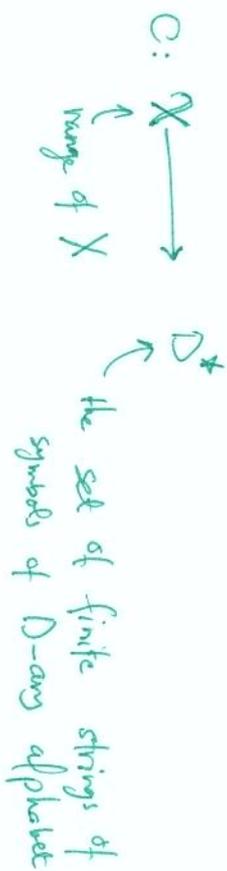
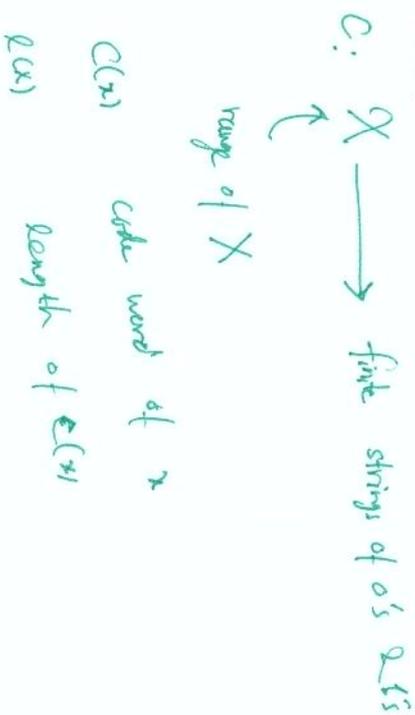


## Codes

Source code for a random variable  $X$  (row)



For our case, it is



~~Code~~ Source Memoryless  
Discrete synchronous

## Code:

~~non-singular~~

1) Non-singular:

$$x \neq x' \Rightarrow C(x) \neq C(x')$$

Non-singularity does not necessarily allow us to decode a sequence of values of  $X$ .

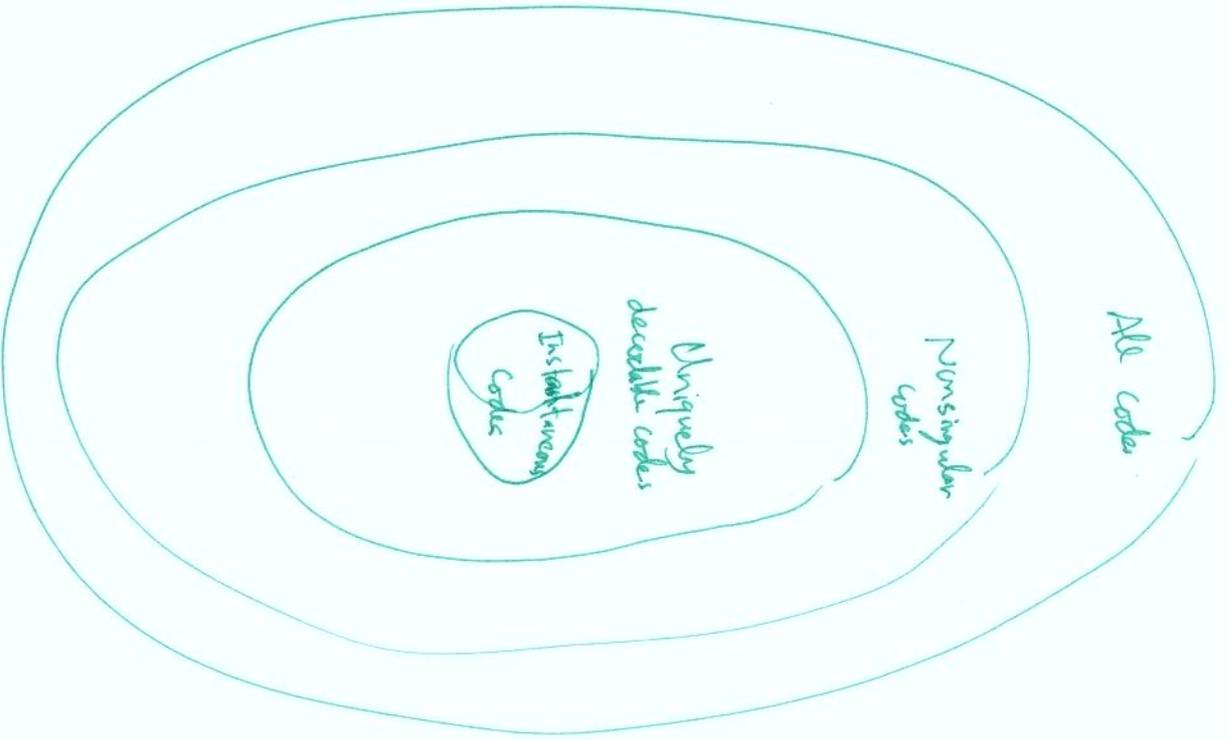
Extension  $C^*$  of a code: Finite strings of 0's & 1's

~~$C(x_1 x_2 \dots x_n)$~~   
such that

$$C(x_1 x_2 \dots x_n) = \underbrace{C(x_1) C(x_2) \dots C(x_n)}_{\text{concatenation of codewords}}$$

2) Uniquely decodable if its extension is not singular

3) prefix code or instantaneous code if no code word is a prefix of another code word



Kraft Inequality:

We would like to construct

- 1) instantaneous code
- 2) with minimum expected length

Set of codeword lengths possible limited by

Kraft Inequality.

