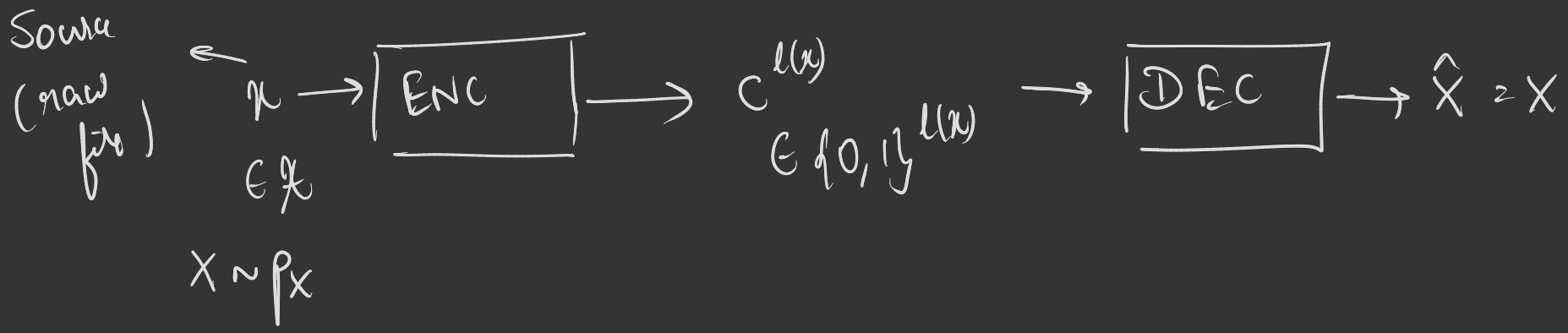
Variable Length Compression

$\mathbb{C}$

x^n	$p(x^n)$	$ENC(x^n)$	$l(x^n)$
000	1/2	0	1
001	1/4	10	2
010	1/8	110	3
011	1/16	1110	4
100	1/32	11110	5
101	1/64	111110	6
110	1/128	1111110	7
111	1/128	1111111	7

$L(\mathbb{C}) = \mathbb{E}\, l(X^n) \approx 1.984$ bits



Nonsingular Codes

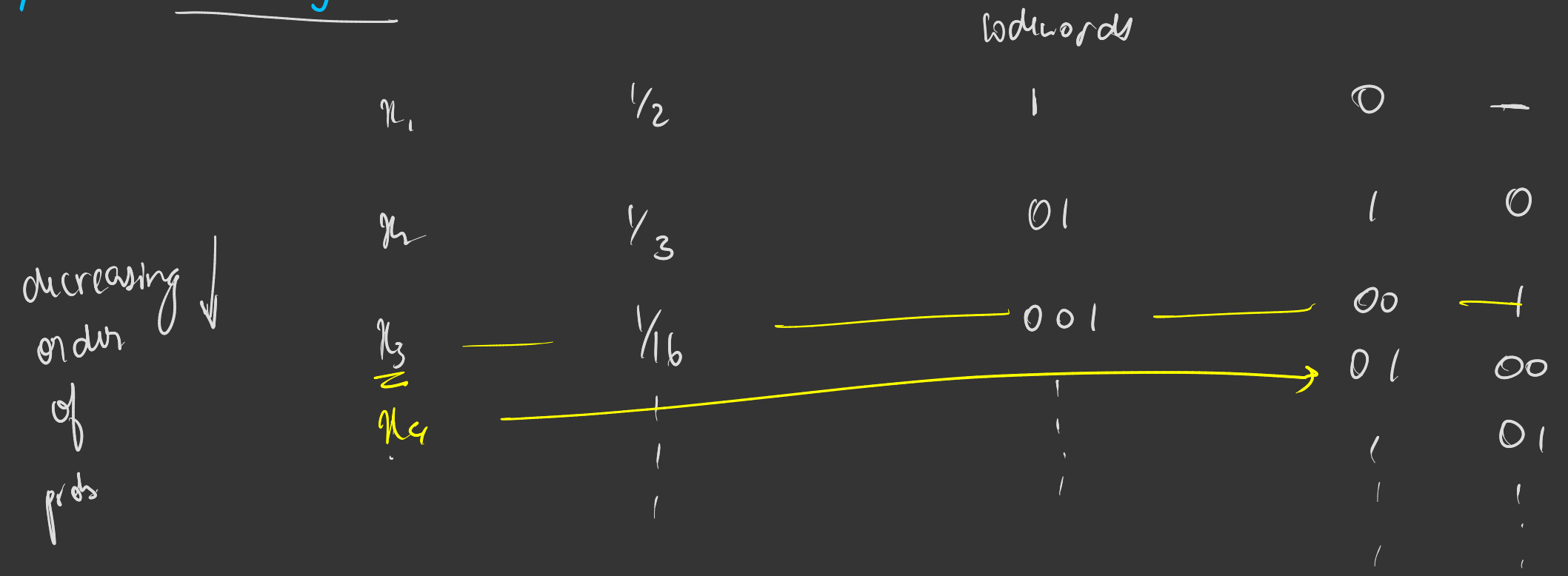
Compression algorithm (code) is nonsingular if

