



上海科技大学
ShanghaiTech University

EE140 Introduction to Communication Systems

Lecture 10

Instructor: Prof. Lixiang Lian

ShanghaiTech University, Fall 2022

Outline

- Source Coding
 - Fixed-length codes for discrete sources
 - Variable-length codes for discrete sources
 - Fixed-to-variable-length codes for discrete sources
 - Variable-to-variable-length codes for discrete sources

Example

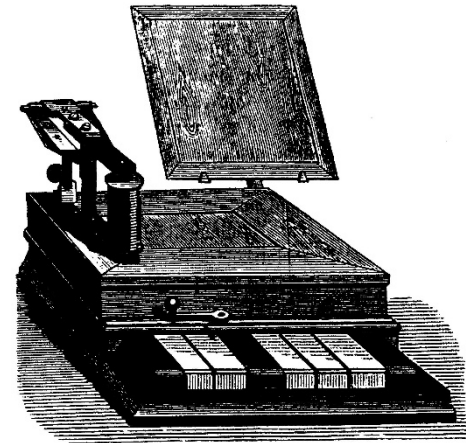
International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

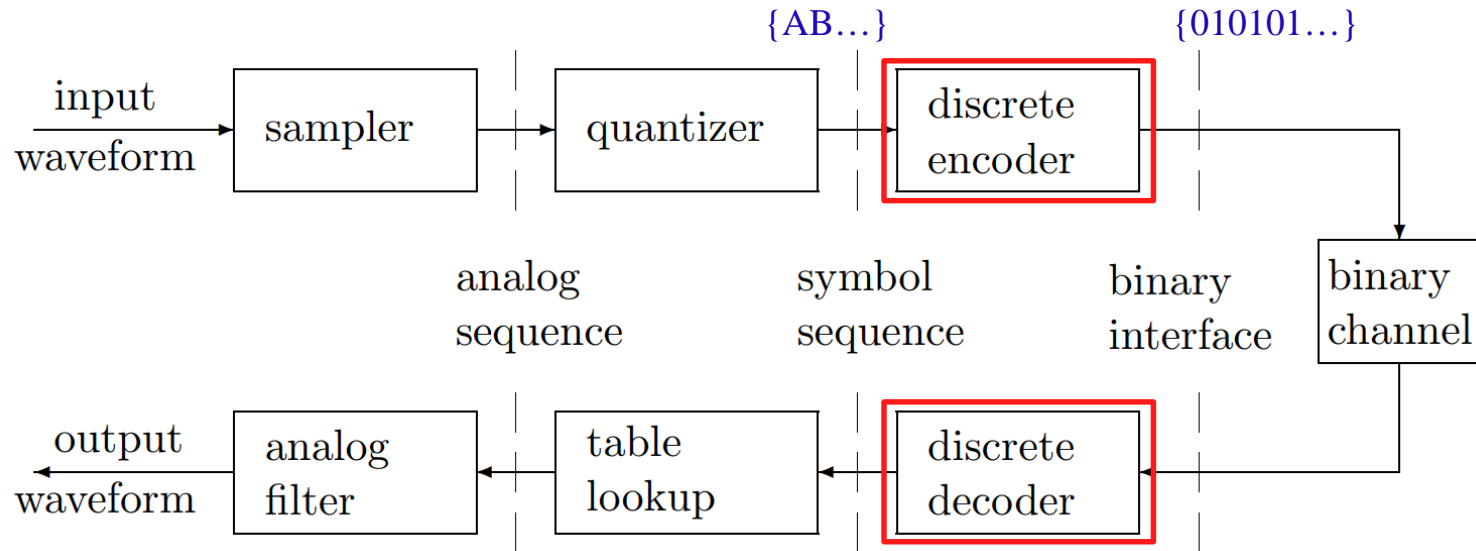
A ● —
B — ● ● ●
C — ● — ●
D — ● ●
E ●
F ● ● — ●
G — — ●
H ● ● ● ●
I ● ●
J ● — — —
K — ● —
L ● — ● ●
M — —
N — ●
O — — —
P ● — — ●
Q — — ● —
R ● — ●
S ● ● ●
T —

U ● ● —
V ● ● ● —
W ● — —
X — ● ● —
Y — ● — —
Z — — ● ●

1 ● — — — —
2 ● ● — — —
3 ● ● ● — —
4 ● ● ● ● —
5 ● ● ● ● ●
6 — ● ● ● ●
7 — — ● ● ●
8 — — — ● ●
9 — — — — ●
0 — — — — —



Layering of Source Coding



- Source coding and decoding for discrete sources
- Goal:
 - uniquely decodable;
 - Use as few binary digits per source symbol as possible.

Discrete Memoryless Sources (DMS)

- The source output is an unending sequence $\{X_1, X_2, \dots\}$ of randomly selected letter from a finite set \mathcal{X} , called the source alphabet.
- Each source output X_1, X_2, \dots is selected from \mathcal{X} using the same P_X .
- Each source output X_k is statistically independent of any other outputs $X_j, \forall j \neq k$.

X_1, X_2, \dots are i.i.d. according to P_X

Fixed-length Codes for Discrete Sources

- Map each symbol x of \mathcal{X} into an L length codeword $\mathcal{C}(x)$.
- Uniquely Decoding:
 - For an alphabet size of M , requires $2^L \geq M$.
- To reduce the encoding bits, choose L as
$$L = \lceil \log_2 M \rceil \longrightarrow \log_2 M \leq L < \log_2 M + 1$$
- Q: when to achieve lower bound?

Fixed-length Codes for Discrete Sources

- Example:

Alphabet $\mathcal{X} = \{a, b, c, d, e, f\}$.

