Instantaneous Codes

- Is the code in which each codeword in any string of codewords can be decoded (reading from left to right) as soon as it is received.
- It will be expressed by code trees (so in the following we study something about graphs and trees).

- **Graph**: a graph \( G = (V, E) \) where
  - \( V = \{ v_1, \ldots, v_s \} \) is a set of vertices (nodes)
  - \( E = \{ e_1, \ldots, e_n \} \) is a set of edges \( e_i : v_k \rightarrow v_j \)

- **Directed graph (Digraph)**: a digraph \( G = (V, E) \) where
  - \( V = \{ v_1, \ldots, v_s \} \) is a set of vertices (nodes)
  - \( E = \{ e_1, \ldots, e_n \} \) is a set of arcs (edges) \( e_i : v_k \rightarrow v_j \)

- **Tree**: a tree is a special digraph with a start distinct node called root.

- **Path**: is a sequence of arcs (edges) \( \cdots \rightarrow \cdots \)

- **Loop**: is a path that starts and ends at the same node

- **Leaf**: is the node with no arcs going out

- **Binary tree**: is the tree in which from each node 2 (or 0) arcs go out.

- **(Binary) Coding tree**: All code words are assigned to leaves
  - Example: \( c_1, c_2, c_3, c_4 \)

**Note that**: non-instantaneous codes cannot be assigned as trees
- Example: \( c_5, c_6 \)
Kraft Inequality

- It examines: What requirements must be met by a code in order for it to be instantaneously decodable

- \( \sum_{i=1}^{n} r^{-l_i} = 1 \) where \( r \) is the size of the code alphabet and \( l_i \) is the length of the codeword

- In case of binary tree (i.e. the code alphabet is \( \{0,1\} \)) we have
  \[ \sum_{i=1}^{n} 2^{-l_i} = 1 \]

Note: (1). K.I. means: if the lengths \( l_1, \ldots, l_n \) satisfy K.I., then there must exist some instantaneous code with these lengths.

(2). K.I. does not mean: any code whose codeword lengths satisfy K.I. must be instantaneous.

Example:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Code</th>
<th>length</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>01</td>
<td>2</td>
</tr>
</tbody>
</table>

\[ \sum_{i=1}^{2} 2^{-l_i} = 2^{-1} + 2^{-2} = 2^{-1} + 2^{-2} = \frac{3}{4} = 1 \]

(2)
⇒ it satisfy K.I., But it is not instantaneous
(because it can not be represented as a binary tree)

(1)
⇒ But by note (1) there must exist some instantaneous code with these lengths (i.e. 1 and 2), which can be

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Code</th>
<th>length</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>11</td>
<td>2</td>
</tr>
</tbody>
</table>

\[ \sum_{i=1}^{2} 2^{-l_i} = \frac{3}{4} = 1 \] and

[Diagram of a binary tree with nodes labeled 0, 1, and 11]
Uniquely Decodable

- If any encoded string has only one possible source string producing it then we have unique decodability

Extension of source code:

A source code $S = \left[ a_1, \ldots, a_m, p_1, \ldots, p_m \right]$ has the $n^{th}$

extension $S^n = \left[ a_1 a_1 \ldots a_1, a_1 a_1 \ldots a_2, \ldots, a_m a_m \ldots a_m, p_1 p_1 \ldots p_1, p_1 p_1 \ldots p_2, \ldots, p_m p_m \ldots p_m \right]$

For example the 2$^{nd}$ extension is

$S^2 = \left[ a_1 a_1, a_1 a_2, a_1 a_3, \ldots, a_m a_m, p_1 p_1, p_1 p_2, p_1 p_3, \ldots, p_m p_m \right]$

Example: Consider the code

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>10</td>
</tr>
<tr>
<td>C</td>
<td>100</td>
</tr>
<tr>
<td>D</td>
<td>1000</td>
</tr>
</tbody>
</table>

The 2$^{nd}$ extension of this code is

$S^2 = \left[ AA AB AC AD BA BB BC BD CA CB CC CD DA DB DC DD 11 110 1100 11000 101 1010 10100 101000 1001 10010 100100 1001000 10001 100010 1000100 10001000 100001 1000010 10000100 100001000 1000001 10000010 100000100 1000001000 10000001 100000010 1000000100 10000001000 \right]$

All codes are different.

- A code is called unequally decodable: if its extension is non-singular

![Relationship between Codes]
McMillan inequality

McMillan inequality extends Kraft's inequality to uniquely decodable codes

i.e. The codeword lengths of any uniquely decodable code must satisfy the Kraft’s inequality

\[ r^{-li} = 1 \]

In case of binary code i.e. \{0, 1\}, \( r = 2 \)

\[ 2^{-li} = 1 \]

Mean code length

- Let \( S \) be a memoryless stationary discrete information source

\[ S = (a_1, a_2, \ldots, a_m, \ p_1, p_2, \ldots, p_m) \]

- Let \( C \) be a uniquely decodable code

- The mean code length \( L \) is given by

\[ L = \sum_{i=1}^{m} p_i l_i = p_1 l_1 + p_2 l_2 + \ldots + p_m l_m \]

\[ = - \sum_{i=1}^{m} p_i \log_2 q_i \]

where \( q_i = 2^{-li} = (\frac{1}{2})^{li} \), \( i = 1, 2, \ldots, m \)

From Kraft – McMillan inequality we have \( q_1 + q_2 + \ldots + q_m = 1 \)
and hence we have

\[ L = - \sum_{i=1}^{m} p_i \log_2 p_i \]
Huffman Code

- With the Huffman code in the binary case the two least probable source output symbols are joined together, resulting in a new message alphabet with one less symbol.

- Huffman Code is also Compact code and satisfies the properties:
  1. Has the shortest mean length among binary instantaneous codes
  2. Optimal tree
  3. Compact Code tree