Lecture 1

There are two forms of Codes

Noiseless Codes	Noisy Codes

The goals between each are a little different.

Noiseless Codes
- Efficiency

Noisy Codes
-Encode "Redundancy"
-Detect and Correct Errors

ISBN Example

An ISBN is a number to search books. If we create an aritificial error, as opposed to searching the ENTIRE database of all books, we can use an Error-Detection code to test this. In this example, we use a big dot product

$$\begin{aligned}
&0, 1, 3, 1, 0, 1, 9, 6, 7, 8\\
&10, 9, 8, 7, 6, 5, 4, 3, 2, 1\\
&0, 9, 24, 2, 0, 5, 36, 18, 14, 8\\
&= 121 \equiv 0(mod 11)
\end{aligned}$$

Broken Error Detected Example

$$\begin{aligned}
&0, 3, 8, 8, 9, 7, 8, 1, 2, 7\\
&10, 9, 8, 7, 6, 5, 4, 3, 2, 1\\
&0, 27, 64, 56, 54, 35, 32, 3, 4, 7\\
&= 282 \equiv 7(mod 11)
\end{aligned}$$

So to explain what is happening the first line is the the ISBN. The second line is the number we multply the first line by. Then, we add the third line together to get the number then we do modular arithmetic. (to note the adding is done 0 + 27 + 64...)

Section 3.1 In Class

Take a finite set $\mathcal{A}$. Call this an alphabet. We can then call $\mathcal{A}^*$ a set of finite sequences. A singlular finite sequence is just a **word**.

Example of words

$$\begin{aligned}\mathcal{A} &= \{0, 1\} \\ \mathcal{A}^* &= \{\emptyset, 0, 1, 00, 01, 11, 10, 001, \dots\}\end{aligned}$$

Each word in $\mathcal{A}^*$ is finite. Now, given a finite set $\mathcal{S}$ call it the source alphabet.

We can now define a code

$$\textbf{Code}\text{: a function st.} \\ f : \mathcal{S} \to \mathcal{A}^*$$

Example of a code

Let $\mathcal{S}$ be the set of spoken English words. Let $\mathcal{A}^*$ be the set of words in the alphabet: $\{A, B, \dots, Z\}$ The code function is then:

$$\begin{aligned}&\mathcal{S} = \{\text{Spoken English}\} \\ &\downarrow \text{f = spelling (encoding)} \\ &\mathcal{A}^* = \{\text{words in} \{A, B, \dots, Z\}\}\end{aligned}$$

Now what is the image of a code? Take e as the set of code word, then

$$e = f(S)$$

Now, let's define a message: We can look at sequences in $\mathcal{S}$ and we call this set $\mathcal{S}^*$. So a message:

> **Message:**
>
> A sequence in $\mathcal{S}^*$ where a word in the source alphabet gets encoded
>
> $f^* : \mathcal{S}^* \to \mathcal{A}^*$

You can think of this like: "Given a message get a string in $\mathcal{A}^*$. This is where concatenation rears its head.

$$f^*(S_1, S_2, \dots, S_n) = f(S_1)f(S_2)\dots f(S_n)$$

Example of Concatenation

$$S = \{A, B, C, D, E\}$$
$$A = \{0, 1, 2\}$$

Our message is $ABCDE \in \mathcal{S}^*$ so we create a mapping like:

$$A \mapsto 0$$
$$B \mapsto 1$$
$$C \mapsto 20$$
$$D \mapsto 21$$
$$E \mapsto 22$$

Which means:

$$f^*(ABCDE) = 01202122 \in \mathcal{A}^*$$

Now, let's define uniquely decipherable

Uniquely Decipherable
a code is UD if:
$f^* : \mathcal{S}^* \to \mathcal{A}^*$
is injective

Small Remark for this, it requires that f is injective but it is not always enough. Think of morse code and how T is ”__” and O is ”___”. If we have OOT how we know its 4 Ts or something else? These can't be uniquely decipherable because multiple things exist.