a set of distinct nonnegative integers, $W$ is well-defined regardless of the order in which we list its arguments (see [Wel54]), so it suffices to show that for fixed $a_2, \ldots, a_w$ the map $a_1 \mapsto W(a_1, a_2, \ldots, a_w)$ is injective.

We proceed by strong induction on $w$.

We start with the case $w = 1$. By definition,

$$W(a_1) = a_1,$$

which is injective.

Now we move to the case $w = 2$. By definition,

$$W(a_1, a_2) = (a_1 \oplus a_2) - 1.$$

For a fixed $a_2$, the map $a_1 \mapsto a_1 \oplus a_2$ is a bijection of the nonnegative integers. Furthermore, since the entries are distinct, $a_1 \neq a_2$, which forces $a_1 \oplus a_2 \geq 1$; therefore subtracting 1 preserves injectivity, and the map is injective.

Now, we proceed with the inductive step. Assume the claim holds for all sizes up to $w - 1$. We will now prove it holds for $w$.

Using the recursive definition of Welter's function, we let:

$$\begin{aligned} x &= W(a_1, a_2, \ldots, a_{w-1}) \\ y &= W(a_2, a_3, \ldots, a_w) \\ z &= W(a_2, a_3, \ldots, a_{w-1}). \end{aligned}$$

The recurrence relation can now be written as

$$W(a_1, a_2, \ldots, a_w) = z \oplus [(x \oplus y) - 1].$$

Here, $z$ and $y$ do not depend on $a_1$, while $x$ does. By the inductive hypothesis, $x$ is injective in $a_1$. Additionally, the two size-$(w-1)$ tuples $(a_1, a_2, \ldots, a_{w-1})$ and $(a_2, a_3, \ldots, a_w)$ differ in exactly one coordinate, so by the inductive hypothesis their $W$-values are distinct; that is, $x \neq y$, and hence $x \oplus y \geq 1$. Therefore $(x \oplus y) - 1$ is a nonnegative integer that is injective in $a_1$, and taking the nim-sum with the fixed value $z$ preserves this injectivity. Thus $W$ is injective in $a_1$, completing the induction.

We now return to the game. Suppose

$$W(a_1, a_2, \ldots, a_w) = 0,$$

and consider any legal move. Such a move decreases exactly one coordinate, replacing some $a_j$ with a value $a_j' \neq a_j$ while keeping all entries distinct. By the injectivity of $W$ in the $j$-th coordinate, the resulting position has $W$-value $\neq 0$. Hence every legal move from a position of $W$-value 0 leads to a position of nonzero $W$-value.

Combining this with the forward direction, the positions of $W$-value 0 are exactly the losing positions, and the desired result follows. ■

Outside of $d = 4$, the resulting codes have no general structure. There are, however, certain special cases which we will look at in more detail.

Recall the Golay codes from the previous section: we know that the lexicode with minimal distance 8 and length 24 is the extended $[24, 12, 8]$ Golay code. If we break down the

weight of each codeword, we see that there are

$$\begin{aligned} 1 \text{ codeword with weight } 0, \\ 759 \text{ codewords with weight } 8, \\ 2576 \text{ codewords with weight } 12, \\ 759 \text{ codewords with weight } 16, \\ \text{and } 1 \text{ codeword with weight } 24. \end{aligned}$$

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

**Theorem 12.** *The words of the lexicode with $w = d = 8$ and $n = 24$ are the blocks, or octads, of a Steiner system $S(5, 8, 24)$.*

See Appendix A for the code used to verify the above result.

We have now obtained one of the Steiner systems that Mathieu studied [Mat73] . It is natural to ask whether we can produce the other one, namely, the $S(5, 6, 12)$ system [Mat61]. The answer is yes, but to do so, we must add another restriction to our lexicode.

We define a *constant weight lexicode with minimum sum* to be a lexicode with minimal distance $d$, weight $w$, length $n$, and minimum sum $s$. The conditions on the codewords are the same for $w$, $d$, and $n$, but they have to satisfy an additional condition based on $s$.

Say $a_i$ is the value of the $i$th digit of a codeword; for a codeword to be considered valid, it has to meet the weight, distance, and length requirements, while also meeting the following:

$$\sum_{a_i=1} i \geq s.$$

Here, addition is in the regular sense.

