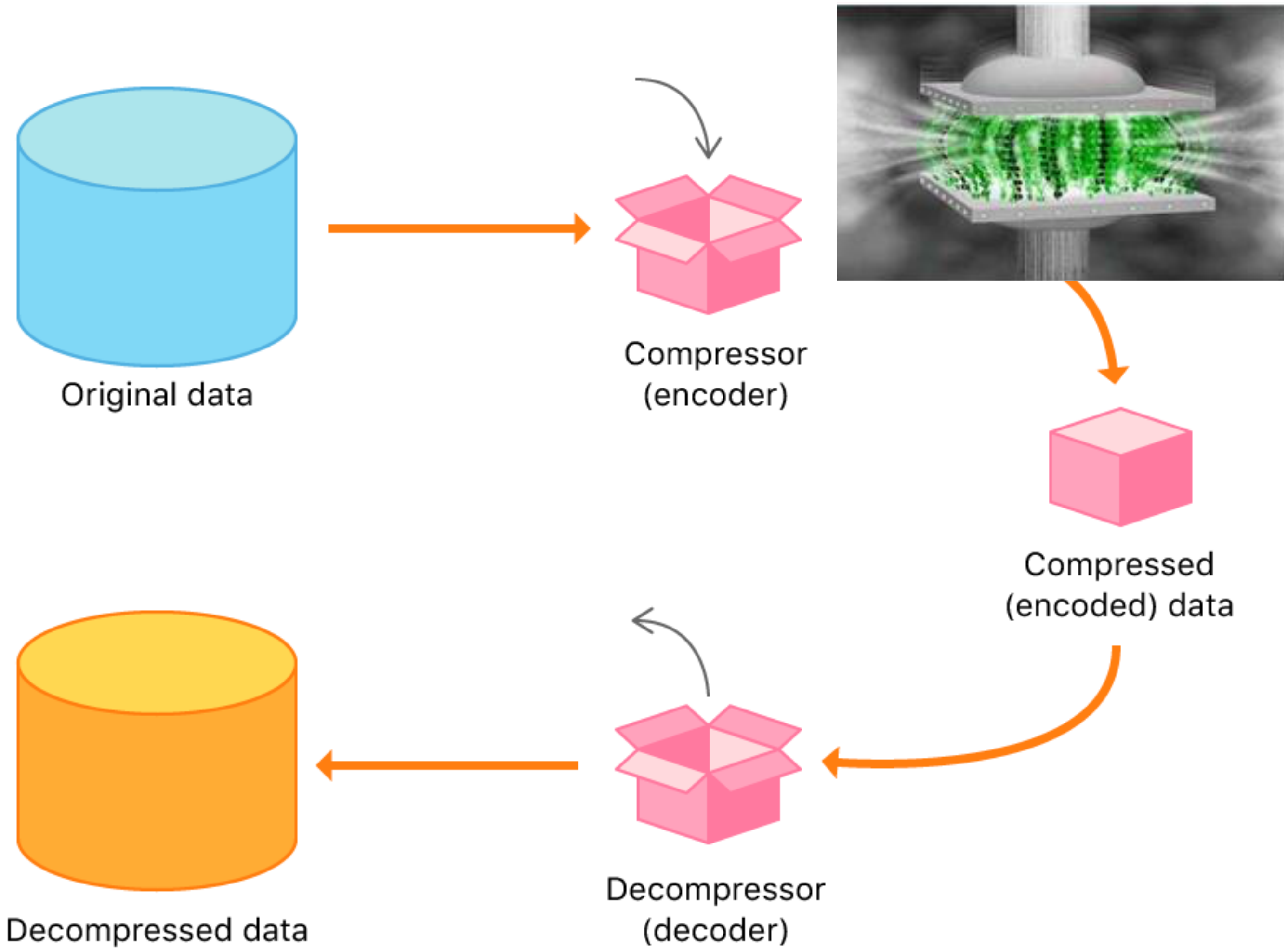
# Source Coding

# What is Data Compression?



# What is data compression?

- Data compression is the **representation** of an **information source** (e.g., data file, speech signal, video signal) as accurately as possible using the **fewest number of bits**

- Compressed data can only be understood if the decoding method is known by the receiver

# Source Coding

**Lossless**
(text, programs)


- **Lossless compression:** the compressed image/video can be decompressed to be identical to the original

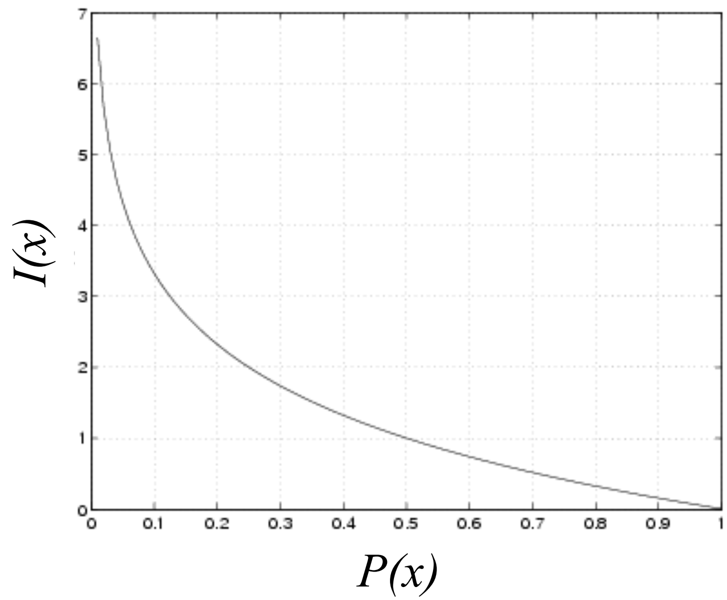
**Lossy**
(images, videos)


- With **lossy compression**, the image is degraded, mathematically not identical to original. Data compression introduces a **distortion** of the source.

# Source Information

- Let $X$ be a discrete random variable taking on values $x_1$, $x_2$, ... $x_j$ from a finite alphabet $A$

- Probabilities of occurrence are $p(x_1)$, $p(x_2)$, ...$p(x_j)$




- The **information** associated with symbol $s_i$ is defined to be

$$I(x_i) = log_2 \frac{1}{p(x_i)}$$

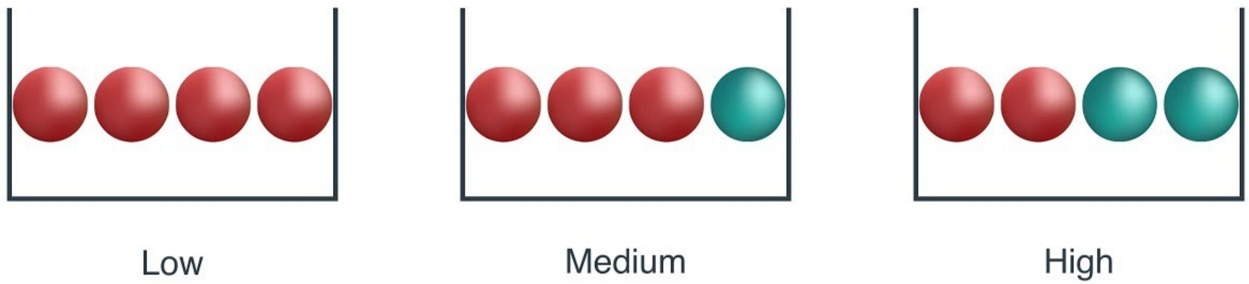
# Entropy

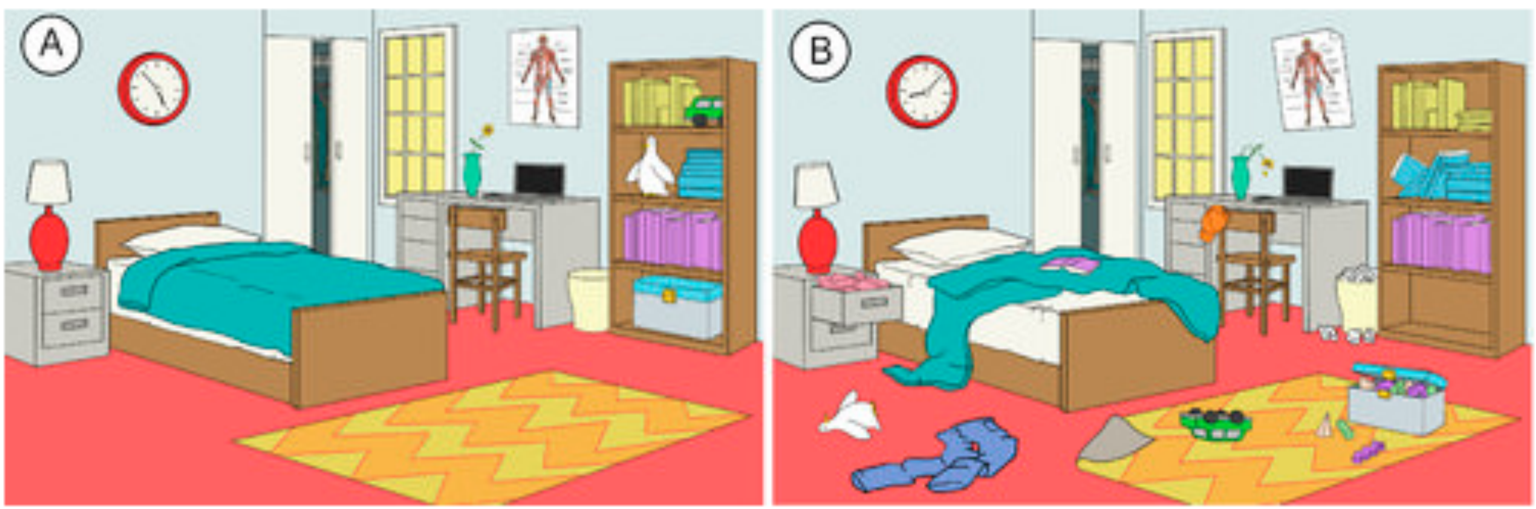
- (First-order) Entropy is a measure of the uncertainty of a random variable.
- **Entropy** is the average information per symbol

