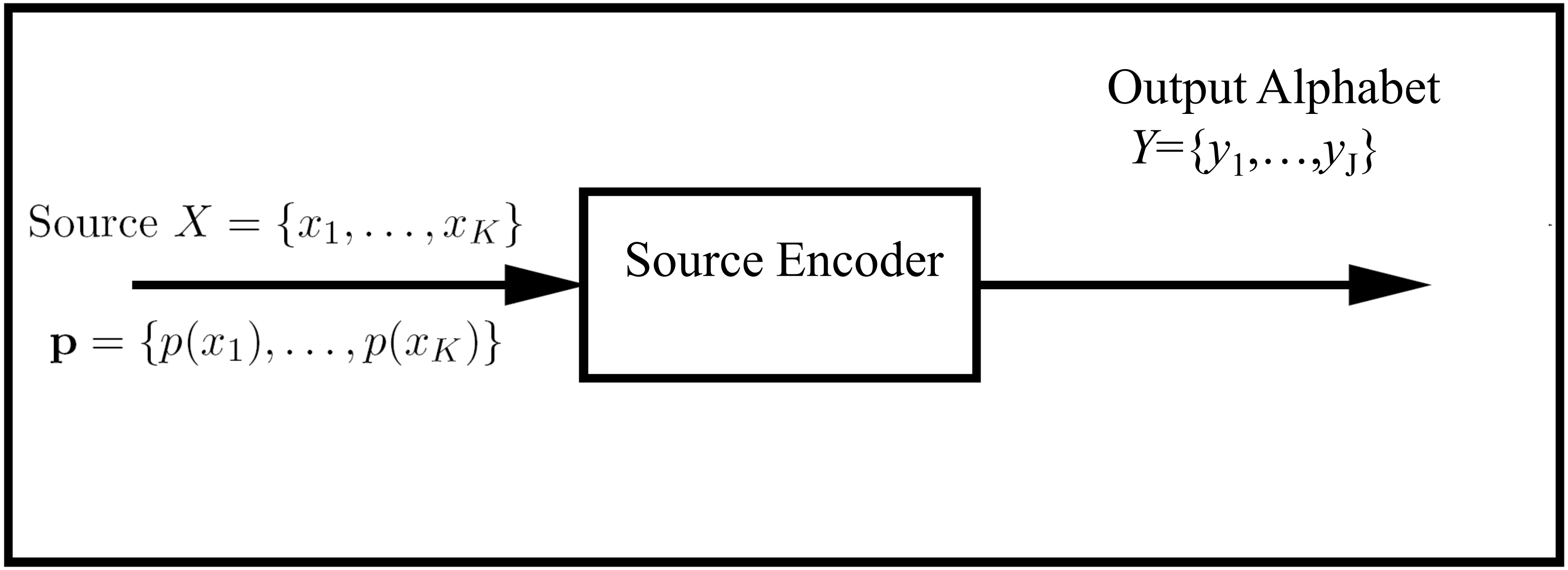
ECE 515
Information Theory

Distortionless Source Coding 1

Source Coding

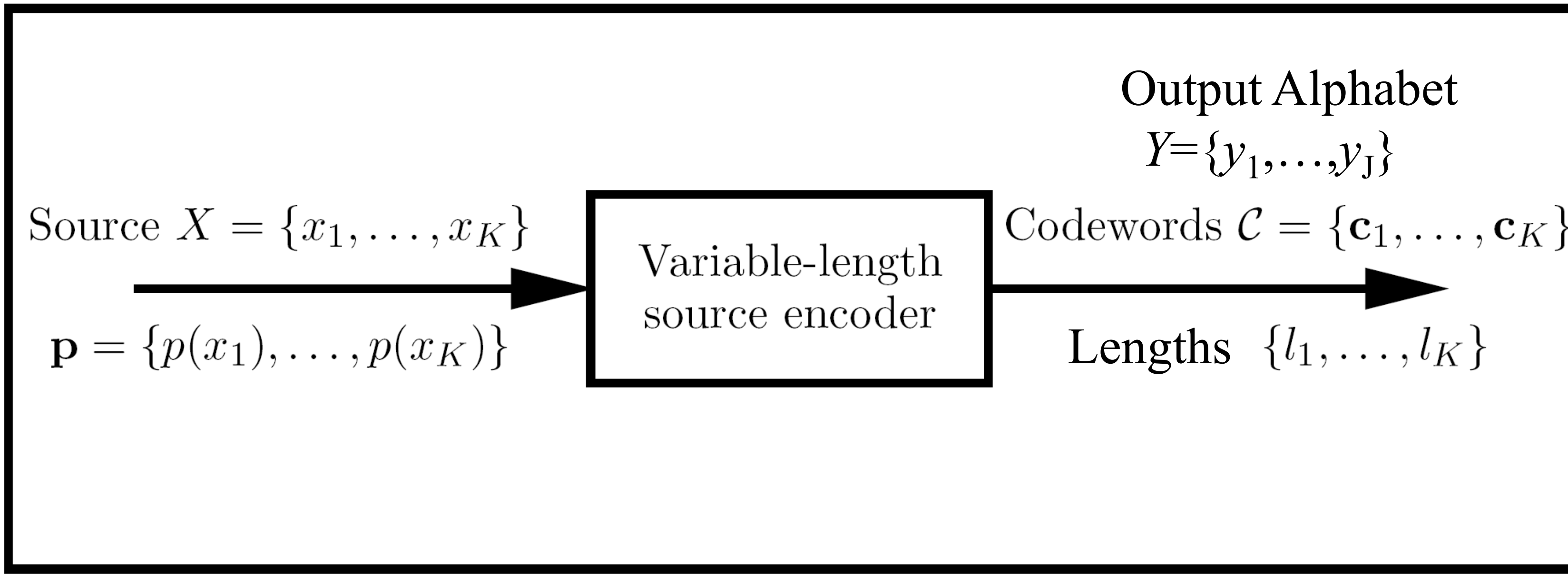


Source Coding

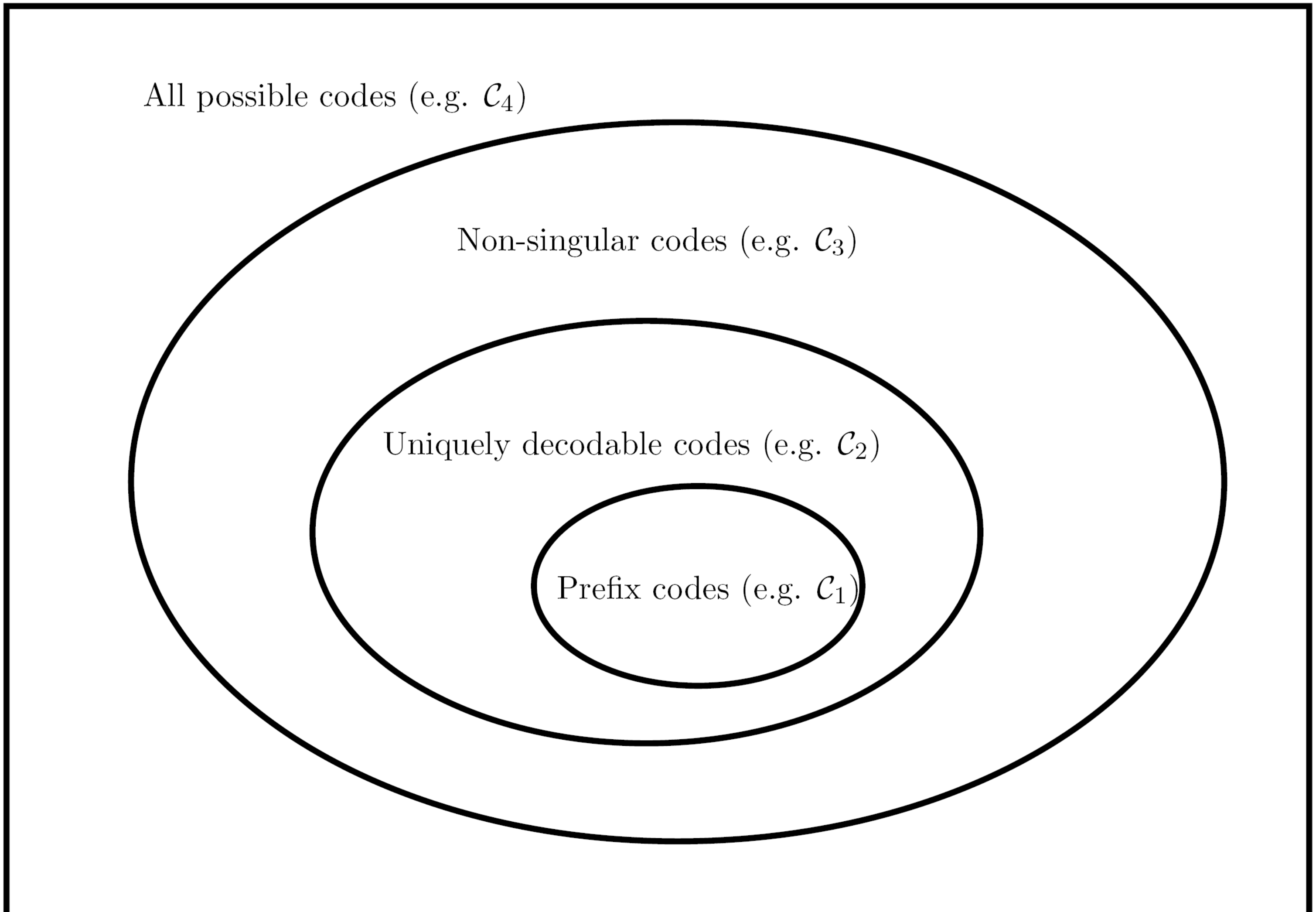
Two requirements

1. The source sequence can be recovered from the encoded sequence with no ambiguity.
2. The average number of output symbols per source symbol is as small as possible.

Variable Length Codes



Variable Length Codes



Variable Length Codes

- Let $K = 4$, X = $\{x_1, x_2, x_3, x_4\}$, $J = 2$
- Prefix code (also prefix-free or instantaneous)

  $C_1=\{0,10,110,111\}$
- Example sequence of codewords: 001110100110
- Decodes to:

  0 0 111 0 10 0 110

  $x_1\ x_1\ \ x_4\ \ x_1\ \ x_2\ \ x_1\ \ x_3$

Instantaneous Codes

- Definition:

  A uniquely decodable code is said to be **instantaneous** if it is possible to decode each codeword in a sequence without reference to succeeding codewords.

  A necessary and sufficient condition for a code to be instantaneous is that no codeword is a **prefix** of some other codeword.

Variable Length Codes

- Uniquely decodable code (which is not prefix)

  C_2={0,01,011,0111}