$a \longrightarrow 000$

$b \longrightarrow 001$

$c \longrightarrow 010$

$e \longrightarrow 011$

$f \longrightarrow 100$

- $L=3$.
- $2^L \geq M \Rightarrow$ Uniquely decoded

Fixed-length Codes for Discrete Sources

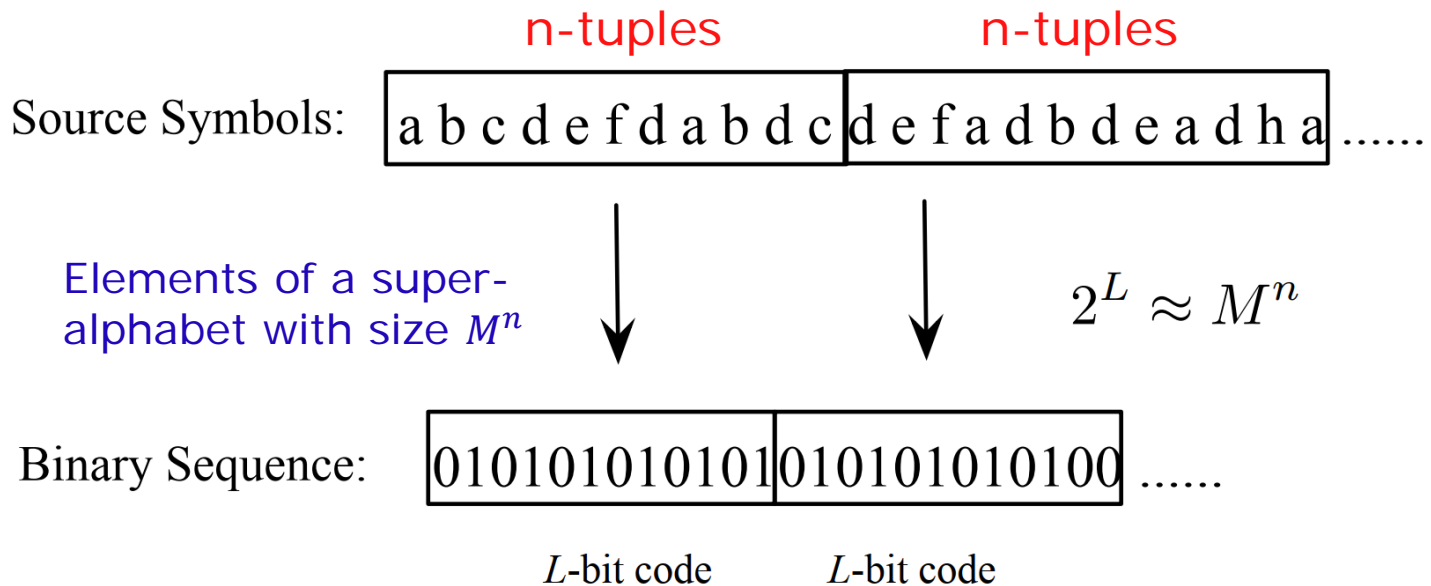
- Example: ASCII Code
 - Maps letters, numbers, etc. into binary 8 bits

Decimal - Binary - Octal - Hex – ASCII
Conversion Chart

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
0	00000000	000	00	NUL	32	00100000	040	20	SP	64	01000000	100	40	@	96	01100000	140	60	`
1	00000001	001	01	SOH	33	00100001	041	21	!	65	01000001	101	41	A	97	01100001	141	61	a
2	00000010	002	02	STX	34	00100010	042	22	"	66	01000010	102	42	B	98	01100010	142	62	b
3	00000011	003	03	ETX	35	00100011	043	23	#	67	01000011	103	43	C	99	01100011	143	63	c
4	00000100	004	04	EOT	36	00100100	044	24	\$	68	01000100	104	44	D	100	01100100	144	64	d
5	00000101	005	05	ENQ	37	00100101	045	25	%	69	01000101	105	45	E	101	01100101	145	65	e
6	00000110	006	06	ACK	38	00100110	046	26	&	70	01000110	106	46	F	102	01100110	146	66	f
7	00000111	007	07	BEL	39	00100111	047	27	'	71	01000111	107	47	G	103	01100111	147	67	g
8	00001000	010	08	BS	40	00101000	050	28	(72	01001000	110	48	H	104	01101000	150	68	h
9	00001001	011	09	HT	41	00101001	051	29)	73	01001001	111	49	I	105	01101001	151	69	i
10	00001010	012	0A	LF	42	00101010	052	2A	*	74	01001010	112	4A	J	106	01101010	152	6A	j
11	00001011	013	0B	VT	43	00101011	053	2B	+	75	01001011	113	4B	K	107	01101011	153	6B	k
12	00001100	014	0C	FF	44	00101100	054	2C	,	76	01001100	114	4C	L	108	01101100	154	6C	l
13	00001101	015	0D	CR	45	00101101	055	2D	-	77	01001101	115	4D	M	109	01101101	155	6D	m
14	00001110	016	0E	SO	46	00101110	056	2E	.	78	01001110	116	4E	N	110	01101110	156	6E	n
15	00001111	017	0F	SI	47	00101111	057	2F	/	79	01001111	117	4F	O	111	01101111	157	6F	o
16	00010000	020	10	DLE	48	00110000	060	30	0	80	01010000	120	50	P	112	01110000	160	70	p
17	00010001	021	11	DC1	49	00110001	061	31	1	81	01010001	121	51	Q	113	01110001	161	71	q
18	00010010	022	12	DC2	50	00110010	062	32	2	82	01010010	122	52	R	114	01110010	162	72	r
19	00010011	023	13	DC3	51	00110011	063	33	3	83	01010011	123	53	S	115	01110011	163	73	s
20	00010100	024	14	DC4	52	00110100	064	34	4	84	01010100	124	54	T	116	01110100	164	74	t
21	00010101	025	15	NAK	53	00110101	065	35	5	85	01010101	125	55	U	117	01110101	165	75	u
22	00010110	026	16	SYN	54	00110110	066	36	6	86	01010110	126	56	V	118	01110110	166	76	v
23	00010111	027	17	ETB	55	00110111	067	37	7	87	01010111	127	57	W	119	01110111	167	77	w
24	00011000	030	18	CAN	56	00111000	070	38	8	88	01011000	130	58	X	120	01111000	170	78	x
25	00011001	031	19	EM	57	00111001	071	39	9	89	01011001	131	59	Y	121	01111001	171	79	y
26	00011010	032	1A	SUB	58	00111010	072	3A	:	90	01011010	132	5A	Z	122	01111010	172	7A	z
27	00011011	033	1B	ESC	59	00111011	073	3B	;	91	01011011	133	5B	[123	01111011	173	7B	{
28	00011100	034	1C	FS	60	00111100	074	3C	<	92	01011100	134	5C	\	124	01111100	174	7C	
29	00011101	035	1D	GS	61	00111101	075	3D	=	93	01011101	135	5D]	125	01111101	175	7D	}
30	00011110	036	1E	RS	62	00111110	076	3E	>	94	01011110	136	5E	^	126	01111110	176	7E	~
31	00011111	037	1F	US	63	00111111	077	3F	?	95	01011111	137	5F	_	127	01111111	177	7F	DEL

More General Fixed Length Codes

Segment the sequence of source symbols into successive blocks of n source symbols at a time.



- Each source n -tuple is encoded into $L = \lceil \log_2 M^n \rceil$ bits.
- For each source symbol, $\bar{L} = L/n$: $\log_2 M \leq \bar{L} < \log_2 M + 1/n$

$$n \rightarrow \infty, \bar{L} \rightarrow \log_2 M.$$

Variable-length Codes for Discrete Sources

- A variable-length source code \mathcal{C} maps each symbol $x \in \mathcal{X}$ into a binary codeword $\mathcal{C}(x)$ of length $l(x)$.

- Example: Given $\mathcal{X} = \{a, b, c\}$

$$a \longrightarrow 0$$

$$b \longrightarrow 10$$

$$c \longrightarrow 11$$

Here, $l(a) = 1, l(b) = 2$ and $l(c) = 2$.

- Problem:
 - No commas or spaces: how to parse the received sequence:
100011010...
 - Buffering: fixed bit rate transmission.

Variable-length Codes for Discrete Sources

- Main Issue: unique decodability
 - \mathcal{C} is uniquely decodable if all concatenations of codewords are distinct

- Uniquely decodable:

Given $\mathcal{X} = \{a, b, c\}$

$a \longrightarrow 0$

$b \longrightarrow 10$

$c \longrightarrow 11$

Q: How to decode 010110?

