Combinatorial Constructions

1. Error-Correcting Codes and Nim

Suppose Alice wants to send a message to Bob. For the sake of simplicity, suppose she wants to sent over a string of bits (Bob can convert them back into alphanumeric characters using some agreed upon conversion method). However, transmitting a message isn't perfect. Oftentimes, bits can get corrupted on the way over. For example, if Alice wants to sent over the string 01101, during transmission, the 3rd and 4th bits could get flipped, so Bob would receive 01010. Obviously they don't want this, so can they devise a method such that Bob, given a message Alice sent through her computer, can recover the message Alice intended to send? This is where error-correcting codes come into play.

An error-correcting code is able to take the received method and give back the original, intended message. However, the complexity of these codes do depend on how many bits can potentially get corrupted during transmission, so for now we focus on the case where at most 1 bit gets corrupted.

There's an easy way to make sure the receiver can decipher the original message: send the intended message three times. We can also send it twice and add a parity bit. But there is a better way to send codes, depending on the length of the string.

Definition (perfect code): A code of length n correcting at most k errors is said to be a *perfect code* if, for every string s of length n, there is exactly one code word c such that s and c differ in at most k places.

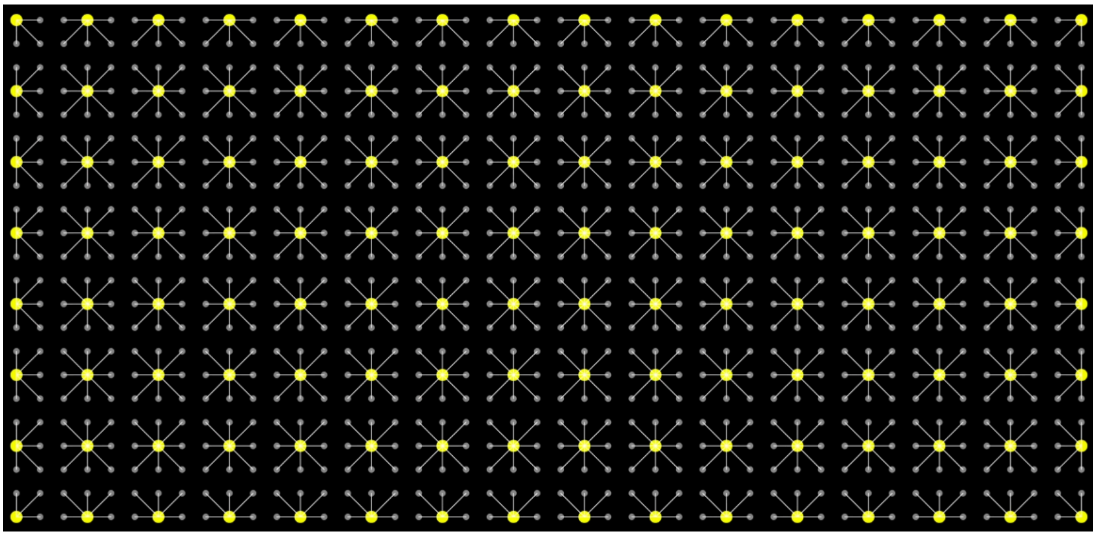

Figure 1: Credit: 3Blue1Brown. The yellow nodes represent the words that are in the perfect code. We assume that during transmission at most one bit gets flipped.

So if $k = 1$, every bit string should be at most one bit flip away from a code word. Given a code word with length n, there are $n + 1$ words reachable via at most one bit flip, so the number of code words is at most $\frac{2^n}{n+1}$. In fact, we need this to be an integer, since otherwise some words aren't within one bit flip from a code word. Thus, the set of possible n is $1, 3, 7, 15, \ldots$. Note that this condition is necessary for a perfect code to exist, but not sufficient. There's no gurantee, for example, that there exists a perfect code for $n = 7$. However, we'll see this is not the case.

We'll use the image to help aid our intuition of what the code should look like. Note that any code word (the yellow nodes), are each 3 but flips away. Why is that?

If we have two code words that only differ by 1 bit, and we receive one of those words, we have no idea whether one of the bits flipped to reach the other one or if there was no error at all. If we have two code words that differ by 2 bits, and we receive a word where one of those bits is flipped, we don't know which bit to flip to get back to the original, since the word we received can be reached from both code words. Thus, if we have a perfect code, each word must differ in at least three bits. This distance has a special name.

