SYMBOL CODE

Coding is the process of mapping from letters in one alphabet to letters in another.

ASCII code

Computers do not process English language directly. Instead, English letters are first mapped into bits, then processed by the computer.

Morse code

The Morse code is a reasonably efficient code for the English alphabet using an alphabet of four symbols: a dot, a dash, a letter space, and a word space.

Decimal to binary conversion

From a source of 10 digits \{0, ..., 9\} to a destination of 2 digits \{0,1\}

Formally, symbol code \( C \) is a mapping \( X \rightarrow D^+ \)

\( X \) = the source alphabet
\( D = \) the destination alphabet
\( D^+ = \) set of all finite strings from \( D \)

Example

\( \{X, Y, Z\} \rightarrow \{0,1\}^+, C(X) = 0, C(Y) = 10, C(Z) = 1 \)

Codeword: the result of a mapping, e.g., 0, 10, 1 ...

Code: a set of valid codewords.

Extension

The extension \( C^+ \) of a code \( C \) is the mapping \( X^+ \rightarrow D^+ \) by concatenating \( C(x_i) \) without punctuation: \( C(x_1x_2...x_n) = C(x_1)C(x_2)...C(x_n) \)

Example: \( C(XXYZXZZY) = 001001110 \) with \( C(X) = 0, C(Y) = 10, C(Z) = 1 \)

Non-singular

Every element from \( X \) maps into a different string in \( D^+ \): \( x_1 \neq x_2 \Rightarrow C(x_1) \neq C(x_2) \)

Uniquely decodable

The extension \( C^+ \) is non-singular, which also means \( C^+(x^+) \) is unambiguous.
Prefix code

Prefix or Instantaneous code is the code that no codeword is a prefix of another
\{Prefix codes\} ⊂ \{Uniquely decodable codes\} ⊂ \{Non – singular codes\}

Examples:

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>0</td>
<td>00</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>X</td>
<td>11</td>
<td>01</td>
<td>10</td>
<td>01</td>
</tr>
<tr>
<td>Y</td>
<td>00</td>
<td>10</td>
<td>110</td>
<td>011</td>
</tr>
<tr>
<td>Z</td>
<td>01</td>
<td>11</td>
<td>1110</td>
<td>0111</td>
</tr>
</tbody>
</table>

C1 is non-singular, but not uniquely decodable code because $C1(WW) = C1(Y)$. Also C1 is
not prefix code while C1(W) is prefix of C1(Y)
C2, C3 is non-singular, uniquely decodable and prefix code.
C4 is non-singular, uniquely decodable but not prefix code since C4(W) is prefix of C4(X).

Code tree

Form a D-ary tree where $D = |D|$

- D branches at each node
- Each branch denotes a symbol $d_i$ in D
- Each leaf denotes a source symbol $x_i$ in X
- $C(x_i) = \text{concatenation of symbols } d_i \text{ along the path from the root to the leaf that represents } x_i$
- Some leaves may be unused

Note: The prefix code implies that no codeword is an ancestor of any other codeword on the
tree.
Examples:

$C(W) = 00, C(X) = 011, C(Y) = 010, C(Z) = 11$
$C(WWYXZ) = 00001001111$
**KRAFT INEQUALITY FOR PREFIX CODES**

For any prefix code, we have \[ \sum_{i=1}^{||X||} D^{-l_i} \leq 1 \] where \( l_i \) is the length of codeword \( i \) and \( D = ||D|| \).

**Proof:**
Consider a D-ary tree in which each node has D children, represent the prefix code. Some leaves may be unused to represent codewords.

- Label each node at depth \( l \) with \( D^{-l} \). At the root, the number is 1.
- We see that each node equals the sum of all its leaves. So the sum of all the end leaves must equals to 1.
- The depth of the codeword is in range of \([1, ||X||]\).
- We do not use more than all the end leaves to represent codewords. So the sum of used end leaves must less than or equals to 1.

Therefore, this sum: \[ \sum_{i=1}^{||X||} D^{-l_i} \leq 1. \]
Equality if all leaves are utilized.