- Non-uniquely decodable:

Given $\mathcal{X} = \{a, b, c\}$

$a \longrightarrow 0$

$b \longrightarrow 1$

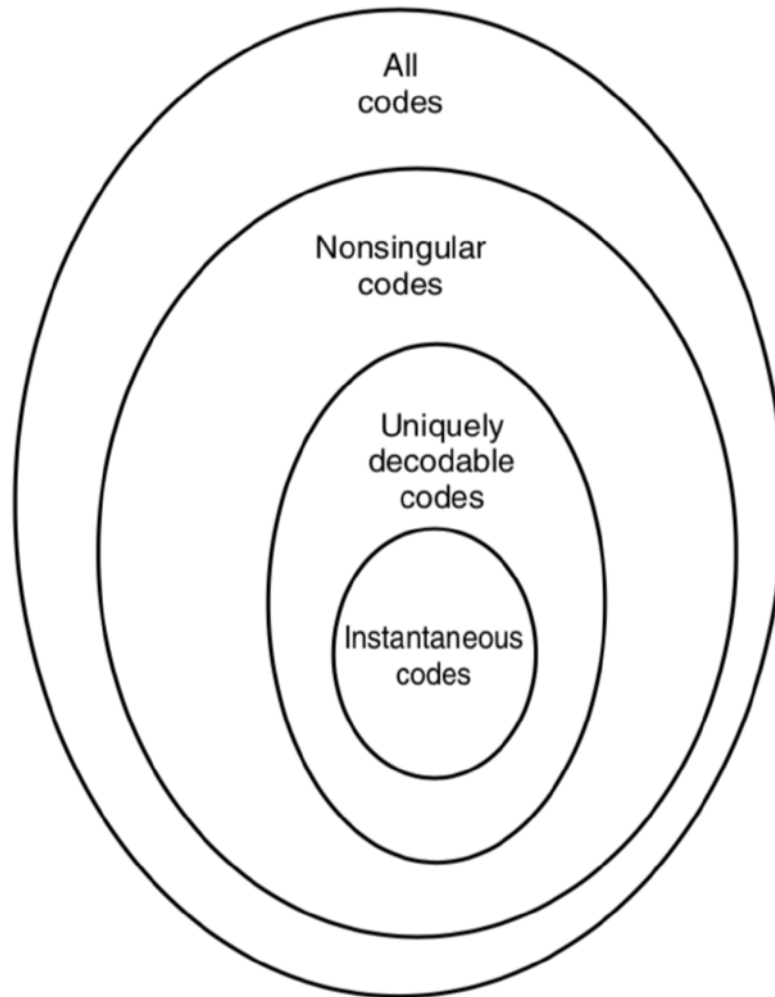
$c \longrightarrow 10$

Classes of Codes

X	Singular	Nonsingular, But Not Uniquely Decodable	Uniquely Decodable, But Not Instantaneous	Instantaneous
1	0	0	10	0
2	0	010	00	10
3	0	01	11	110
4	0	10	110	111

- Nonsingular: $x \neq x' \Rightarrow c(x) \neq c(x')$
- instantaneous code or prefix-free code

Classes of Codes



Requirements:

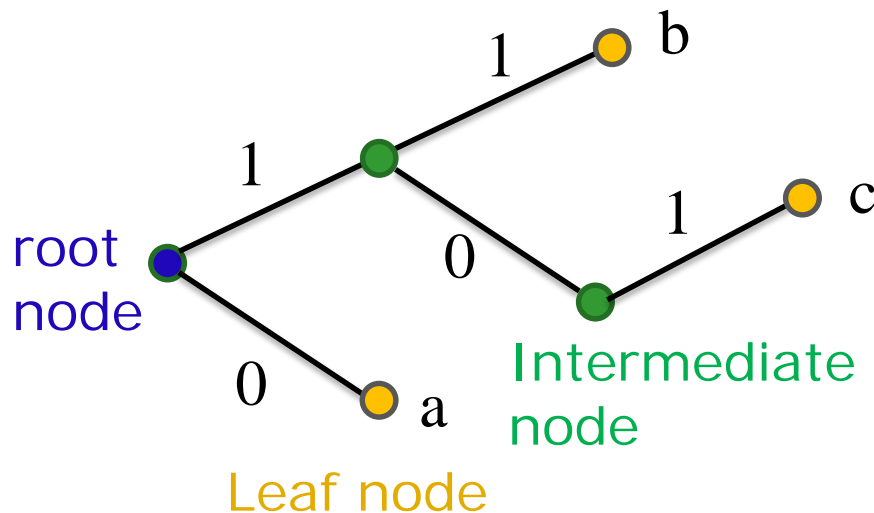
1. Uniquely decodable
2. Instantaneous
3. Systematic design method
4. Benchmark on the lowest possible bits/symbol than can be achieved.

Prefix-free codes

- A code is prefix-free if no codeword is a prefix of any other codeword.
- Example:
 - $\{0,01,11\}$ is not prefix-free, $\{0,10,11\}$ is prefix-free.
 - Fixed-length code is prefix-free.
- Q: Why prefix-free code is uniquely decodable?

Prefix-free codes

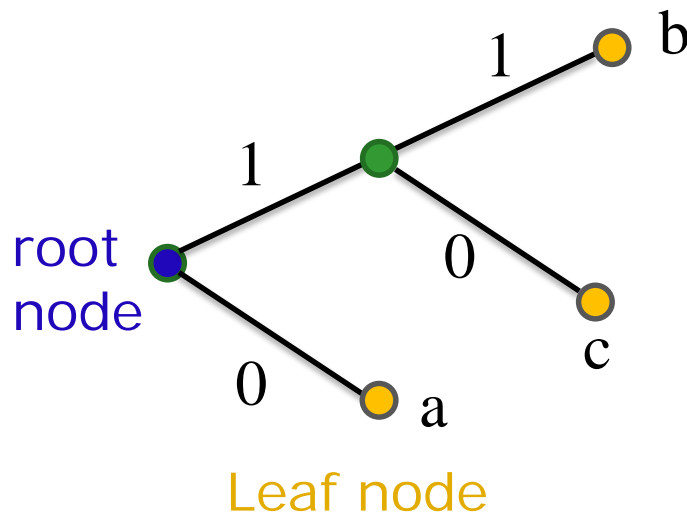
- A prefix-free code \mathcal{C} can be presented as **binary code tree** which grow from a root on the left to leaves on the right representing the codewords.



$a \rightarrow 0$
 $b \rightarrow 11$
 $c \rightarrow 101$

Prefix-free codes

- A prefix-free code \mathcal{C} can be presented as **binary code tree** which grow from a root on the left to leaves on the right representing the codewords.