- Example sequence of codewords: 001110100110
- Decodes to:

  0 0111 01 0 011 0

  x_1 x_4 x_2 x_1 x_3 x_1

Variable Length Codes

- Non-singular code (which is not uniquely decodable)

  $C_3=\{0,1,00,11\}$
- Example sequence of codewords:

  001110100110
- Decodes to:

  0 0 1 1 1 0 1 0 0 1 1 0

  $x_1\ x_1\ x_2\ x_2\ x_2\ x_1\ x_2\ x_1\ x_1\ x_2\ x_2\ x_1$

  00 11 1 0 1 00 11 0

  $x_3\ x_4\ x_2\ x_1\ x_2\ x_3\ x_4\ x_1$

Variable Length Codes

- Singular code

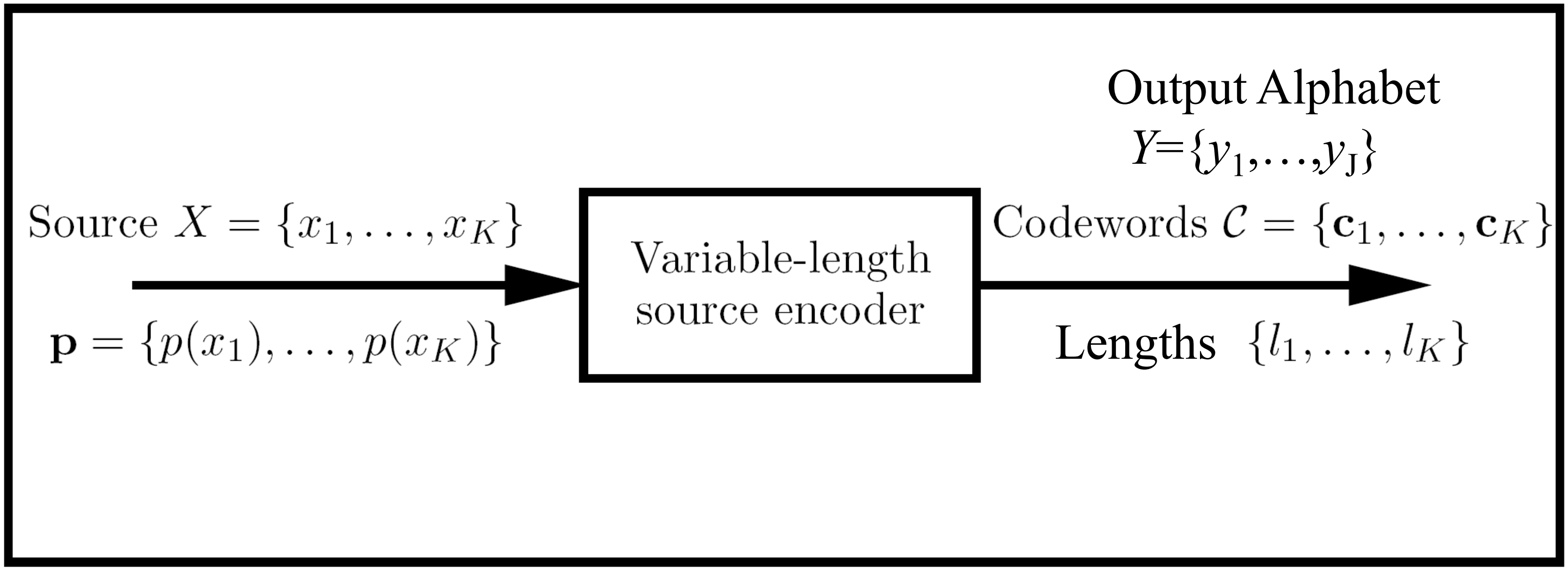
  $C_4=\{0,10,11,10\}$
- Example sequence of codewords: 001110100110
- Decodes to:

  0 0 11 10 10 0 11 0

  $x_1\ x_1\ x_3\ x_2\ x_2\ x_1\ x_3\ x_1$

  $x_1\ x_1\ x_3\ x_4\ x_2\ x_1\ x_3\ x_1$

Variable Length Codes



Variable Length Codes

Source Symbol	Codeword	Codeword Length
x_1	$\mathbf{c}_1 = (c_{1,1}, c_{1,2}, \ldots, c_{1,l}, \ldots, c_{1,l_1})$	l_1
x_2	$\mathbf{c}_2 = (c_{2,1}, c_{2,2}, \ldots, c_{2,l}, \ldots, c_{2,l_2})$	l_2
$\vdots$	$\vdots$	$\vdots$
x_k	$\mathbf{c}_k = (c_{k,1}, c_{k,2}, \ldots, c_{k,l}, \ldots, c_{k,l_k})$	l_k
$\vdots$	$\vdots$	$\vdots$
x_K	$\mathbf{c}_K = (c_{K,1}, c_{K,2}, \ldots, c_{K,l}, \ldots, c_{K,l_K})$	l_K

