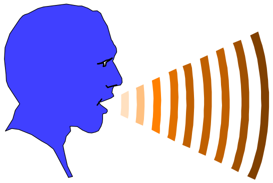


Video Processing & Communications

Foundation of Video Coding
Part I: Overview and Binary Encoding

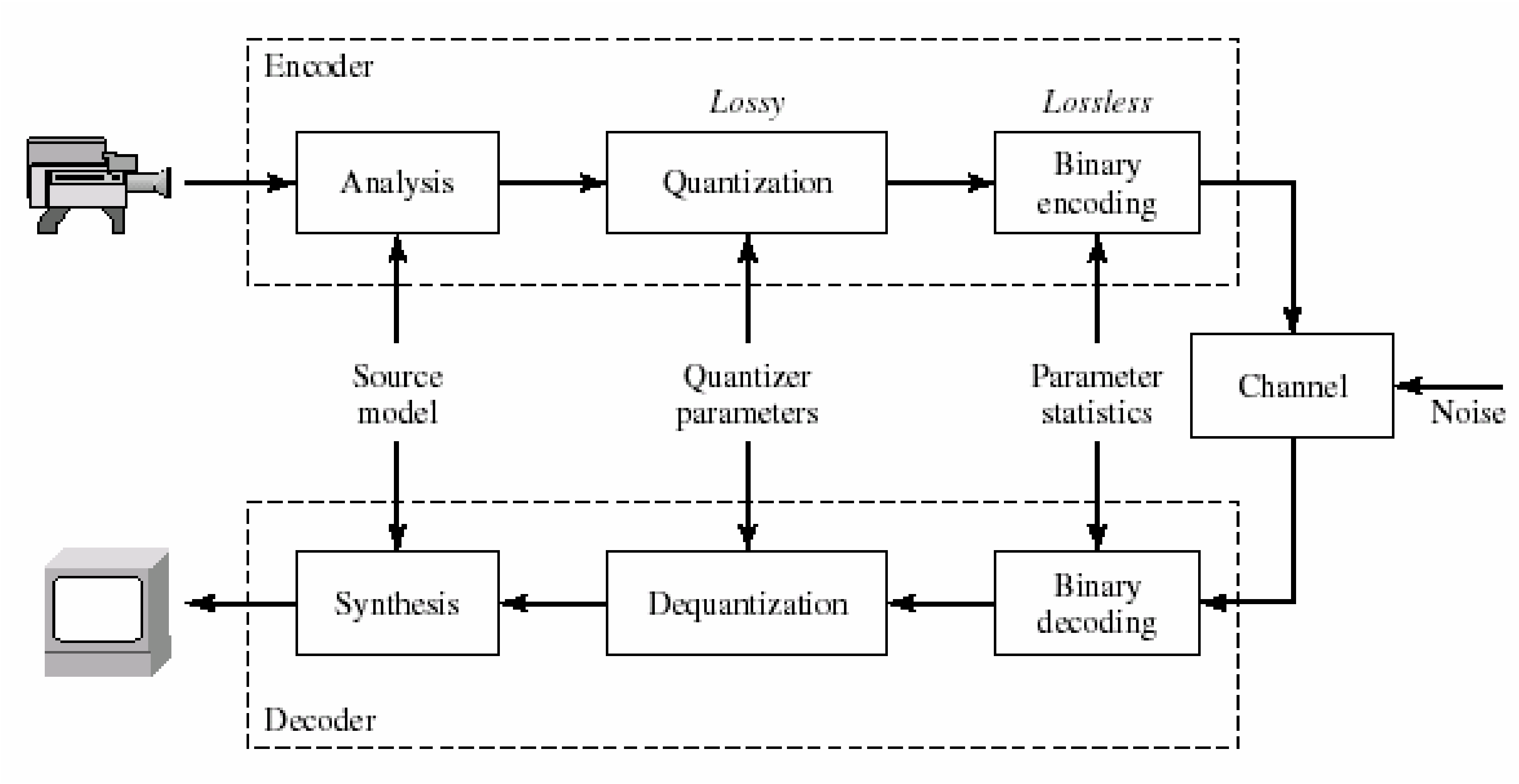
Yao Wang
Polytechnic University, Brooklyn, NY11201
http://eeweb.poly.edu

Based on: Y. Wang, J. Ostermann, and Y.-Q. Zhang, Video Processing and Communications, Prentice Hall, 2002.

Outline

- Overview of video coding systems
- Review of probability and information theory concepts
- Binary encoding
  - Information theory bounds
  - Huffman coding
  - Arithmetic coding