$$H(X) = \sum_{i=1}^{J} p(x_i) log_2 \frac{1}{p(x_i)}$$

- Entropy is expressed in bits
- Entropy of a fair coin toss is 1 bit
- Average length of the shortest description of the random variable

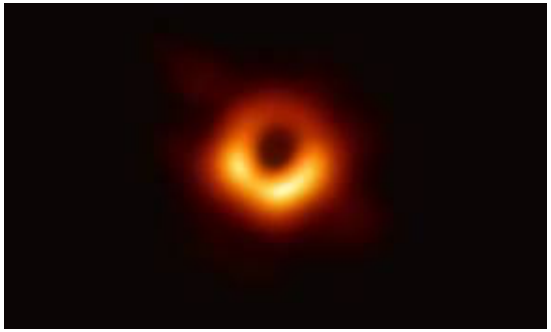
# Entropy



# Source Information

- Sunrise Time – not that much informative



- Black Hole Pic – highly informative

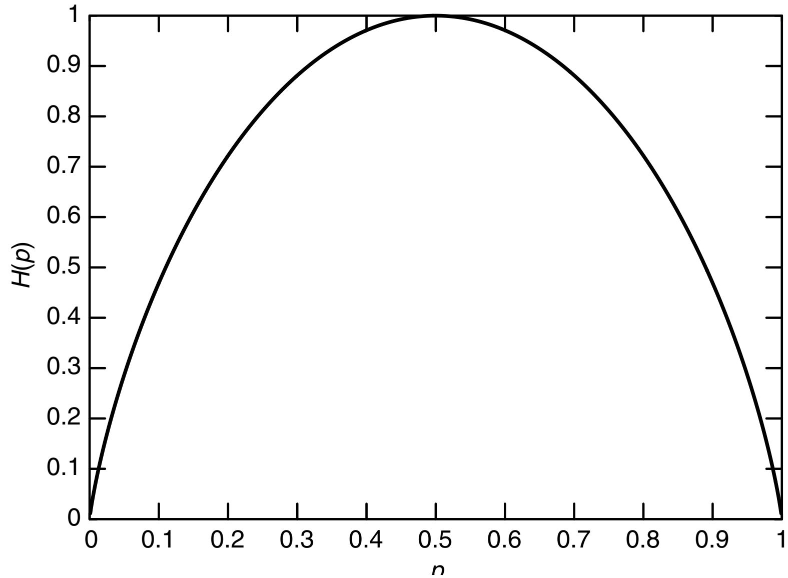
# Example 1

Let $$X = \begin{cases} 1 & \text{with probability } p, \\ 0 & \text{with probability } 1-p \end{cases}$$

Then $$H(X) = -p\, log_2 p \;-(1-p) log_2 (1-p) \stackrel{\text{def}}{=} H(p)$$

Entropy of a fair coin toss is 1 bit.

Or in other words p=0.5 → H(p)=1



# Example 2

Let

$$X = \begin{cases} a & \text{with probability} \frac{1}{2}, \\ b & \text{with probability} \frac{1}{4}, \\ c & \text{with probability} \frac{1}{8}, \\ d & \text{with probability} \frac{1}{8}. \end{cases}$$

Then, the entropy of $X$ is

$$H(X) = -\frac{1}{2}\, log_2 \frac{1}{2} - \frac{1}{4}\, log_2 \frac{1}{4} - \frac{1}{8}\, log_2 \frac{1}{8} - \frac{1}{8}\, log_2 \frac{1}{8} = \frac{7}{4} \text{ bits}$$

What is the minimum number of questions you need to determine $X$?

"Is X = a?"

"Is X = b

"Is X = c?"

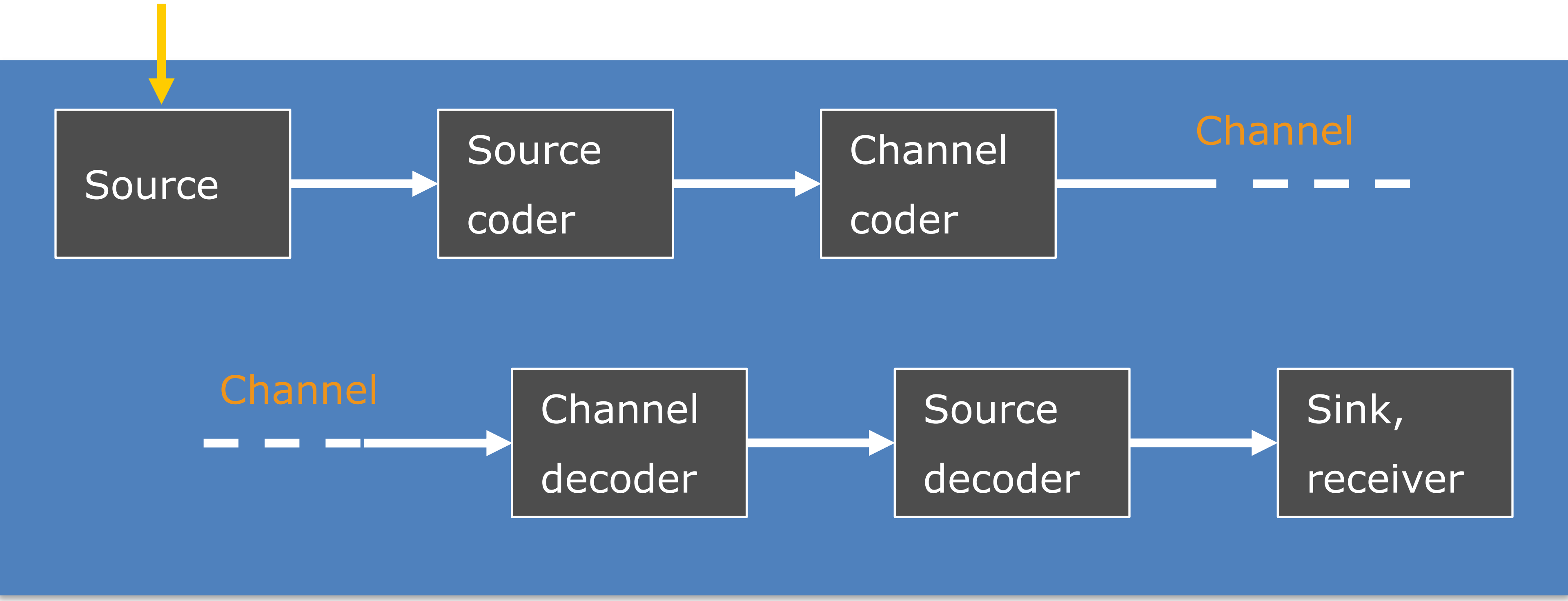
The resulting expected number of binary questions required is $1.75$.