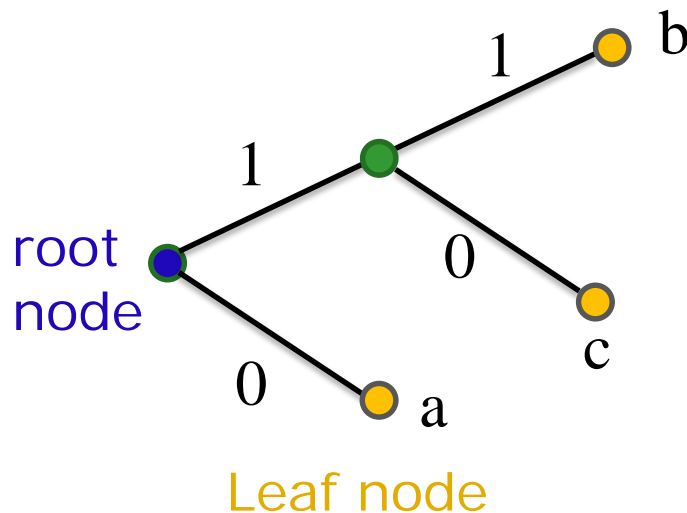


$a \rightarrow 0$
 $b \rightarrow 11$
 $c \rightarrow 10$

- Full prefix-free code:
- no codeword can be shortened and no codeword can be added.

Prefix-free codes

- Prefix-free codes are uniquely decodable.
- Source decoder: start at the left and parse whenever a leaf in the tree is reached.



Received sequence is

1010011...

c c a b

Kraft Inequality

- The kraft inequality is a condition determining on the existence of prefix-free codes with a given set of codeword lengths $\{l(x), x \in \mathcal{X}\}$.

Theorem (Kraft): Every prefix-free code for an alphabet \mathcal{X} with codeword lengths $\{l(x), x \in \mathcal{X}\}$ satisfies

$$\sum_{x \in \mathcal{X}} 2^{-l(x)} \leq 1$$

Proof: p24 of
Gallagar' book

- Conversely, if the inequality holds, then a prefix-free code with lengths $\{l(x)\}$ exists.
- A prefix-free code is full iff the equality above holds.

Minimum \bar{L} for Prefix-free Codes

- Objective: Choose $l(x)$ to minimize \bar{L} .
 - The expected length of codeword (bits per source symbol):

$$E(L) = \bar{L} = \sum_x p_X(x) l(x).$$

- Find \bar{L}_{min} over $l(x)$ s.t. Kraft Inequality $\sum_{x \in \mathcal{X}} 2^{-l(x)} \leq 1$.
- Example:
 - Consider an alphabet $\{a,b,c,d\}$ with symbol probabilities $\{1/2, 1/4, 1/8, 1/8\}$
 - Prefix-free code $\{0, 10, 110, 111\}$: $\bar{L} = \frac{1}{2} + \frac{1}{4} * 2 + \frac{1}{8} * 3 + \frac{1}{8} * 3 = 1.75$.
 - Prefix-free code $\{110, 0, 10, 111\}$: $\bar{L} = \frac{1}{2} * 3 + \frac{1}{4} * 1 + \frac{1}{8} * 2 + \frac{1}{8} * 3 = 2.375$ bits.

Minimum \bar{L} for Prefix-free Codes

- Entropy bounds on \bar{L}

Theorem 2.5.1 (Entropy bounds for prefix-free codes). *Let X be a discrete random symbol with symbol probabilities p_1, \dots, p_M . Let \bar{L}_{\min} be the minimum expected codeword length over all prefix-free codes for X . Then*

$$H[X] \leq \bar{L}_{\min} < H[X] + 1 \quad \text{bit/symbol.} \quad (2.7)$$

Furthermore, $\bar{L}_{\min} = H[X]$ if and only if each probability p_j is an integer power of 2.

- Proof:

- We first show that $H(X) \leq \bar{L}$ for all prefix-free codes.

- $H(X) - \bar{L} = \sum \left(p_j \log \frac{1}{p_j} - p_j l_j \right) = \sum_{j=1}^M p_j \left(\log \frac{2^{-l_j}}{p_j} \right) \leq$

$$\log e \sum_{j=1}^M p_j \left(\frac{2^{-l_j}}{p_j} - 1 \right) = \log e \left(\sum_{j=1}^M 2^{-l_j} - 1 \right) \leq 0$$

Kraft Inequality

Minimum \bar{L} for Prefix-free Codes

- Entropy bounds on \bar{L}

Theorem 2.5.1 (Entropy bounds for prefix-free codes). *Let X be a discrete random symbol with symbol probabilities p_1, \dots, p_M . Let \bar{L}_{\min} be the minimum expected codeword length over all prefix-free codes for X . Then*

$$H[X] \leq \bar{L}_{\min} < H[X] + 1 \quad \text{bit/symbol.} \quad (2.7)$$

Furthermore, $\bar{L}_{\min} = H[X]$ if and only if each probability p_j is an integer power of 2.

- Proof:

- We first show that $H(X) \leq \bar{L}$ for all prefix-free codes.
- Equality holds when $2^{-l_j} = p_j$ or $l_j = -\log p_j, \forall j$.

$\bar{L}_{\min} = H(X)$ iff p_j is an integer power of 2 for all j .

Minimum \bar{L} for Prefix-free Codes

- Entropy bounds on \bar{L}

Theorem 2.5.1 (Entropy bounds for prefix-free codes). *Let X be a discrete random symbol with symbol probabilities p_1, \dots, p_M . Let \bar{L}_{\min} be the minimum expected codeword length over all prefix-free codes for X . Then*

$$H[X] \leq \bar{L}_{\min} < H[X] + 1 \quad \text{bit/symbol.} \quad (2.7)$$

Furthermore, $\bar{L}_{\min} = H[X]$ if and only if each probability p_j is an integer power of 2.

- Proof:

- We show that a prefix-free code exists with $\bar{L} < H(X) + 1$.

- Choose $l_j = \lceil -\log p_j \rceil \Rightarrow \underbrace{-\log p_j}_{\text{integer part}} \leq l_j < \underbrace{-\log p_j + 1}_{\text{fractional part}}.$