Components in a Coding System



Video Coding Techniques Based on Different Source Models

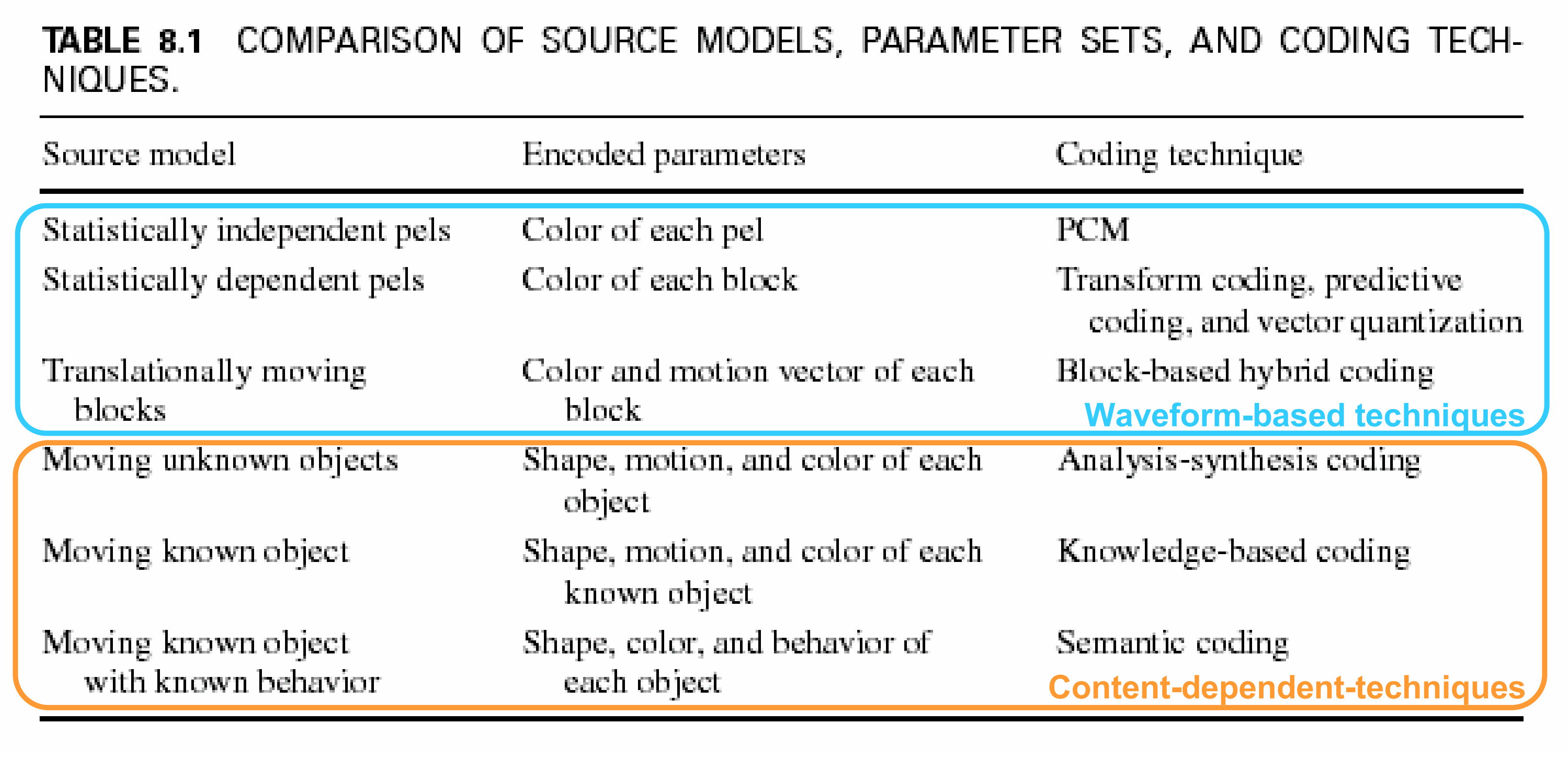


TABLE 8.1 COMPARISON OF SOURCE MODELS, PARAMETER SETS, AND CODING TECHNIQUES.

Source model	Encoded parameters	Coding technique
Statistically independent pels	Color of each pel	PCM
Statistically dependent pels	Color of each block	Transform coding, predictive coding, and vector quantization
Translationally moving blocks	Color and motion vector of each block	Block-based hybrid coding
Moving unknown objects	Shape, motion, and color of each object	Analysis-synthesis coding
Moving known object	Shape, motion, and color of each known object	Knowledge-based coding
Moving known object with known behavior	Shape, color, and behavior of each object	Semantic coding

Waveform-based techniques

Content-dependent-techniques

Statistical Characterization of Random Sources

- Source: a random sequence (discrete time random process), $\mathcal{F} = \{\mathcal{F}_n\}$
  - Ex 1: an image that follows a certain statistics
    - F_n represents the *possible value* of the n-th pixel of the image, ***n****=(m,n)*
    - f_n represents the *actual value* taken
  - Ex 2: a video that follows a certain statistics
    - F_n represents the *possible value* of the n-th pixel of a video, ***n****=(k,m,n)*
    - f_n represents the actual value taken
  - Continuous source*:* F_n takes continuous values (analog image)
  - Discrete source: F_n takes discrete values (digital image)
- Stationary source: statistical distribution invariant to time (space) shift
- Probability distribution
  - probability mass function (pmf) or probability density function (pdf): $p_{\mathcal{F}_n}(f) \quad p(f)$
  - Joint pmf or pdf: $p_{\mathcal{F}_{n+1},\mathcal{F}_{n+2},\ldots,\mathcal{F}_{n+N}}(f_1, f_2, \ldots, f_N) \quad p(f_1, f_2, \ldots f_N)$
  - Conditional pmf or pdf: $p_{\mathcal{F}_n|\mathcal{F}_{n-1},\mathcal{F}_{n-2},\ldots,\mathcal{F}_{n-M}}(f_{M+1} \mid f_M, f_{M-1}, \ldots, f_1) \quad p(f_{M+1} \mid f_M, f_{M-1}, \ldots, f_1)$

Entropy of a RV

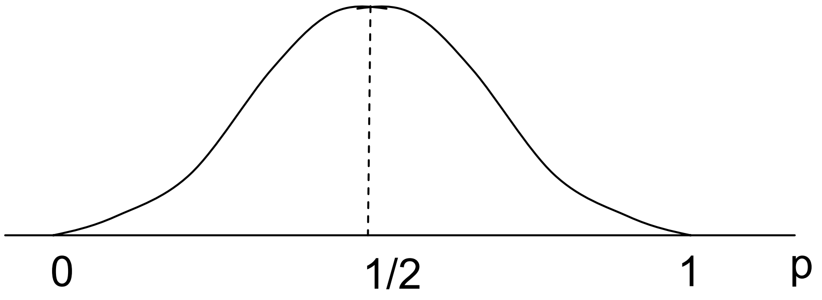
- Consider RV $F=\{f_1,f_2,\ldots,f_K\}$, with probability $p_k=Prob.\{F= f_K\}$
- Self-Information of one realization f_k : $H_k= -\log(p_k)$
  - $p_k=1$: always happen, no information
  - P_k ~0: seldom happen, its rea of information

$$H(\mathcal{F}) = -\sum_{f\in\mathcal{A}} p_{\mathcal{F}}(f)\log_2 p_{\mathcal{F}}(f).$$

- Entropy = average information:
  - Entropy is a measure of uncertainty or information content, unit=bits
  - Very uncertain -> high information content