Average Codeword Length

$$\mathrm{L(C)} = \sum_{k=1}^{K} \mathrm{p}(x_k) l_k$$

Two Binary Prefix Codes

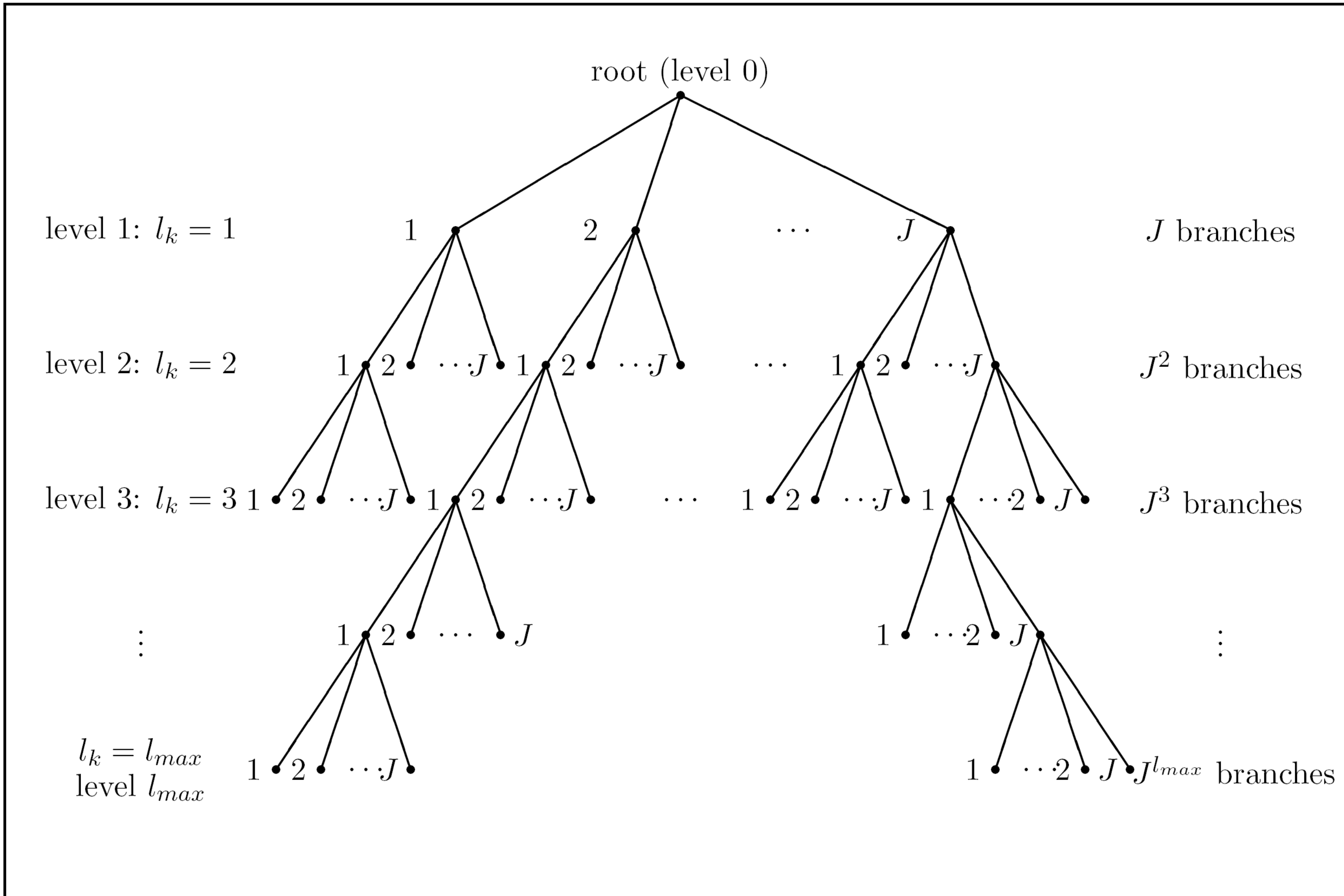
- Five source symbols: x_1, x_2, x_3, x_4, x_5
- K = 5, J = 2

- $c_1 = 0$, $c_2 = 10$, $c_3 = 110$, $c_4 = 1110$, $c_5 = 1111$
  - codeword lengths 1,2,3,4,4
- $c_1 = 00$, $c_2 = 01$, $c_3 = 10$, $c_4 = 110$, $c_5 = 111$
  - codeword lengths 2,2,2,3,3

Kraft Inequality for Prefix Codes

$$\sum_{k=1}^{K} J^{-l_k} \leq 1$$

Code Tree



root (level 0)
level 1: $l_k = 1$
1
2
$\cdots$
J
J branches
level 2: $l_k = 2$
J^2 branches
level 3: $l_k = 3$
J^3 branches
$l_k = l_{max}$
level l_{max}
$J^{l_{max}}$ branches

Five Binary Codes

Source symbols	Code A	Code B	Code C	Code D	Code E
x_1	00	0	0	0	0
x_2	01	100	10	100	10
x_3	10	110	110	110	110
x_4	11	111	111	11	11

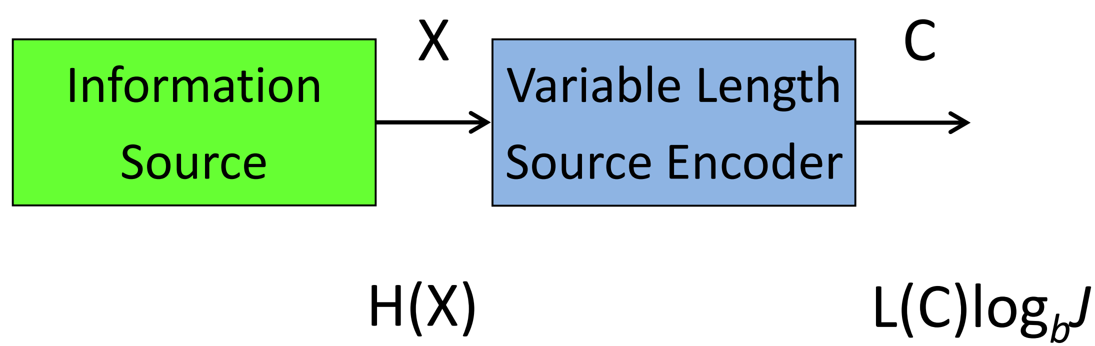
Ternary Code Example

- Ten source symbols: $x_1, x_2, \ldots, x_9, x_{10}$
- $K = 10, J = 3$

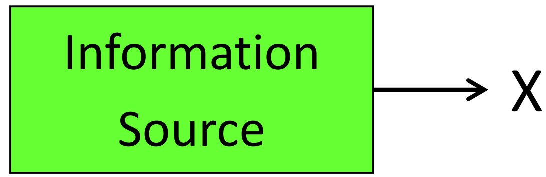
- $l_k = 1,2,2,2,2,2,3,3,3,3$
- $l_k = 1,2,2,2,2,2,3,3,3$
- $l_k = 1,2,2,2,2,2,3,3,4,4$

Average Codeword Length Bound

$$L(C) \geq \frac{H(X)}{\log_b J}$$



Four Symbol Source



- $p(x_1) = 1/2$ $p(x_2) = 1/4$ $p(x_3) = p(x_4) = 1/8$
- $H(X) = 1.75$ bits

x_1	0	x_1	00
x_2	10	x_2	01
x_3	110	x_3	10
x_4	111	x_4	11

$L(C) = 1.75$ bits $L(C) = 2$ bits

Code Efficiency

$$\zeta = \frac{\mathrm{H(X)}}{\mathrm{L(C)}\log_b J} \leq 1$$

- First code $\zeta = 1.75/1.75 = 100\%$
- Second code $\zeta = 1.75/2.0 = 87.5\%$

Compact Codes

A code C is called **compact** for a source X if its average codeword length L(C) is less than or equal to the average length of all other uniquely decodable codes for the same source and alphabet Y size *J*.

Codeword Lengths

$$\mathrm{H}(X) = -\sum_{k=1}^{K} \mathrm{p}(x_k) \log \mathrm{p}(x_k)$$

$$\mathrm{L}(C) = \sum_{k=1}^{K} \mathrm{p}(x_k) l_k$$

Upper and Lower Bounds for a Compact Code

$$\frac{H(X)}{\log_b J} \leq L(C) < \frac{H(X)}{\log_b J} + 1$$

if $b = J$

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

The Shannon Algorithm

- Order the symbols from largest to smallest probability
- Choose the codeword lengths according to

$$l_k = \lceil -\log_J \mathrm{p}(x_k) \rceil$$

- Construct the codewords according to the cumulative probability P_k

$$P_k = \sum_{i=1}^{k-1} \mathrm{p}(x_i)$$

expressed as a base J number

Example

- $K = 10$, $J = 2$
- $p(x_1) = p(x_2) = 1/4$
- $p(x_3) = p(x_4) = 1/8$
- $p(x_5) = p(x_6) = 1/16$
- $p(x_7) = p(x_8) = p(x_9) = p(x_{10}) = 1/32$

Converting Decimal Fractions to Binary

- To convert a fraction to binary, multiply it by 2
- If the integer part is 1, the binary digit is 1, otherwise it is 0
- Delete the integer part
- Continue multiplying by 2 and obtaining binary digits until the resulting fractional part is 0 or the required number of binary digits have been obtained

Example

- Convert $5/8 = 0.625_{10}$ to binary
- $2 \times 0.625 = 1.25 = 1 + 0.25$ MSB
- $2 \times 0.250 = 0.50 = 0 + 0.50$
- $2 \times 0.500 = 1.00 = 1 + 0.00$ LSB
- $0.625_{10} = 0.101_2$