$$x \neq x' \Rightarrow \text{ENC}(x) \neq \text{ENC}(x')$$

$$\forall x, x'$$

Distinct symbols are assigned distinct codewords

Optimal nonsingular codes



$p(x_i) \geq p(x_{i+1})$

x_{10}

00 — x_4
$x_3 x_4$

potential codewords $\{0, 1, 00, 10, 11, 01, 000, 001, \ldots\}$

① $l(x_i) = \lfloor \log_2 i \rfloor$ ⟶ Claim (Check!)

↓

length of codeword for symb with i^{th} larger prob.

② $p(x_i) \leq \frac{1}{i}$ → Claim

↓

If not

$$\sum_{j=1}^{i} p(x_i)$$

$$= p(x_1) + p(x_2) + \cdots + p(x_i)$$

$$> \frac{1}{i} + \frac{1}{i} + \frac{1}{i} + \cdots + \frac{1}{i} > 1$$

$$\Rightarrow \quad i \leq \frac{1}{p(x_i)}$$

$p(x_1) \leq 1$

$p(x_2) \leq \frac{1}{2}$

$p(x_3) \leq \frac{1}{3}$

$$\mathbb{E}\, l(X) = \sum_{i=1}^{m} p(x_i)\, l(x_i)$$

$$= \sum_{i=1}^{m} p(x_i) \lfloor \log_2 i \rfloor$$

$$\approx \sum_{i=1}^{m} p(x_i) \left\lfloor \log_2 \frac{1}{p(x_i)} \right\rfloor$$

$$\approx \sum_{i=1}^{m} p(x_i) \log_2 \frac{1}{p(x_i)} = H(X)$$

Entropy

Extension of a code

$\{x_1, x_2, \cdots x_m\} = \chi$

$\{a, b, c, \cdots z\} = \chi$

a — 0

b — 10

c — 110

d — 111

apple

Single letter $\longrightarrow$ Multiletter code

ENC(a) = 0 , ENC(b) = 10 ...

ENC(ada) = ENC(a) ENC(d) ENC(a) $\longrightarrow$ Extension

= 0 111 0

Extension :

$$ENC(x_1\ x_2\ \cdots\ x_n) = ENC(x_1)\ ENC(x_2)\ \cdots\ ENC(x_n)$$

ENC(a d b a c c a)

$$= ENC(a)\ ENC(d)\ ENC(b)\ \ldots$$

Optimal non singular code

a	0
b	1
c	00
d	01

Extension

$aa \mapsto 00$

$bb \mapsto 11$

$ab \mapsto 01$

⋮

Extension $\quad ENC : \mathcal{X} \rightarrow \{0,1\}^*$

(For any set A, A^* denotes all possible finite-length sequences from A)

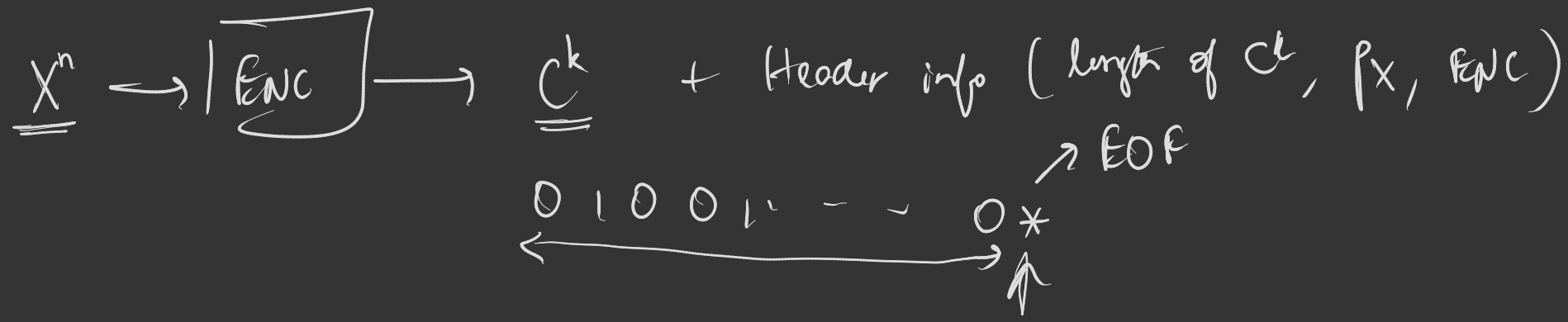
$\{0,1\}^* = \{-, 0, 1, 00, 01, 10, 11, 000, 001, \ldots\}$

