Source Coding Theorem

- Source encoding

⇨ Efficient representation of data → compaction

⇨ Be uniquely decodable

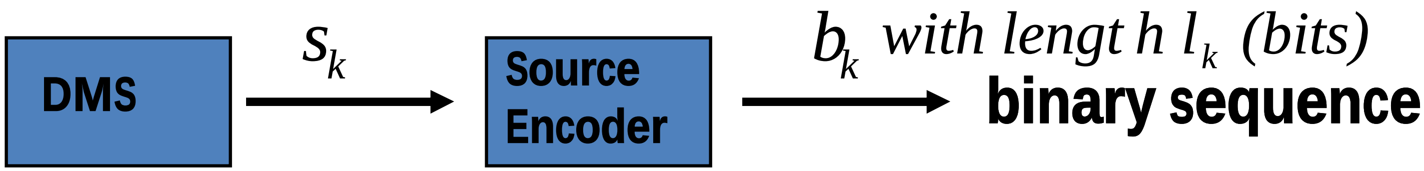
⇨ Need of statistics of the source

(There is an algorithm called “Lempel-Ziv” for unknown statistics of the source)

→ (Another frequent method is Run-Length code)

Source Coding Theorem (cont')

- Variable length code ⟷ Fixed length code



⇨ The average code-Length, $\bar{L}$, is

$$\bar{L} = \sum_{k=0}^{K-1} p_k l_k$$

⇨ The coding efficiency, $\eta = \frac{L_{min}}{\bar{L}}$

where L_{min} is the minimum possible value of $\bar{L}$

Shannon's first theorem : Source-coding theorem

- Given a dms of entropy H(*S*), the average code-word length $\overline{L}$ for any source coding is

$$\overline{L} \geq H(S)$$

i.e.) $L_{\min} = H(S) \quad \& \quad \eta = \frac{H(S)}{\overline{L}}$

Practical Source Coding

- Prefix coding

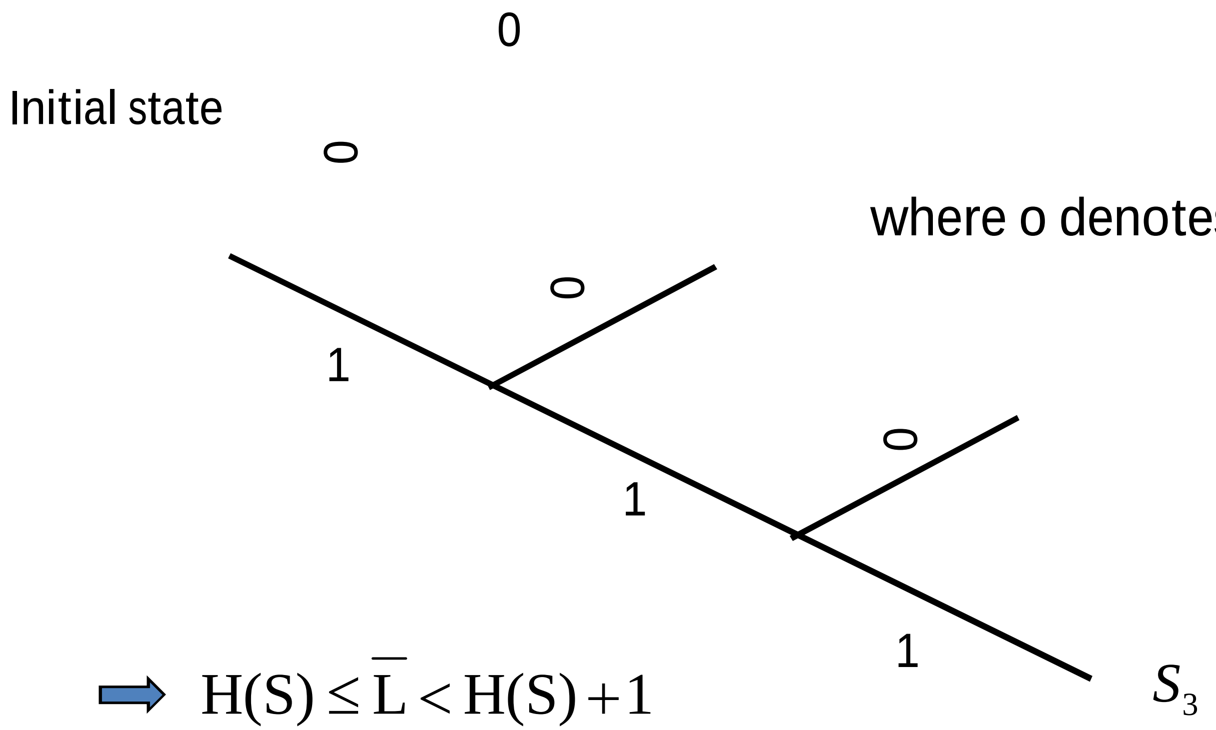
⇨ Def. : A code in which no code-word is the prefix of any other code-word

Ex)

Symbols	P_k	code1( )	code2( )	Code3( )
S_0	0.5	0	0	0
S_1	0.25	1	1 0	0 1
S_2	0.125	0 0	1 1 0	0 1 1
S_3	0.125	1 1	1 1 1	0 1 1 1

Practical Source Coding (cont’)

- Decoding



where o denotes terminal state

- $\mathrm{H(S)} \le \overline{\mathrm{L}} < \mathrm{H(S)} + 1$

Equality holds under one condition that $P_k = 2^{-l_k}$

Huffman Coding

- Property
  - ➡ a prefix code
  - ➡ average word length $\overline{L}$
    - ⟶ to fundamental limit , H(S)
  - ➡ optimum
- Algorithm shown by ex.

Symbol	P_k(stage1)	Stage 2	Stage 3	stage4
S_0	0.4	0.4	0.4	0.6
S_1	0.2	0.2	0.4	0.4
S_2	0.2	0.2	0.2	
S_3	0.1	0.2		
S_4	0.1			

Huffman Coding (cont')

- The result is

Symbol	P_k	Code - word
S_0	0.4	0 0
S_1	0.2	1 0
S_2	0.2	1 1
S_3	0.1	0 1 0
S_4	0.1	0 1 1

⇨ Then, $\bar{L} = 2.2$

while, H(S) = 2.12193

- Huffman encoding is not unique.

1) 0, 1 or 1, 0 → trivial

Practical Source Coding (cont')

Set the combined symbol with equal prob. ,

a) as high as possible

or

b) as low as possible

⟶ **get the same average code-word length, but with different variance**