Example

Symbol	$p(x_k)$	P_k	l_k	Codeword
x_1	$1/4$	0	2	00
x_2	$1/4$	$1/4$	2	01
x_3	$1/8$	$1/2$	3	100
x_4	$1/8$	$5/8$	3	101
x_5	$1/16$	$3/4$	4	1100
x_6	$1/16$	$13/16$	4	1101
x_7	$1/32$	$7/8$	5	11100
x_8	$1/32$	$29/32$	5	11101
x_9	$1/32$	$15/16$	5	11110
x_{10}	$1/32$	$31/32$	5	11111

Shannon Algorithm

- $p(x_1) = .4$ $p(x_2) = .3$ $p(x_3) = .2$ $p(x_4) = .1$
- $H(X) = 1.85$ bits

Shannon Code	Alternate Code
x_1 00	x_1 0
x_2 01	x_2 10
x_3 101	x_3 110
x_4 1110	x_4 111
$L(C) = 2.4$ bits	$L(C) = 1.9$ bits
$\zeta = 77.1\%$	$\zeta = 97.4\%$

Shannon's Noiseless Coding Theorem

$$\frac{N\mathrm{H}(\mathrm{X})}{\log_b J} \le \mathrm{L}_N(\mathrm{C}) < \frac{N\mathrm{H}(\mathrm{X})}{\log_b J} + 1$$

$$\frac{\mathrm{H}(\mathrm{X})}{\log_b J} \le \frac{\mathrm{L}_N(\mathrm{C})}{N} < \frac{\mathrm{H}(\mathrm{X})}{\log_b J} + \frac{1}{N}$$

Shannon's Noiseless Coding Theorem

If $b = J$

$$N\mathrm{H}(\mathrm{X}) \leq \mathrm{L}_N(\mathrm{C}) < N\mathrm{H}(\mathrm{X}) + 1$$

$$\mathrm{H}(\mathrm{X}) \leq \frac{\mathrm{L}_N(\mathrm{C})}{N} < \mathrm{H}(\mathrm{X}) + \frac{1}{N}$$

Robert M. Fano (1917-2016)

The Fano Algorithm

- Arrange the symbols in order of decreasing probability
- Divide the symbols into *J* approximately equally probable groups
- Each group receives one of the *J* code symbols as the **first** codeword symbol
- This division process is repeated within the groups as many times as possible

Example

Symbol	$p(x_k)$					
x_1	1/4	0	0			
x_2	1/4	0	1			
x_3	1/8	1	0	0		
x_4	1/8	1	0	1		
x_5	1/16	1	1	0	0	
x_6	1/16	1	1	0	1	
x_7	1/32	1	1	1	0	0
x_8	1/32	1	1	1	0	1
x_9	1/32	1	1	1	1	0
x_{10}	1/32	1	1	1	1	1

Shannon Algorithm vs Fano Algorithm

- $p(x_1) = .4$ $p(x_2) = .3$ $p(x_3) = .2$ $p(x_4) = .1$
- $H(X) = 1.85$ bits

Shannon Code	Fano Code
x_1 00	x_1 0
x_2 01	x_2 10
x_3 101	x_3 110
x_4 1110	x_4 111
$L(C) = 2.4$ bits	$L(C) = 1.9$ bits
$\zeta = 77.1\%$	$\zeta = 97.4\%$

Upper Bound for the Fano Code $J \in \{2,3\}$

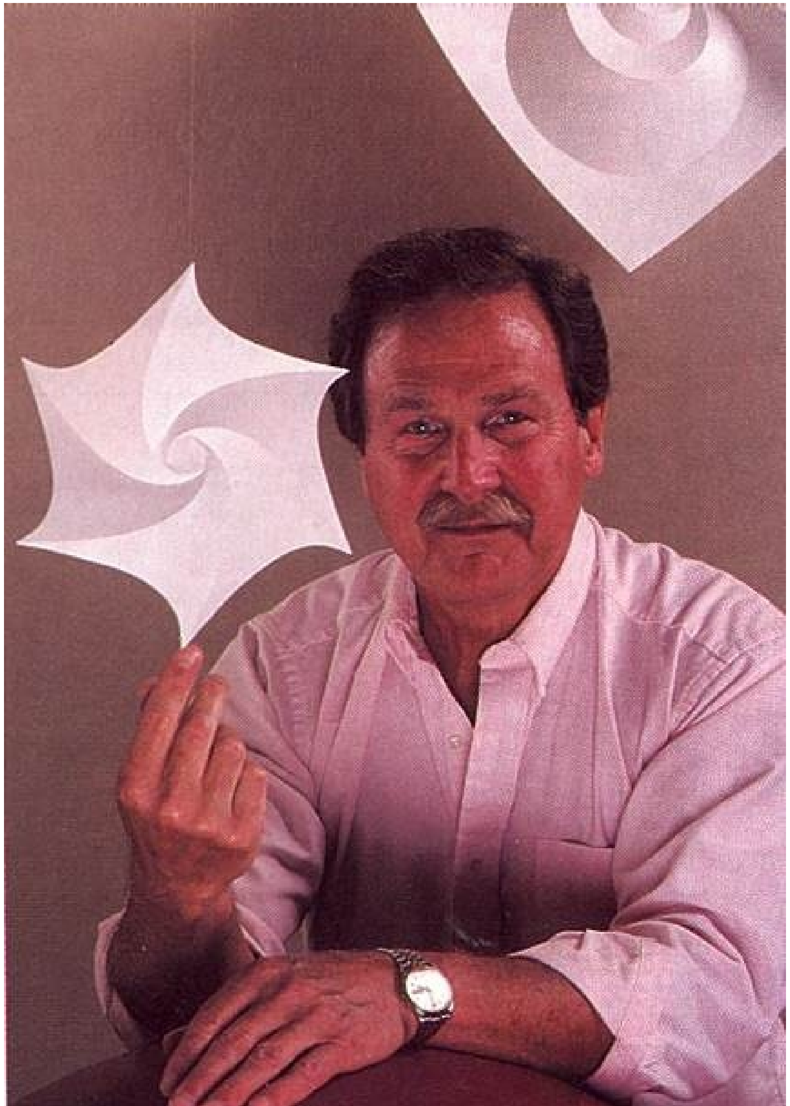
$$L(C) \le \frac{H(X)}{\log_b J} + 1 - p_{\min}$$

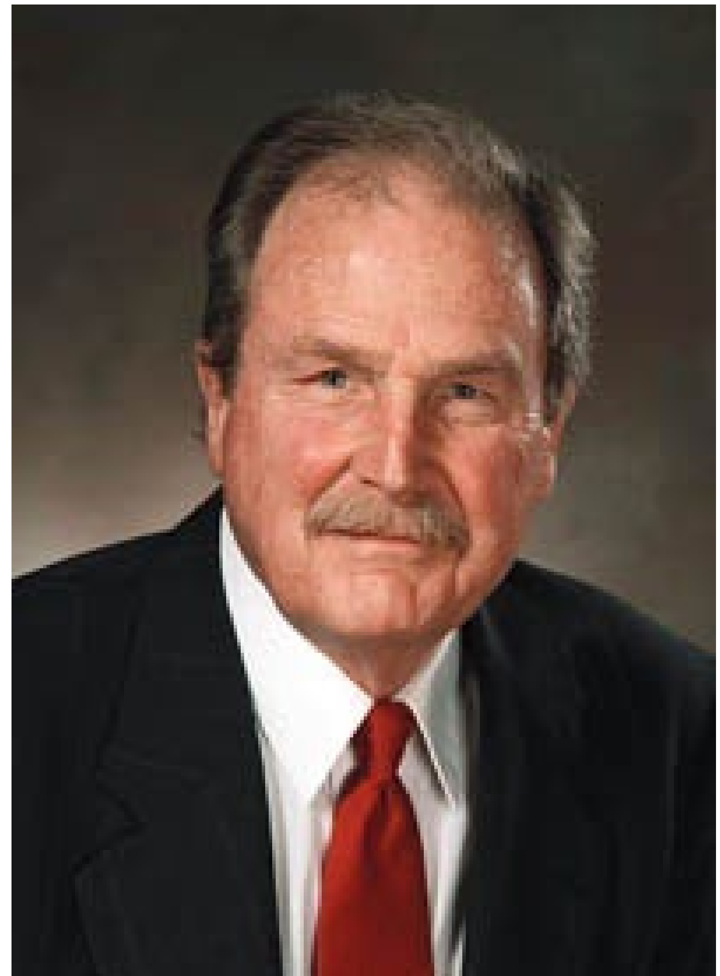
where $p_{\min}$ is the smallest nonzero symbol probability

if $b = J$

$$L(C) \le H(X) + 1 - p_{\min}$$

David A. Huffman (1925-1999)