$\sum_j 2^{-l_j} \leq 1 \Rightarrow \text{Prefix-free code exists.}$

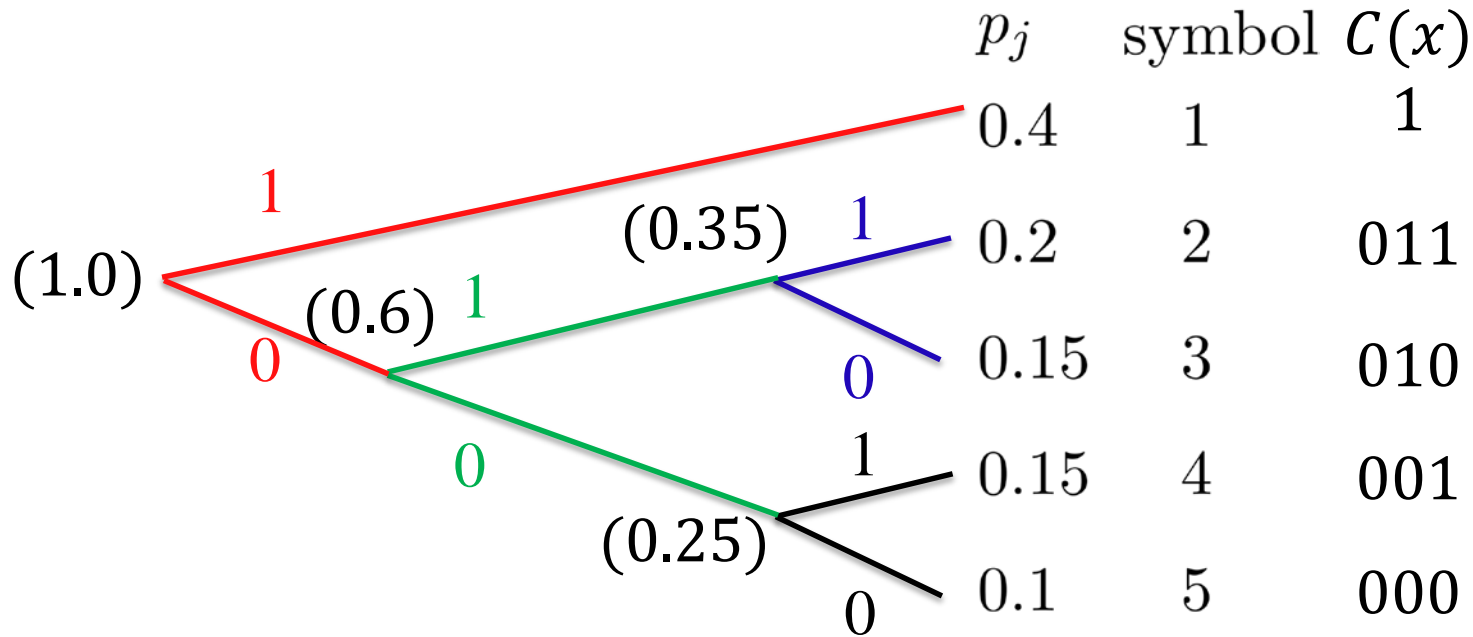
- $\bar{L}_{\min} \leq \bar{L} \Rightarrow \bar{L}_{\min} < H(X) + 1.$

$$\bar{L} < \sum_{j=1}^M -p_j \log p_j + 1 = H(X) + 1$$

Huffman Coding

- Above theorem suggested that good codes have length $l_j \approx -\log p_j$ (Note: Many researchers trying to find code using this way, but it turns out to work not well!!)
- Any optimal source code should satisfy:
 - if $p_i > p_j$, then $l_i \leq l_j$.
 - Optimal prefix-free codes has a **full code tree**.
 - Let X be a random symbol with a pmf satisfying $p_1 \geq p_2 \geq \dots \geq p_M$. There is an optimal prefix-free code for X in which the codewords for $M - 1$ and M are siblings and have maximal length within the code.

Huffman Coding



$$\bar{L} = 0.4 * 1 + 0.6 * 3 = 2.2 \text{ bits}$$

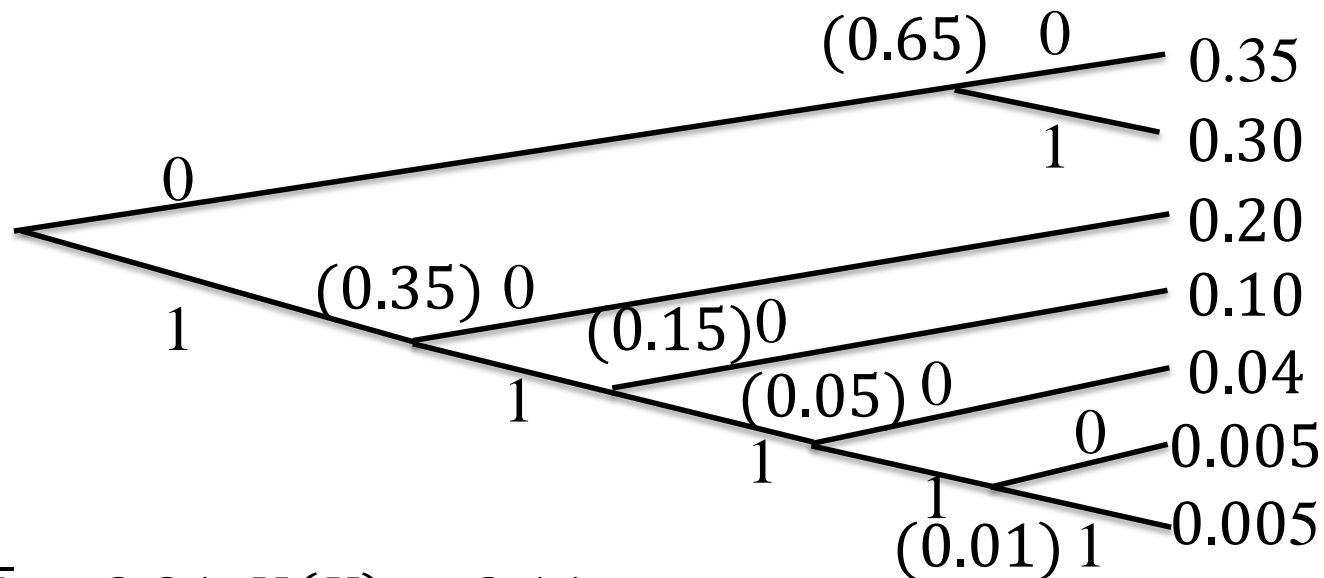
$$H(X)$$

$$= -0.4 * \log 0.4 - 0.2 * \log 0.2 - 2 * 0.15 * \log 0.15 - 0.1 * \log 0.1 = 2.1464 \text{ bits}$$

Huffman Coding

Problem: Given a DMS $X \in \{a_1, a_2, \dots, a_7\}$, with probability $\{0.35, 0.30, 0.20, 0.10, 0.04, 0.005, 0.005\}$.

- Design a Huffman code for this source
- Find \bar{L} , average codeword length
- Determine the efficiency of the code $\eta = \frac{H(X)}{\bar{L}}$