**The minimum expected number of binary questions required to determine $X$ lies between $H(X)$ and $H(X) + 1$.**

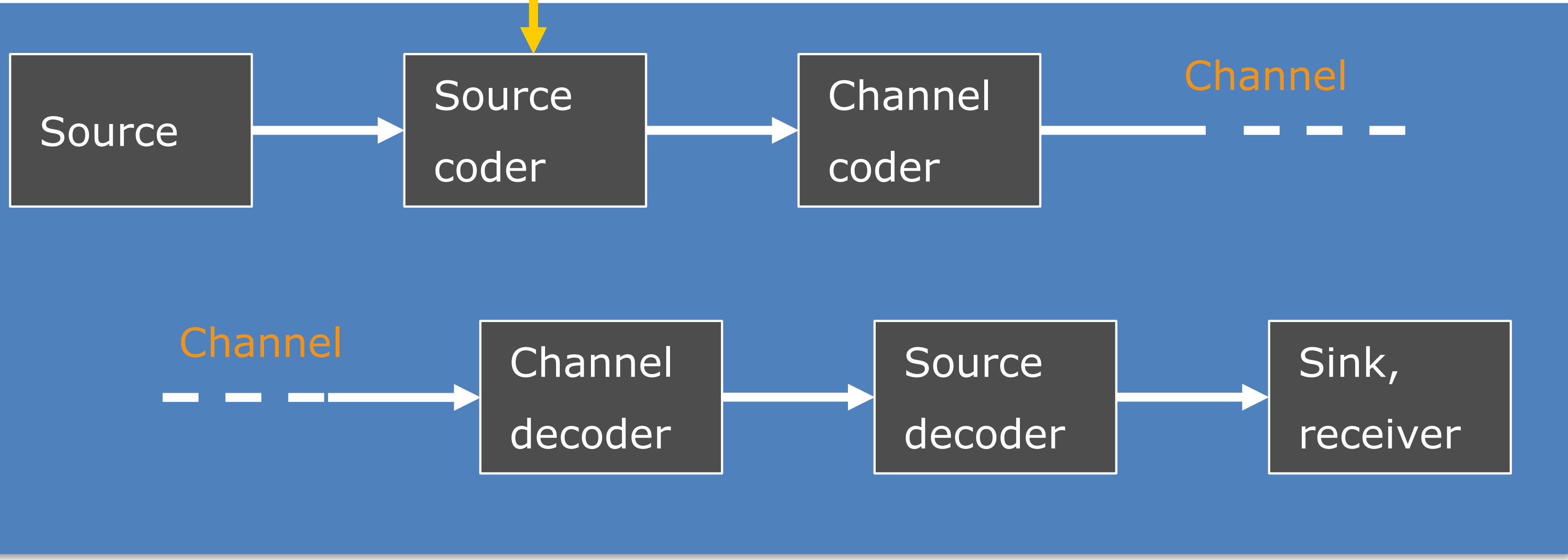
# Is Entropy that Important?

**YES!**

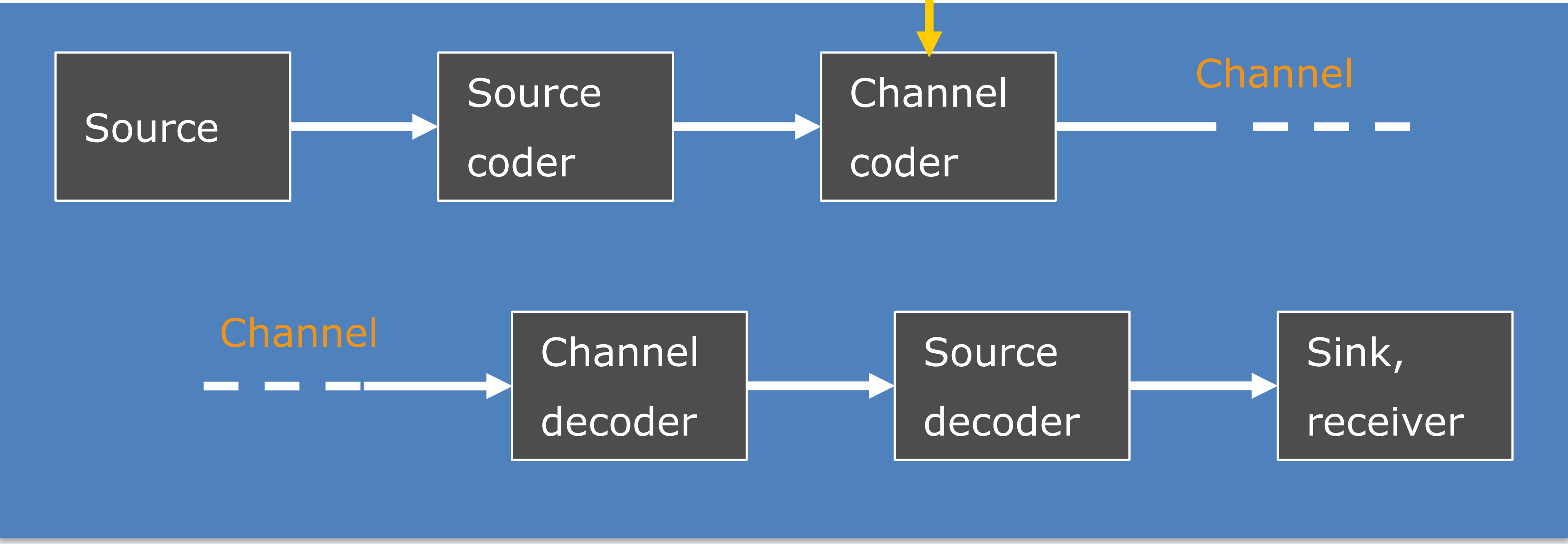
Any source of information

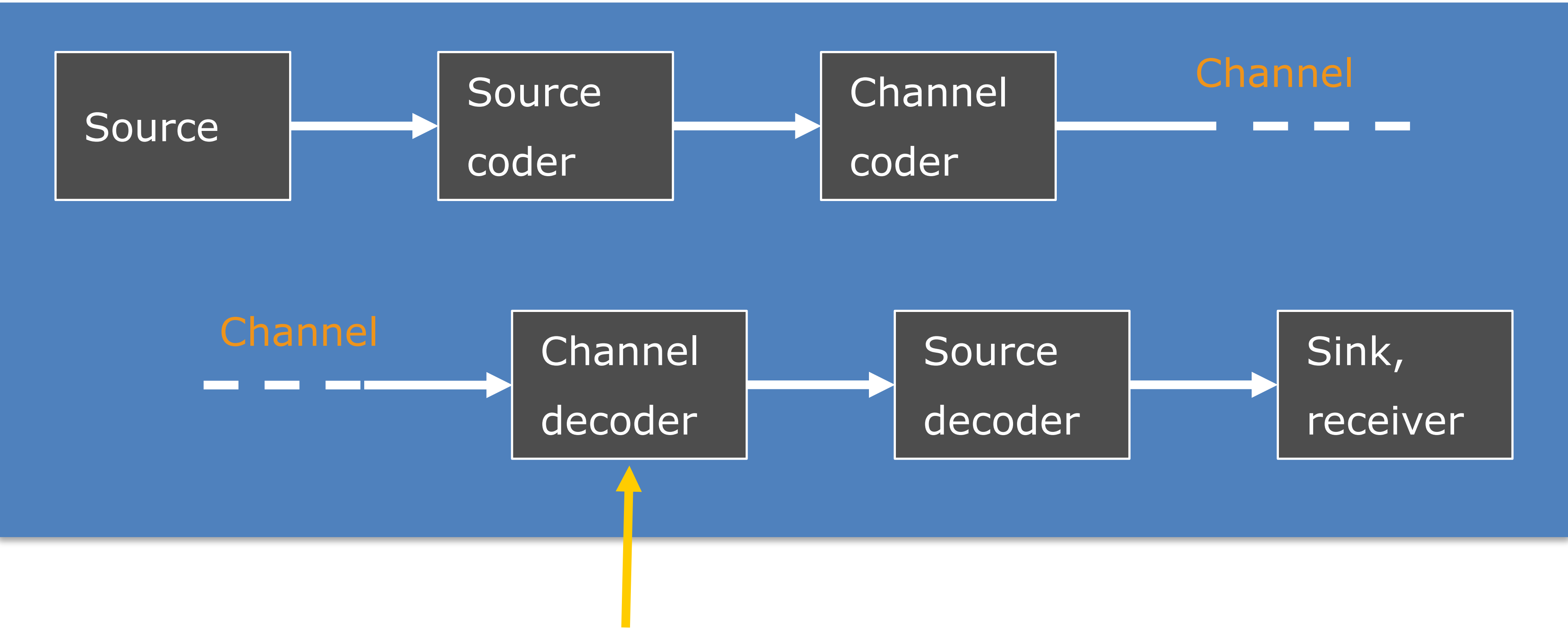


Change to an efficient representation,
i.e., data compression.