Example: Two Possible Symbols

- Example: Two possible outcomes
  - Flip a coin, F={“head”,”tail”}: $p_1=p_2=1/2$: H=1 (highest uncertainty)
  - If the coin has defect, so that $p_1=1$, $p_2=0$: H=0 (no uncertainty)
  - More generally: $p_1=p$, $p_2=1-p$,
    - $H=-(p \log p+ (1-p) \log (1-p))$



Another Example: English Letters

- 26 letters, each has a certain probability of occurrence
  - Some letters occurs more often: “a”,”s”,”t”, …
  - Some letters occurs less often: “q”,”z”, …
- Entropy ~= information you obtained after reading an article.
- But we actually don’t get information at the alphabet level, but at the word level!
  - Some combination of letters occur more often: “it”, “qu”,…

Joint Entropy

- Joint entropy of two RVs:
  - Uncertainty of two RVs together

$$H(\mathcal{F},\mathcal{G}) = -\sum_{f\in\mathcal{A}_f}\sum_{g\in\mathcal{A}_g} p_{\mathcal{F},\mathcal{G}}(f,g)\log_2 p_{\mathcal{F},\mathcal{G}}(f,g).$$

$$H(\mathcal{F},\mathcal{G}) \le H(\mathcal{F}) + H(\mathcal{G})$$

- N-th order entropy
  - Uncertainty of N RVs together

$$\begin{aligned} H_N(\mathcal{F}) &= H(\mathcal{F}_1,\mathcal{F}_2,\ldots,\mathcal{F}_N) \\ &= -\sum_{[f_1,f_2,\ldots,f_N]\in\mathcal{A}^N} p(f_1,f_2,\ldots,f_N)\log_2 p(f_1,f_2,\ldots,f_N). \end{aligned}$$

- Entropy rate (lossless coding bound)
  - Average uncertain per RV

$$\bar{H}(\mathcal{F}) = \lim_{N\to\infty}\frac{1}{N}H_N(\mathcal{F}) = \lim_{N\to\infty}\bar{H}_{C,N}(\mathcal{F}).$$

Conditional Entropy

- Conditional entropy between two RVs:
  - Uncertainty of one RV given the other RV

$$\begin{aligned} H(\mathcal{F} \mid \mathcal{G}) &= \sum_{g \in \mathcal{A}_g} p_{\mathcal{G}}(g) H(\mathcal{F} \mid g) \\ &= -\sum_{g \in \mathcal{A}_g} p_{\mathcal{G}}(g) \sum_{f \in \mathcal{A}_f} p_{\mathcal{F} \mid \mathcal{G}}(f \mid g) \log_2 p_{\mathcal{F} \mid \mathcal{G}}(f \mid g). \end{aligned}$$

$$H(\mathcal{F}) \geq H(\mathcal{F} \mid \mathcal{G}) \qquad H(\mathcal{F}, \mathcal{G}) = H(\mathcal{G}) + H(\mathcal{F} \mid \mathcal{G})$$

- M-th o

$$\begin{aligned} H_{C,M}(\mathcal{F}) &= H(\mathcal{F}_{M+1} \mid \mathcal{F}_M, \mathcal{F}_{M-1}, \ldots, \mathcal{F}_1) \\ &= \sum_{[f_1, f_2, \ldots, f_M] \in \mathcal{A}^M} p(f_1, f_2, \ldots, f_M) H(\mathcal{F}_{M+1} \mid f_M, f_{M-1}, \ldots, f_1) \end{aligned}$$

$$\begin{aligned} &H(\mathcal{F}_{M+1} \mid f_M, f_{M-1}, \ldots, f_1) \\ &= -\sum_{f_{M+1} \in \mathcal{A}} p(f_{M+1} \mid f_M, f_{M-1}, \ldots, f_1) \log_2 p(f_{M+1} \mid f_M, f_{M-1}, \ldots, f_1). \end{aligned}$$

$$\bar{H}(\mathcal{F}) \leq H_{C,N-1}(\mathcal{F}) \leq \frac{1}{N} H_N(\mathcal{F}) \leq H_1(\mathcal{F}).$$

Example: 4-symbol source

- Four symbols: “a”,”b”,”c”,”d”
- pmf:

$$\mathbf{p}^T = [0.5000, 0.2143, 0.1703, 0.1154]$$

- 1st order conditional pmf: $q_{ij}=Prob(f_i|f_j)$

$$[\mathbf{Q}] = \begin{bmatrix} 0.6250 & 0.3750 & 0.3750 & 0.3750 \\ 0.1875 & 0.3125 & 0.1875 & 0.1875 \\ 0.1250 & 0.1875 & 0.3125 & 0.1250 \\ 0.0625 & 0.1250 & 0.1250 & 0.3125 \end{bmatrix}$$

- 2nd order pmf:

$$p(f_{n-1}, f_n) = p(f_{n-1})q(f_n \mid f_{n-1}).$$

$$\text{Ex.} \quad p("ab") = p("a")q("b"/"a") = 0.5 * 0.1875 = 0.0938$$

- Go through how to compute H_1, H_2, $H_{c,1}$.

Mutual Information

- Mutual information between two RVs :
  - Information provided by *G* about *F*

$$\boxed{I(\mathcal{F};\mathcal{G}) = \sum_{f\in\mathcal{A}_f}\sum_{g\in\mathcal{A}_g} p_{\mathcal{F},\mathcal{G}}(f,g)\log_2\frac{p_{\mathcal{F},\mathcal{G}}(f,g)}{p_{\mathcal{F}}(f)p_{\mathcal{G}}(g)}.}$$

