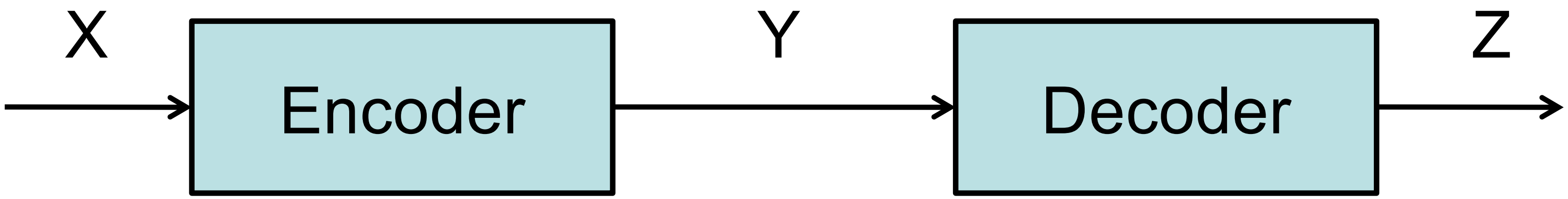
Lossless Coding

- Data processing
- Source code
- Prefix code
- Kraft inequality
- Optimal code
- Noiseless source coding theorem
- Huffman code

Encoding and Decoding

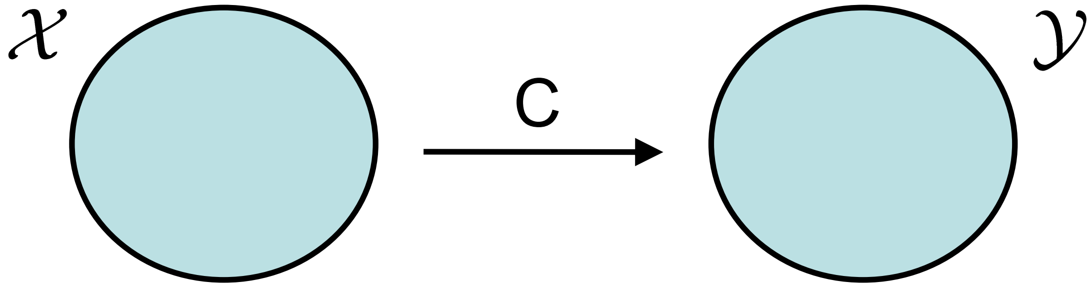
- Encoding and decoding as data processing



- Lossless coding: $Z = X$
- Data processing inequality: $H(X) \leq I(X;Y)$

Binary Source Code

- A binary source code C for a random variable X is a mapping from $\mathcal{X}$ to $\mathcal{Y}$, the set of finite length strings of binary symbols.



- The expected length R(C) of a source code C for a random variable X with PMF f_X is

