

* BINARY TREE

REPRESENTATION OF

CODE:

Useful For Design
as well a decoding
of codes.

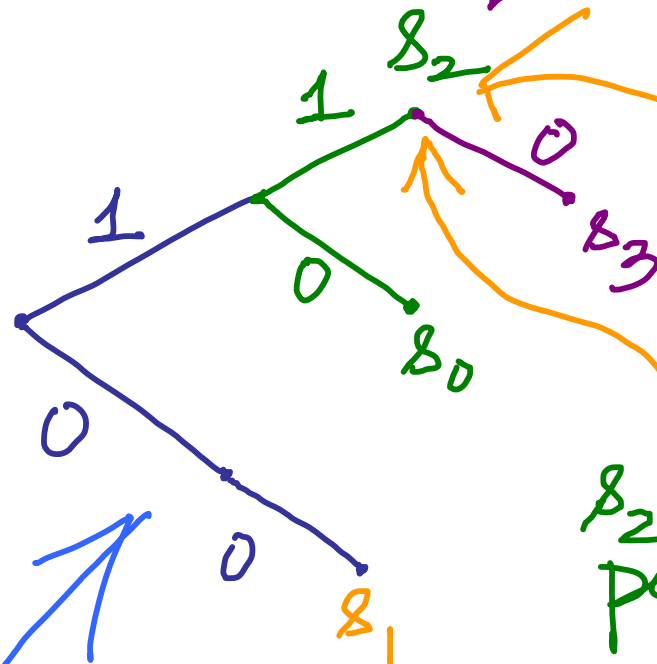
Consider the code:

x_0	—	10
x_1	—	00

s_2 — 11
 s_3 — 110

Binary Tree:

Binary Tree representation for above code



s_2 is an ancestor node of s_3 .

s_2 lies along path to s_3

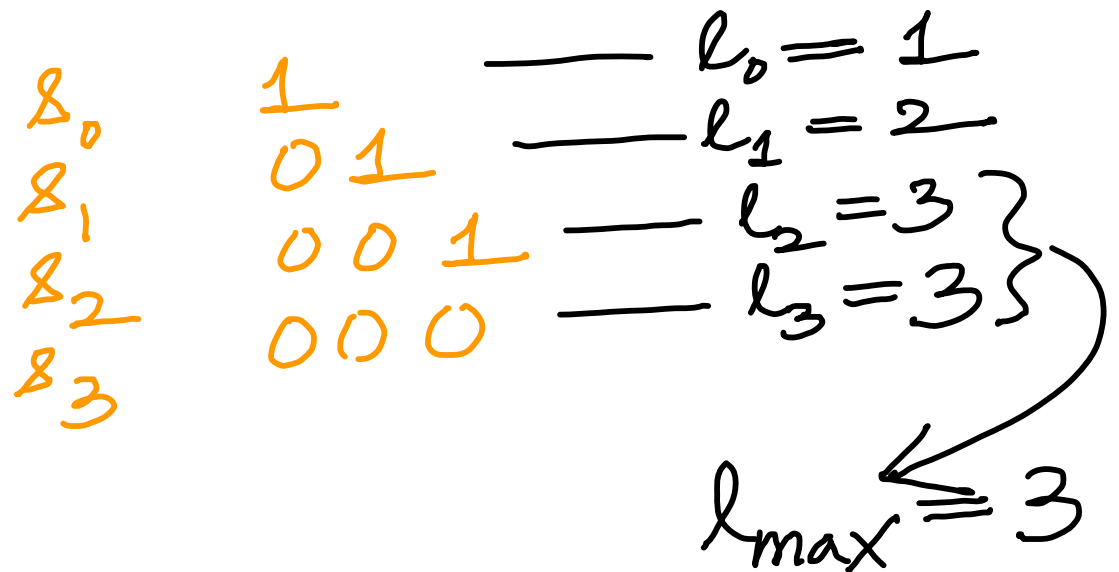
s_2 — 11
 s_3 — 110 } s_2 forms a "prefix" of s_3 .