$$I(\mathcal{F};\mathcal{G}) = H(\mathcal{F}) - H(\mathcal{F}\,|\,\mathcal{G})$$

$$I(\mathcal{F};\mathcal{G}) \leq H(\mathcal{F})$$

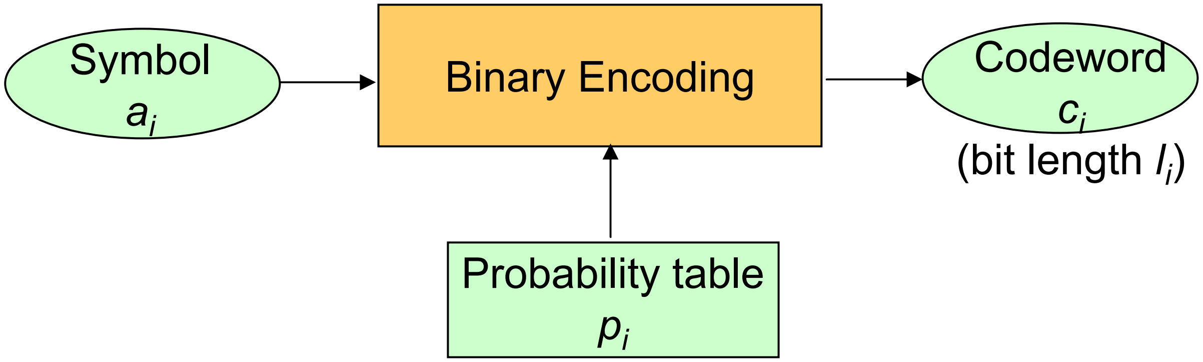
$$I(\mathcal{F};\mathcal{G}) = H(\mathcal{F}) + H(\mathcal{G}) - H(\mathcal{F},\mathcal{G})$$

- M-th order mutual information (lossy coding bound)

$$I_N(\mathcal{F};\mathcal{G}) = \sum_{[f_1,f_2,\ldots,f_N]\in\mathcal{A}_f^N}\ \sum_{[g_1,g_2,\ldots,g_N]\in\mathcal{A}_g^N} p(f_1,f_2,\ldots,f_N,g_1,g_2,\ldots,g_N)\cdot\log_2\frac{p(f_1,f_2,\ldots,f_N,g_1,g_2,\ldots,g_N)}{p(f_1,f_2,\ldots,f_N)p(g_1,g_2,\ldots,g_N)}$$

Lossless Coding (Binary Encoding)

- Binary encoding is a necessary step in any coding system
  - Applied to
    - original symbols (e.g. image pixels) in a discrete source,
    - or converted symbols (e.g. quantized transformed coefficients) from a continuous or discrete source
- Binary encoding process (scalar coding)



Bit rate (bit/symbol): $R = \sum_{a_i \in \mathcal{A}} p(a_i) l(a_i).$

Bound for Lossless Coding

- Scalar coding: $H_1(\mathcal{F}) \leq \bar{R}_1(\mathcal{F}) \leq H_1(\mathcal{F}) + 1.$
  - Assign one codeword to one symbol at a time
  - Problem: could differ from the entropy by up to 1 bit/symbol
- Vector coding:
  - Assign one codeword for each group of N symbols
  - Larger N -> Lower Rate, but higher complexity

$$H_N(\mathcal{F}) \leq R^N(\mathcal{F}) \leq H_N(\mathcal{F}) + 1 \qquad H_N(\mathcal{F})/N \leq \bar{R}_N(\mathcal{F}) \leq H_N(\mathcal{F})/N + 1/N.$$

$$\lim_{N \to \infty} \bar{R}_N(\mathcal{F}) = \bar{H}(\mathcal{F}).$$

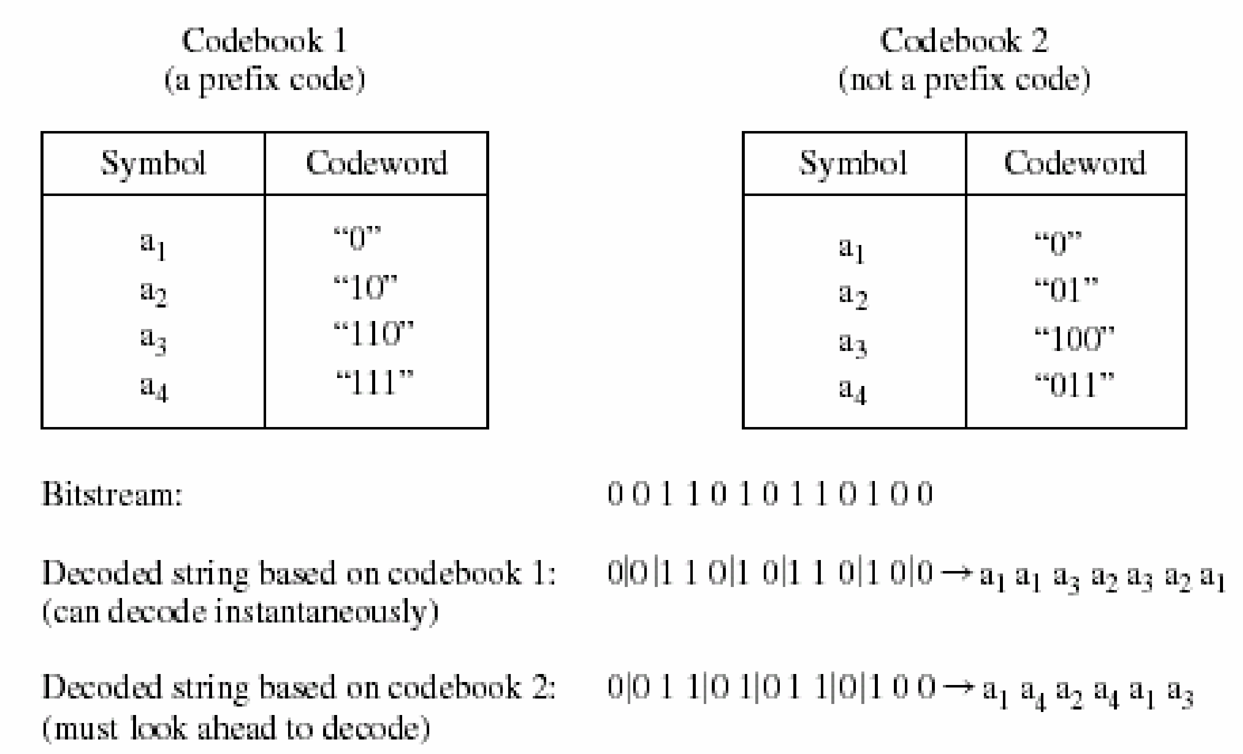
- Conditional coding (context-based coding)
  - The codeword for the current symbol depends on the pattern (context) formed by the previous M symbols

$$H^m_{C,M}(\mathcal{F}) \leq \bar{R}^m_{C,M}(\mathcal{F}) \leq H^m_{C,M}(\mathcal{F}) + 1, \qquad H_{C,M}(\mathcal{F}) \leq \bar{R}_{C,M}(\mathcal{F}) \leq H_{C,M}(\mathcal{F}) + 1.$$

$$\bar{H}(\mathcal{F}) \leq \lim_{M \to \infty} R_{C,M}(\mathcal{F}) \leq \bar{H}(\mathcal{F}) + 1.$$