Extension of ENC :

$\{ ENC(\underline{x}) : \underline{x} \in \mathcal{X}^* \}$

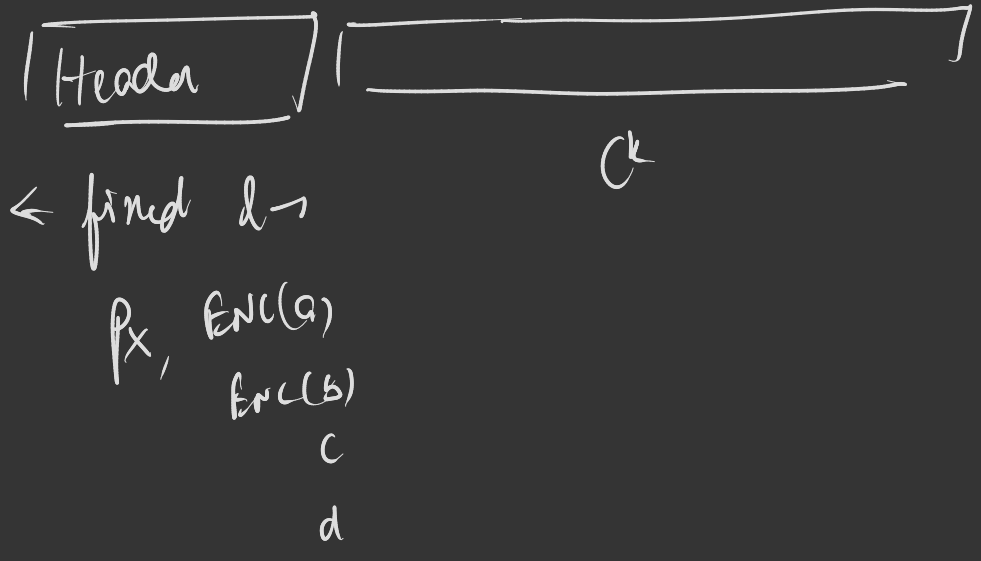
$\{ ENC(a), ENC(b) \cdots ENC(d), ENC(aa), ENC(ab) \cdots ENC(dd)$

$ENC(aaa) \cdots \}$



We have a mechanism for identifying end of compressed file.

abcda



Uniquely decodable code

UDC is a code for which the extension is nonsingular

$\Rightarrow \forall\, x^n, \tilde{x}^m, \quad ENC(x^n) \neq ENC(\tilde{x}^m)$

$\forall n, \forall m, \forall x^n, \forall \tilde{x}^m$

$x^n \neq \tilde{x}^m$

a	0
b	1
c	00
d	01

$ENC(d) = ENC(ab)$

$\therefore$ Not uniquely dec.

a	0
b	10
c	100
d	101

$ENC(c) = ENC(ba)$

a 0

b 1 0

c 1 1 0

d 1 1 1

→ Uniquely decodable

Prefix-free or instantaneous codes (Prefix code)

A code is prefix-free if no codeword is a prefix of another

①

a	0
b	101
c	110
d	111

Prefix free

②

a	0
b	10
c	100
d	101

NOT P.F

③

a	0
b	1
c	00
d	01

NOT P.F

→ 10 1|1 0

Is every prefix code uniquely decodable?

YES!