- ``It was the most singular moment in my life. There was the absolute lightning of sudden realization.''
  – David Huffman

- ``Is that all there is to it!''
  – Robert Fano

The Binary Huffman Algorithm

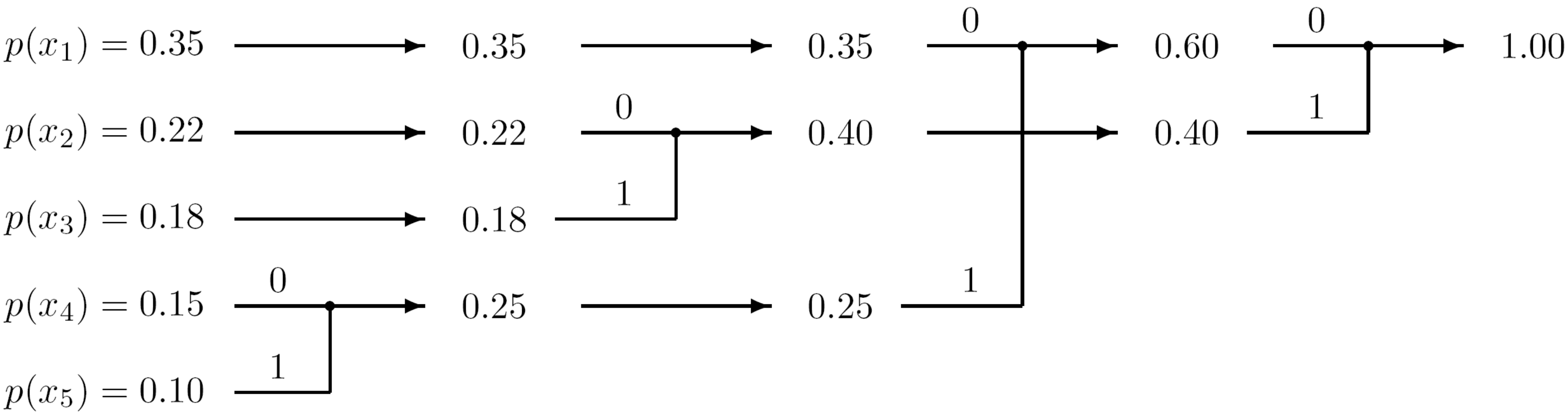
1. Arrange the K symbols of the source X in order of decreasing probability.
2. Assign a 1 to the last digit of the Kth codeword c_K and a 0 to the last digit of the (K-1)th codeword c_{K-1}. Note that this assignment is arbitrary.
3. Form a new source X´ with $x'_k = x_k$, $k = 1, \ldots, K-2$, and
   $x'_{K-1} = x_{K-1} \cup x_K$ $\quad$ $p(x'_{K-1}) = p(x_{K-1}) + p(x_K)$
4. Set $K = K-1$.
5. Repeat Steps 1 to 4 until all symbols have been combined.

To obtain the codewords, trace back to the original symbols.

Five Symbol Source

- p(x_1)=.35 p(x_2)=.22 p(x_3)=.18 p(x_4)=.15 p(x_5)=.10

- H(X) = 2.2 bits



Symbol x_k	Probability $p(x_k)$	Codeword $\mathbf{c}_k = (c_{k,1}, \cdots, c_{k,l_k})$	Length l_k
x_1	$p(x_1) = 0.35$	0 0	2
x_2	$p(x_2) = 0.22$	1 0	2
x_3	$p(x_3) = 0.18$	1 1	2
x_4	$p(x_4) = 0.15$	0 1 0	3
x_5	$p(x_5) = 0.10$	0 1 1	3

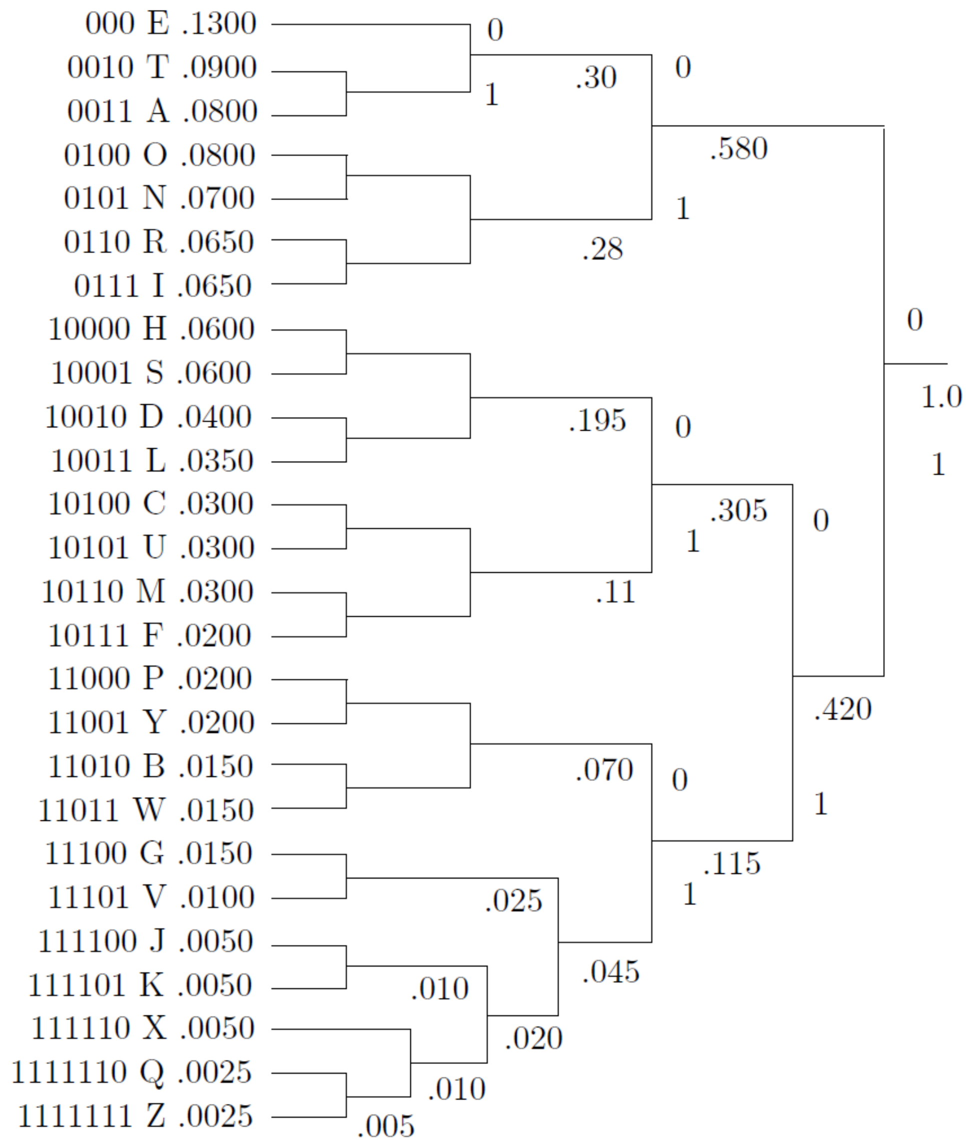
L(C) = 2.25 bits ζ = 97.8%

Shannon and Fano Codes

- $p(x_1)=.35$ $p(x_2)=.22$ $p(x_3)=.18$ $p(x_4)=.15$ $p(x_5)=.10$
- $H(X) = 2.2$ bits

Shannon Code	Fano Code
x_1 00	x_1 00
x_2 010	x_2 01
x_3 100	x_3 10
x_4 110	x_4 110
x_5 1110	x_5 111
$L(C) = 2.75$ bits	$L(C) = 2.25$ bits
$\zeta = 80.4\%$	$\zeta = 97.8\%$

