Coding Theory

Chapter 10: Coding Theory – I

Chapter 11: Coding Theory – II

The *Art* of Doing SCIENCE and Engineering

Richard W. Hamming

Doğan Altunbay

Burak Şenol

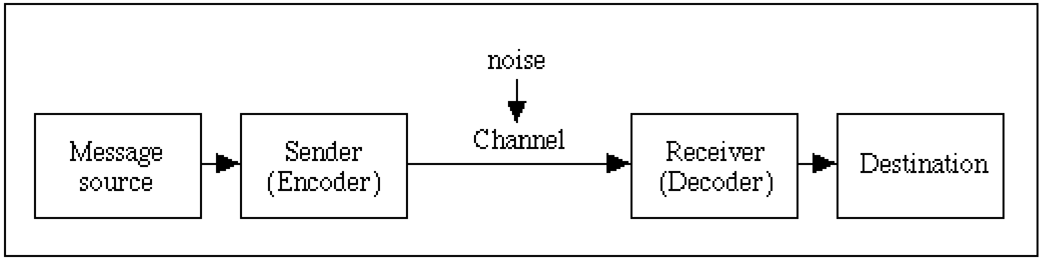
Fethi Burak Sazoğlu

Yusuf Aytaş

Content

- This chapter focuses on how the information is represented
- Discuss the transmission of information, which is encoding and decoding symbols of a message
- How to define encodings that has “good” code length
- Affect of noise in transmission and correcting errors

Transmission

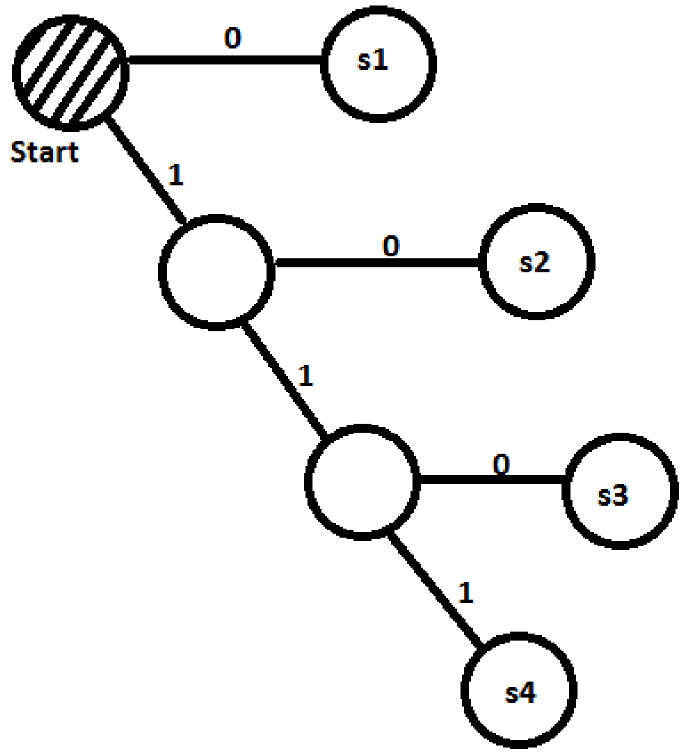


- **Source encoding**: Transforming the message symbols to a standard representation.
- **Channel encoding**: Standard representation is transformed to adapt the transmitting channel.
- The channel is supposed to have **random noise** added.
- **Decoding**
  - Channel to standard
  - Standard to Source

Coding

- A symbol of the alphabet is represented with a set of codes, i.e. binary codes
  - Ex: s1 = 0, s2 = 00, s3 = 01, s4 = 11
- A code is uniquely decodable if it can be split into a single set of symbols.
  - Ex: 0011 is not uniquely decodable.
    - Possible decodings: s1s1s4, s2s4
  - Ex: given s1 = 0, s2 = 10, s3 = 110, s4 = 111
    - 11011100 is uniquely decodable
      - Decoding: s3s4s1s1

Decoding Tree



Goodness of Code

- The most obvious measure of goodness is the average length of codes.
  - *We can compute the average code length using the below formula*
    - $L = \sum_{i=1}^{q} p_i l_i$
  - The smaller values of average code length is desired for a good encoding

Kraft Inequality

- Gives the limit on the lengths of codes.

