3F1 Information Theory, Lecture 2

Jossy Sayir

UNIVERSITY OF CAMBRIDGE
Department of Engineering

Michaelmas 2013, 27 November 2013

Variable length codes

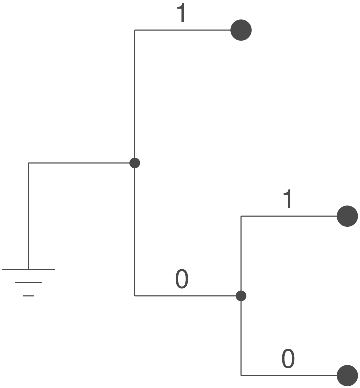
Variable length words in the English language

- and
- ophtalmologist
- to
- top
- tops
- topsy

Prefix Free Codes

Prefix Condition

No codeword may be the prefix of another.

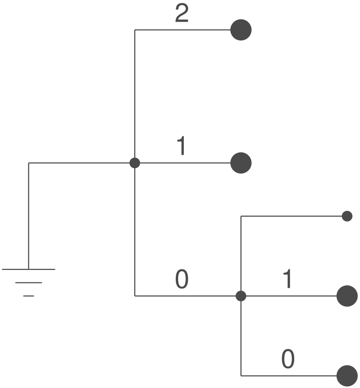


Source Symbol	Code-word
A	1
B	01
C	00

Prefix Free Codes

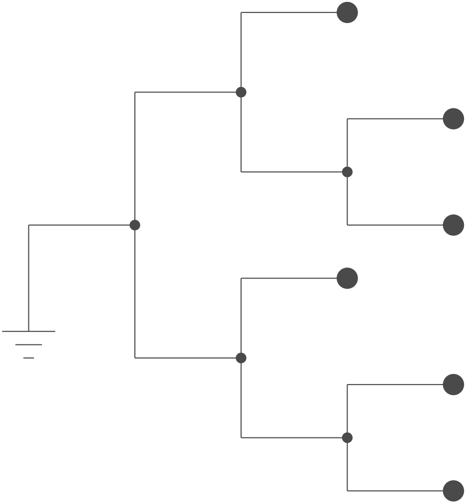
Prefix Condition

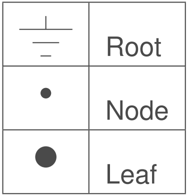
No codeword may be the prefix of another.



Source Symbol	Code-word
A	2
B	1
C	01
D	00

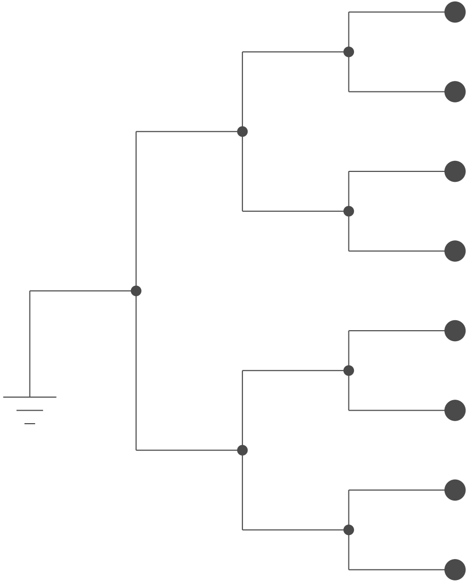
Rooted D-ary tree





Leaves = Codewords

Full Rooted *D*-ary tree of depth 3



Existence of a Prefix-Free Code

Kraft's Inequality

There exists a prefix-free code with lengths $w_1, w_2, \ldots, w_N$ if and only if

$$\sum_i D^{-w_i} \leq 1.$$

If the condition is satisfied with equality, no leaves are unused in the equivalent D-ary tree.

A DEVICE FOR QUANTIZING, GROUPING, AND CODING AMPLITUDE-MODULATED PULSES

by

Leon G. Kraft, Jr.

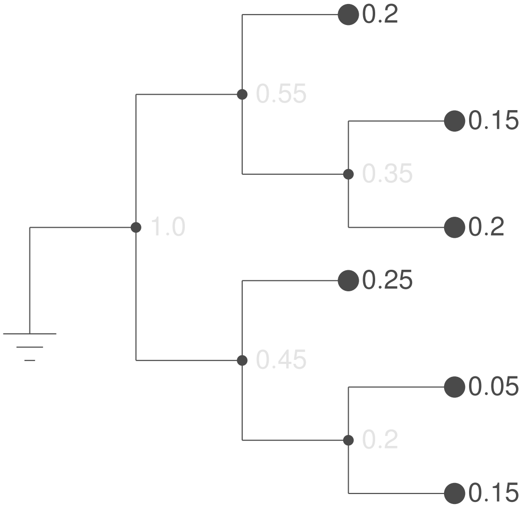
B.S. in E.E., University of Pennsylvania (1944)