Huffman Code for the English Alphabet



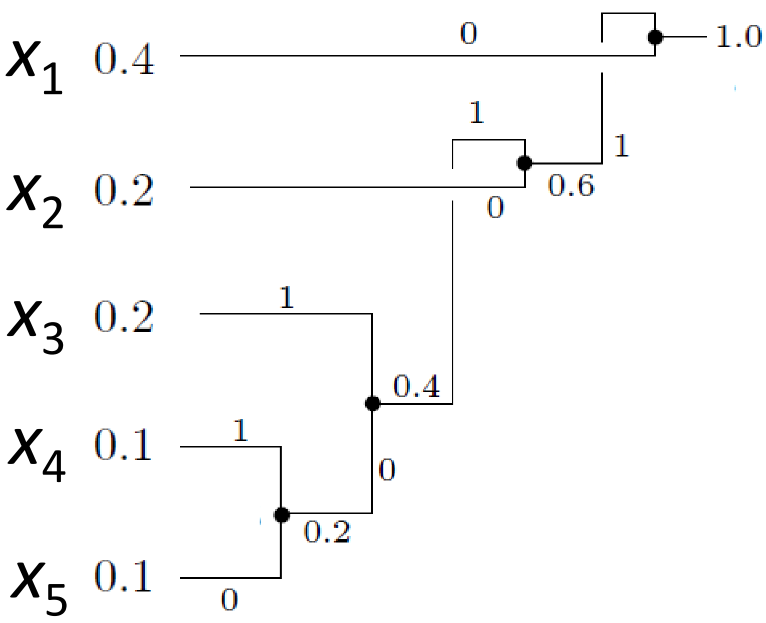
Six Symbol Source

- $p(x_1)=.4$ $p(x_2)=.3$ $p(x_3)=.1$ $p(x_4)=.1$ $p(x_5)=.06$ $p(x_6)=.04$
- $H(X) = 2.1435$ bits

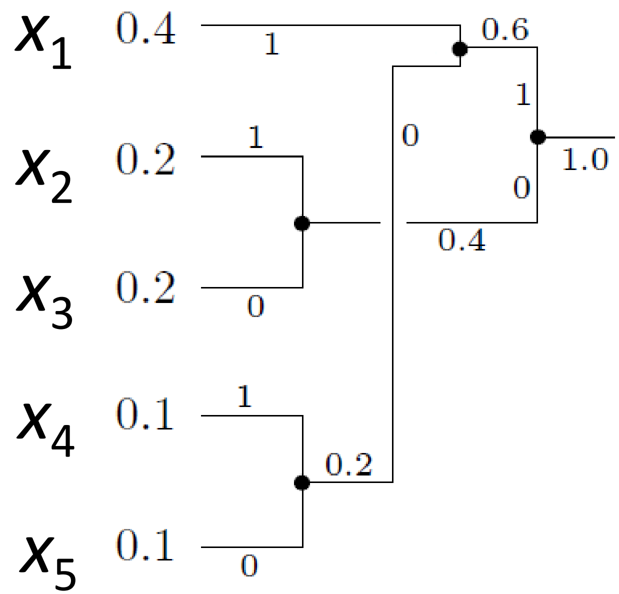
	First Code		Second Code
x_1	1	x_1	1
x_2	00	x_2	00
x_3	0100	x_3	010
x_4	0101	x_4	0110
x_5	0110	x_5	01110
x_6	0111	x_6	01111

Second Five Symbol Source

- p(x_1)=.4 p(x_2)=.2 p(x_3)=.2 p(x_4)=.1 p(x_5)=.1
- H(X) = 2.1219 bits



C_1



C_2

Second Five Symbol Source

	C_1	C_2
x_1	0	11
x_2	10	01
x_3	111	00
x_4	1101	101
x_5	1100	100

Which code is preferable?

Second Five Symbol Source

- p(x_1)=.4 p(x_2)=.2 p(x_3)=.2 p(x_4)=.1 p(x_5)=.1
- H(X) = 2.122 bits   L(C) = 2.2 bits

- variance of code C_1

  σ_1^2 = 0.4(1-2.2)^2+0.2(2-2.2)^2+0.2(3-2.2)^2+0.2(4-2.2)^2 = 1.36
- variance of code C_2

  σ_2^2 = 0.8(2-2.2)^2+0.2(3-2.2)^2 = 0.16

Midterm Test

- Friday, October 21, 2022
- During class time (11:30 – 12:20)
- Counts for 20% of the final mark
- Aids allowed
  - One page of notes on 8.5” × 11.5” paper (both sides)
  - Calculator
- Cellphones, tablets, laptops, or any other electronic devices are NOT ALLOWED

Nonbinary Codes

- The Huffman algorithm for nonbinary codes (J>2) follows the same procedure as for binary codes except that J symbols are combined at each stage.
- This requires that the number of symbols in the source X is $K'=J+c(J-1)$, $K' \geq K$

$$c = \left\lceil \frac{K-J}{J-1} \right\rceil$$

Nonbinary Example

- J=3 K=6
- p(x_1)=1/3 p(x_2)=1/6 p(x_3)=1/6 p(x_4)=1/9 p(x_5)=1/9 p(x_6)=1/9
- H(X) = 1.544 trits

Nonbinary Example

- J=3 K=6
- $c = \left\lceil \frac{K-J}{J-1} \right\rceil$=2 so $K'=J+c(J-1)=3+2(2)=7$
- Add an extra symbol x_7 with $p(x_7)=0$
- $p(x_1)=1/3$ $p(x_2)=1/6$ $p(x_3)=1/6$ $p(x_4)=1/9$ $p(x_5)=1/9$ $p(x_6)=1/9$ $p(x_7)=0$

Nonbinary Example with an Extra Symbol

x_1	1
x_2	00
x_3	01
x_4	02
x_5	20
x_6	21
x_7	22

L(C) = 1.667 trits

H(X) = 1.544 trits

ζ = 92.6%

Nonbinary Example with no Extra Symbol

x_1	1
x_2	01
x_3	02
x_4	000
x_5	001
x_6	002

L(C) = 2.0 trits

H(X) = 1.544 trits

ζ = 77.2%

Codes for Different Output Alphabets

- K=13

- $p(x_1)=1/4$ $p(x_2)=1/4$
  $p(x_3)=1/16$ $p(x_4)=1/16$ $p(x_5)=1/16$ $p(x_6)=1/16$
  $p(x_7)=1/16$ $p(x_8)=1/16$ $p(x_9)=1/16$
  $p(x_{10})=1/64$ $p(x_{11})=1/64$ $p(x_{12})=1/64$ $p(x_{13})=1/64$

- J=2 to 13

Codes for Different Output Alphabets

$p(x_i)$	x_i	J: 13	12	11	10	9	8	7	6	5	4	3	2
$\frac{1}{4}$	x_1	0	0	0	0	0	0	0	0	0	0	0	00
$\frac{1}{4}$	x_2	1	1	1	1	1	1	1	1	1	1	1	01
$\frac{1}{16}$	x_3	2	2	2	2	2	2	2	2	2	20	200	1000
$\frac{1}{16}$	x_4	3	3	3	3	3	3	3	3	30	21	201	1001
$\frac{1}{16}$	x_5	4	4	4	4	4	4	4	4	31	22	202	1010
$\frac{1}{16}$	x_6	5	5	5	5	5	5	5	50	32	23	210	1011
$\frac{1}{16}$	x_7	6	6	6	6	6	6	60	51	33	30	211	1100
$\frac{1}{16}$	x_8	7	7	7	7	7	70	61	52	34	31	212	1101
$\frac{1}{16}$	x_9	8	8	8	8	80	71	62	53	40	32	220	1110
$\frac{1}{64}$	x_{10}	9	9	9	90	81	72	63	54	41	330	221	111100
$\frac{1}{64}$	x_{11}	A	A	A0	91	82	73	64	550	42	331	2220	111101
$\frac{1}{64}$	x_{12}	B	B0	A1	92	83	74	65	551	43	332	2221	111110
$\frac{1}{64}$	x_{13}	C	B1	A2	93	84	75	66	552	44	333	2222	111111
Average length	$L(C)$	1	$\frac{33}{32}$	$\frac{67}{64}$	$\frac{17}{16}$	$\frac{9}{8}$	$\frac{19}{16}$	$\frac{5}{4}$	$\frac{87}{64}$	$\frac{23}{16}$	$\frac{25}{16}$	$\frac{131}{64}$	$\frac{25}{8}$

