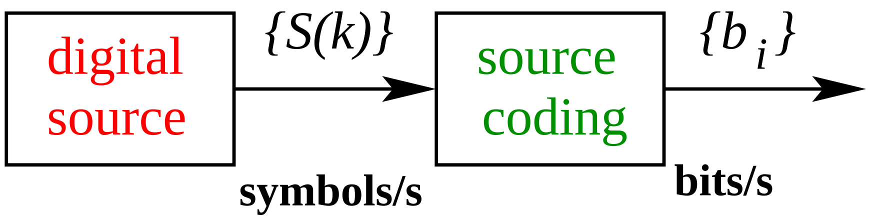
Revision of Lecture 3



- Source is defined by
  1. Symbol set: $\mathcal{S} = \{m_i, 1 \le i \le q\}$
  2. Probability of occurring of m_i: p_i, $1 \le i \le q$
  3. Symbol rate: R_s [symbols/s]
  4. Interdependency of $\{S(k)\}$ (memory or memoryless source)

- We have completed discussion on digital sources using information theory
  - Entropy $\surd$
  - Information rate $\surd$
  - Efficient coding: memoryless sources $\surd$; memory sources how ?

- But we know how not to code memory sources: code a memory source $\{S(k)\}$ directly by entropy coding is a bad idea
  - in fact, information theory we just learnt tell us how to code memory source efficiently

Source Coding Visit

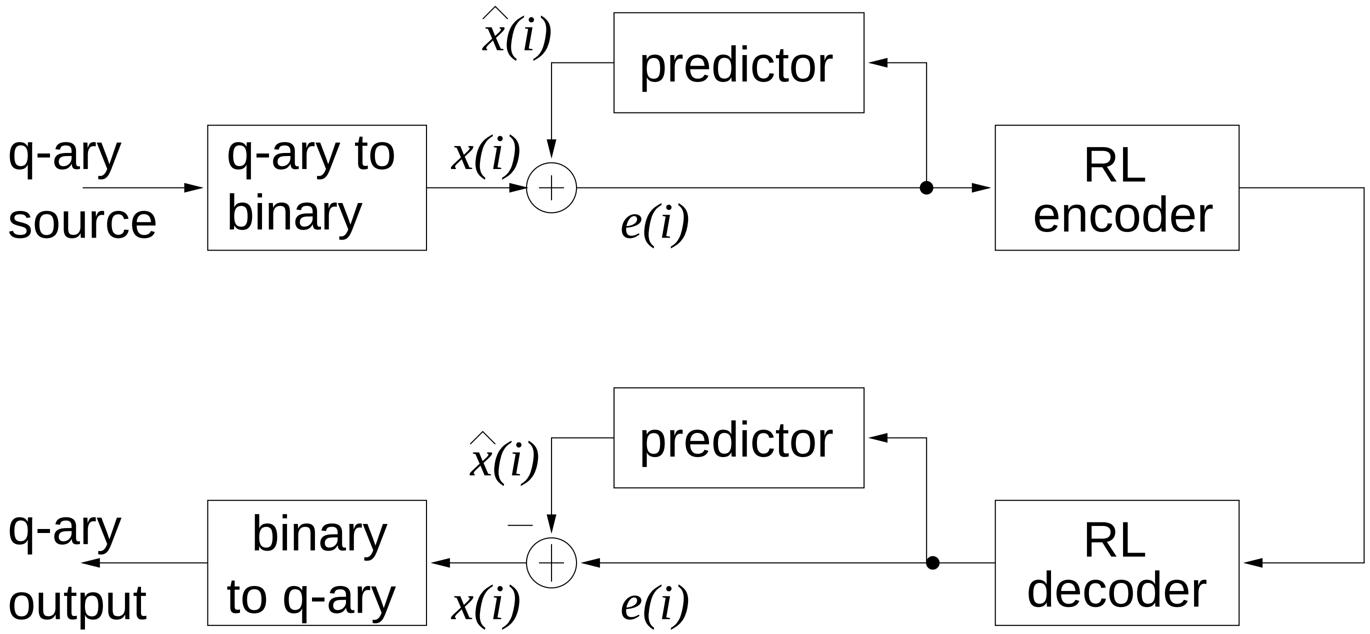
- Transmit at certain rate R_b requires certain amount of resource – bandwidth and power
  - The larger the rate R_b, the larger the required resource
- Source coding aims to get R_b as small as possible, ideally close to the source information rate
  - Information rate, a fundamental physical quantity of the source, tells you how many bits/s of information the source really needs to send out
  - If R_b is close to information rate, source coding is most efficient
- Memoryless source $\{S(k)\}$, entropy coding on $\{S(k)\}$ is most efficient, as data rate R_b is as close to source information rate as possible
- For memory source $\{S(k)\}$, information theory also tells us how to code $\{S(k)\}$ most efficiently