$$R(C) = \sum_{x \in \mathcal{X}} f_X(x) l(x),$$

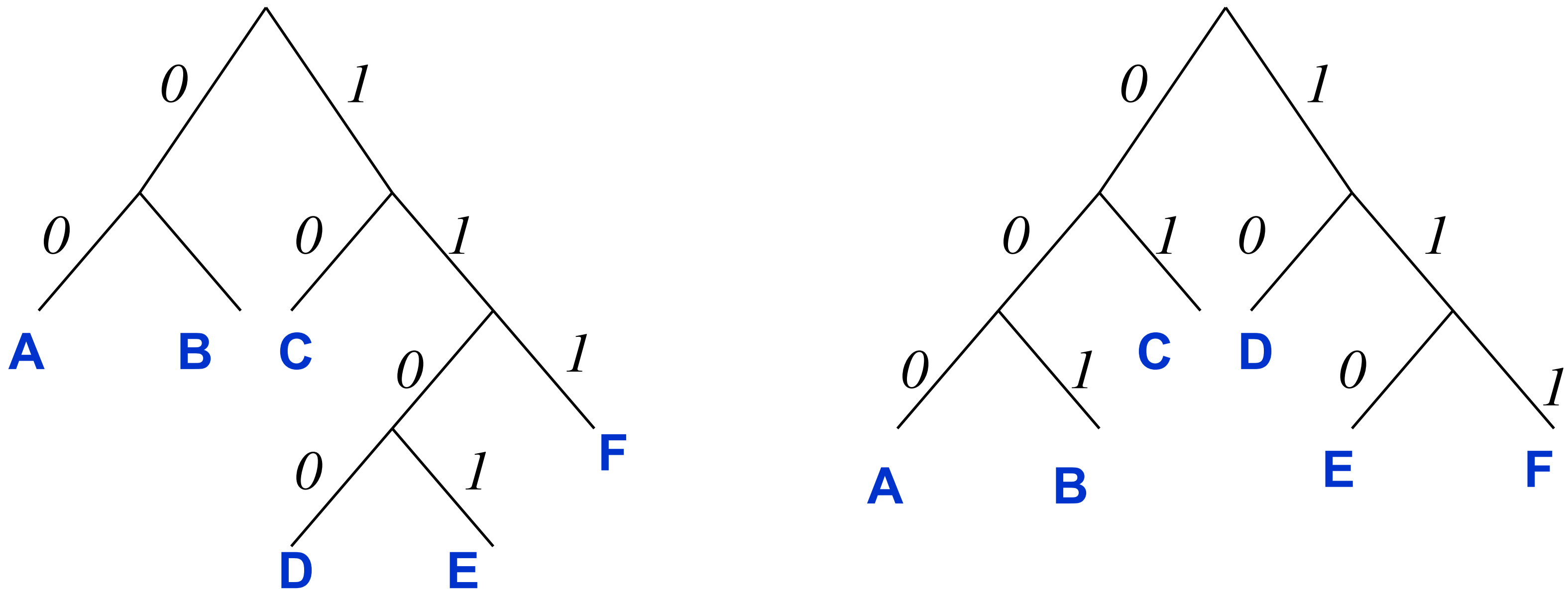
where l(x) is the length of the codeword associated with x.

Example: 20 Questions

- *Alice* thinks of an outcome (from a finite set), but does not disclose her selection.
- *Bob* asks a series of yes-no questions to uniquely determine the outcome chosen. The goal of the game is to ask as few questions as possible **on average**.
- **Our goal:** Design the best strategy for *Bob.*

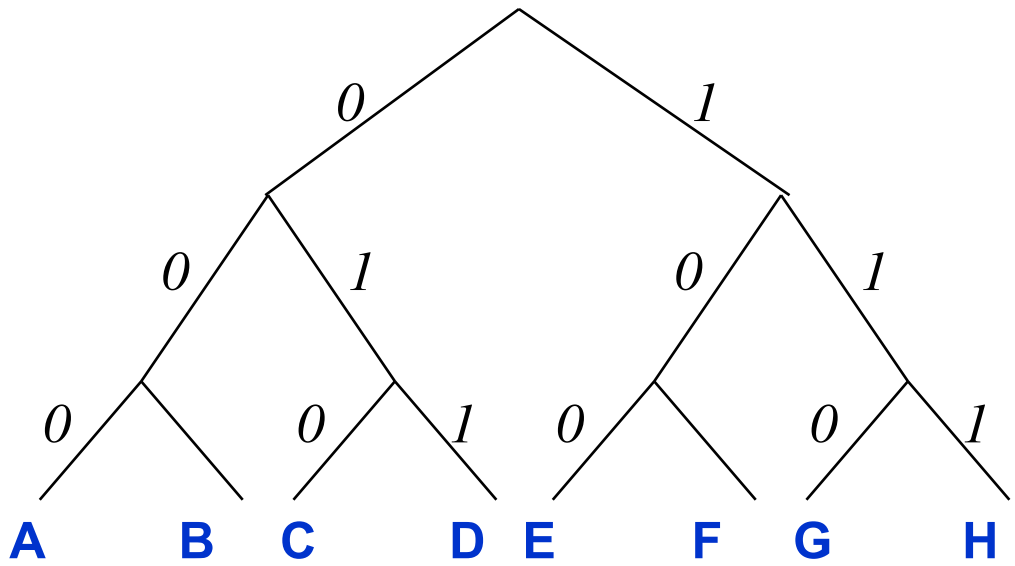
Example: 20 Questions

- Observation: The collection of questions and answers yield a binary code for each outcome.