$x(1)\ x(2)\ \cdots\ x(n)$

$\tilde{x}(1)\ \tilde{x}(2)\ \cdots\ \tilde{x}(m)$

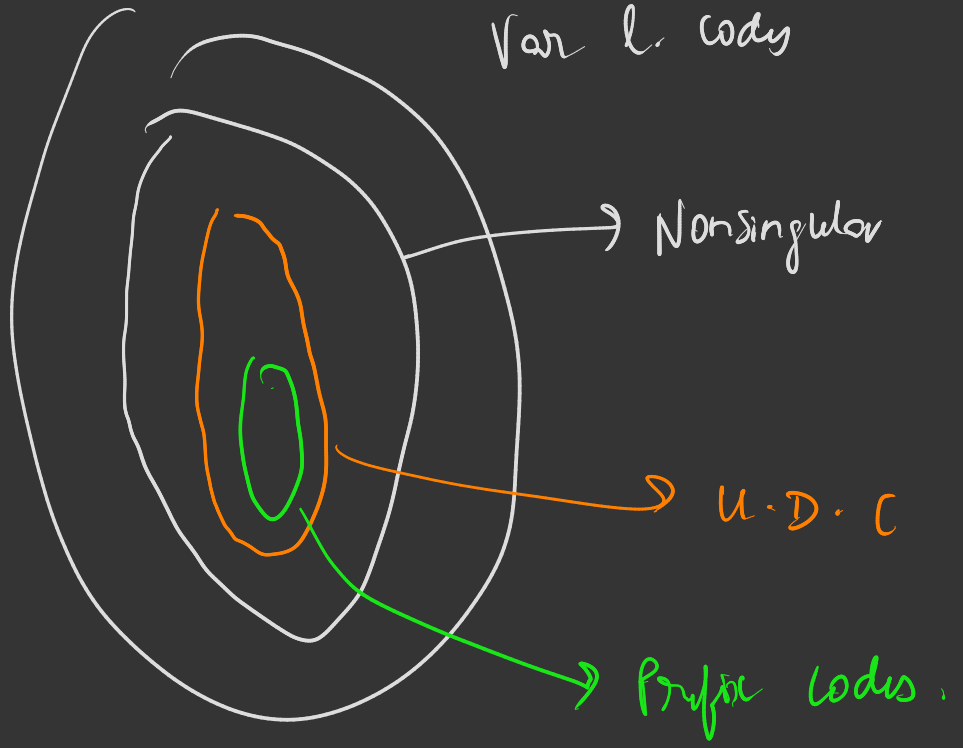
$ENC(x(1)\ x(2))$

$ENC(\tilde{x}(1)\ \tilde{x}(2)\ \tilde{x}(3))$

$\Rightarrow$ Either $ENC(x(1)) = ENC(\tilde{x}(1))$ only if $x(1) = \tilde{x}(1)$

OR $ENC(x(1))$ is a prefix of $ENC(\tilde{x}(1))$ ✗

$ENC(\tilde{x}(1))$ is a prefix of $ENC(x(1))$ ✗


Var l. cody
Nonsingular
U.D.C
Prefix codes.

Summary

1. Nonsingular codes — No two symbols have the same codeword

2. Uniquely decodable codes — Every extension is nonsingular

3. Prefix codes — No codeword is a prefix of another

Example

$\mathcal{X}$	C_1 (Singular)	C_2 (Nonsingular, Not U.D, Not P.F)	C_3 (Nonsingular, Not P.F.)	C_4 (Non-singular U.D, Prefix free)
a	0	0	00	0
b	00	010	10	10
c	10	01	11	110
d	00	10	110	111

ENC(ca) = 010

ENC(b) = 010

ENC(ab) = 0010

ENC(ac) = 0011

110 111 00 110 ...

Even # of 0's ⇒ a's

1000001

Odd # of 1's ⇒ c--- cb

+0

a 00

b 10

c 11

d 110

Even # of 1's + ends ⇒ c's

Even # of 1's + zeros ⇒

odd # of 0's → c-- c d a -- a

even # of 0's → c-- c a -- a

Odd # of 0's → Never happens

000 --- 0

a --- a

odd

11 --- 10

c--- c s

a 00

b 10

c 11

d 110

① In this eg, the delay in decoding the first sym depends on # 0's (arbitrarily large)

② Decoding may be complicated.

Tree representation of a code

Codebook : Set of all codewords

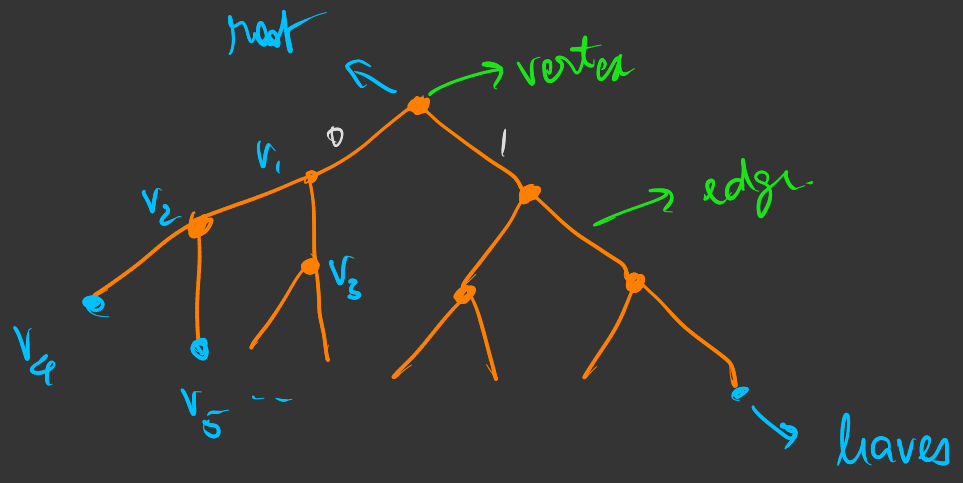
$$\mathbb{C} = \{ ENC(x) : x \in \mathcal{X} \}$$