Remove Redundancy

- Key to get close to information rate $H \cdot R_s$ is to remove **redundancy**
  - Part of $S(k)$ can be predicted by $\{S(k-i)\}$, $i \geq 1 \rightarrow$ When coding $S(k)$ this temporal redundancy can be **predicted** from $\{S(k-i)\}$, $i \geq 1$
  - By removing the predictable part, resulting residual sequence $\{\varepsilon(k)\}$ is near independent or **uncorrelated**, can thus be coded by an entropy coding
- Speech samples are highly correlated, i.e. containing lots of redundancy
  - Predictive speech coding builds a predictive model like $s(k) = \sum_{j=1}^{N} a_j s(k-j) + \varepsilon(k) \rightarrow \varepsilon(k)$ contains new information unpredictable from $s(k-j)$, $1 \leq j \leq N$
  - $\{\varepsilon(k)\}$ can then be coded with far smaller rate, and you send $\{\varepsilon(k)\}$ together with the model parameters a_j, $1 \leq j \leq N$, not $\{s(k)\}$
- Video, containing a sequence of frames or pictures sent at certain frame rate, has
  - Inter-frame (temporal) redundancy and intra-frame (spatial) redundancy
  - Video coding involves remove these large temporal and spatial correlations

Predictive Scheme with Run-Length Coding (RLC)

- Following predictive scheme exploits partial predictability of memory source:
  - Convert q-ary source to binary sequence $\{x(i)\}$ by BCD
  - Build a predictor $\{\widehat{x}(i)\}$ for it
  - If prediction is successful, the "residual" binary sequence $\{e(i)\}$ mostly contains zeros, and this property is exploited in RLC



- This scheme widely used in video coding: by removing inter-frame and intra-frame correlations as much as possible, resulting binary sequence contains most zeros
  - RLC then **compresses** it into another much **shorter** binary sequence for transmission

Run Length Coding Table

- Code words with fixed length of n bits are formed from a bit stream (encoder input pattern) of upto $l \le N-1 = 2^n - 2$ successive zeros followed by a one or zero:

length of 0-run l	encoder input pattern (length $= \min\{N, l+1\}$)	encoder output codeword (fixed n bits)
0	1	$00\cdots000$
1	01	$00\cdots001$
2	001	$00\cdots010$
3	0001	$00\cdots011$
$\vdots$	$\vdots$	$\vdots$
$N-2$	$0\cdots01$	$11\cdots101$
$N-1$	$00\cdots01$	$11\cdots110$
$N=2^n-1$	$00\cdots00$	$11\cdots111$

- Assumption is input bit stream contains mostly "0"s, i.e. $p = P(\text{"0"})$ is very high

- Thus encoder on average reduces the word length

RLC Efficiency

- Code word length after run length coding: n bits;
- Average code word length d before coding with $N = 2^n - 1$:

$$d = \sum_{l=0}^{N-1} (l+1) \cdot p^l \cdot (1-p) + N \cdot p^N = 1 + p + p^2 + \cdots p^{N-1} = \frac{1-p^N}{1-p}$$

where p is the probability of a bit is '0'

- Therefore compression ratio $C = d/n$
- A numerical example: $p = 0.95$, $n = 5$ $(N = 31)$

$$C = \frac{d}{n} = \frac{1-p^N}{n(1-p)} \approx \frac{15.92}{5} \approx 3.18$$

RLC Re-exam Again

- RLC is widely used in various applications, so let us exam RLC more closely

  Input patterns have variable lengths, $2^n - 1$ bits to just 1 bit, depending on length of "**0**" runs before "**1**"; while output codewords have fixed length of n bits