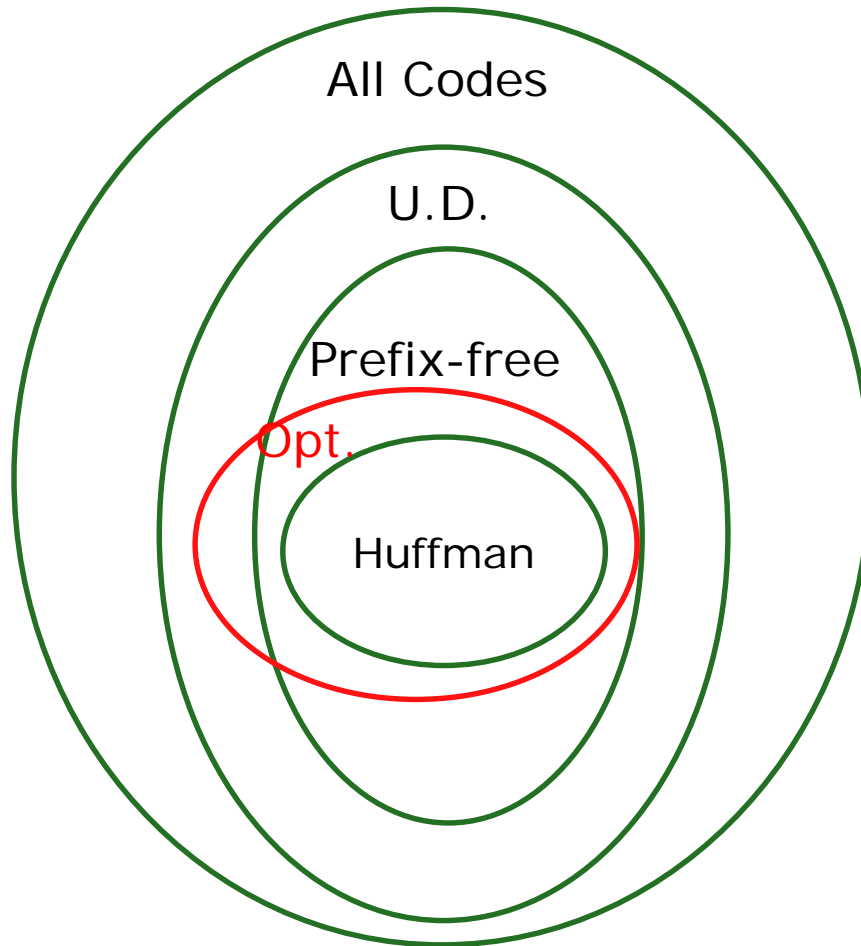


$$\bar{L} = 2.21, H(X) = 2.11.$$

Optimality of Huffman Code (Skip)

- Huffman is optimal for symbol-to-symbol coding with a known input probability distribution.
- Proof: (an optimal code for the reduced random symbol X' yields an optimal code for X)
 - Huffman algorithm chooses an optimal code tree by starting with two least likely symbols, specifically M and $M-1$.
 - Let X' be the reduced RV from X (Combining the two smallest probability symbols). Let \bar{L}' be the expected length of X' . Then the optimal L satisfies
$$\bar{L} = \bar{L}' + p_{M-1} + p_M.$$
(Extending the codeword $C'(M-1)$ into two sibling for $M-1$ and M)
 - $\bar{L}_{min} = \bar{L}'_{min} + p_{M-1} + p_M$
 - Using Huffman algorithm, an optimal code for X' yields an optimal code for X . Prove X'' to X' and so forth, down to a binary symbol.

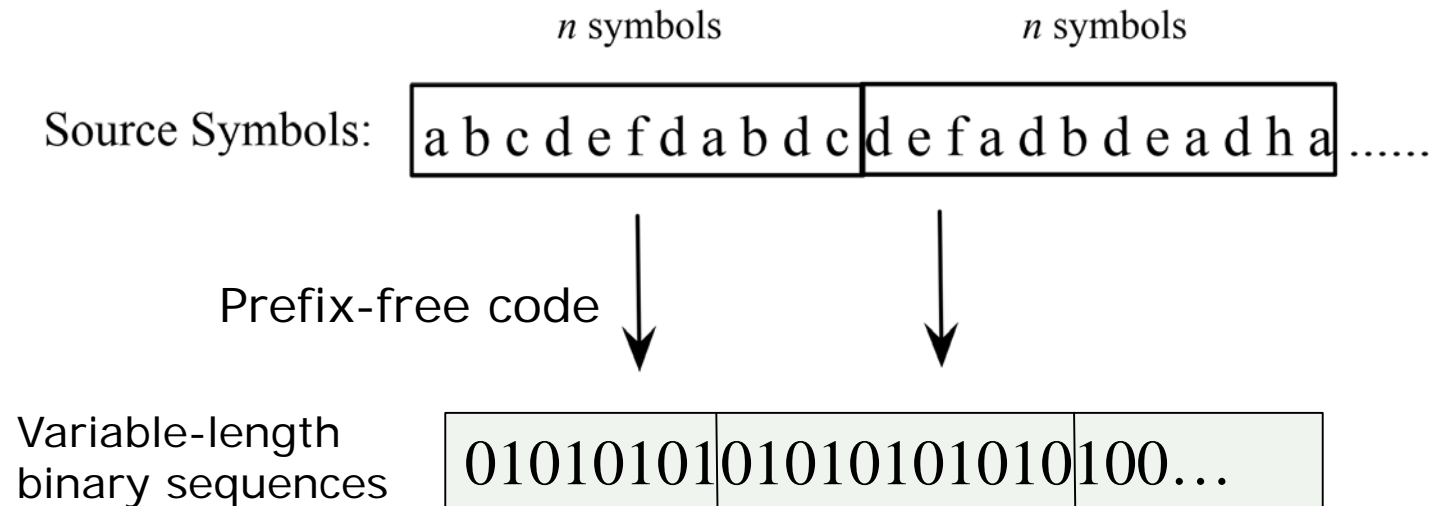
Optimality of Huffman Code



- No code is uniquely optimal.
- The set of lengths of an optimal code is not unique.
- Not every optimal u.d. code is Huffman code.
- Not every optimal prefix-free code is Huffman code.

Huffman Code for Encoding a Block

- Fixed-to-variable length code



- Q: what is the minimum number of bits per source symbol?

Huffman Code for Encoding a Block

- The minimum number of bits per source symbol over all prefix-free code for X^n is

$$H(X) \leq \bar{L}_{min,n} < H(X) + \frac{1}{n}.$$

- Proof:
 - $H(X^n) = H(X_1, \dots, X_n) = nH(X)$, with X_i i.i.d $\sim P_x$
 - Take $X^n \in \mathcal{X}^n$ as a big “source RV”
 - By $H(X) \leq \bar{L}_{min} < H(X) + 1$, we have

$$nH(X) = H(X^n) \leq \bar{L}(X^n)_{min} < H(X^n) + 1 = nH(X) + 1$$

- $nH(X) \leq \bar{L}_n < nH(X) + 1$

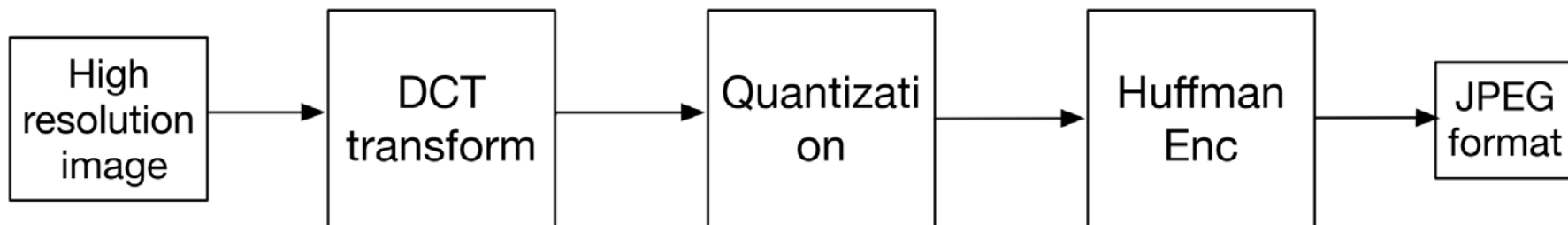
Huffman Code for Encoding a Block

- Example: A DMS $\{x_1, x_2, x_3\}$ has probability $\{0.45, 0.35, 0.2\}$.