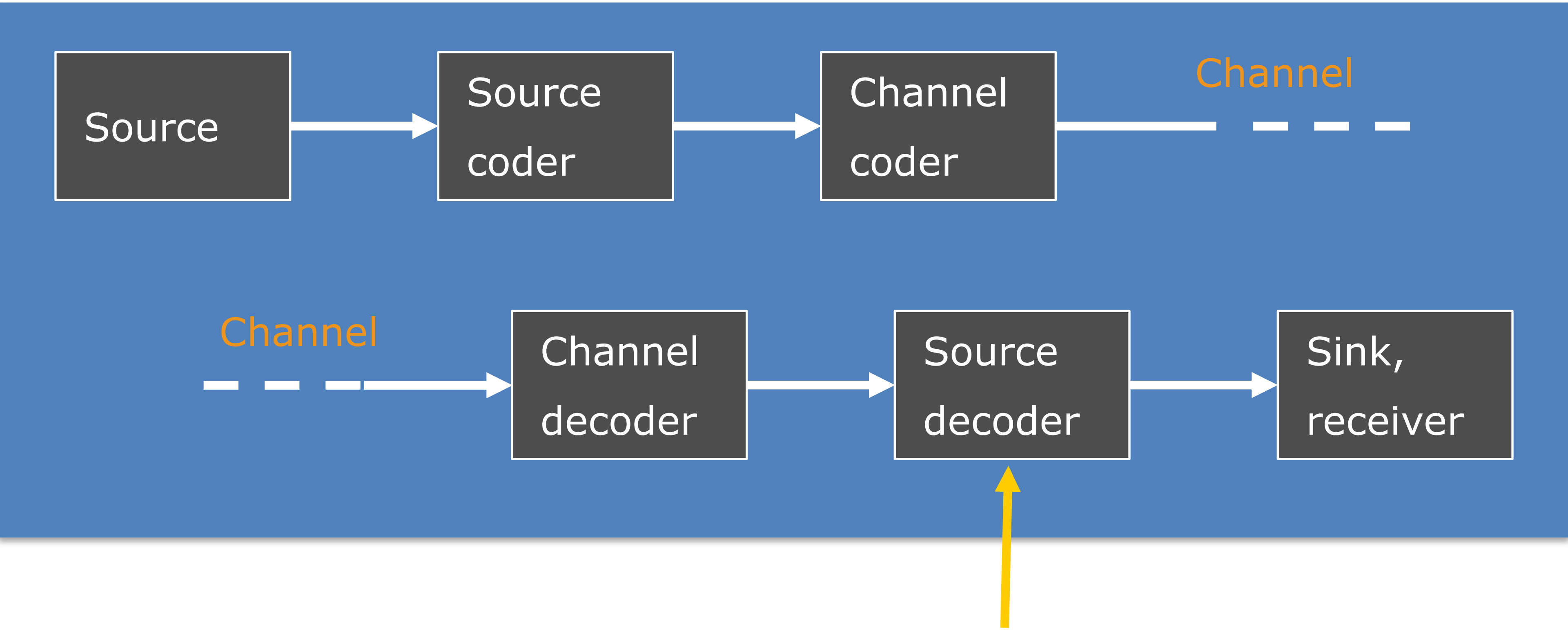


Change to an efficient representation for, transmission, i.e., error control coding.

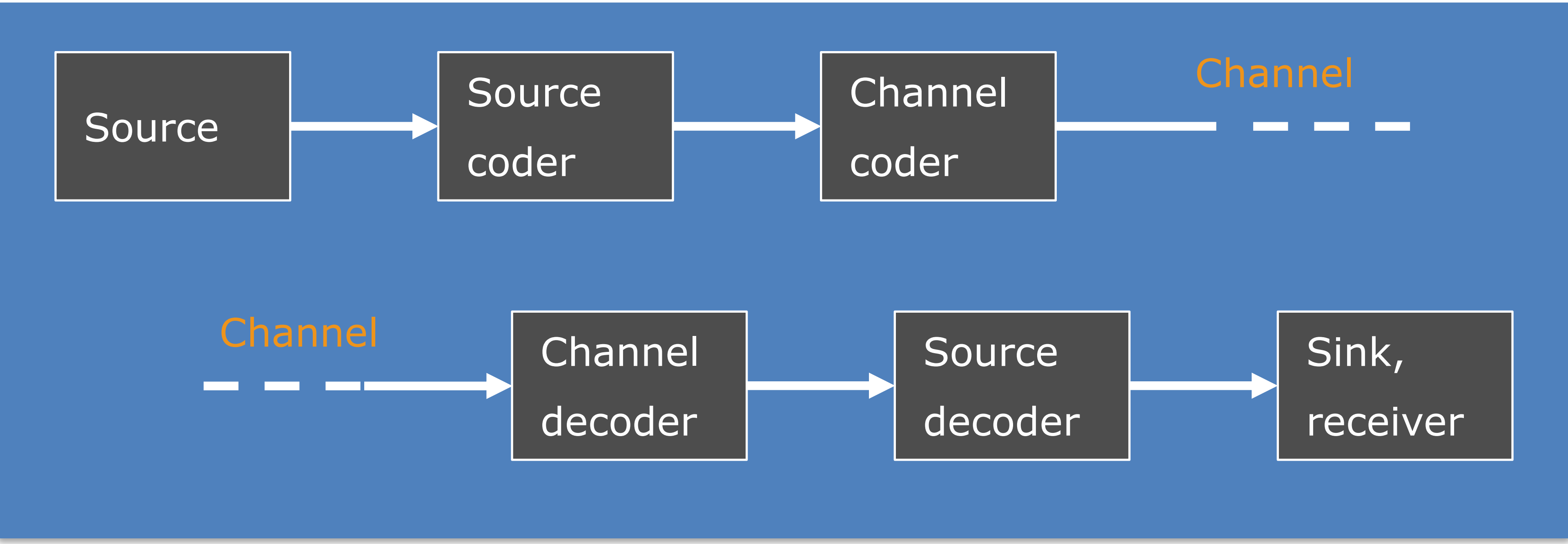





Recover from channel distortion.