Definition (Hamming distance): Let C be a code. For two code words $c, d \in C$, the *Hamming distance* between c and d is the number of positions at which c and d differ.

In particular, for a perfect code that corrects errors of at most one bit, every pair of code words must have Hamming distance at least 3.

In the case of $n = 7$, we obtain the following perfect code:

0000000	0000111	0011001	0011110
0101010	0101101	0110011	0110100
1001011	1001100	1010010	1010101
1100001	1100110	1111000	1111111

In fact, these binary strings that form the perfect code for $n = 7$ have a surprising connection to Nim. Suppose we have a binary string $\cdots a_3 a_2 a_1$. Then the digit a_n denotes the number of piles of size n. Note that since two piles of the same size in Nim are equivalent to 0 and are irrelevant to determine who wins, given a general Nim position, we throw out every pair of piles of the same size. Thus every Nim position can indeed be written as a binary string.

The moves we're allowed to do in Nim is to remove a pile or remove some amount of stones from a pile. What this does to the binary representation of the game is either change a 1 to a 0 or change a 1 to a 0 while adding 1 to a lower bit, respectively. In other words, we can change one or two bits, as long as the number we create in binary decreases in value.

What we claim is that the codes in our perfect code for $n = 7$ are exactly the $\mathcal{P}$ positions in Nim where each pile has size at most 7, and we can show this using the partition theorem.

Theorem: The $\mathcal{P}$ positions of Nim when each pile has size at most 7 are exactly the code words in the perfect code for $n = 7$ that corrects at most one bit flip.

Proof: We move through the binary strings in lexographic order. First put $0000000 \in \mathcal{P}$ into $\mathscr{P}$. Then, if we reach a string that is Hamming distance at least 3 away from every word currently in $\mathscr{P}$, put it in $\mathscr{P}$. Otherwise put it in $\mathscr{N}$. Thus, the first few strings in $\mathscr{N}$ are

$$0000001, 0000010, 0000011, \ldots$$

In order to prove the theorem, we must show that there is a move from every $\mathscr{N}$ position to a $\mathscr{P}$ position, and from a $\mathscr{P}$ position we can only reach a $\mathscr{N}$ position.

Suppose we're at a $\mathscr{N}$ position. By construction, there is some word before it in $\mathscr{P}$ that it is Hamming distance at most 2 away from (since being in $\mathscr{N}$ is defined by not being at least Hamming distance 3 away from every string in $\mathscr{P}$). Thus we can reach a $\mathscr{P}$ position from every $\mathscr{N}$ position.

Now suppose we're at a $\mathscr{P}$ position. By constrcution, every word lexographically before it in $\mathscr{P}$ is at least Hamming distance 3 away from it. Thus, it's impossible to reach one of those positions in a turn, since we can only change at most 2 bits. Thus, every move from a $\mathscr{P}$ position leads to a $\mathscr{N}$ position. ■

Note also that when two code words are nim-summed (equivalenty XORed), we get another code word. We can interpret this in terms of Nim by noting that the sum of two $\mathcal{P}$ position is still a $\mathcal{P}$ position. This turns out to be a more general phenomena in code words with certain bases and minimal Hamming distances that we'll explore later, where nim-summing the code words is replaced with adding them as vectors and taking the components modulo the base we're working in.

1.1. Heap Games and Turning sets

Before we delve into more general codes, we discuss heap games, which will be useful for discussing these more general codes.

In a heap game, a position consists of a set of heaps of any size. A position with a pile of size k is denoted P_k, and a position with piles of size $a, b, c, \ldots$ is denoted with $P_a + P_b + P_c + \cdots$. Let R be a set of sets. A set in R is $\{h, i, j, \ldots\}$ where $h > i > j > \cdots$. We refer to R as the set of *turning sets*. A turn in a game consists of picking a turning set in R, and replacing P_h with $P_i + P_j + \cdots$. For example, in Nim with at most 7 stones in a pile, we have

$$R = \{\{1,0\}, \{2,0\}, \{2,1\}, \{3,0\}, \{3,1\}, \ldots\{7,5\}, \{7,6\}\}$$

Since a heap game is impartial, we can use Grundy values. In particular, if we can move from P to $Q, R, \ldots$, then

$$\mathcal{G}(P) = \operatorname{mex}\{\mathcal{G}(Q), \mathcal{G}(R), \ldots\}$$

and

$$\mathcal{G}(P_a + P_b + \cdots) = \mathcal{G}(P_a) \oplus \mathcal{G}(P_b) \oplus \cdots$$

1.2. Codes in Base B

Suppose we're given a number $N = \sum \zeta_i B^i$ written in base B, where $\zeta_i = 0, 1, \ldots, B - 1$, and we're given a finite set of turning sets. Then a legal move is to replace N with $N' = \zeta_i' B^i$, where

1. $N' < N$
2. the set of i such that $\zeta_i' \neq \zeta_i$ is a turning set.