Letter pair	Probability	Self-information	Code
x_1x_1	0.2025	2.312	10
x_1x_2	0.1575	2.676	001
x_2x_1	0.1575	2.676	010
x_2x_2	0.1225	3.039	011
x_1x_3	0.09	3.486	111
x_3x_1	0.09	3.486	0000
x_2x_3	0.07	3.850	0001
x_3x_2	0.07	3.850	1100
x_3x_3	0.04	4.660	1101

- Using symbol-to-symbol Huffman code: $\bar{L} = 1.55$, $\eta = 97.6\%$
- Using Block Huffman code:
 - $\bar{L}_2 = 3.067$, $\bar{L} = \bar{L}_2/2 = 1.534 < 1.55$
 - $\eta = 98.6\% > 97.6\%$

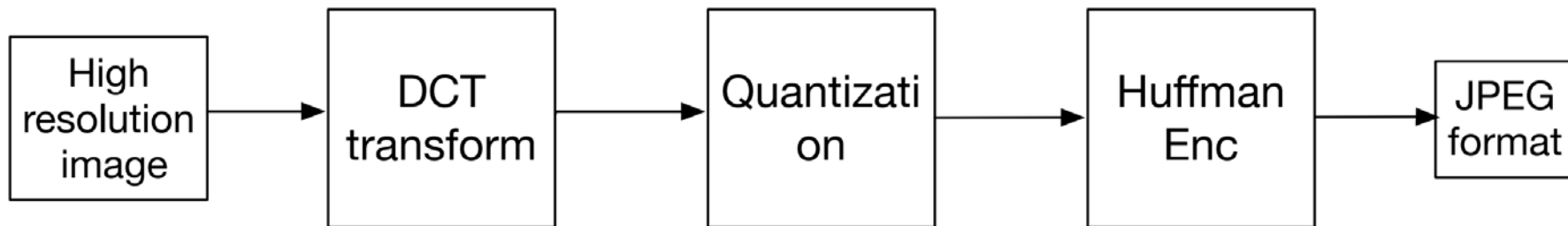
Application of Huffman Code



- Discrete Cosine Transform (widely used in video and audio compression): $y = Cx$, where C is an $n \times n$ transformation matrix:

$$C = \sqrt{\frac{2}{n}} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \dots & \frac{1}{\sqrt{2}} \\ \cos \frac{\pi}{2n} & \cos \frac{3\pi}{2n} & \dots & \cos \frac{(2n-1)\pi}{2n} \\ \cos \frac{2\pi}{2n} & \cos \frac{6\pi}{2n} & \dots & \cos \frac{2(2n-1)\pi}{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \cos \frac{(n-1)\pi}{2n} & \cos \frac{(n-1)3\pi}{2n} & \dots & \cos \frac{(n-1)(2n-1)\pi}{2n} \end{bmatrix}$$

Application of Huffman Code



- Discrete Cosine Transform (widely used in video and audio compression): $y = Cx$, where C is an $n \times n$ transformation matrix:



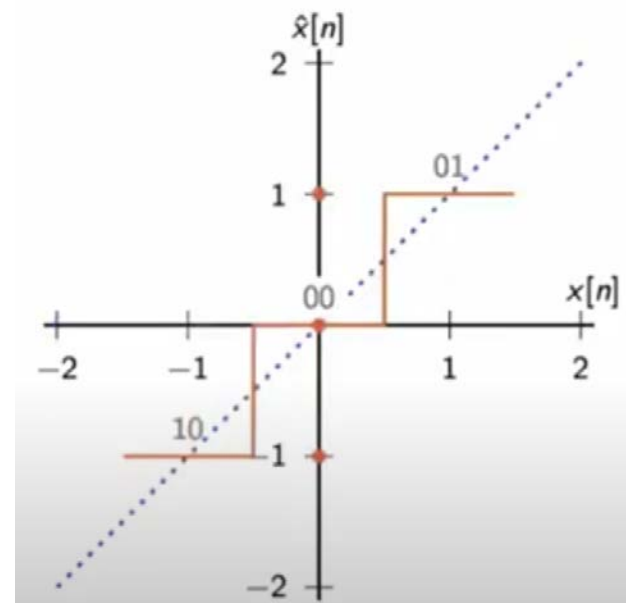
Application of Huffman Code

- Smart quantization: $\hat{c}[k_1, k_2] = \text{round}(c[k_1, k_2]/Q[k_1, k_2])$

Table I the default JPEG quantization table (DQT)

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

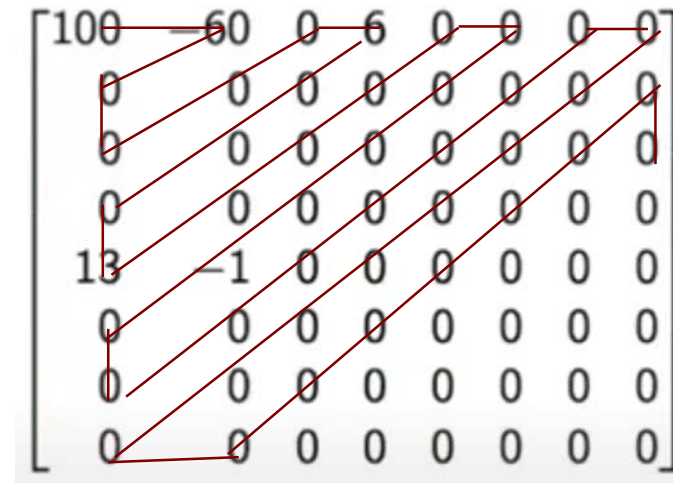
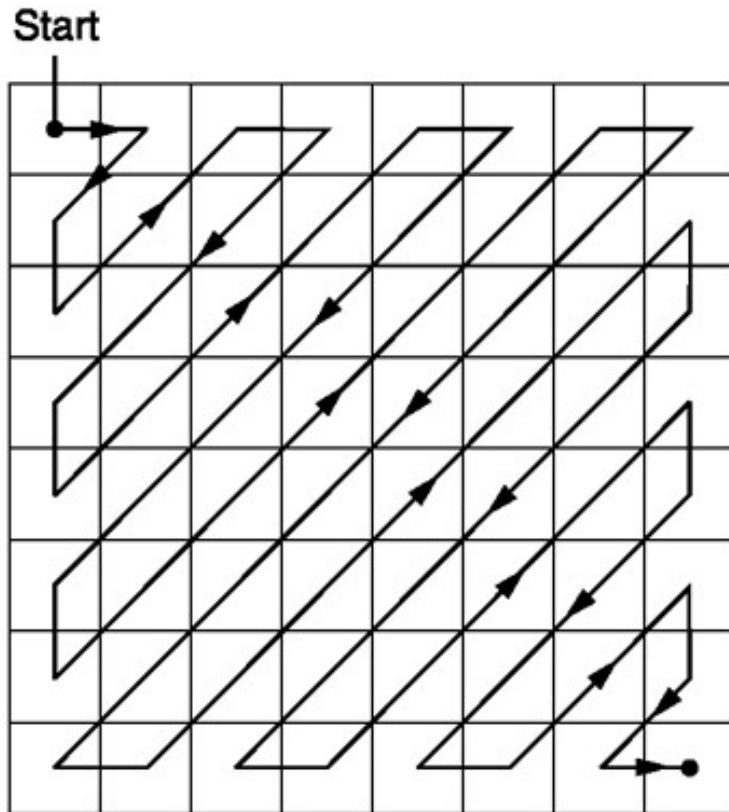
$Q =$