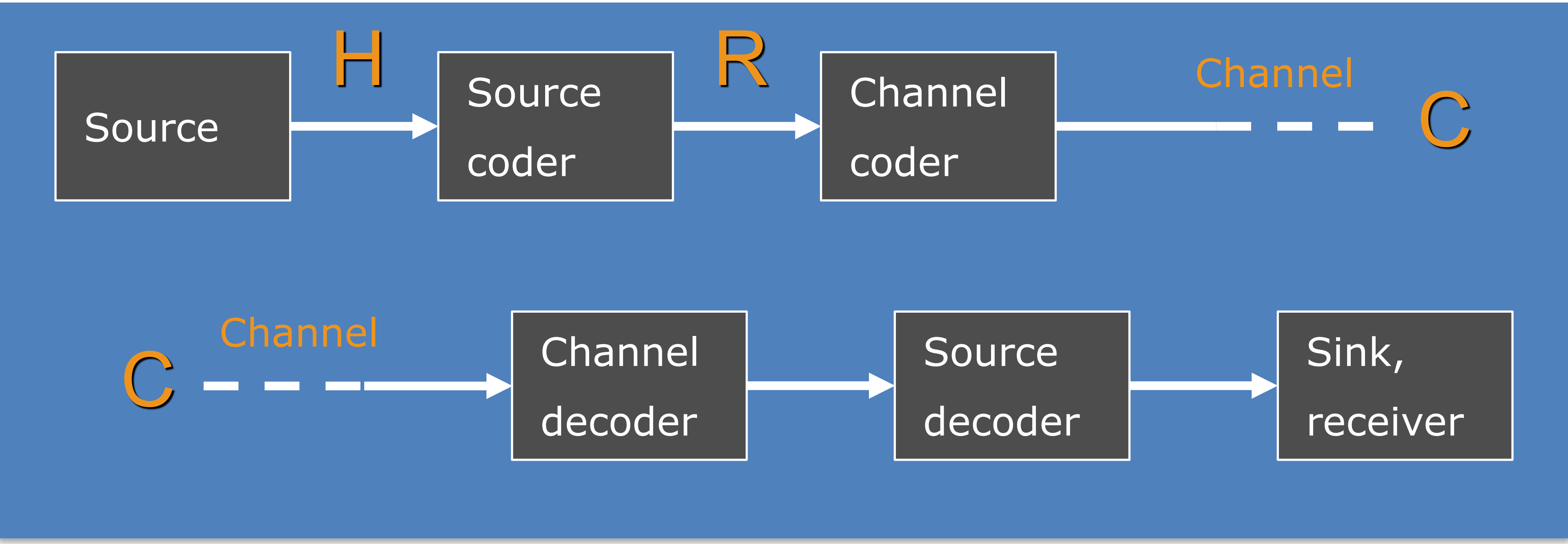



Reconstruct the source



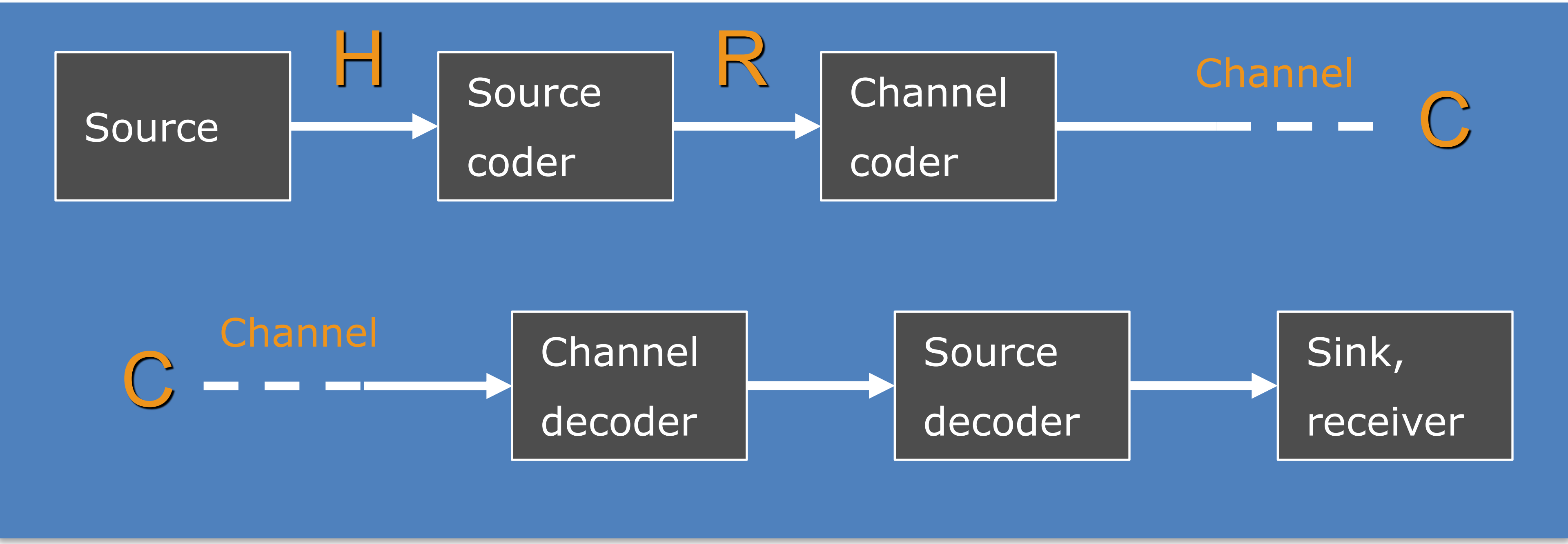

The channel is anything transmitting or storing information – a radio link, a cable, a disk, a CD, a piece of paper, …

# Fundamental Entities




H: The information content of the source.
R: Rate from the source coder.
C: Channel capacity.

# Fundamental Theorems




Shannon 1: Error-free transmission possible if $R \geq H$ and $C \geq R$.

Source coding theorem (simplified)

Channel coding theorem (simplified)

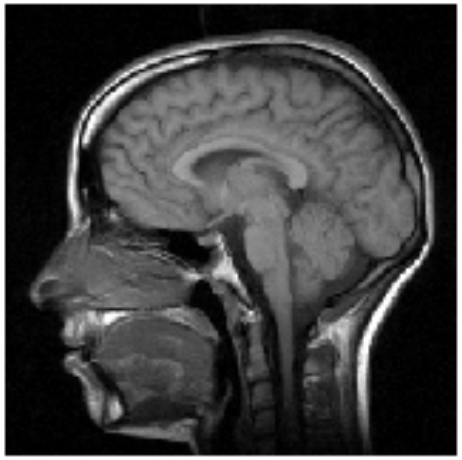
# Key concept in data compression

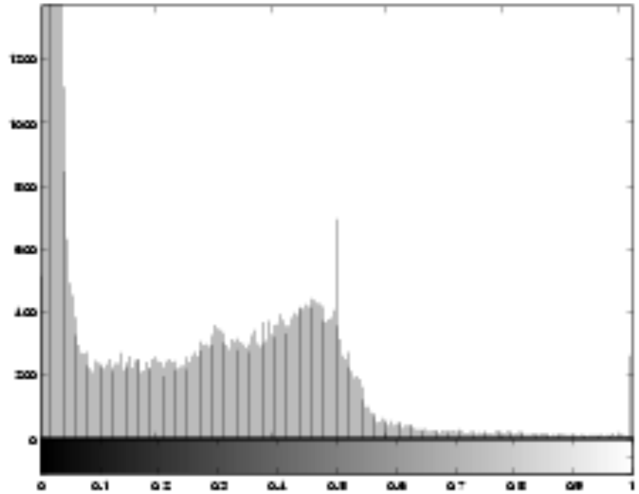
“Data compression can be achieved by assigning **short descriptions** to the **most frequent outcomes of the data source**, and necessarily longer descriptions to the less frequent outcomes.” – Thomas & Cover

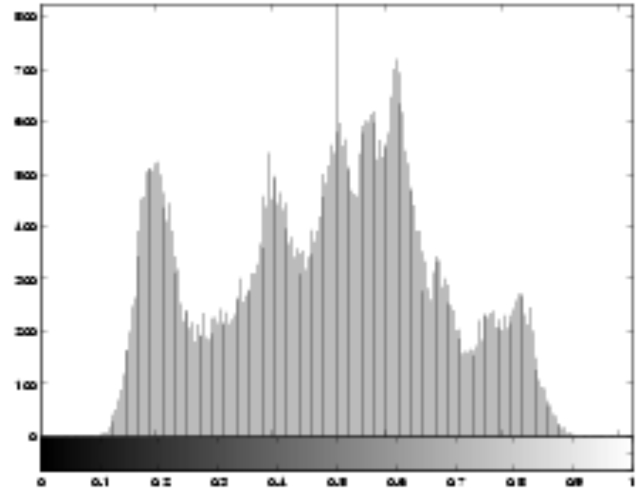
**Why?**

# Coding Redundancy

- Some things are more common than others:







# Short words are common

- Short representations for common things, and long representations for uncommon
- This concept is already familiar in many ways
- The 19 most common words in the English language are:

  the, of, are, I, and, you, a, can, to, he,

  her, that, in, was, is, has, it, him, his

# Long words are uncommon

- Very long words are generally very uncommon:
  - Antidisestablishmentarianism
  - Floccipoccinihilipilification
  - Pneumonoultramicroscopicsilicovolcanicconiosis

# Morse Code – Common Letters

| Letter | Frequency per 1000 in English | International Morse Code |
|---|---|---|
| E | 130 | · |
| T | 93 | – |
| N | 78 | – · |
| R | 77 | · – · |
| I | 74 | · · |

# Morse Code – Uncommon Letters

| Letter | Frequency per 1000 English | International Morse Code |
|---|---|---|
| X | 5 | —··— |
| K | 3 | —·— |
| Q | 3 | — —·— |
| J | 2 | ·— — — |
| Z | 1 | — —·· |

# Source Coding

$$X \xrightarrow{\mathcal{X}:\{x_1,\ x_2,\ \dots\ x_j\ \}} \boxed{\text{Source coding}} \xrightarrow{\mathcal{D}:\{0,\ 1,\ 00,\ 01,\ 10,\ 11,\ 000,\ \dots\}} \mathcal{D}^*$$

- Let $X$ be a random variable taking on values $x_1,\ x_2,\ \dots\ x_j$ from a finite alphabet $\mathcal{X}$
- Let $\mathcal{D}^*$ be the set of finite length strings of symbols from a D-ary alphabet
- For binary, $\mathcal{D}^* = \{0,1,00,01,10,11,000,\dots\}$
- $\boxed{\text{A \textbf{source code} } C \text{ for the random variable } X \text{ is a mapping from } \mathcal{X} \text{ to } \mathcal{D}^*}$
- $C(x)$ denotes the codeword corresponding to $x$
- $l(x)$ denotes the length of $C(x)$

# Source Coding

EXAMPLE:

C(red) = 00, C(blue) = 11 is a source code for $\mathcal{X}$ = {red, blue} with alphabet $\mathcal{D} = \{0,1\}$

# Expected Length

- The expected length of the code is:

$$L(X) = \sum_{x \in \mathcal{X}} p(x) l(x)$$

- ASCII is a **fixed-length** code (FLC)

a → 10000011 and A → 10000001

- **The same number of bits (7) is used to represent each symbol**

# Variable Length Coding

- If we want to **reduce the number of bits required to represent different messages**, we should use different numbers of bits to represent different symbols

- Use fewer bits for things that occur more often

- This would be a ***variable length code*** **(VLC)**

# Examples of Codes

| Input letter | Prob. | Code 1 | Code 2 | Code 3 | Code 4 |
|---|---|---|---|---|---|
| A | 1/2 | 0 | 0 | 1 | 0 |
| B | 1/4 | 0 | 1 | 01 | 01 |
| C | 1/8 | 1 | 00 | 001 | 011 |
| D | 1/8 | 10 | 11 | 000 | 0111 |
| L(C) | | 1.125 | 1.25 | 1.75 | 1.875 |

# Problems with these codes

- Code 1: Two input symbols have the same codeword
- Code 2: This problem is fixed. But suppose the decoder receives 00. What was the input? C or AA?
- Codes 3 and 4 look OK. Code 3 is shorter.
- Is that the best we can do?