Now, we have the following theorem:

**Theorem 13.** *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 $S(5, 6, 12)$.*

This theorem is also verified by computer; the code can be found in Appendix A. The game that produces this code is known as Mathematical Blackjack; more details can be found in [BCG82]. We also note that this Steiner system is closely related to the ternary Golay code discussed earlier.

## 6. Applications and Further Reading

Lexicodes are not just a triviality; rather, they have a wide variety of applications. For example, the Hamming codes that we discussed in earlier sections form the backbone of modern error correction, and without them, modern digital communication would be heavily prone to errors. The extended binary Golay code (and the very closely connected system $S(5, 8, 24)$) were used to ensure that the photos sent to Earth from Voyager 1 were not distorted or misinterpreted, while the ternary Golay code has applications in quantum computing and quantum error correction. The other Steiner system that we looked at, $S(5, 6, 12)$, has

modern-day applications in cryptography.

There is of course much more to cover in terms of lexicodes and error-correcting codes. Recall that we spent some time looking at nim-addition and nim-multiplication; Conway covers the details of those operations, as well as a different way of looking at lexicodes, in [Con90]. More details on error-correcting codes and their foundation can be found in [Ham50], and one can look at the Steiner systems mentioned in this paper in [Wit38].

In addition, 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. However, new progress is being made in this regard: Irie, in addition to helping uncover the construction for the ternary Golay code, has also done work on the conditions needed for base $B$ lexicodes to be linear [Iri26]. This suggests that there may be more to discover about the structure of lexicodes with bases that are not powers of 2.

Ultimately, lexicodes and game theory give us another way to look at error-correcting codes, and can help give us insight on how these codes work. As research into lexicodes continues, it is entirely possible that we may end up discovering more fascinating results that help us see the connections between game theory and error-correcting codes.

## Acknowledgments

The author would like to thank Simon Rubinstein-Salzedo and Trenton von Sosen for their help in the creation of this paper.

## Appendix A. Programs used to Verify Theorems

The programs used to verify Theorems 12 and 13, as well as part of Theorem 8, can be found in the Github repository linked here. The programs in question were made using the help of Claude Opus 4.8.

## References

[BCG82] Elwyn R. Berlekamp, John H. Conway, and Richard K. Guy. *Winning Ways for Your Mathematical Plays*, volume 1. Academic Press, London, 1982.

[Con76] John H. Conway. *On Numbers and Games*. Number 6 in London Mathematical Society Monographs. Academic Press, London and New York, 1976.

[Con90] John H. Conway. Integral lexicographic codes. *Discrete Mathematics*, 83(2–3):219–235, 1990.

[CS86] John H. Conway and Neil J. A. Sloane. Lexicographic codes: Error-correcting codes from game theory. *IEEE Transactions on Information Theory*, 32(3):337–348, 1986.

[Gru39] P. M. Grundy. Mathematics and games. *Eureka*, 2:6–8, 1939.

[Ham50] Richard W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, 1950.

[Iri26] Yuki Irie. On linear lexicographic codes: Ninth column construction of the ternary golay code, 2026.

[Mat61] Émile Mathieu. Mémoire sur l'étude des fonctions de plusieurs quantités, sur la manière de les former et sur les substitutions qui les laissent invariables. *Journal de Mathématiques Pures et Appliquées*, 6:241–323, 1861.

[Mat73] Émile Mathieu. Sur la fonction cinq fois transitive de 24 quantités. *Journal de Mathématiques Pures et Appliquées, Série 2*, 18:25–46, 1873.

[MS77] Florence J. MacWilliams and Neil J. A. Sloane. *The Theory of Error-Correcting Codes*. Elsevier, 1977.

[Ple68] Vera Pless. On the uniqueness of the golay codes. *Journal of Combinatorial Theory*, 5(3):215–228, 1968.

[Spr35] Roland P. Sprague. Über mathematische Kampfspiele. *Tohoku Mathematical Journal*, 41:438–444, 1935.

[Wel54] C. P. Welter. The theory of a class of games on a sequence of squares, in terms of the advancing operation. *Indagationes Mathematicae (Proceedings)*, 57:194–200, 1954.

[Wit38] Ernst Witt. Über Steinersche Systeme. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 12(1):265–275, 1938.