$$\mathbb{C} = \{ 00, 10, 11, 110 \}$$

a	00
b	10
c	11
d	110

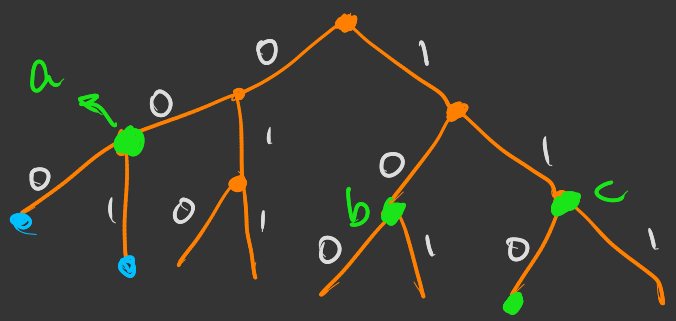
l_{max}

2, 2, 2, 3



v_1 is a parent of v_2 & v_3 ; v_2 & v_3 are children of v_1

v_1 is an ancestor of v_4, v_5 , v_4 & v_5 are descendants of v_1



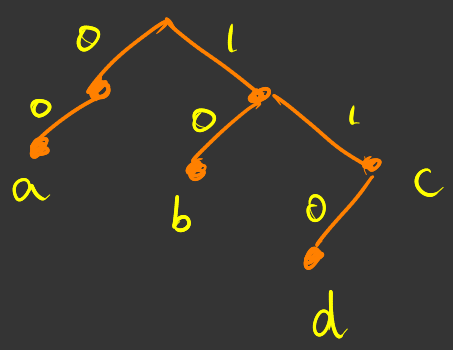
$\uparrow$ l_{max} $\downarrow$

a 00
b 10
c 11
d 110

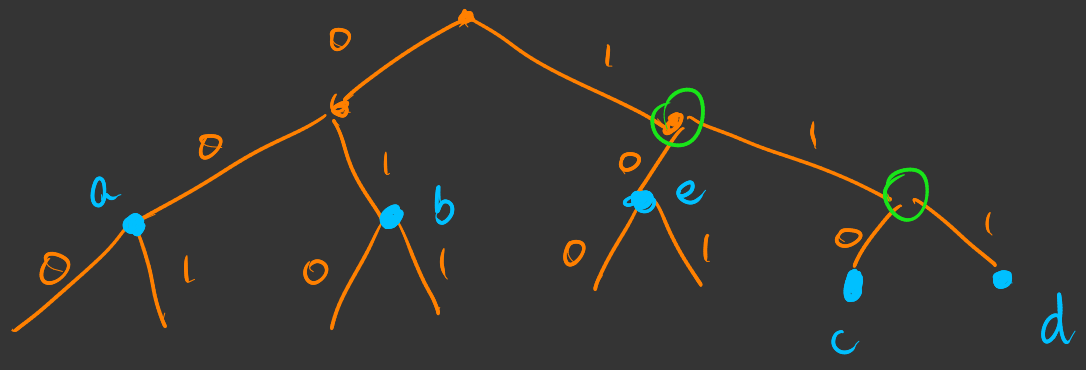
l_{max}

Tree enumerates all binary strings of length l_{max}.

Code tree



The codebook is a set of vertices on a binary tree of depth l_{max}.