# Introduction to Information Coding

Dr Laura Toni
l.toni@ucl.ac.uk

University College London

# Examples of Codes

| Input letter | Prob. | Code 1 | Code 2 | Code 3 | Code 4 |
|---|---|---|---|---|---|
| A | 1/2 | 0 | 0 | 1 | 0 |
| B | 1/4 | 0 | 1 | 01 | 01 |
| C | 1/8 | 1 | 00 | 001 | 011 |
| D | 1/8 | 10 | 11 | 000 | 0111 |
| L(C) | | 1.125 | 1.25 | 1.75 | 1.875 |

00 → AA

00 → C

# Problems with these codes

- Code 1: Two input symbols have the same codeword
- Code 2: This problem is fixed.  But suppose the decoder receives 00.  What was the input?  C or AA?
- Codes 3 and 4 look OK.  Code 3 is shorter.
- Is that the best we can do?

# More Examples of Codes

| Input Letter | Singular | Non-sing *Not* U.D. | U.D. *Not* prefix | Prefix (Instant.) |
|---|---|---|---|---|
| A | 0 | 0 | 10 | 0 |
| B | 0 | 010 | 00 | 10 |
| C | 0 | 01 | 11 | 110 |
| D | 0 | 10 | 110 | 111 |

# Optimal Code

$$H(X) = \sum_{x_i \in X} p(x_i) \cdot \cancel{1} \log \underbrace{\frac{1}{p(x_i)}}_{p(x_i)^{-1}} = -\sum p(x_i) \log p(x_i)$$

$$\underbrace{\phantom{p(x_i)^{-1}}}_{-\log p(x_i)}$$

$$H(X) = -\sum p(x_i) \log(x_i)$$

$$L(X) = \sum_{x_i \in X} p(x_i)\, l(x_i)$$

$$\boxed{L(X) \geq H(X)}$$

$$H(X) \leq L^* < H(X) + 1$$

# Optimal Code

**Theorem 5.3.1** *The expected length $L$ of any instantaneous $D$-ary code for a random variable $X$ is greater than or equal to the entropy $H_D(X)$; that is,*

$$L \geq H_D(X), \tag{5.21}$$

*with equality if and only if $D^{-l_i} = p_i$.*

$$-\sum p_i \log_D p_i = H_D(X)$$

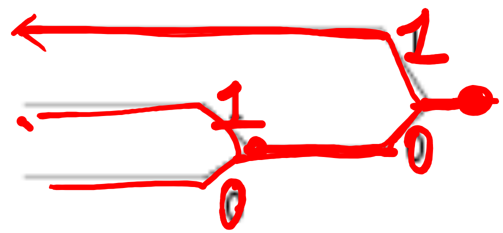
**Theorem 5.4.1** *Let $l_1^*, l_2^*, \ldots, l_m^*$ be optimal codeword lengths for a source distribution $\mathbf{p}$ and a $D$-ary alphabet, and let $L^*$ be the associated expected length of an optimal code ($L^* = \sum p_i l_i^*$). Then*

$$H_D(X) \leq L^* < H_D(X) + 1. \tag{5.33}$$

# Huffman Coding Example

- Example:

| input symbols | Probabilities | Binary Codewords |
|---|---|---|
| A | 3/4 | 1 |
| B | 3/16 | 01 |
| C | 1/16 | 00 |




- The entropy of the source is H = 1.012 bps
- Average length of this code is L = 1.25 bps
- The efficiency of the code is:

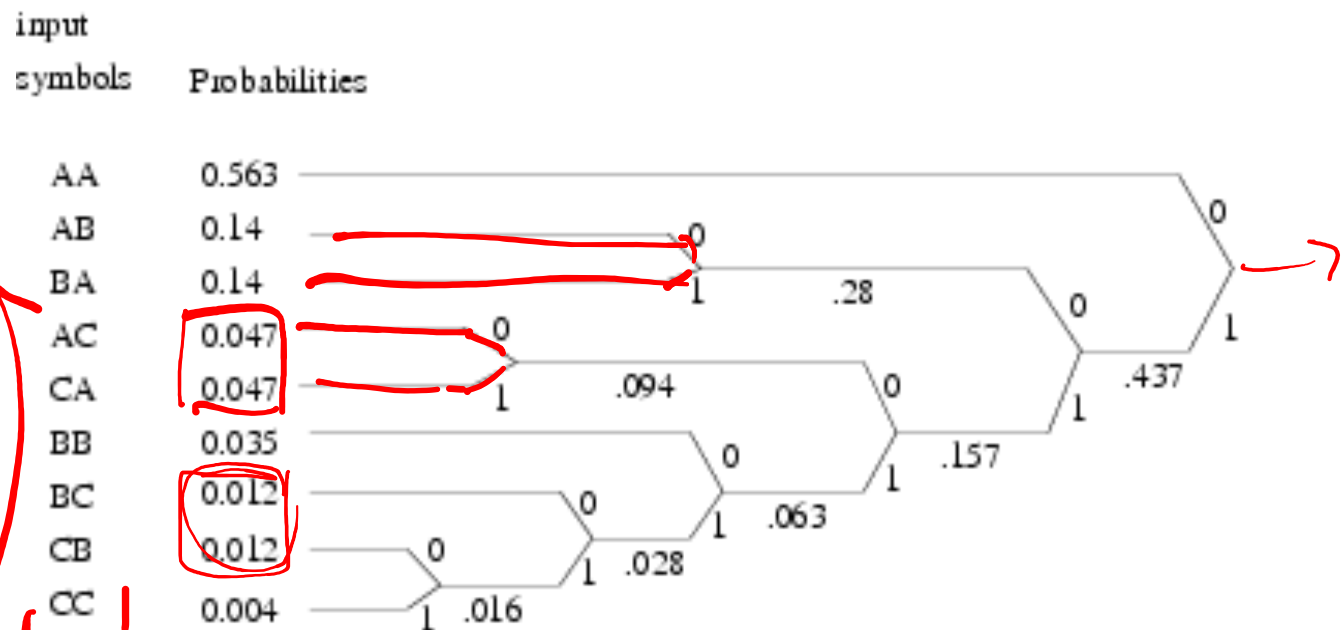
$H < L < H+1$

$1.012 \quad 1.25 \quad 2.012$

$$efficiency = \frac{H}{L} = \frac{1.012}{1.25} \approx 0.81$$

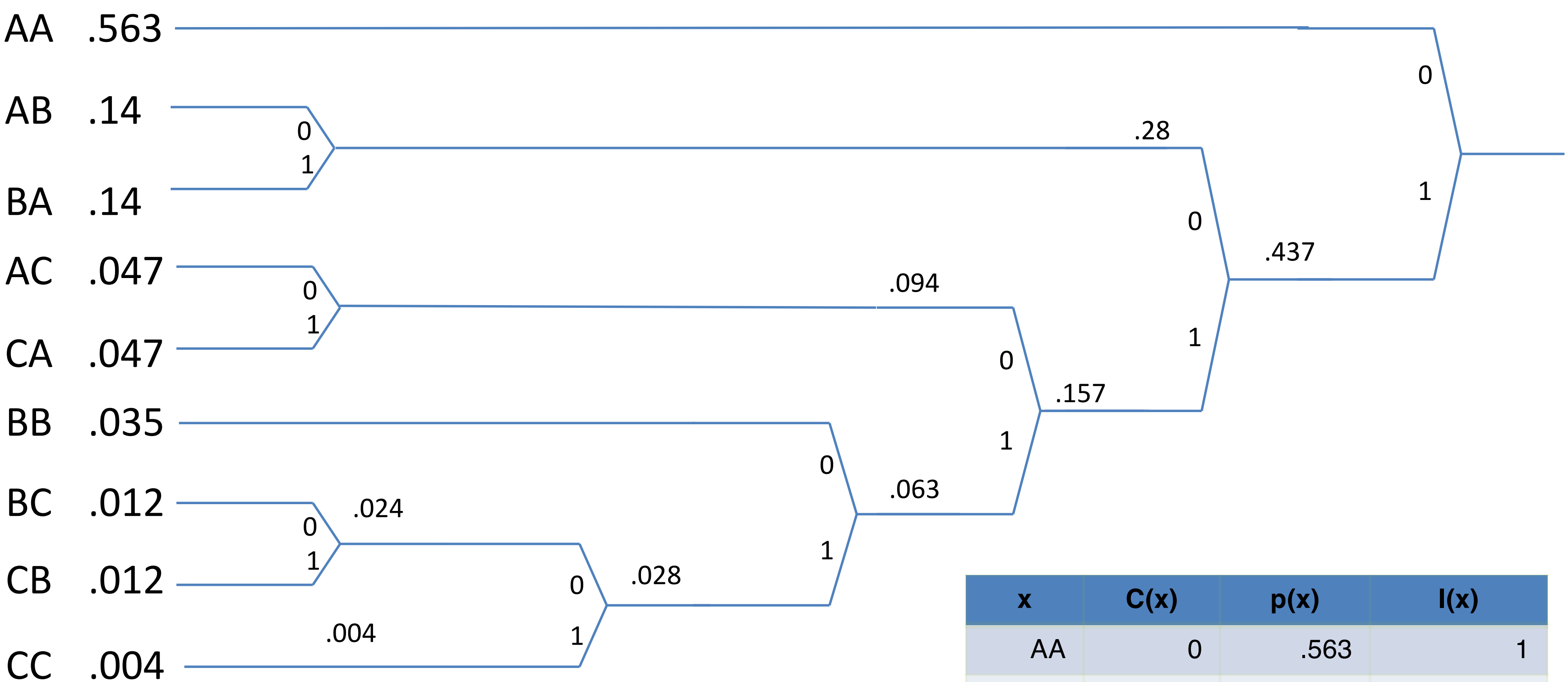
# Huffman Coding Example Extended

- If we take pairs, get more efficient code:




- Expected length now 1.0375 bps
- Efficiency is up to 1.012/1.0375 = 0.97
- **Code/decode the message: AAABAAAAACBAAABA**

# EXAMPLE




| x | C(x) | p(x) | l(x) |
|---|---|---|---|
| AA | 0 | .563 | 1 |
| AB | 100 | .14 | 3 |
| BA | 101 | .14 | 3 |
| AC | 1101 | .047 | 4 |
| CA | 1101 | .047 | 4 |
| BB | 1110 | .035 | 4 |
| BC | 111101 | .012 | 6 |
| CB | 111101 | .012 | 6 |
| CC | 11111 | 0.004 | 5 |

$L(x)=2.0830$

# Outline

**Introduction to Information compression**

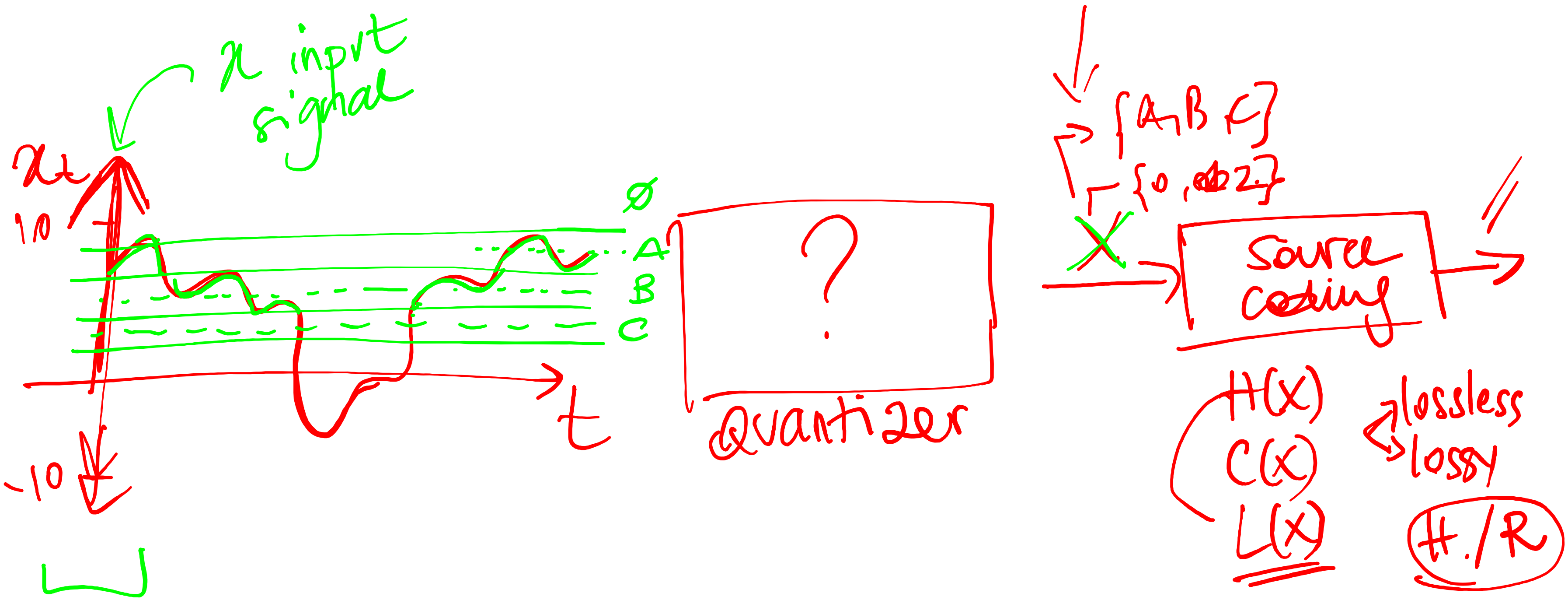
- Source Coding
- Information and Entropy
- Variable length coding
- Quantization

**Multimedia Systems**

- Image and Lossy Compression
  - Transforms
  - JPEG Quantization
  - JPEG Lossless Compression

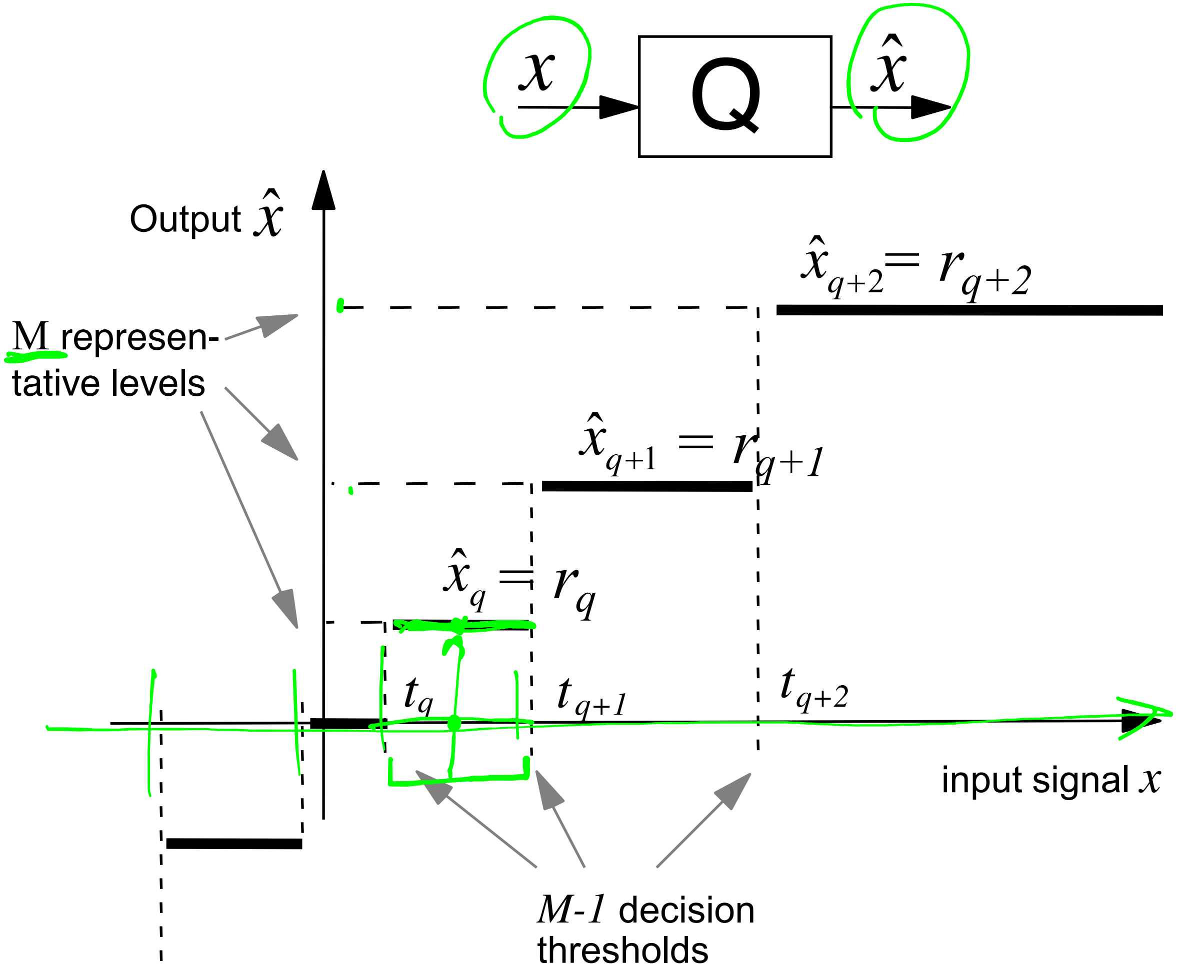
- Video Compression
  - Motion Compensation