Binary Encoding: Requirement

- A good code should be:
  - Uniquely decodable
  - Instantaneously decodable – prefix code



Codebook 1
(a prefix code)

Symbol	Codeword
a_1	"0"
a_2	"10"
a_3	"110"
a_4	"111"

Codebook 2
(not a prefix code)

Symbol	Codeword
a_1	"0"
a_2	"01"
a_3	"100"
a_4	"011"

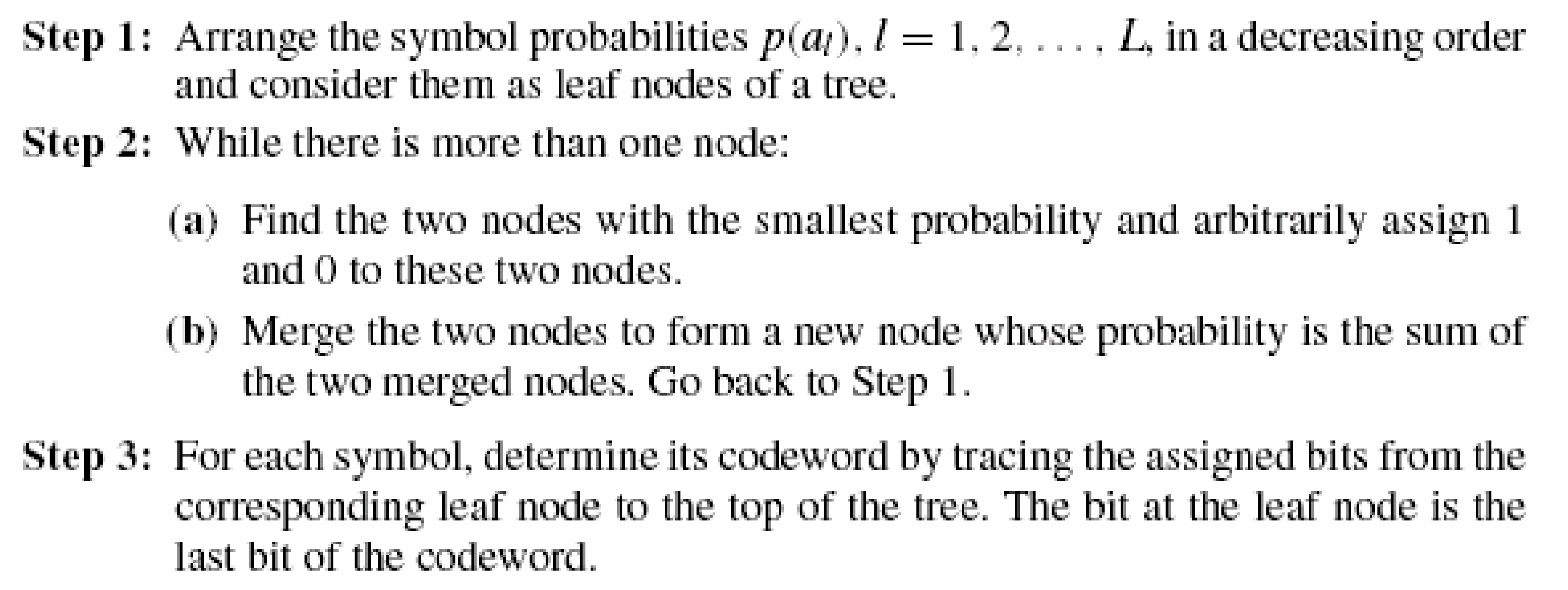
Bitstream: 0 0 1 1 0 1 0 1 1 0 1 0 0

Decoded string based on codebook 1:
(can decode instantaneously)
0|0|1 1 0|1 0|1 1 0|1 0|0 → $a_1\ a_1\ a_3\ a_2\ a_3\ a_2\ a_1$

Decoded string based on codebook 2:
(must look ahead to decode)
0|0 1 1|0 1|0 1 1|0|1 0 0 → $a_1\ a_4\ a_2\ a_4\ a_1\ a_3$

Huffman Coding

- Idea: more frequent symbols -> shorter codewords
- Algorithm:



Step 1: Arrange the symbol probabilities $p(a_l)$, $l = 1, 2, \ldots, L$, in a decreasing order and consider them as leaf nodes of a tree.

Step 2: While there is more than one node:

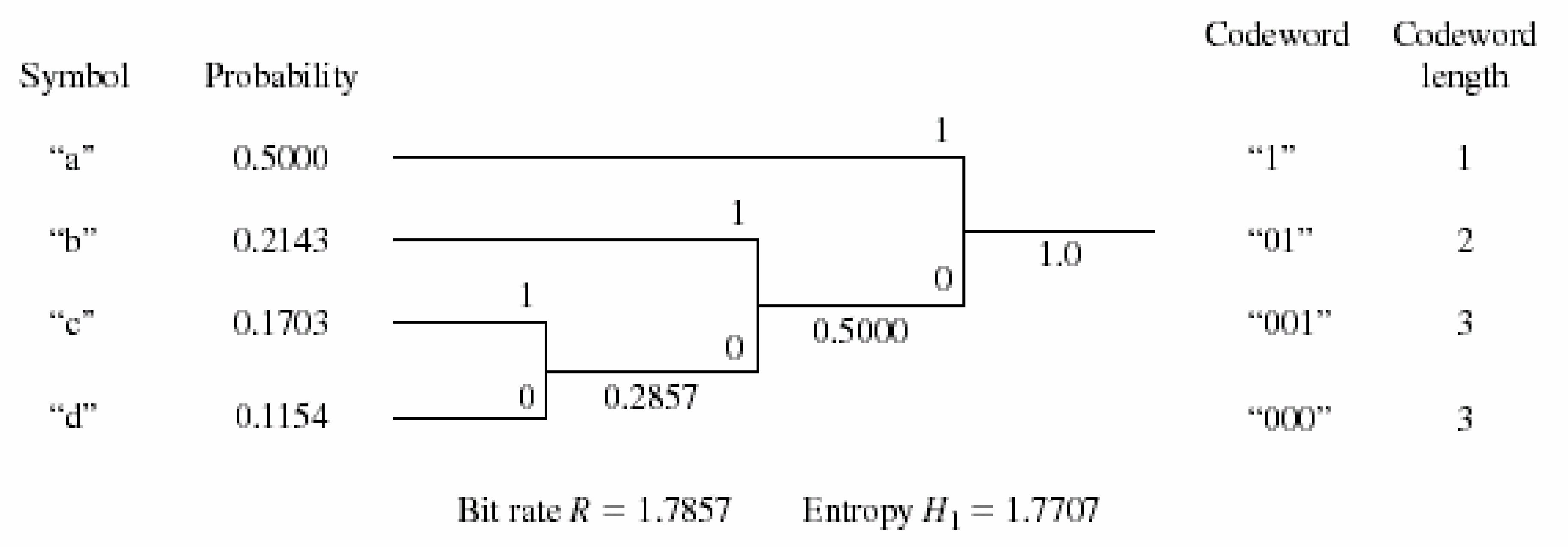
(a) Find the two nodes with the smallest probability and arbitrarily assign 1 and 0 to these two nodes.

(b) Merge the two nodes to form a new node whose probability is the sum of the two merged nodes. Go back to Step 1.

Step 3: For each symbol, determine its codeword by tracing the assigned bits from the corresponding leaf node to the top of the tree. The bit at the leaf node is the last bit of the codeword.

- Huffman coding generate prefix code ☺
- Can be applied to one symbol at a time (scalar coding), or a group of symbols (vector coding), or one symbol conditioned on previous symbols (conditional coding)

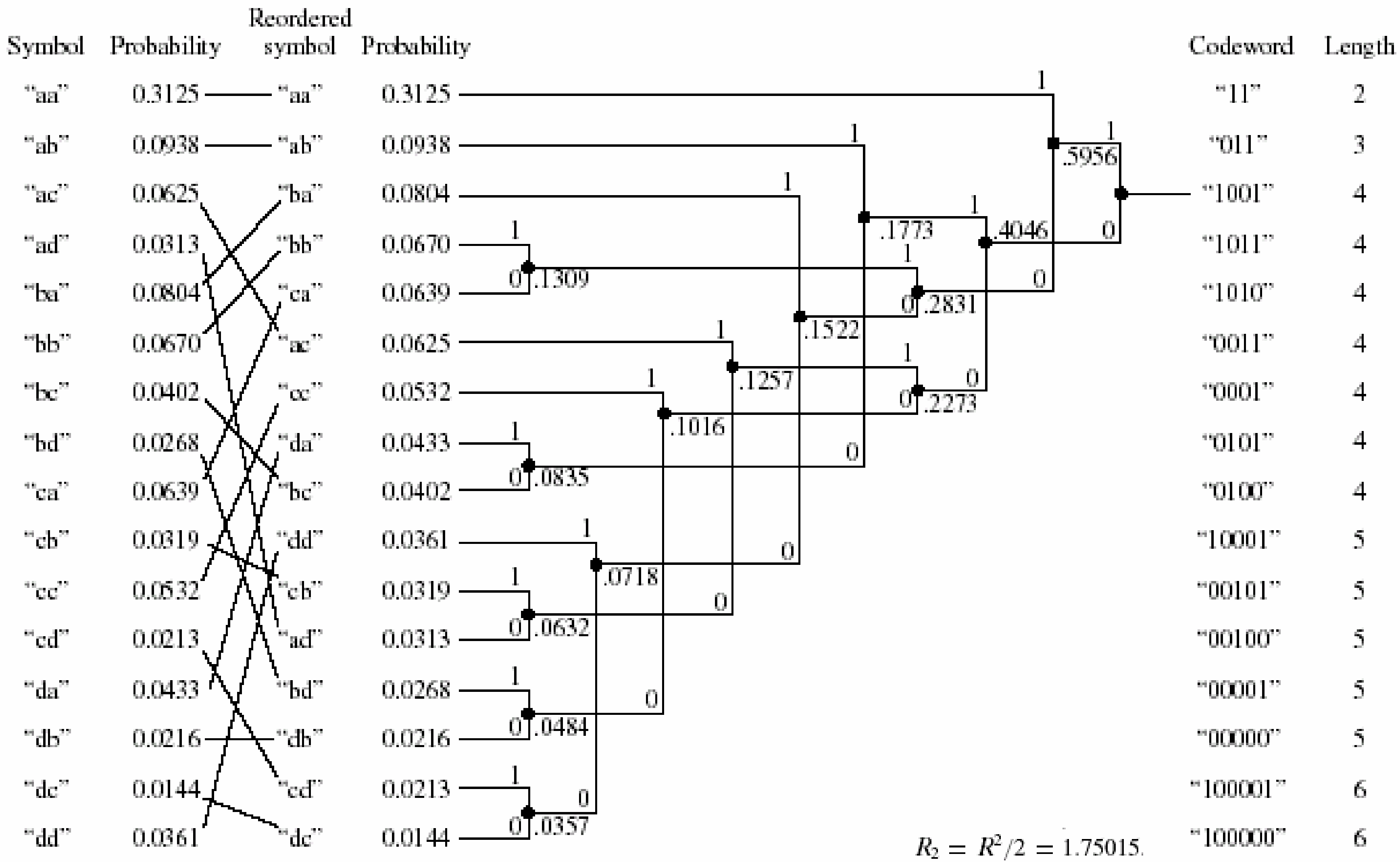
Huffman Coding Example: Scalar Coding



Symbol	Probability	Codeword	Codeword length
“a”	0.5000	“1”	1
“b”	0.2143	“01”	2
“c”	0.1703	“001”	3
“d”	0.1154	“000”	3