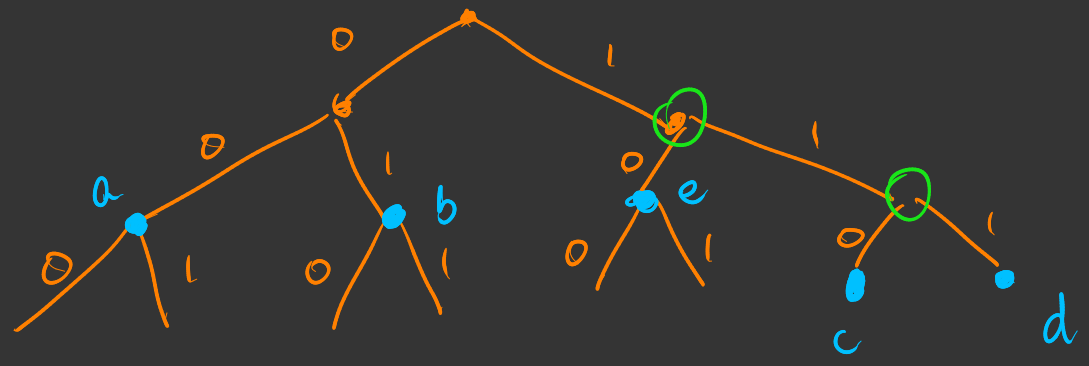


a	0 0
b	0 1
c	1 1 0
d	1 1 1
e	1 0

Prefix-free

Observation: Prefix $\Leftrightarrow$ ancestor on code tree / complete binary tree of depth l_{max}

Prefix-free code $\Leftrightarrow$ No codeword is an ancestor of any other.

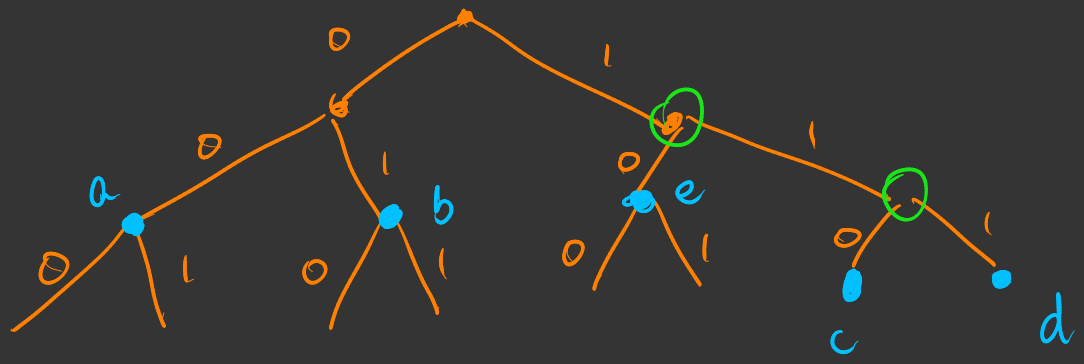


Decoding: Code tree

① Start at root & do:

② if next bit is 0, go left
, right

if you hit codeword/labeled vertex, output sym, go to step ①



0 0 1 1 1 1 0 1 1 0

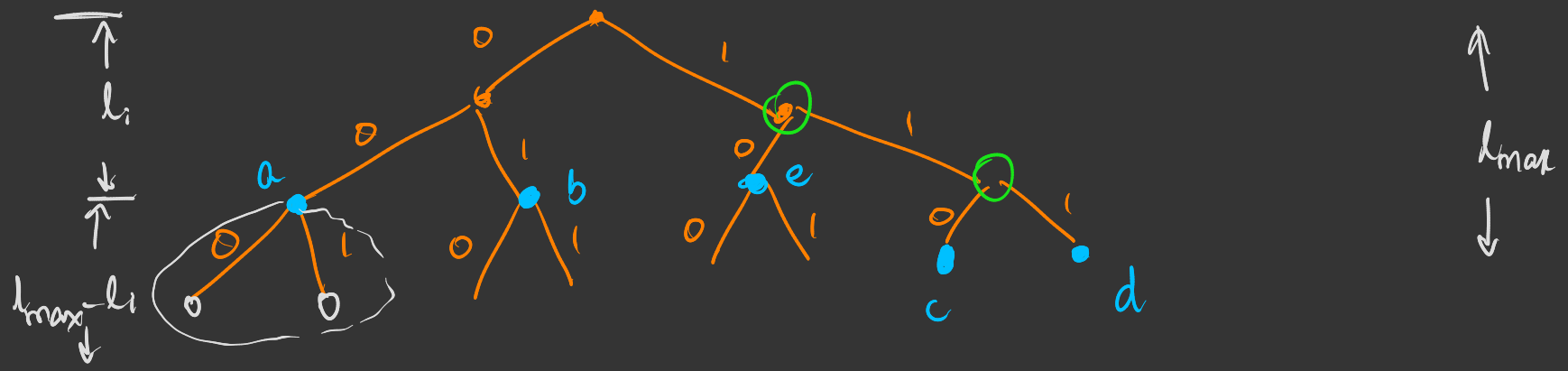
a d e c

Delay = 0

Instantaneous

a d e c

Necessary and sufficient conditions on the codeword lengths of prefix codes



Codeword lengths = $\{l_1, l_2 \cdots l_m\}$ $\quad |\mathcal{X}| = m$

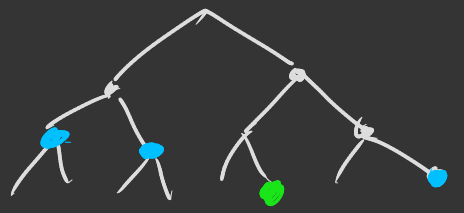
$\{2, 2, 2, 3, 3\}$

Take any codeword $\rightarrow l_i$. Count the # of leaves that are descendants.

$2^{l_{max} - l_i}$

How many c/w can be ancestors of a leaf?

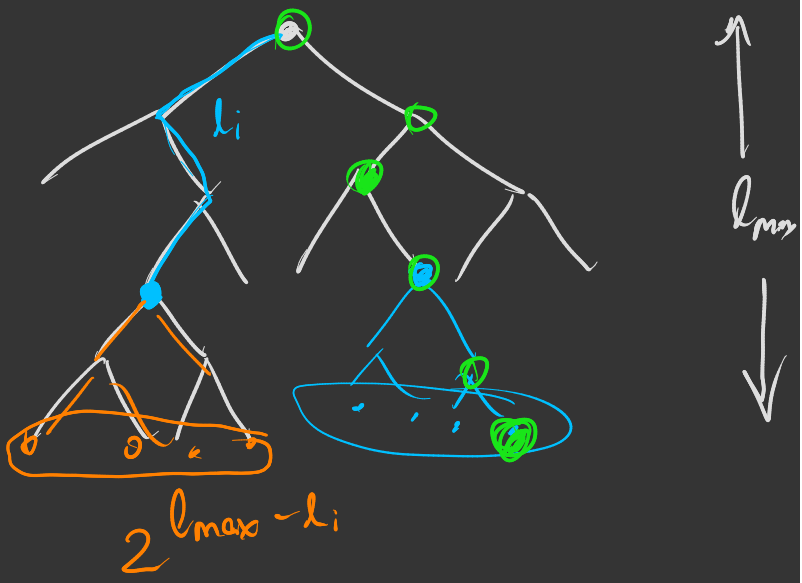
$\sum_{\text{all codewords}}$ # leaves that are descendants of c/w



$\sum_{i=1}^{m}$ # of leaves — " — of c/w i $\leq$ $2^{l_{max}}$

total # of leaves in the tree.

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



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

Kraft's inequality.

Every prefix code must satisfy ↗

For each symbol i codeword i S_i = set of all leaf descendants

$$i, j : \quad S_i \cap S_j = \emptyset$$

$$\bigcup_{i=1}^{m} S_i \subseteq \{\text{all leaves}\}$$

$$2^{l_{max}} \geq \left| \bigcup_{i=1}^{m} S_i \right| = |S_1| + |S_2| + \cdots + |S_m|$$