# Why do we need quantization?

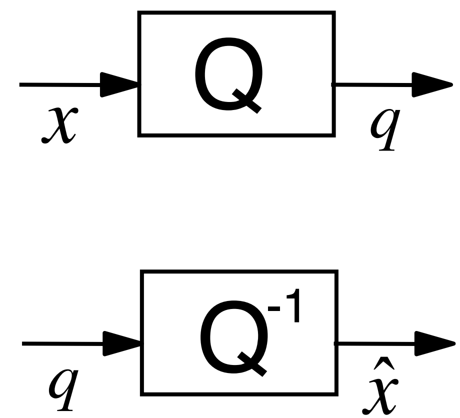


# Scalar Quantization (SQ)

Input-output characteristic of a scalar quantizer




Sometimes, this convention is used:

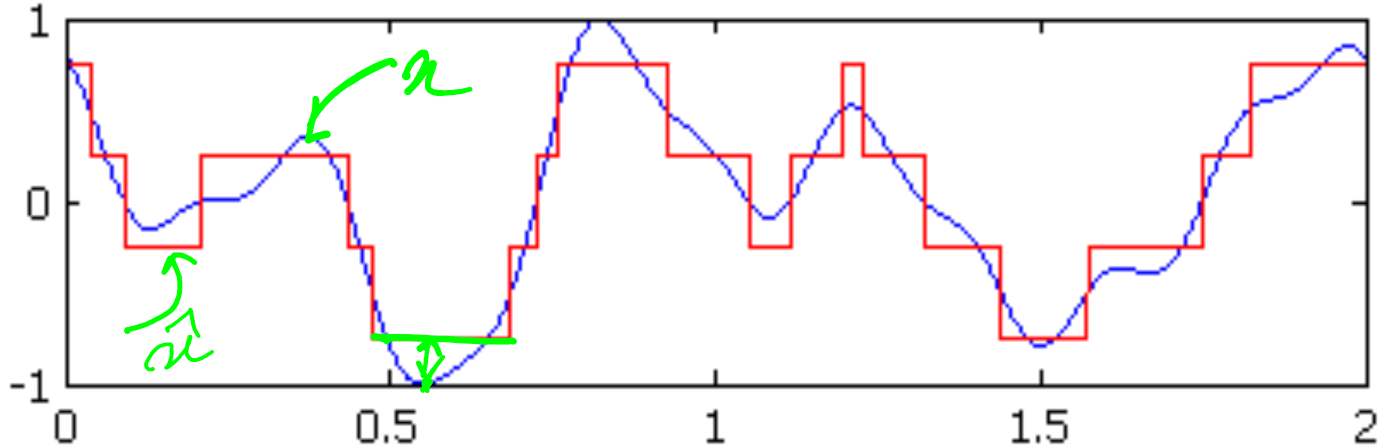



$\rightarrow$ M ↑↑ , lower quantiz err , ↑↑ |X| $\curvearrowright$ C(X) → R^M

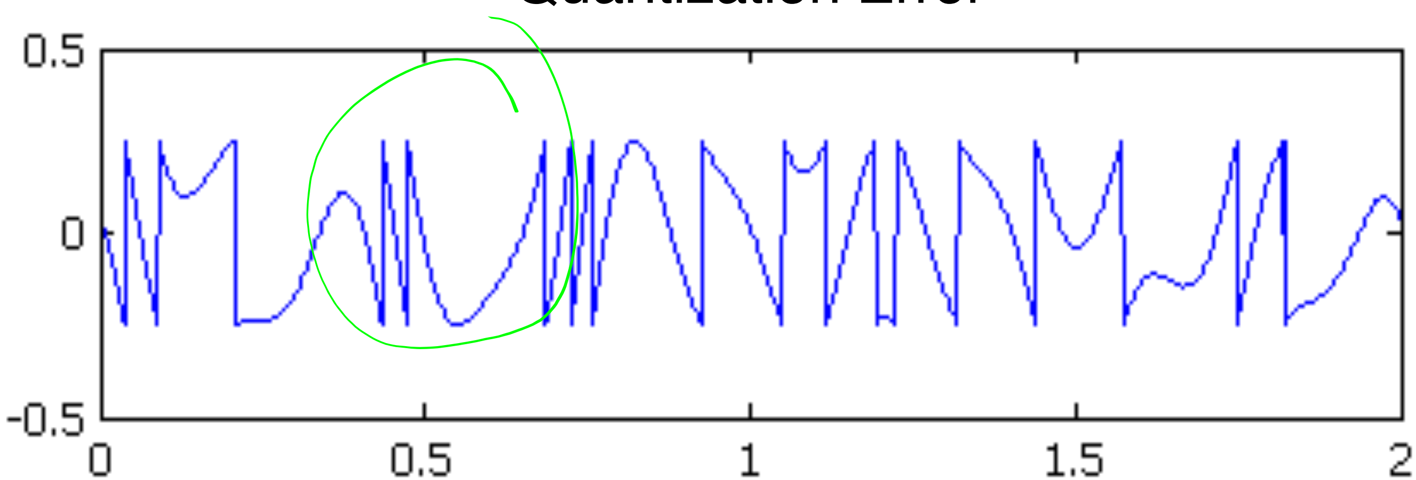
x

# Example of Quantized Waveform

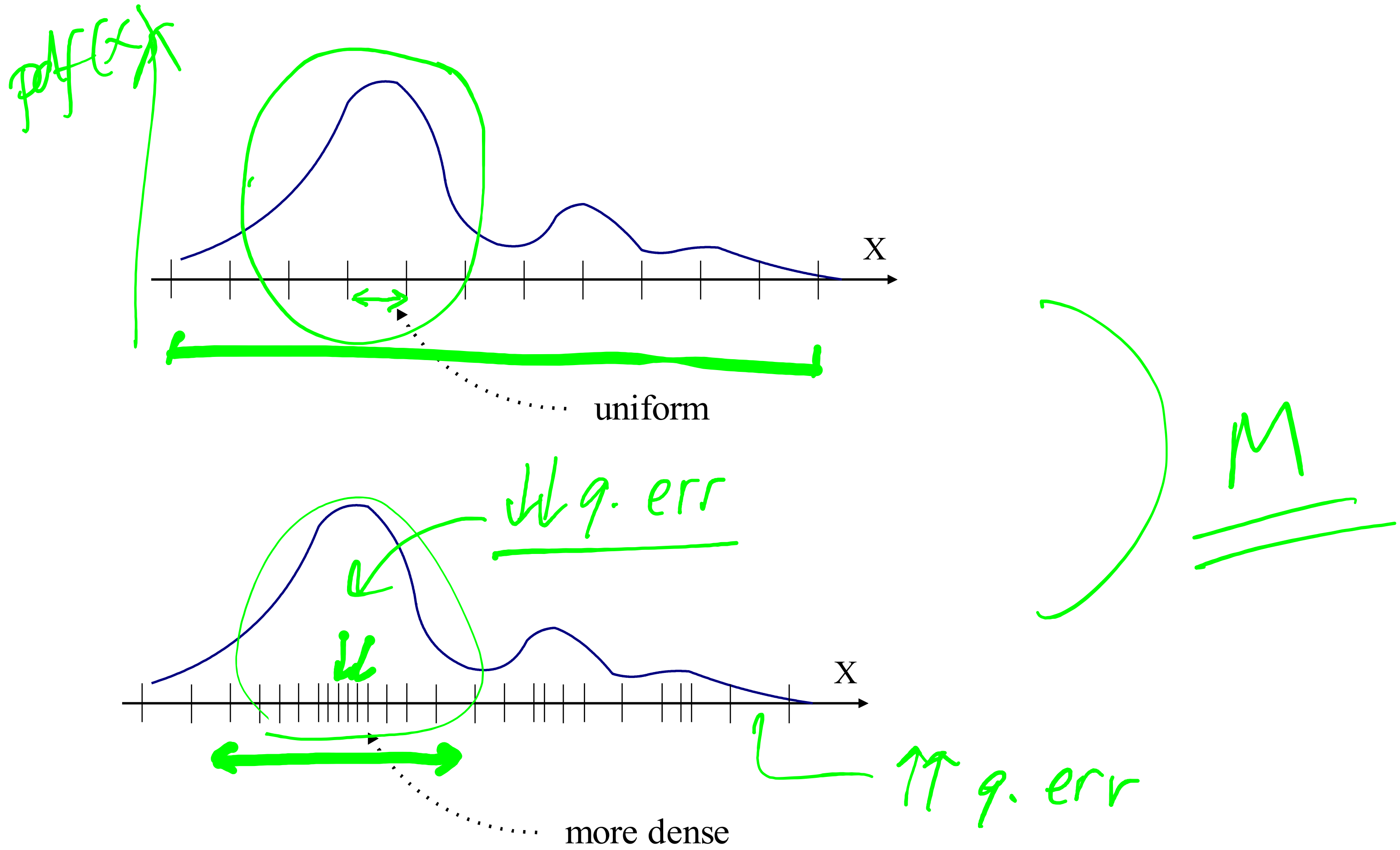
Original and Quantized Signal




Quantization Error



# Scalar Quantization




- Uniform quantization is not always the best

# Thank You