Kraft inequality:

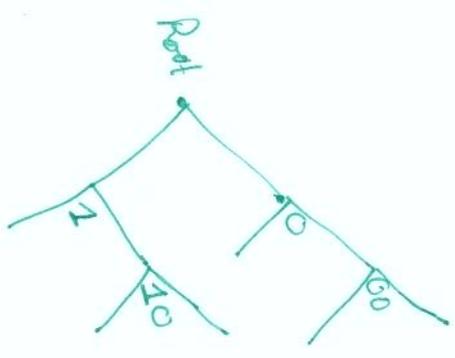
For any instantaneous code, the codeword lengths must satisfy

$$\sum 2^{-l_i} \leq 1$$

Conversely, given a set of words that satisfy this inequality, there exists an instantaneous code with these word lengths.

Proof:

Consider a binary tree  
Each node has 2 children nodes



Each codeword is represented by a leaf on the tree  
 The prefix condition means that no codeword is ancestor of any other codeword on the tree  
 So each codeword eliminates descendants

Let  $l_{max}$  be the length of the longest codeword

Consider all nodes of the tree at level  $l_{max}$

Some are codewords  
Some are descendants of codewords  
Some are neither

These sets are disjoint

The total ~~number of~~ leaves number of nodes =  $Q_{l_{max}}$

A codeword at level  $l_i$  has  $Q_{l_{max}-l_i}$  descendants

at level  $l_{max}$

$$S_0 \sum_{Q_{l_{max}-l_i}} \leq Q_{l_{max}}$$

i.e.

$$\boxed{\sum_{Q_{l_i}} \leq 1}$$

Converse:

Given a set of codewords of lengths

$$l_1, l_2, \dots, l_m$$

that satisfy Kraft inequality.

Construct a tree of depth  $l_{max}$

Choose a branch of depth  $l_1$  & let that be codeword 1; eliminate its descendants

Choose the a branch of depth  $l_2$  & let that be codeword 2; eliminate its descendants as codeword 2; until the  $n$ th codeword is

constructed.

We can show that uniquely decodable codes also satisfy the Kraft inequality with

This means that in designing codes with minimum expected length, there is really no advantage to have uniquely decodable codes & we can focus on prefix codes.

~~Optimal Codes:~~

Lower bound on Expected Length of a code

How low can the average length of an instantaneous code be?

$$L \geq H(X)$$

Note that

$$\begin{aligned} L - H(X) &= \sum p_i l_i - \sum p_i \log \frac{1}{p_i} \\ &= - \sum p_i \log 2^{-l_i} + \sum p_i \log \frac{1}{p_i} \\ &= - \left( \sum p_i \log \frac{2^{-l_i}}{p_i} \right) \end{aligned}$$

$$\begin{aligned} &\geq - \log \sum p_i \frac{2^{-l_i}}{p_i} \quad (*) \\ &= + \log \left( \frac{1}{\sum 2^{-l_i}} \right) \end{aligned}$$

Kraft's inequality

$$\boxed{L - H(X) \geq 0}$$

(\*) follows from the fact that  $\log$  is concave  $\log$  is above the average



# Optimal codes

Let's find the optimal set of length  $l_1, \dots, l_m$  of an instantaneous code that will minimize ~~the~~ <sup>the</sup> average length of the code.

So we should minimize  $L = \sum p_i l_i$

over integers  $l_1, \dots, l_m$  satisfying  $\sum 2^{-l_i} \leq 1$

Ignore the integer constraints & relax the inequality to an equality (this will result in a lower  $L$ )

We have a constrained minimization problem

$$J = \sum p_i l_i + \lambda \left( \sum 2^{-l_i} - 1 \right)$$

$$\frac{\partial J}{\partial p_i} = p_i + \lambda (l_n 2^{-l_i}) = 2^{-l_i n 2}$$

Set to zero to get

$$2^{-l_i} = \frac{p_i}{\lambda 2^{ln 2}}$$

Use the constraint  $\sum 2^{-l_i} = 1$

to get  $\lambda =$

So the optimal length is

$$l_i^* = -\log p_i$$

The corresponding min avg. length is  $L^* = \sum p_i l_i = -\sum p_i \log p_i = H(X)$  which is our lower bound.