- Which strategy (=code) is better?

Fixed Length Codes



- Average description length for K outcomes $l_{av} = \log_2 K$
- Optimum for equally likely outcomes
- Verify by modifying tree

Variable Length Codes

- If outcomes are NOT equally probable:
  - Use shorter descriptions for likely outcomes
  - Use longer descriptions for less likely outcomes
- Intuition:
  - Optimum balanced code trees, i.e., with equally likely outcomes, can be pruned to yield unbalanced trees with unequal probabilities.
  - The unbalanced code trees such obtained are also optimum.
  - Hence, an outcome of probability *p* should require about

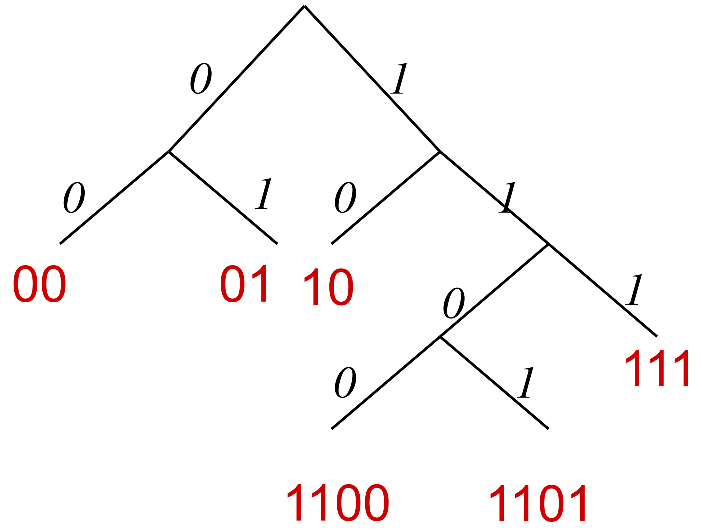
$$\log_2\left(\frac{1}{p}\right) \text{ bits}$$

Variable Length Codes

- Given IID random process $\{X_n\}$ with alphabet A_X and PMF $f_X(x)$
- Task: assign a distinct code word, c_x, to each element, $x \in \mathrm{A}_X$, where c_x is a string of $\|c_x\|$ bits, such that each symbol x_n can be determined from a sequence of concatenated codewords c_{x_n}
- Codes with the above property are said to be **"uniquely decodable"**
- Prefix codes
  - No code word is a prefix of any other codeword
  - Uniquely decodable, symbol by symbol, in natural order *0, 1, 2, . . . , n, . . .*

Binary Trees and Prefix Codes

- Each binary tree can be converted into a prefix code by traversing the tree from root to leaves.



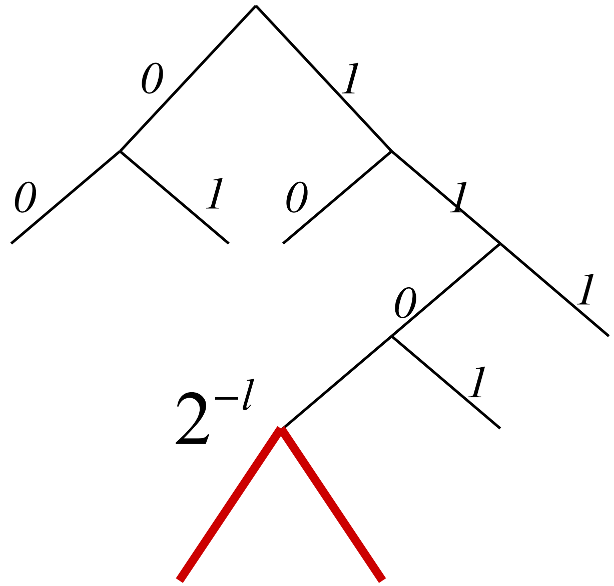
- Each prefix code corresponding to a binary tree meet McMillan condition with equality

$$3 \cdot 2^{-2} + 2 \cdot 2^{-4} + 2^{-3} = 1$$

$$\boxed{\sum_{x \in \mathrm{A}_X} 2^{-\|c_x\|} = 1}$$

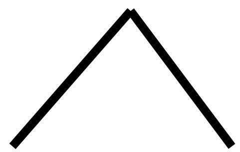
Binary Trees and Prefix Codes

- Augmenting binary tree by two new nodes does not change McMillan sum.
- Pruning binary tree does not change McMillan sum.