**KRAFT INEQUALITY FOR UNIQUELY DECODABLE CODES**

For any uniquely decodable code with codewords of lengths \( l_1, l_2, ..., l_{||X||} \) then \[ \sum_{i=1}^{||X||} D^{-l_i} \leq 1 \]

**Proof:**
Let \( S = \sum_{i=1}^{||X||} D^{-l_i} \) and \( M = \max l_i \), then for any \( N \), we have
\[
S^N = \left( \sum_{i=1}^{||X||} D^{-l_i} \right)^N \text{ now we consider the codeword concatenating from N elements in X}
\]
\[
= \sum_{x_1=1}^{||X||} \sum_{x_2=1}^{||X||} \cdots \sum_{x_N=1}^{||X||} D^{-(l_{i_1} + l_{i_2} + \cdots + l_{i_N})}
\]
\[
= \sum_{x \subseteq X^N} D^{-l(x)} \cdot c(x) \text{, where } c(x) \text{ is the number of codewords of length } l(x). \text{ The maximum length of codewords with } X^N = M \cdot N
\]
We note that: for uniquely decodable code, for each fixed length \( l(x) \), the number of codewords
can not exceed the possible number of codewords $D^{|x|}$ then
$c(x) \leq D^{|x|}$ and plug in
$\Rightarrow S^N \leq \sum_{M,N} 1 = M \cdot N$

Here, we can choose any $N$ large so $S \leq 1$.

**CONVERSE OF KRAFT INEQUALITY**

If $\sum_{i=1}^{|X|} D^{-l_i} \leq 1$, which means that the code is unqiuetly decodable then there exists a prefix code
with codelengths $l_1, l_2, ... l_{|X|}$.

**Proof:**
1) Without loss of generality, we can reorder $l_1 \leq l_2 \leq ... \leq l_{|X|}$
2) Consider a codeword as $0.d_1d_2...d_l$
3) $C_k = \sum_{i=1}^{k-1} D^{-l_i}, C_k$ would have $l_k$ digits (we could add more '0' to the end of $C_k$).

For any $j < k$ or ($l_j < l_k$): $C_k = C_j + \sum_{i=1}^{k-1} D^{-l_i} \geq C_j + D^{-l_j}$
$\Rightarrow C_j$ cannot be a prefix of $c_k$ because they differ in the first $l_j$ digits
$\Rightarrow$ prefix code.

**Example:**

Suppose $L = [2; 2; 3; 3; 3]$ and $\sum_{i=1}^5 2^{-l_i} = 1/4 + 1/4 + 1/8 + 1/8 + 1/8 = .875 < 1$. So there
exists a prefix code:

<table>
<thead>
<tr>
<th>$l_k$</th>
<th>$c_k = \sum_{i=1}^{k-1} D^{-l_i}$</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>$0 = 0.002$</td>
<td>00</td>
</tr>
<tr>
<td>2</td>
<td>$0.25 = 0.012$</td>
<td>01</td>
</tr>
<tr>
<td>3</td>
<td>$0.5 = 0.1002$</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>$0.625 = 0.1012$</td>
<td>101</td>
</tr>
<tr>
<td>3</td>
<td>$0.75 = 0.1102$</td>
<td>110</td>
</tr>
</tbody>
</table>

**OPTIMAL CODE**

- If $l(C(x))= \text{the length of } C(X)$, then $C$ is optimal if $L_C = E[l(C(X))] = \sum x_i p(x_i \cdot l(x_i))$
  is as small as possible for a given $p(X)$.

- Suppose $C$ is a uniquely decodable code $\rightarrow L_C \geq H(X)/\log_2 D$ (unit = bits/symbol code)

  **Proof:**

  $L_C - H(X)/\log_2 D = \sum x_i p(x_i \cdot l(x_i))$
  $= E[\log_D D^{-l(x_i)} + E[\log_D p(x)]]$
  $\Rightarrow$ $c = \sum x_i D^{-l(x_i)} \leq 1$ and $q(x) = \frac{D^{-l(x_i)}}{c}$
  $\Rightarrow$ $L_C - H(X)/\log_2 D = E[-\log_D c \cdot q(x) + \log_D p(x)]$
  $= E[\log_D \frac{p(x)}{q(x)} - \log_D c]$
= \log_D 2 \cdot E[\log p(x)/q(x)] - E[\log_D c]
= \log_D 2 \cdot D(p||q) - E[\log_D c]

While \log_D 2 > 0 and Relative entropy \( D(p||q) \geq 0 \)
and \log_D c \leq 0 because \( c \leq 1 \)
then \( L_C - H(X)/\log_2 D \geq 0 \).

Equality when \( D(p||q) = 0 \) and \( c = 1 \) \( \Rightarrow  p(x) = q(x), \forall x \) and \( \sum_x D^{-l(x)} = 1 \)
\( \Rightarrow p(x) = D^{-l(x)} \)
\( \Rightarrow l(x) = \log_D \frac{1}{p(x)}. \)

**SUMMARY**

- Symbol code
  - Non-singular
  - Uniquely decodable
  - Prefix

- Kraft inequality for uniquely decodable code
\[ \sum_{i=1}^{k-1} D^{-l_i} \leq 1 \]
- If a code is uniquely decodable then we can always construct a prefix code with the same lengths

- Lower bound for any uniquely decodable code
\[ L_C = E[l(C(X))] \geq H(X)/\log_2 D \]