$$= 2^{l_{max}-l_1} + 2^{l_{max}-l_2} - \cdots -$$

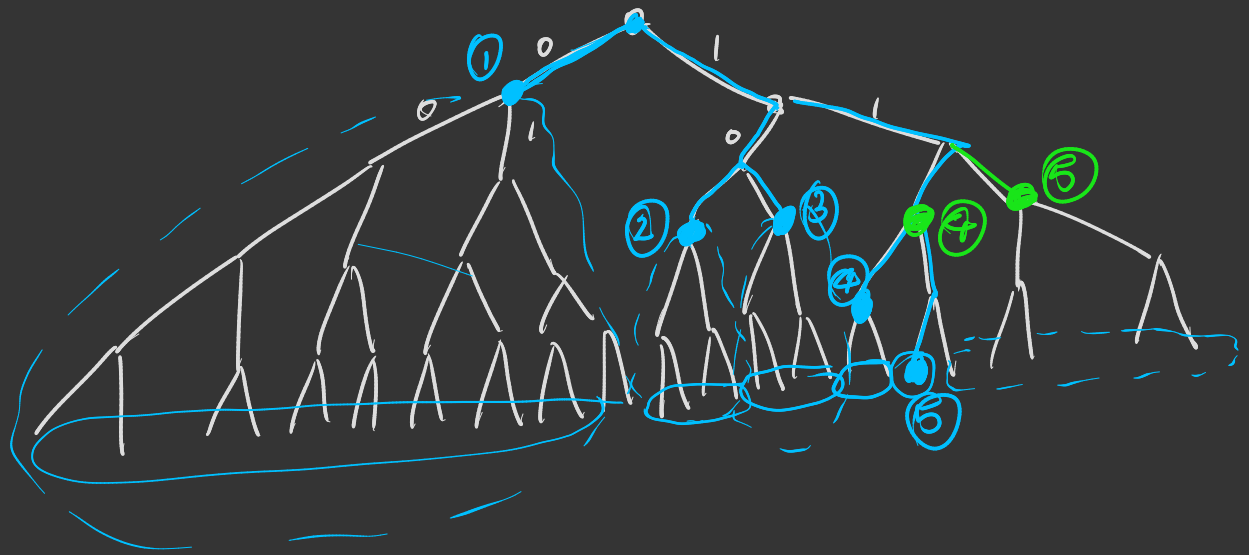
$C = \{0, 00, 10\}$

Kraft's ineq $\sum_{i=1}^{n} 2^{-l_i} = 2^{-1} + 2^{-2} + 2^{-2}$

$= 1$

$L = \{1, 2, 2\}$

$C = \{0, 10, 11\}$



① 0

② 1 0 0

③ 1 0 1

④ 1 1 0 0

⑤ 1 1 0 1 0

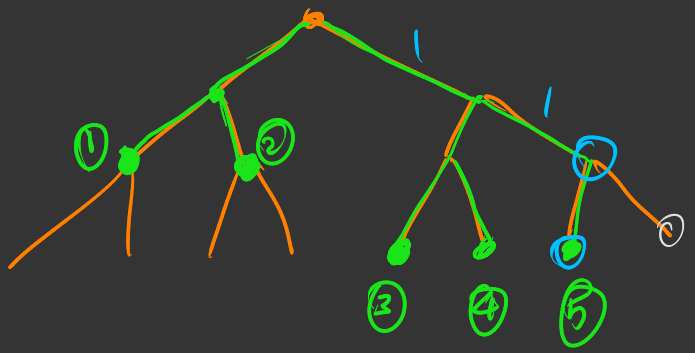
$\{l_1, l_2, l_3 \cdots l_m\}$

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

$\{1, 3, 3, 4, 5\}$

$$\sum_{i=1}^{5} 2^{-l_i} = 0.84\ldots < 1$$

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



$\{2, 2, 3, 3, 3\}$

$l_{max} = 3$

1 00

2 01

3 100

4 101

5 11 0

$$\sum_{i=1}^{n} 2^{-l_i} = 0.875 < 1$$

Kraft sum

If Kraft sum < 1, then some of the leaves have no codeword ancestors.

Kraft's inequality

For any prefix code, the codeword lengths $\{l_1 \cdots l_m\}$ must satisfy:

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

If $\{l_1 \cdots l_m\}$ is any set of integers satisfying

$$\sum_{i=1}^{m} 2^{-l_i} \leq 1,$$

we can construct a prefix code with these lengths.

Extended Kraft's inequality

Same result holds even for countably infinite m.

(A set is countable if there is a 1-1 map with $\mathbb{Z}$)

Shannon Code

$\mathcal{X}$ → alphabet $|\mathcal{X}| = m$

Consider any pmf $(p_1, p_2 - - , p_m)$ $(0.2, 0.1, 0.4, 0.3)$

Goal: Construct a prefix-free code.

with small expected length

$$\sum_{i=1}^{m} p_i l_i$$

Know that $\sum_{i=1}^{m} p_i = 1$

$$\sum_{i=1}^{m} 2^{-l_i} \leq 1 \qquad l_1\ l_2 \cdots l_m$$

$$l_i = \left\lceil \log_2 \frac{1}{p_i} \right\rceil$$

$$\left(\log_2 \frac{1}{p_i}\right) + 1 \geq l_i \geq \log_2 \frac{1}{p_i} \quad \Rightarrow \quad -l_i \leq \log_2 p_i$$

$$2^{-l_i} \leq p_i$$

$$\sum_{i=1}^{m} 2^{-l_i} \leq \sum_{i=1}^{m} p_i = 1$$

$$\sum_{i=1}^{m} p_i l_i = \sum_{i=1}^{m} p_i \left\lceil \log_2 \frac{1}{p_i} \right\rceil \leq \sum_{i=1}^{m} p_i \left(\log_2 \left(\frac{1}{p_i}\right) + 1 \right)$$

$$\leq H(X) + 1 \quad \text{bit}$$