NOT a prefix Free code

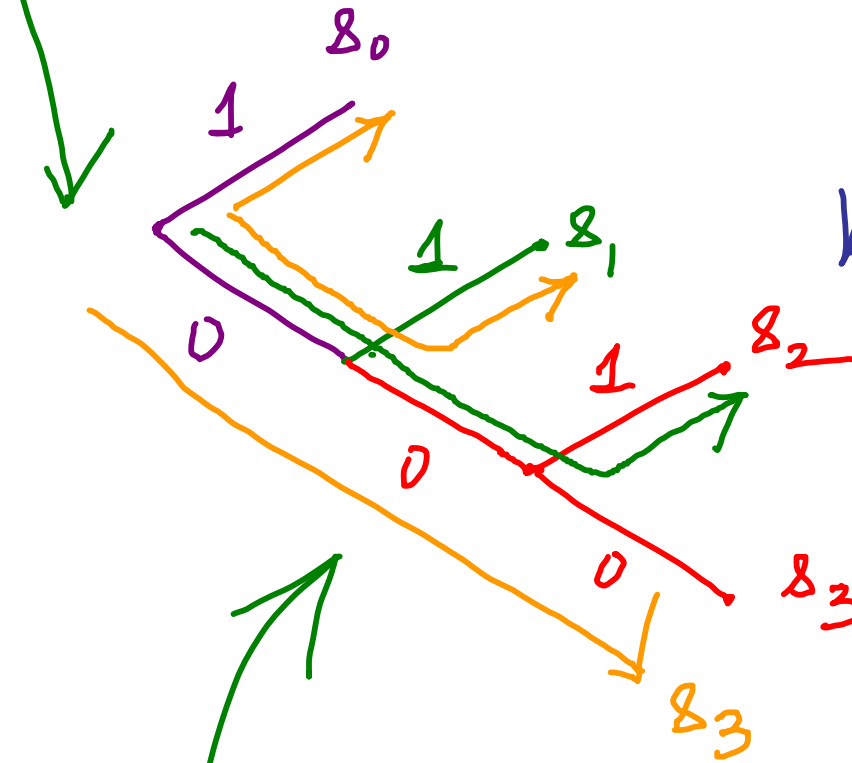
In general: IF any codeword lies along the path to another codeword, then it is NOT a prefix-free code.

Ex:

Consider Now.



Binary Tree.



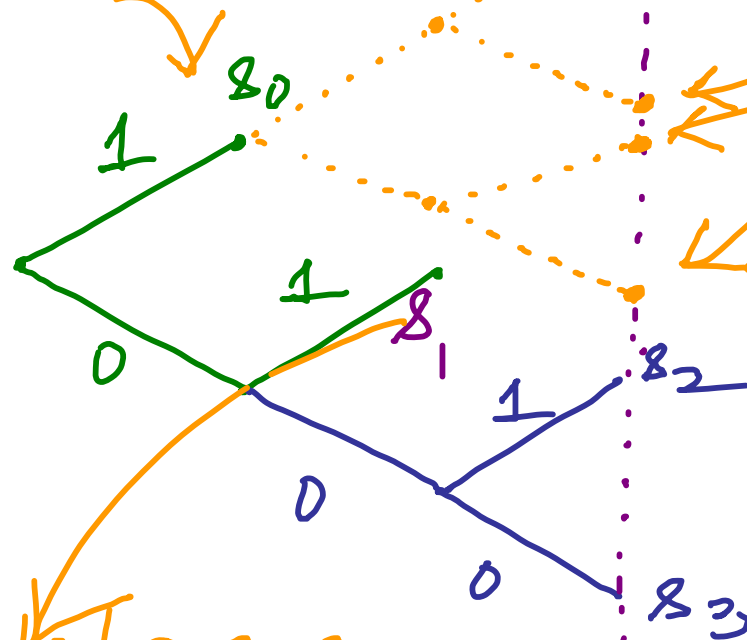
No codeword is an ancestor node for another codeword.

Therefore, this is a prefix Free code.

All codewords are on leaf nodes.
No codeword is on the path of another codeword.

$l_{\max} = 3$
 Max nodes
 $= 2^3 = 8$

level $l_0 = 1$



Number of nodes eliminated at level 3
 $= 2^{(l_{\max} - l_i)}$
 $= 2^{3-1}$
 $= 2^2 = 4$

leads to removal of $2^{3-2} = 2$ nodes

maximum number of bits in codeword

Binary Tree depth $= 3$
 $= l_{\max}$

Number of levels in Binary Tree representation $= 3$

In prefix Free code, each
codeword for s_i corresponds
to $2^{l_{\max} - l_i}$ Distinct
codewords at depth $l_{\max}$
of the Binary Tree.

From this, we observe that,

$$\sum_{i=0}^{M-1} 2^{(l_{\max} - l_i)} \leq 2^{l_{\max}}$$

= Maximum
Number of
nodes at Depth $l_{\max}$

$$\sum_{i=0}^{M-1} 2^{l_{\max} - l_i} \leq 2^{l_{\max}}$$

$$\Rightarrow \cancel{2^{l_{\max}}} \sum_{i=0}^{M-1} \cancel{2^{-l_i}} \leq \cancel{2^{l_{\max}}}$$

$$\Rightarrow \boxed{\sum_{i=0}^{M-1} 2^{-l_i} \leq 1}$$

Kraft Inequality
For prefix Free code.

$l_0 \quad l_1 \quad l_2 \quad \dots \quad l_{M-1}$
 $l_0 \quad l_1 \quad \dots \quad l_{M-1}$

codeword lengths.

$$\sum_{i=0}^{M-1} 2^{-l_i} \leq 1$$

Kraft Inequality.

Converse Also Holds:

IF l_i satisfy condition above, There exists a prefix free code with codeword lengths $= l_i$.