Since the  $l_i$ 's are integers, the optimum  $l_i$ 's we can choose are

$$l_i = \lceil -\log p_i \rceil$$

$$= \lceil \log \frac{1}{p_i} \rceil$$

Now this code is a prefix code because

it satisfies the Kraft inequality  $\lceil \log \frac{1}{p_i} \rceil$

$$\sum 2^{-l_i} = \sum 2^{-\lceil \log \frac{1}{p_i} \rceil}$$

$$\leq \sum 2^{-(\log \frac{1}{p_i})} = \sum 2^{\log p_i}$$

$$= \sum_{p_i=1} 1$$

Now this code allows us to construct an upper bound on the average length of the code possible. Specifically, not that since

$$l_i = \lceil -\log p_i \rceil$$

we have that

$$\log \frac{1}{p_i} \leq l_i < \log \frac{1}{p_i} + 1$$

Multiplying both sides by  $p_i$  & sum over  $i$  we

get

$$\sum p_i \log \frac{1}{p_i} \leq \sum p_i l_i < \sum p_i (\log \frac{1}{p_i} + 1)$$

or

$$H(X) \leq L < H(X) + 1$$

Let  $L^*$  be the ~~optimal~~ ~~possible~~ (minimum) average ~~len~~ expected length of the optimal code,

then

$$L^* \leq L < H(X) + 1$$

& we know from previous developments that

$$H(X) \leq L^*$$

So the expected length of the optimal code also satisfies

$$H(X) \leq L^* < H(X) + 1$$

i.e. there is an overhead of at most 1 bit (due to the fact ~~that~~  $\log 1/p_i$  is not always an integer)

How do we reduce this overhead.

Instead of ~~concatenating~~ ~~a~~ symbol of ~~of~~ ~~stream~~

~~from~~  $X$ , encoding  $n$  super-symbols

from the concatenation of  $n$  symbols from  $X$ .

Take the case of  $n=2$ , then from above

$$H(X_1, X_2) \leq L_2 < H(X_1, X_2) + 1$$

$$\begin{aligned} \text{But } H(X_1, X_2) &= H(X_1) + H(X_2) \\ &= 2H(X_1) \end{aligned}$$

Now the ~~also~~ ~~expected~~ code word length per symbol

$$L_2 = \frac{1}{2} L$$

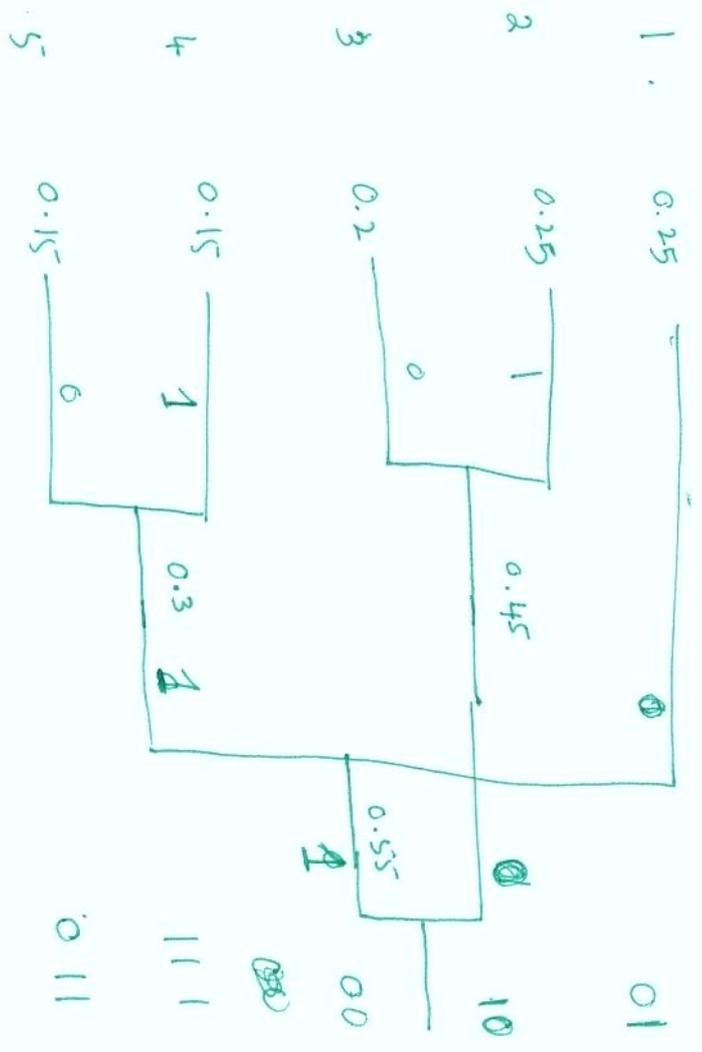
$$\leq \frac{1}{2} (H(X_1) + 1/n)$$

$$\leq \frac{1}{2} L_2$$

$$\leq L_n < H(X_1) + 1/n$$

In general

# Huffman Coding



The Huffman code gives the shortest expected length prefix code

Any other code cannot give a

shorter expected length,  
 But we will not prove that