$$2^{-(l+1)} + 2^{-(l+1)} = 2^{-l}$$

- McMilllan sum for simplest binary tree



$$2^{-1} + 2^{-1} = 1$$

Kraft Inequality

- For any prefix code (instantaneous code) over a binary alphabet, the codeword length l_i must satisfy the inequality

$$\sum_i 2^{-l_i} \leq 1$$

- Necessary condition for uniquely decodable code
- Sufficient condition that a prefix code exists

Optimal Code

- Constrained optimization problem

$$R = \sum_i p_i l_i \quad \text{s.t.} \quad \sum_i 2^{-l_i} \leq 1$$

- Unconstrained optimization problem

$$J = \sum_i p_i l_i + \lambda \left(\sum_i 2^{-l_i} - 1 \right)$$

- Optimal codeword length

$$l_i^* = -\log_2 p_i$$

- Shannon code

$$l_i = \lceil -\log_2 p_i \rceil$$

Noiseless Source Coding Theorem

- Consider IID random process $\{X_n\}$ (or “source”) where each sample X_n (or “symbol”) possesses identical entropy *H(X)*
- *H(X)* is the entropy rate of the random process.
- **Noiseless Source Coding Theorem (Shannon, 1948):**
  – The entropy *H(X)* is a lower bound for the average word length *R* of a decodable variable-length code for the symbols.
  – Conversely, the average word length *R* can approach *H(X),* if sufficiently large blocks of symbols are encoded jointly.
- Redundancy of a code: $\boxed{\rho = R - H(X) \geq 0}$

Instantaneous Variable Length Encoding without Redundancy

- A code without redundancy, i.e.,

$$R = H(X)$$

requires all individual code word lengths

$$l_{\alpha_k} = -\log_2 f_X(\alpha_k)$$

- All probabilities would have to be binary fractions:

$$f_X(\alpha_k) = 2^{-l_{\alpha_k}}$$

Example

α_i	$P(\alpha_i)$	**redundant code**	**optimum code**
α_0	**0.500**	**00**	**0**
α_1	**0.250**	**01**	**10**
α_2	**0.125**	**10**	**110**
α_3	**0.125**	**11**	**111**

$$H(X) = 1.75 \text{ bits}$$

$$R = 1.75 \text{ bits}$$

$$\rho = 0$$

Redundancy of Prefix Code for General Distribution

- **Theorem:** For any distribution f_X, a prefix code may be found, whose rate R satisfies

$$\boxed{H(X) \leq R < H(X) + 1}$$

- Proof:
  - Left hand inequality: Shannon's noiseless coding theorem
  - Right hand inequality:

$$\text{Choose code word lengths} \quad \|c_x\| = \lceil -\log_2 f_X(x) \rceil$$

$$\text{Resulting rate} \qquad R = \sum_{x \in \mathcal{A}_X} f_X(x) \lceil -\log_2 f_X(x) \rceil$$

$$< \sum_{x \in \mathcal{A}_X} f_X(x) \left(1 - \log_2 f_X(x)\right)$$

$$= H(X) + 1$$

Huffman Code

- Design algorithm for variable length codes proposed by Huffman (1952) always finds a code with minimum redundancy.
- Obtain code tree as follows:

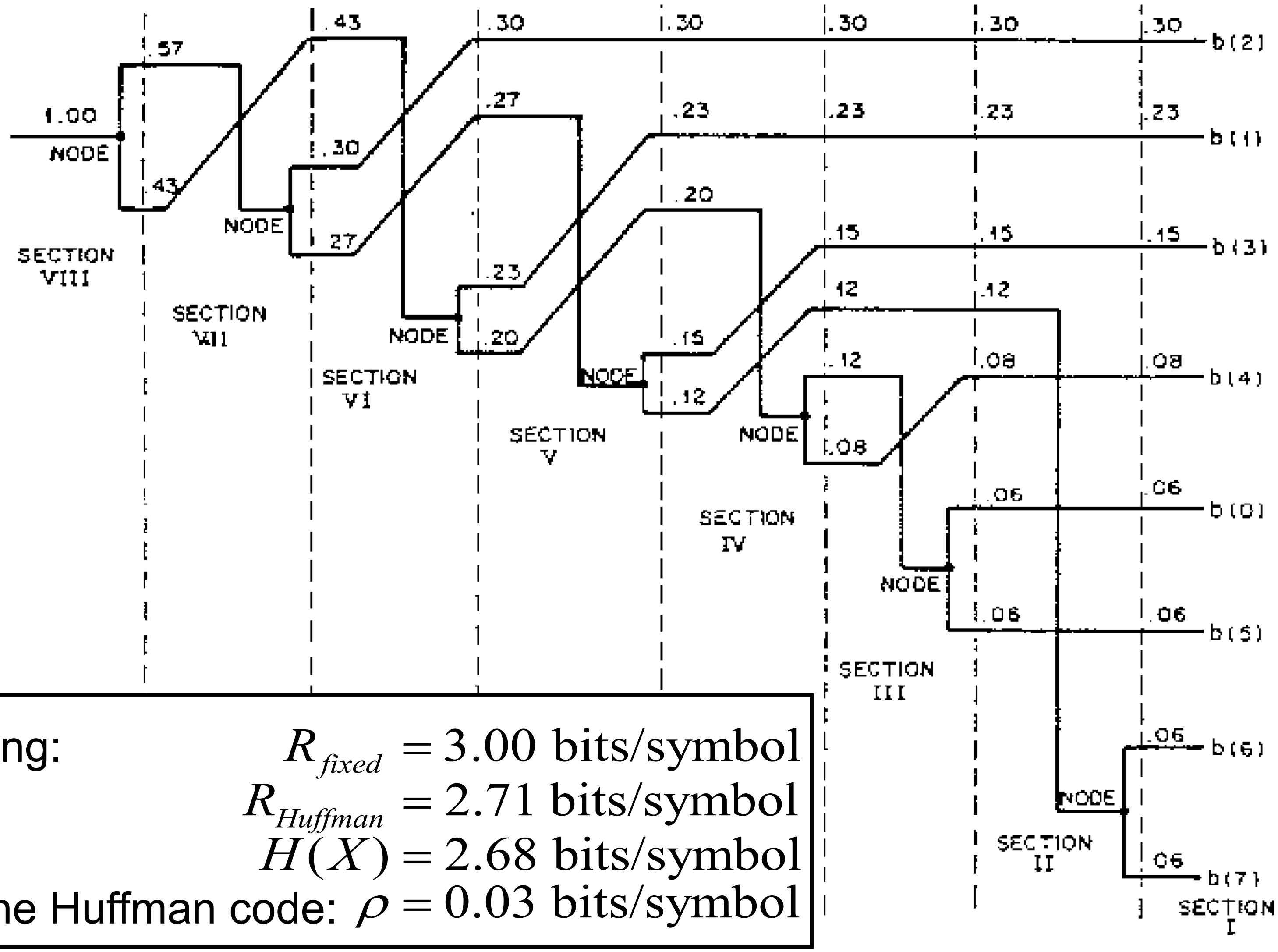
1 Pick the two symbols with lowest probabilities and merge them into a new auxiliary symbol.

2 Calculate the probability of the auxiliary symbol.

3 If more than one symbol remains, repeat steps **1** and **2** for the new auxiliary alphabet.

4 Convert the code tree into a prefix code.

Example: Huffman Code



Fixed length coding: $R_{fixed} = 3.00$ bits/symbol

Huffman code: $R_{Huffman} = 2.71$ bits/symbol

Entropy $H(X) = 2.68$ bits/symbol

Redundancy of the Huffman code: $\rho = 0.03$ bits/symbol