Bit rate $R = 1.7857$ Entropy $H_1 = 1.7707$

Huffman Coding Example: Vector Coding

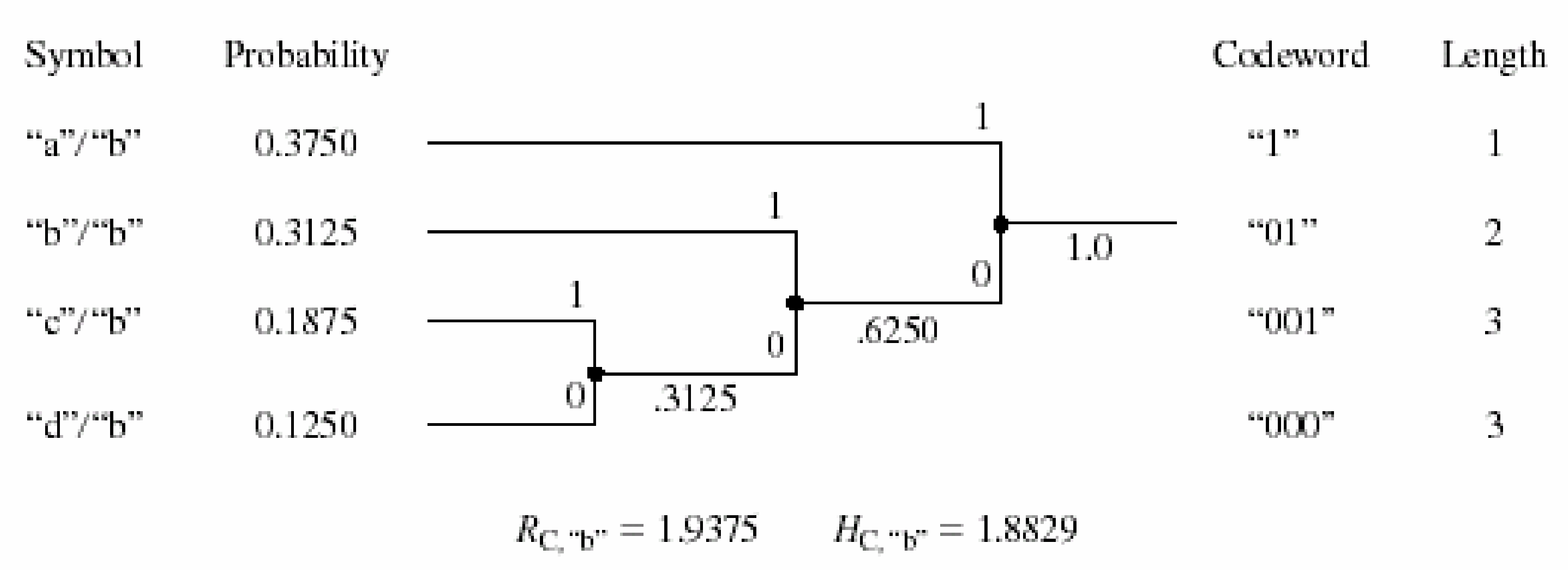


Symbol	Probability	Reordered symbol	Probability	Codeword	Length
"aa"	0.3125	"aa"	0.3125	"11"	2
"ab"	0.0938	"ab"	0.0938	"011"	3
"ac"	0.0625	"ba"	0.0804	"1001"	4
"ad"	0.0313	"bb"	0.0670	"1011"	4
"ba"	0.0804	"ca"	0.0639	"1010"	4
"bb"	0.0670	"ac"	0.0625	"0011"	4
"bc"	0.0402	"cc"	0.0532	"0001"	4
"bd"	0.0268	"da"	0.0433	"0101"	4
"ca"	0.0639	"bc"	0.0402	"0100"	4
"cb"	0.0319	"dd"	0.0361	"10001"	5
"cc"	0.0532	"cb"	0.0319	"00101"	5
"cd"	0.0213	"ad"	0.0313	"00100"	5
"da"	0.0433	"bd"	0.0268	"00001"	5
"db"	0.0216	"db"	0.0216	"00000"	5
"dc"	0.0144	"cd"	0.0213	"100001"	6
"dd"	0.0361	"dc"	0.0144	"100000"	6

$R_2 = R^2/2 = 1.75015.$

$R^2 = 3.5003$ $H_2 = 3.4629$

Huffman Coding Example: Conditional Coding



Symbol	Probability	Codeword	Length
“a”/“b”	0.3750	“1”	1
“b”/“b”	0.3125	“01”	2
“c”/“b”	0.1875	“001”	3
“d”/“b”	0.1250	“000”	3

$R_{C,\text{“b”}} = 1.9375 \qquad H_{C,\text{“b”}} = 1.8829$

$$R_{C,"a"} = 1.5625, R_{C,"b"} = R_{C,"c"} = R_{C,"d"} = 1.9375, R_{C,1} = 1.7500$$

$$H_{C,"a"} = 1.5016, H_{C,"b"} = H_{C,"c"} = H_{C,"d"} = 1.8829, H_{C,1} = 1.6922$$