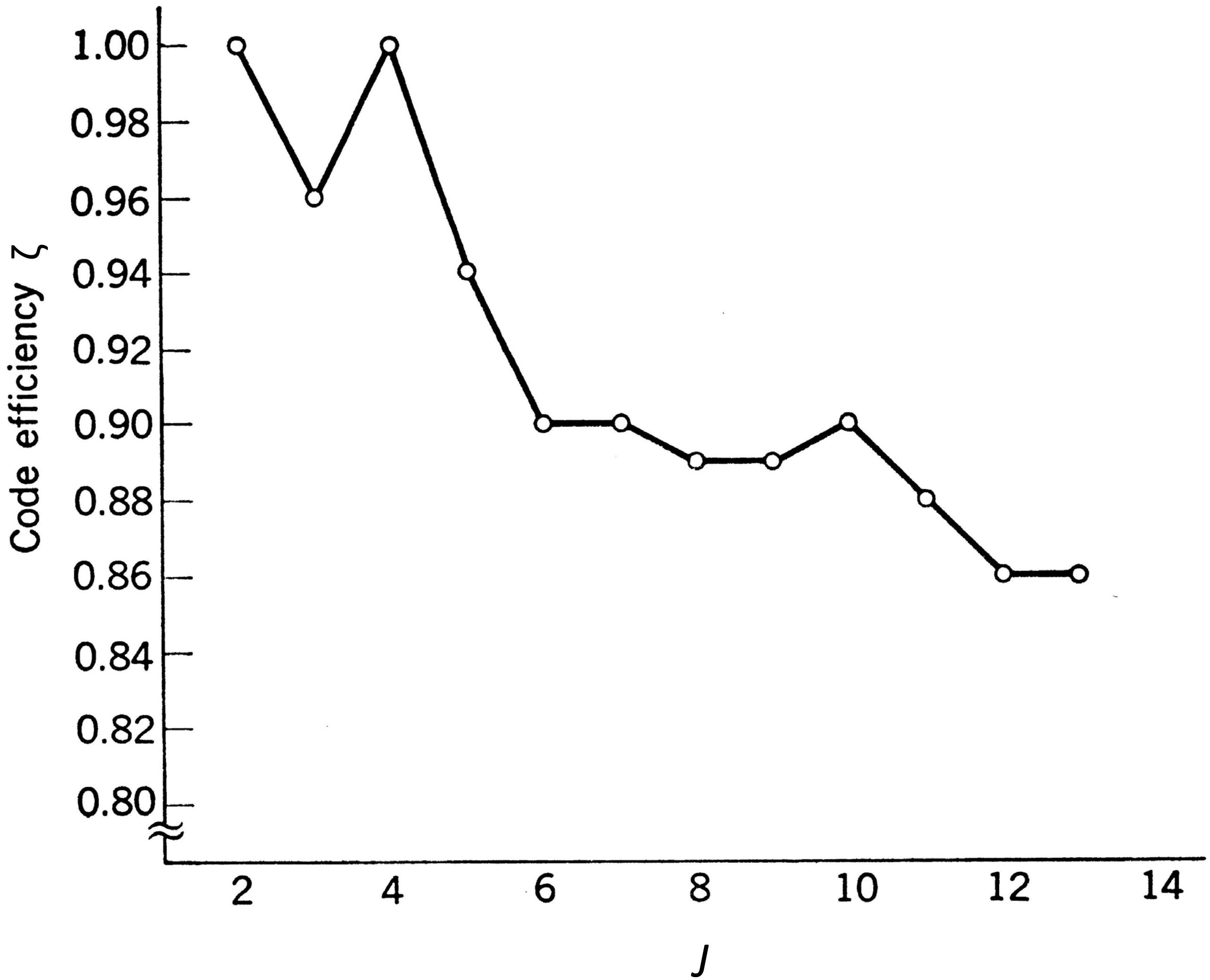
Codes for Different Output Alphabets

J	L(C)
2	3.125
3	2.047
4	1.563
5	1.438
6	1.359
7	1.250
8	1.188
9	1.125
10	1.063
11	1.047
12	1.031
13	1.000

Codes for Different Output Alphabets

J	L(C)	ζ
2	3.125	1.000
3	2.047	0.963
4	1.563	1.000
5	1.438	0.936
6	1.359	0.889
7	1.250	0.891
8	1.188	0.877
9	1.125	0.876
10	1.063	0.885
11	1.047	0.863
12	1.031	0.845
13	1.000	0.844

Code Efficiency



Binary and Quaternary Codes

x_1	00	0
x_2	01	1
x_3	1000	20
x_4	1001	21
x_5	1010	22
x_6	1011	23
x_7	1100	30
x_8	1101	31
x_9	1110	32
x_{10}	111100	330
x_{11}	111101	331
x_{12}	111110	332
x_{13}	111111	333

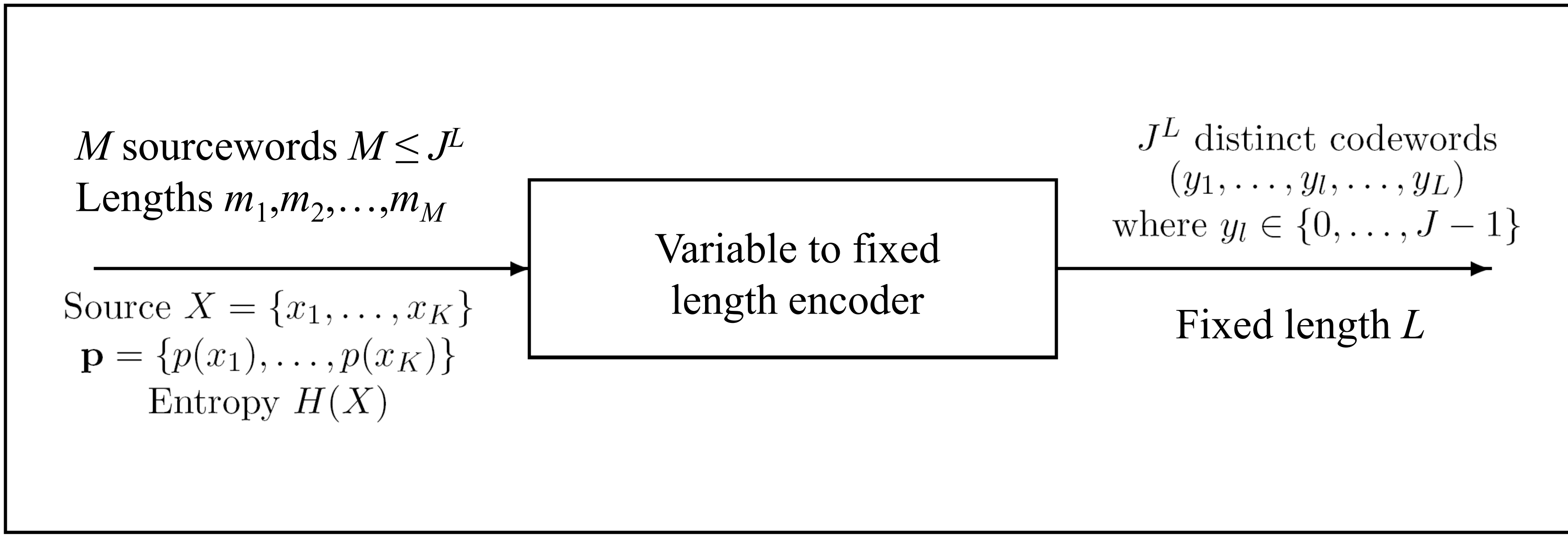
Huffman Codes

- Symbol probabilities must be known a priori
- The redundancy of the code

  $L(C)-H(X)$ (for $J=b$)

  is typically nonzero
- Error propagation can occur
- Codewords have variable length

Variable to Fixed Length Codes



Variable to Fixed Length Codes

- Two questions:
  1. What is the best mapping from sourcewords to codewords?
  2. How to ensure unique encodability?