$$K = \sum_{i=1}^{q} \frac{1}{2^{l_i}} \leq 1$$

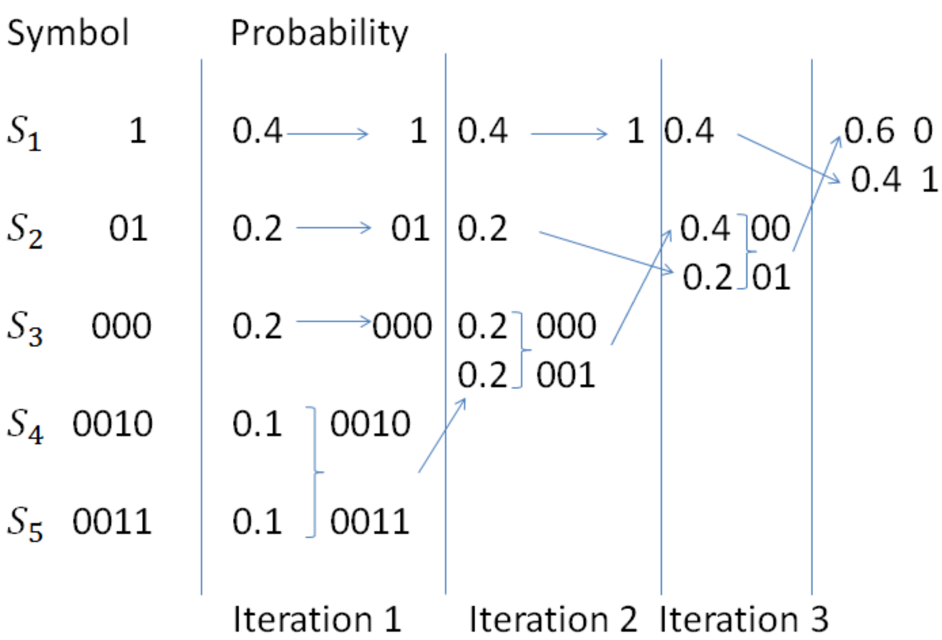
- Inequality says that if there are too many short symbols the sum will be too large.

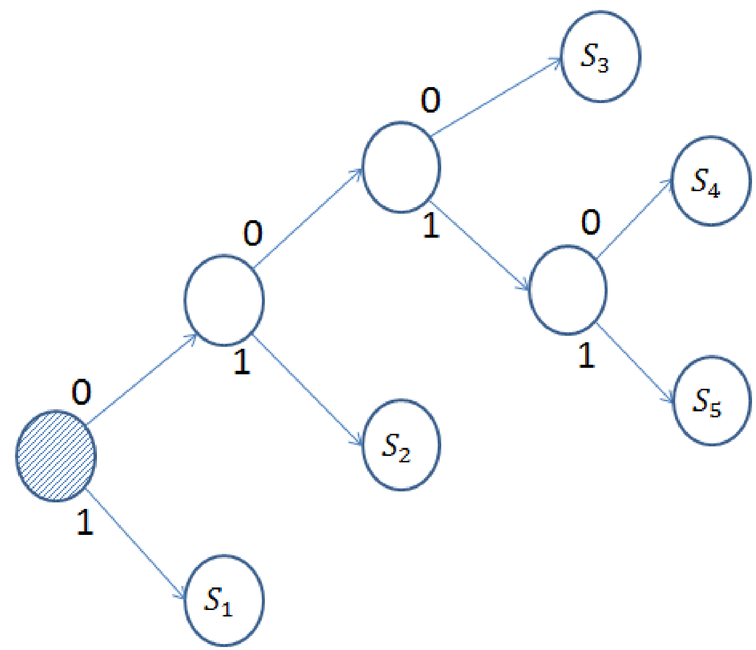
How to Obtain L

- L : average code length
- s_i: input symbols
- p_i:probability of occurring for s_i
- l_i: length of s_i
- Following inequalities must hold to attain minL (Huffman)
  - $p_1 \geq p_2 \geq \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \geq p_q$
  - $l_1 \leq l_2 \leq \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \leq l_q$

Example Huffman Code

- p(s_1) = 0.4
- p(s_2) = 0.2
- p(s_3) = 0.2
- p(s_4) = 0.1
- p(s_5) = 0.1
- Merge 2 least frequent symbols by dropping the last bit into 1 symbol with probability of their sum, until 2 symbols with length 1 is obtained


Symbol
Probability
S_1 1
S_2 01
S_3 000
S_4 0010
S_5 0011
0.4 1
0.2 01
0.2 000
0.1 0010
0.1 0011
0.4 1
0.2 01
0.2 000
0.2 001
0.4
0.4 00
0.2 01
0.6 0
0.4 1
Iteration 1
Iteration 2
Iteration 3


S_1
S_2
S_3
S_4
S_5
0
1
0
1
0
1
0
1

When to use Huffman coding?

- Different probabilities of the symbols are better than equal probabilities.
- Equal probability
  - Huffman process generates symbols of equal size.
  - L is equal to length of a symbol s.
- Different probability (Comma code)
  - 0, 10, 110, ………,11….110, 11….111.
  - In this case the value of L is smaller.

Noise Model (Error Detection)

- Single error is found by parity check.
- If the channel has white noise:
  - Each position in the message has the same probability of error.
  - These errors are uncorrelated. (Independent)
- Then,
  - No error = $(1-p)^n$
  - 1 error = n*p$*\ (1-p)^{n-1}$
  - 2 error = (n*(n-1)/2)*$p^2 * (1-p)^{n-2}$

Error Detection in Alphanumeric Case

- Code a sequence of symbols from an alphabet of 26 letter, 10 digits and space.
- Weighted Code:
- Assign numbers 0, 1, 2,…,36 to 0, 1,…, 9, A, B, …, Z, space.
- Compute $(\sum_{k=1}^{n} k * s_k)\ modulo\ 37$
- s_k: value for k^{th} symbol.

Weighted Code

- Used in inventory part names
- The error is caught at transmission time.
- Effective in human errors when contrasted to white noise.
- Future will increasingly concerned with information in the form of symbols.

Conclusion

- We had a brief overview on the Shannon's Theory, Kraft's Inequality and Huffman Encoding.
- We got a sense of what to consider in designing systems to transmit information through time or through space.