We construct a *lexicode* by using a similar greedy algorithm to the code of length 7. We move through numbers in base B in lexographic order. We reject a number $\ldots\zeta_3\zeta_2\zeta_1$ if for some number already in the code, the collection of i where the numbers differe is a turning set. Otherwise we accept the code.

We can check and see that indeed this same algorithm can we used on the base 2 lexicode with 7 numbers, where we use out finite number of Nim turning sets.

Now we have the following theorem, due to Conway and Sloane.

Theorem: For any turning set and any base, the winning moves in the game are to move to positions corresponding to the codewords in the lexicode.

Proof: This proof is essentially the same as the proof we used for the base 2 length 7 code. Let the code constructed be the $\mathscr{P}$ positions, and everything else be a $\mathscr{N}$ position. We need to show that every move from a $\mathscr{P}$ positions goes to $\mathscr{N}$, and there exists a move from every $\mathscr{N}$ position to a $\mathscr{P}$ position, and then we're done by the partition theorem.

If we have a position $N \in \mathscr{P}$, then a legal move consists of moving to N', with the collection of differing positions being a turning set and $N' < N$. Thus we must have rejected N' from the code, and so any move from N will move to a position in $\mathscr{N}$.

Now suppsoe we have a position $N \in \mathscr{N}$. Then by construction, there msut be a smaller number N' in $\mathscr{P}$ for which the collection on positions they differ at is a turning set. Thus there exists a move from $\mathscr{N}$ to $\mathscr{P}$. ∎

Example: Let $B = 8$ and let the turning sets all be of size 1 and 2. Then applying the greedy algorithm, we get the following code:

$$\begin{gathered} 0000 \\ 0111 \\ 0222 \\ \vdots \\ 0777 \\ 1012 \\ 1103 \\ \vdots \end{gathered}$$

Note that turning sets consisting of all possible turning sets with size $1, 2, 3, \ldots, d - 1$ will create a code that has minimal Hamming distance d.

We also have the following proposition:

Proposition: Given a base 2 code defined by any family of turning sets, then the code is closed under component wise mod 2 addition.

Proof: Since the code words are all $\mathcal{P}$ positions, and this componenwise addition is just the nimsum operation, we're essentially adding two $\mathcal{P}$ positions together, which again must be a $\mathcal{P}$ position. ∎

2. Wythoff's Nim

We now pivot to an intersection of combinatorial game theory and number theory.

The game of Wythoff's Nim is played on two piles. On a move, you can either take as many stones as you want from one pile, or the same numbers of stones from both piles. Positions in Wythoff Nim are represented as (a, b). Thus, the possible positions that can be moved to are (a', b), (a, b'), and $(a - k, b - k)$, where $0 \leq a' < a$, $0 \leq b' < b$, and $1 \leq k \leq \min(a, b)$.

From here, we can find a few small $\mathcal{P}$ positions. Note that if $(a, b) \in \mathcal{P}$, then $(b, a) \in \mathcal{P}$. Thus we can just focus on $a \leq b$. The first few $\mathcal{P}$ positions are

$$(0, 0), (1, 2), (3, 5), (4, 7), (6, 10), (8, 13).$$

If we plot these points on the plane, they lie very close to two lines through the origin, one with slope φ and the other with slope $\frac{1}{\varphi}$, where φ is the golden ratio. This suggests that that $\mathcal{P}$ positions have coordinates $(\lfloor \varphi n \rfloor, \lfloor \varphi^2 n \rfloor)$.

There are two useful lemmas that can be used to prove this.

Lemma: The $\mathcal{P}$ positions of Wythoff's Nim are the pairs of the form (a_n, b_n) for $n \geq 0$, which are defined recursively by

$$a_n = \operatorname{mex}\{a_i, b_i : 0 \leq i < n\} \quad \text{and} \quad b_n = a_n + n.$$

Proof: We use the partition theorem. Let the positions that are created by the recursion be the $\mathscr{P}$ positions, and everything else be the $\mathscr{N}$ positions.