Average Bit Rate

$$\text{ABR} = \frac{\text{average codeword length}}{\text{average sourceword length}}$$

$$= \frac{L}{\text{L(S)}}$$

$$\text{L(S)} = \sum_{i=1}^{M} \text{p}(s_i) m_i$$

M - number of sourcewords

s_i - sourceword i

m_i - length of sourceword i

p(s_i) - probability of sourceword i

Average Bit Rate

- For fixed to variable length codes

$$\text{ABR} = \frac{\text{average codeword length}}{\text{average sourceword length}}$$

$$= \frac{\text{L(C)}}{1} \text{ or } \frac{\text{L}_N\text{(C)}}{N}$$

- Design criterion: minimize L(C) or L_N(C)
  – minimize the ABR

Variable to Fixed Length Codes

- Design criterion: minimize the Average Bit Rate

$$\text{ABR} = \frac{L}{\text{L(S)}}$$

- ABR ≥ H(X) (L(C) ≥ H(X) for fixed to variable length codes)
- L(S) should be as large as possible so that the ABR is close to H(X)

Code Efficiency

- Fixed to variable length codes

$$\zeta = \frac{H(X)}{L(C)} \leq 1$$

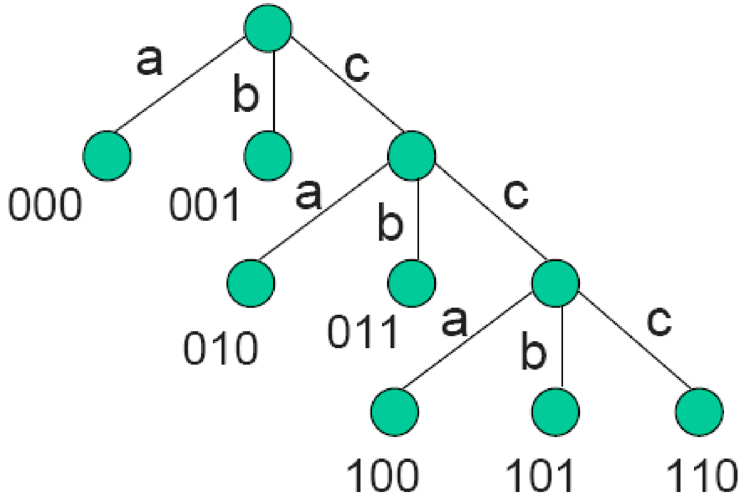
- Variable to fixed length codes

$$\zeta = \frac{H(X)}{ABR} \leq 1$$

Binary Tunstall Code K=3, L=3

Let x_1 = a, x_2 = b and x_3 = c

a	000
b	001
ca	010
cb	011
cca	100
ccb	101
ccc	110



Unused codeword is 111

Tunstall Codes

Tunstall codes must satisfy the Kraft inequality

$$\sum_{i=1}^{M} K^{-m_i} \leq 1$$

M - number of sourcewords
K - source alphabet size
m_i - length of sourceword i

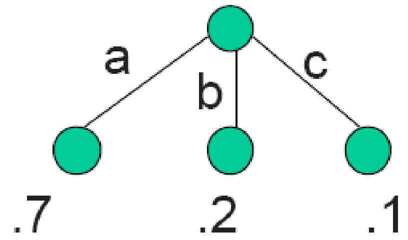
Binary Tunstall Code Construction

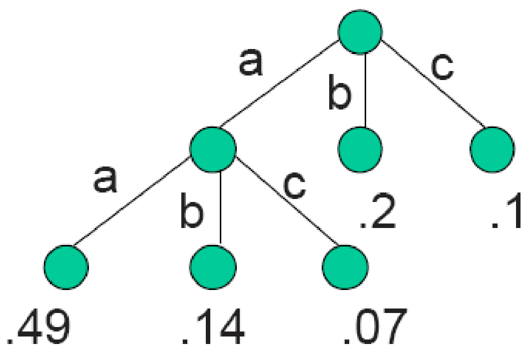
- Source X with *K* symbols
- Choose a codeword length *L* where $2^L > K$

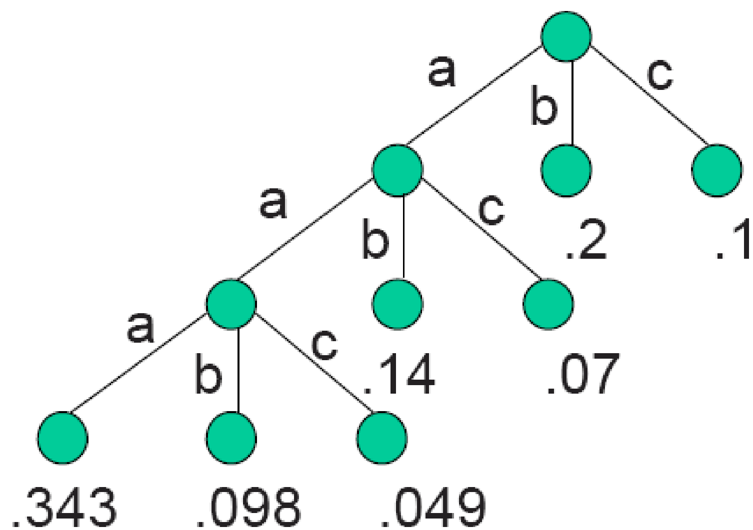
1. Form a tree with a root and *K* branches labelled with the symbols
2. If the number of leaves is greater than $2^L - (K-1)$, go to Step 4
3. Find the leaf with the highest probability and extend it to have *K* branches, go to Step 2
4. Assign codewords to the leaves

K=3, *L*=3

p(a) = .7, p(b) = .2, p(c) = .1







ABR = 3/[3 (.343 + .098 + .049) + 2 (.14 + .07) + .2 + .1]

= 1.37 bits per symbol

H(X) = 1.16 bits per symbol

ζ = H(X)/ABR = 84.7%

The Codewords

aaa	000
aab	001
aac	010
ab	011
ac	100
b	101
c	110

- What if a or aa is left at the end of the sequence of source symbols?
  - there are no corresponding codewords
- Solution: use the unused codeword 111
  - a   1110 or 111 000
  - aa   1111 or 111 001

Tunstall Codes for a Binary Source

- $L = 3$, $K = 2$, $J = 2$, $p(x_1) = 0.7$, $p(x_2) = 0.3$
- $J^L = 8$

Seven sourcewords	Eight sourcewords	Codewords
$x_1x_1x_1x_1x_1$	$x_1x_1x_1x_1x_1$	000
$x_1x_1x_1x_1x_2$	$x_1x_1x_1x_1x_2$	001
$x_1x_1x_1x_2$	$x_1x_1x_1x_2$	010
$x_1x_1x_2$	$x_1x_1x_2$	011
x_1x_2	$x_1x_2x_1$	100
x_2x_1	$x_1x_2x_2$	101
x_2x_2	x_2x_1	110
	x_2x_2	111

- The end of the sequence of source symbols can be

  x_1, x_2, x_1x_1 , $x_1x_1x_1$, or $x_1x_1x_1x_1$

- With M=7 sourcewords the codeword 111 is unused so they can be assigned as follows
  - x_1 111 000
  - x_2 111 001
  - x_1x_1 111 010
  - $x_1x_1x_1$ 111 011
  - $x_1x_1x_1x_1$ 111 100

Huffman Code for a Binary Source

- $N = 3$, $K = 2$, $p(x_1) = 0.7$, $p(x_2) = 0.3$
- Eight sourcewords
- A = $x_1x_1x_1$ p(A) = .343 00
- B = $x_1x_1x_2$ p(B) = .147 11
- C = $x_1x_2x_1$ p(C) = .147 010
- D = $x_2x_1x_1$ p(D) = .147 011
- E = $x_2x_2x_1$ p(E) = .063 1000
- F = $x_2x_1x_2$ p(F) = .063 1001
- G = $x_1x_2x_2$ p(G) = .063 1010
- H = $x_2x_2x_2$ p(H) = .027 1011

Code Comparison

- H(X) = .8813
- Tunstall Code *L*=3 (7 codewords)
  ABR = .9762 ζ = 90.3%
- Tunstall Code *L*=3 (8 codewords)
  ABR = .9138 ζ = 96.4%
- Huffman Code *N*=1 (2 codewords)
  L(C) = 1.0 ζ = 88.1%
- Huffman Code *N*=3 (8 codewords)
  $L_3(C)/3$ = .9087 ζ = 97.0%

Error Propagation

- Received Huffman codeword sequence

  00 11 00 11 00 11 …

  A B A B A B …

- Sequence with one bit error

  011 1001 1001 1 …

  D F F …

Error Propagation

- The corresponding Tunstall codeword sequence

  000 110 001 000 110 001 …

  $x_1x_1x_1x_1x_1\ x_2x_1\ x_1x_1x_1x_1x_2$ …

- Sequence with one bit error

  010 110 001 000 110 001 …

  $x_1x_1x_1x_2\ x_2x_1\ x_1x_1x_1x_1x_2$ …