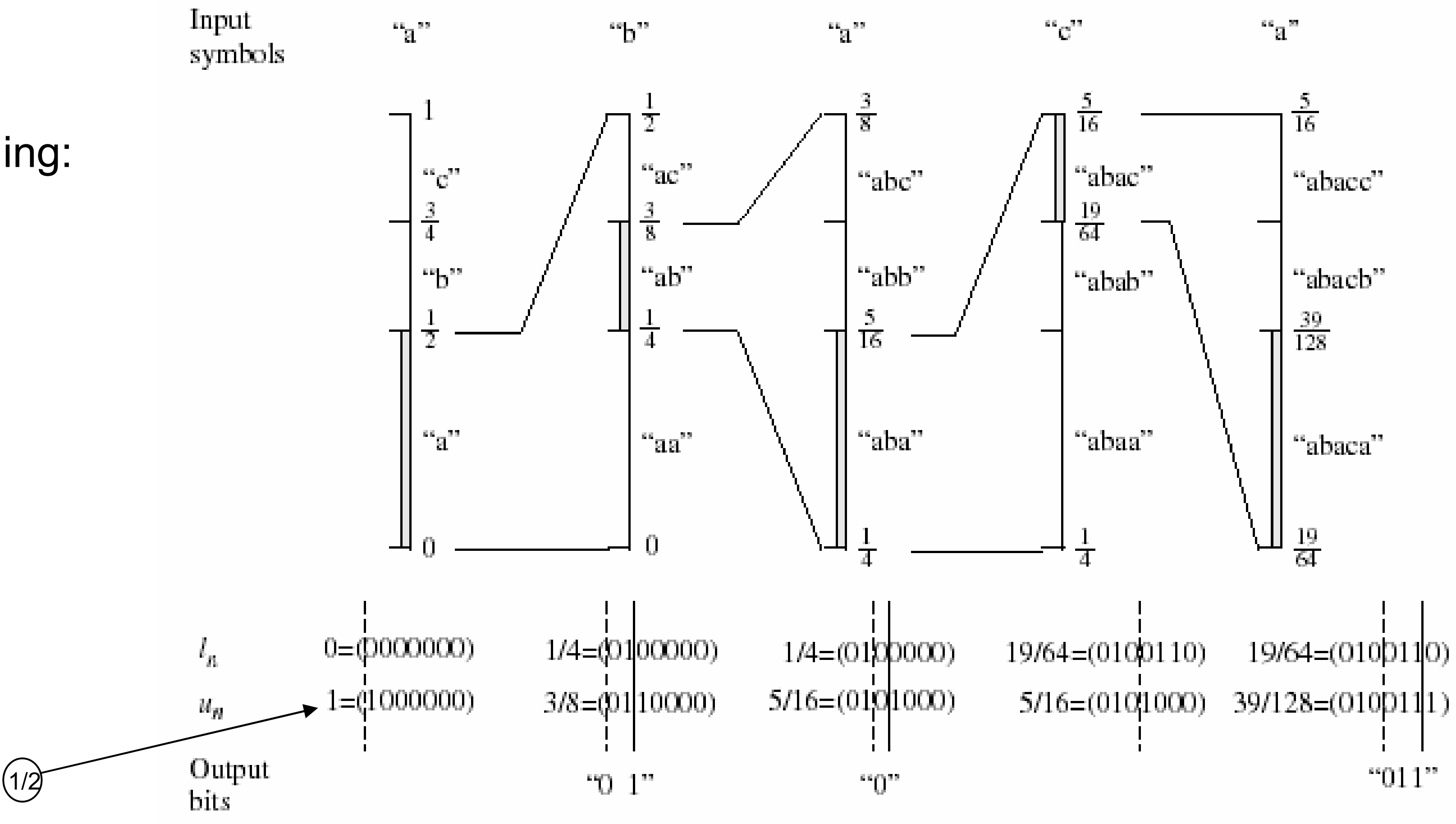
Arithmetic Coding

- Basic idea:
  - Represent a sequence of symbols by an interval with length equal to its probability
  - The interval is specified by its lower boundary (*l*), upper boundary (*u*) and length *d* (=probability)
  - The codeword for the sequence is the common bits in binary representations of *l* and *u*
  - *A more likely sequence=a longer interval=fewer bits*
- The interval is calculated sequentially starting from the first symbol
  - The initial interval is determined by the first symbol
  - The next interval is a subinterval of the previous one, determined by the next symbol

$$d_n = d_{n-1} * p_l; \quad l_n = l_{n-1} + d_{n-1} * q_{l-1}; \quad u_n = l_n + d_n.$$

P(a)=1/2
P(b)=1/4
P(c)=1/4

Encoding:



(a)

Decoding:

Received bits	Interval	Decoded symbol
"0"	[0,1/2)	"a"
"01"	[1/4,1/2)	—
"010"	[1/4,3/8)	"b"
"0100"	[1/4,5/16)	"a"
"01001"	[9/32,5/16)	—
"010011"	[19/64,5/16)	"c"
…	…	…

(b)

Implementation of Arithmetic Coding

- Previous example is meant to illustrate the algorithm in a conceptual level
  - Require infinite precision arithmetic
  - Can be implemented with finite precision or integer precision only
- For more details on implementation, see
  - Witten, Neal and Cleary, “Arithmetic coding for data compression”, J. ACM (1987), 30:520-40
  - Sayood, *Introduction to Data Compression*, Morgan Kaufmann, 1996

Huffman vs. Arithmetic Coding

- Huffman coding
  - Convert a fixed number of symbols into a variable length codeword
  - Efficiency:

$$H_N(\mathcal{F})/N \le \bar{R}_N(\mathcal{F}) \le H_N(\mathcal{F})/N + 1/N.$$

  - To approach entropy rate, must code a large number of symbols together
  - Used in all image and video coding standards

- Arithmetic coding
  - Convert a variable number of symbols into a variable length codeword
  - Efficiency:

$$H_N(\mathcal{F})/N \le R \le H_N(\mathcal{F})/N + 2/N,$$

N is sequence length

  - Can approach the entropy rate by processing one symbol at a time
  - Easy to adapt to changes in source statistics
  - Integer implementation is available, but still more complex than Huffman coding with a small N
    - Not widely adopted in the past (before year 2000)
  - Used as advanced options in image and video coding standards
  - Becoming standard options in newer standards (JPEG2000,H.264)

Summary

- Coding system:
  - original data -> model parameters -> quantization-> binary encoding
  - Waveform-based vs. content-dependent coding
- Characterization of information content by entropy
  - Entropy, Joint entropy, conditional entropy
  - Mutual information
- Lossless coding
  - Bit rate bounded by entropy rate of the source
  - Huffman coding:
    - Scalar, vector, conditional coding
    - can achieve the bound only if a large number of symbols are coded together
    - Huffman coding generates prefix code (instantaneously decodable)
  - Arithmetic coding
    - Can achieve the bound by processing one symbol at a time
    - More complicated than scalar or short vector Huffman coding

Homework

- Reading assignment:
  - Sec. 8.1-8.4 (excluding Sec. 8.3.2,8.8.8)
- Written assignment
  - Prob. 8.1,8.3,8.5,8.6,8.7
  - (for Prob. 8.7, you don't have to complete the encoding for the entire sequence. Do the first few letters.)