First we show no move from an $\mathscr{P}$ positions goes to an $\mathscr{P}$ position. Suppose there's a move from (a_n, b_n) to some (a_m, b_m) or (b_m, a_m). Since the minimum of the coordinated can't increase, we must have $m < n$. By construction, we also can't have $a_n = a_m$ or $a_n = b_m$. Thus the move must be removing stones from both piles. However, this also can't work, since when we remove the same number of stones from both piles, the difference betwene the piles stays the same, and by construction we have $|a_m - b_m| = m < n = |a_n - b_n|$.

Now suppose we're at a $\mathscr{N}$ position, which means the positions isn't equal to (a_n, b_n) or (b_n, a_n). We need to show there exists a move to a $\mathscr{P}$ position. Without loss of generality, suppose we're at (a, b) with $a \leq b$.

First suppose $a = a_n$. If $b > b_n$, then we can move to (a_n, b_n), which is a $\mathscr{P}$ position. Otherwise, $a \leq b < b_n$. Let $m = b - a$. Then $m < n$, so $a_m < a_n$, so there's a move to (a_m, b_m).

Now suppose $a \neq a_n$. Then by construction, we must have $a = b_n$. Thus $a_n < a = b_n$, so there's a move to (b_n, a_n), as desired. ■

Lemma (Rayliegh's Theorem): Let α and β be positive irrationals such that $\frac{1}{\alpha} + \frac{1}{\beta} = 1$. Then the sets

$$A = \{\lfloor n\alpha \rfloor : n > 0\} \quad \text{and} \quad B = \{\lfloor n\beta \rfloor : n > 0\}$$

partition the positive integers.

Proof: Call and elements in $(A \cap B)$ a collision and elements in $\mathbb{Z}_{>0} \setminus (A \cup B)$ an anti-collision. We need to show that both of these sets are empty.

Suppose for the sake of contradiction that there exists a collision $j = \lfloor m\alpha \rfloor = \lfloor n\beta \rfloor$. Thus

$$j < m\alpha < j + 1 \quad \text{and} \quad j < n\beta < j + 1.$$

We don't have equality on the left since α, β are irrational. Dividing each by α, β and adding yields

$$\frac{j}{\alpha} + \frac{j}{\beta} = j < m + n < j + 1 = \frac{j+1}{\alpha} + \frac{j+1}{\beta}.$$

However, $m + n$ is an integer, and there are no integers between j and $j + 1$, so we have a contradiction.

Now suppose j is an anti-collision. Thus there exists m, n such that

$$m\alpha < j \quad \text{and} \quad j + 1 < (m + 1)\alpha,$$
$$n\beta < j \quad \text{and} \quad j + 1 < (n + 1)\beta.$$

Again, equality on the right is impossible. Dividing by the irrationals again and adding the corresponding inequalities yield

$$m + n < j \quad \text{and} \quad j + 1 < m + n + 2 \Rightarrow m + n < j < m + n + 1,$$

and again this is a contradiction. ■

Theorem: The $\mathcal{P}$ positions of Wythoff's Nim are positions of the form (a, b) and (b, a) with

$$a = \lfloor \varphi n \rfloor \quad \text{and} \quad b = \lfloor \varphi^2 n \rfloor.$$

Proof: We have $\frac{1}{\varphi} + \frac{1}{\varphi^2} = \frac{1}{\varphi^2}(\varphi + 1) = \frac{1}{\varphi^2}(\varphi^2) = 1$, where we use $\varphi^2 - \varphi - 1 = 0$ in the third equality. Thus Rayleigh's theorems shows than the sequences $a_n = \lfloor \varphi n \rfloor$ and $b_n = \lfloor \varphi^2 n \rfloor$ are partitions of the positive integers. Note also that

$$\lfloor \varphi^2 n \rfloor = \lfloor \varphi n + n \rfloor = \lfloor \varphi n \rfloor + n.$$

Thus

$$\lfloor \varphi n \rfloor = \operatorname{mex}\{\lfloor \varphi m \rfloor, \lfloor \varphi^2 m \rfloor : m < n\} \quad \text{and} \quad \lfloor \varphi^2 n \rfloor = \lfloor \varphi n \rfloor + n.$$

Then the first lemma finishes. ■