- nonuniform bit allocation strategy

Application of Huffman Code

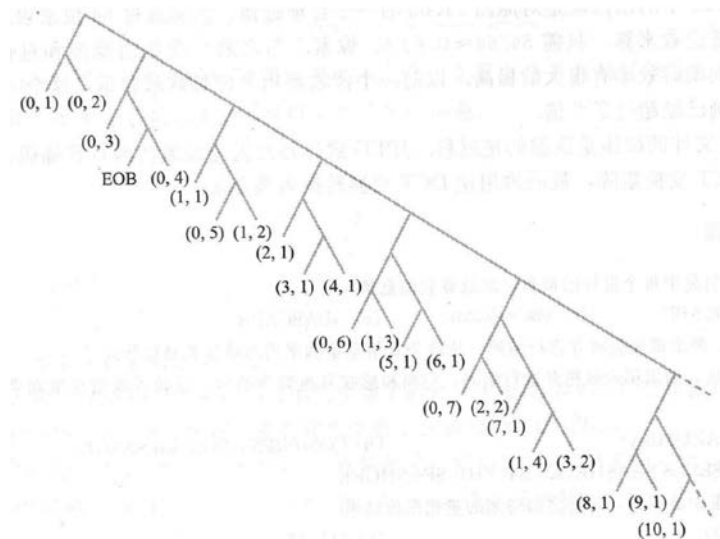
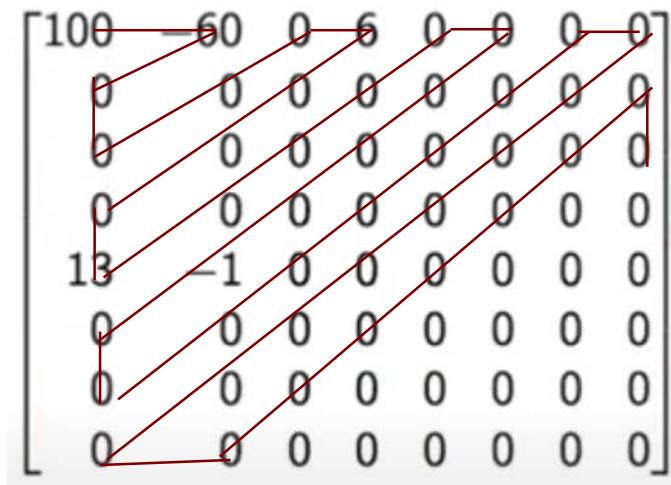
- Entropy coding:



100,-60,0,0,0,0,6,0,0,0,13,0,0,0,0,0,0,0,-1,0,0,0,0,...

Application of Huffman Code

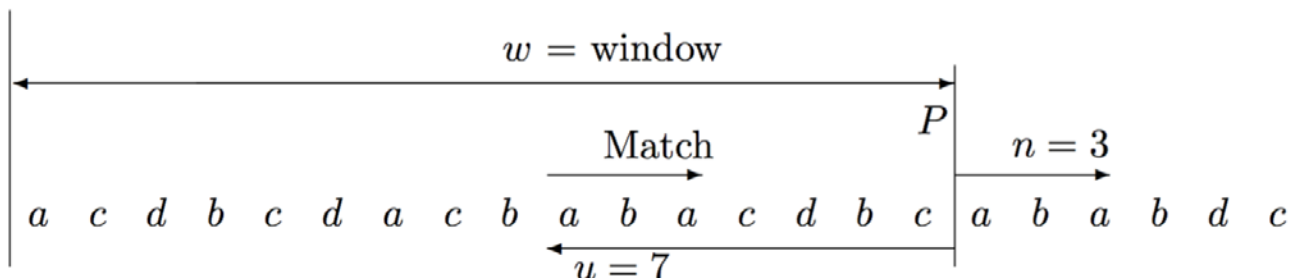
- Entropy coding:
 - Runlength Encoding: $[(r,s),c]$
 - $[(0,7),100],[(0,6),-60],[(4,3),6],[(3,4),13],[(8,1),-1],[(0,0)]$
 - $(r,s) \in \mathcal{A}, |\mathcal{A}| = 256$.
 - Some pairs are much more common than others!
 - Huffman coding to save the bit rate.



Lempel-Ziv Data Compression

- Variable-to-variable-length codes
- Don't require prior knowledge of source statistics
- Adapt to minimize average code length
- Widely used in practice (GIF format)


Lempel-Ziv Data Compression



Set window size w

- ① Encode the first w symbols in a fixed length code, without compression
- ② Set pointer $P = w$
- ③ Find the largest $n \geq 2$ such that $x_{P+1}^{P+n} = x_{P+1-u}^{P+n-u}$ for some $u \in [1, w]$.
 x_{P+1}^{P+n} is encoded by encoding n and u (p. 53)
 - Encode n into a codeword from the unary-binary code
 - Encode $u \leq w$ using fixed-length code of length $\log w$
- ④ Set the pointer P to $P + n$ and go to step (3). Iterate forever

Lempel-Ziv Data Compression

$\lceil \log_2 n \rceil$ zeroes


n	prefix	base 2 exp.	codeword
1		1	1
2	0	10	010
3	0	11	011
4	00	100	00100
5	00	101	00101
6	00	110	00110
7	00	111	00111
8	000	1000	0001000

unary-binary code of n

Lossless Source Coding Theorem

Shannon's First Theorem: Let X denote a DMS with entropy $H(X)$, there exists a lossless source code for this source at any rate $R > H(X)$. There exists no lossless code for this source at rates less than $H(X)$.

Achievability:

Recall prefix-free code: Given a DMS X , the minimum expected codeword length for all prefix-free code satisfies

$$H(X) \leq \bar{L}_{\min} < H(X) + 1$$

Converse: If $R < H(X)$, then the error probability approaches 1 for large n . See p. 44 [Gallager'44].

Understanding by AEP: "Typical" sequence \rightarrow "Typical" set.

$H(X)$ is a lowerbound to \bar{L} over all uniquely decodable encoding techniques.

Summary

- Lossless Source Coding (Shannon's First Theorem)
- Fixed-length code (ignore the source distribution)
 - $\log_2 M \leq L < \log_2 M + 1$
- Fixed-to-fixed-length code: $\log_2 M \leq \bar{L} < \log_2 M + \frac{1}{n}$
- Variable-length code (parse problem \rightarrow prefix-free code \rightarrow Kraft-inequality \rightarrow Huffman-code)
 - Entropy bound for prefix-free code: $H(X) \leq \bar{L}_{min} < H(X) + 1.$
- Fixed-to-variable-length code: $H(X) \leq \bar{L}_{min,n} < H(X) + \frac{1}{n}.$
- Variable-to-variable-length code: Lemp-Ziv code
- Lossless Source Coding Theorem: $R > H(X)$ is sufficient to decode X



Thanks for your kind attention!

Questions?