Input pattern

00...0000	00...0001	00...001	...	01	1
2^n−1+0 bits	2^n−2+1 bits	2^n−3+1 bits		1+1 bits	0+1 bits

$\Downarrow$ RLC

Output codeword

11...111	11...110	11...101	...	00...001	00...000
n bits	n bits	n bits		n bits	n bits

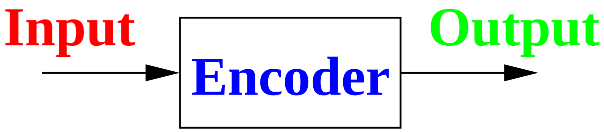
- Shannon-Fano and Huffman: inputs have fixed length while outputs variable lengths

  RLC appears very different from Shannon-Fano and Huffman or is it?

- RLC, Shannon-Fano and Huffman encodings are **lossless** or **entropy** encodings

Lossless Encodings Comparison

- Lossless or entropy encodings



Shannon−Fano
Huffman
RLC (for binary data most 0s)

- Same principle:

rare input pattern/message/symbol coded with large output codeword
large probability coded with small codeword

- Shannon-Fano and Huffman: input fixed length $\longrightarrow$ output variable length

  RLC: input variable length $\longrightarrow$ output fixed length

- It is the ratio

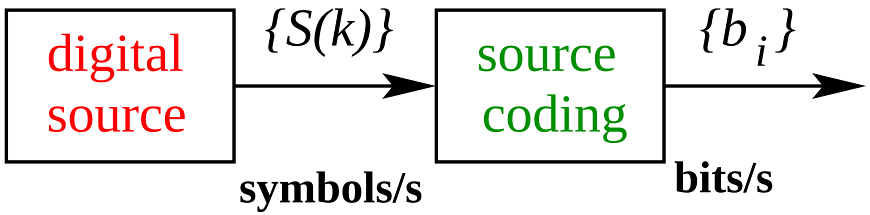
$$\textbf{ratio} = \frac{\textbf{output length}}{\textbf{input length}}$$

small **probability** $\longrightarrow$ large **ratio** large **probability** $\longrightarrow$ small **ratio**

Lecture 4 Summary

- According to information theory, key to efficient source coding for digital source with memory is first to remove redundancy
  - Efficient coding means coding rate R_b close to information rate $H \cdot R_s$
  - Predictable part of current $S(k)$ can be constructed and removed from $S(k)$
  - Resulting "residual" sequence $\varepsilon(k) = S(k) - \widehat{S}(k)$ is near white, and entropy coding can be applied to it
- Run-length encoding: a lossless or entropy encoding method
  - For binary digital source, where binary sequence contains most zeros, i.e. Prob$(\mathbf{0})$ is close to 1, and Prob$(\mathbf{1})$ is very small
  - RLC "compresses" such a binary sequence into a much shorter binary sequence
  - Compression ratio of RLC
  - Comparison with Shannon-Fano and Huffman entropy encoding methods

Digital Source Summary



- Digital source is defined by
  1. Symbol set: $\mathcal{S} = \{m_i, 1 \leq i \leq q\}$
  2. Probability of occurring of m_i: p_i, $1 \leq i \leq q$
  3. Symbol rate: R_s [symbols/s]
  4. Interdependency of $\{S(k)\}$
- Information content of alphabet m_i: $I(m_i) = -\log_2(p_i)$ [bits]
- Entropy: quantifies average information conveyed per symbol
  - Memoryless sources: $H = -\sum_{i=1}^{q} p_i \cdot \log_2(p_i)$ [bits/symbol]
  - 1st-order memory (1st-order Markov) sources with transition probabilities p_{ij}

$$H = \sum_{i=1}^{q} p_i H_i = -\sum_{i=1}^{q} p_i \sum_{j=1}^{q} p_{ij} \cdot \log_2(p_{ij}) \text{ [bits/symbol]}$$

- Information rate: tells you how many bits/s information the source really needs to send out
  - Information rate $R = R_s \cdot H$ [bits/s]
- Efficient source coding: get rate R_b as close as possible to information rate R
  - Memoryless source: apply entropy coding, such as Shannon-Fano and Huffman, and RLC if source is binary with most zeros
  - Generic sources with memory: remove redundancy first, then apply entropy coding to "residauls"