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE

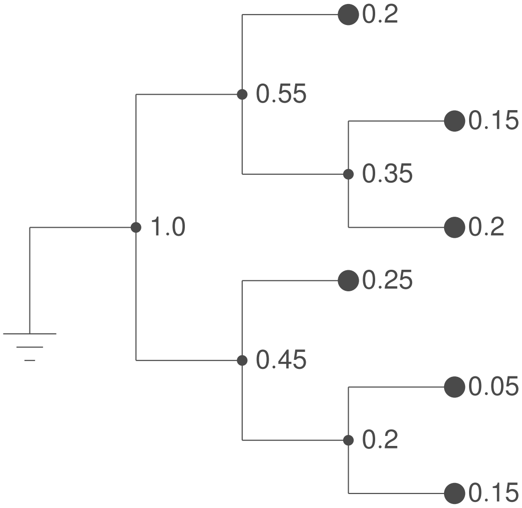
at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY (1949)

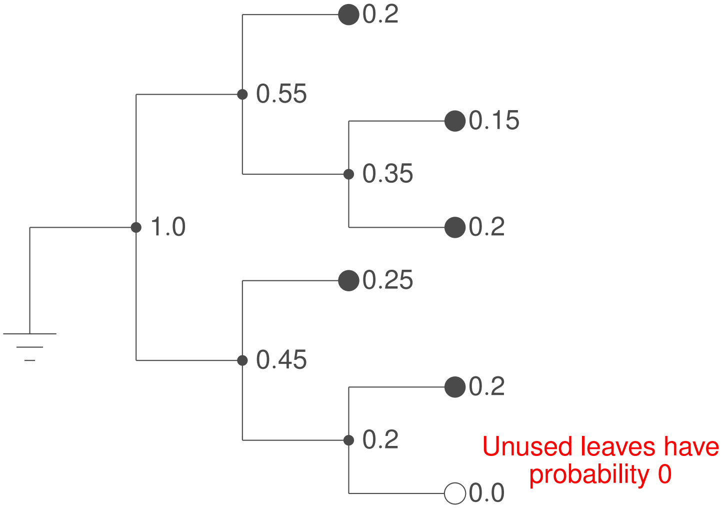
Rooted D-ary tree with probabilities



Rooted D-ary tree with probabilities



Rooted D-ary tree with probabilities



A few useful quantities

Average Codeword Length

$$L = E[W] = \sum_i p_i w_i$$

where p_i are the leaf (or source symbol) probabilities, w_i are the leaf depths (or codeword lengths), and W is a random variable that indicates the codeword length.

Leaf Entropy

$$H_{\text{leaf}} = H(X) = -\sum_i p_i \log p_i$$

is the leaf (or source) entropy.

A few useful results

Let $\tilde{p}_1, \tilde{p}_2, \ldots, \tilde{p}_K$ be the node probabilities (i.e., excluding the leaves, where $\tilde{p}_1 = 1.0$ always).

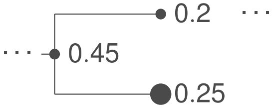
Path Length Lemma

$$L = E[W] = \sum_i \tilde{p}_i$$

Proof: a leaf at depth w_i with probability p_i contributes p_i to the sum probability of w_i nodes on the path from the leaf to the root, and hence its probability p_i is counted w_i times in the sum above, resulting in an equivalent expression for the average codeword length in the previous slide.

A few useful results

For any node, let its branching entropy H_i be the entropy of the probabilities of its children divided by its own node probability, e.g.



$$H_i = -\frac{0.2}{0.45}\log\frac{0.2}{0.45} - \frac{0.25}{0.45}\log\frac{0.25}{0.45}$$

Leaf Entropy Theorem

$$H(X) = H_{\text{leaf}} = \sum_i \tilde{p}_i H_i$$

A fundamental result (Shannon, 1948)

Converse Source Coding Theorem

$$L = E[W] \geq \frac{H(X)}{\log D}$$

You cannot design a prefix free code with an average length less than the entropy of the source.

Shannon-Fano Coding

Codeword lengths

$$w_i = \lceil -\log_D p_i \rceil$$

Feasibility

$$\sum_i D^{-w_i} \leq 1$$

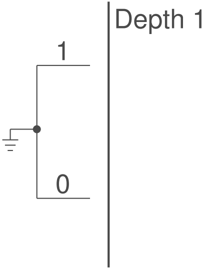
Performance

$$L = E[W] < \frac{H(X)}{\log D} + 1$$

Fano's version of S-F Coding

x	$P_X(x)$	$-\log P_X(x)$	$\lceil -\log P_X(x) \rceil$
A	.05	4.32	5
B	.10	3.32	4
C	.15	2.74	3
D	.20	2.32	3
E	.20	2.32	3
F	.30	1.74	2

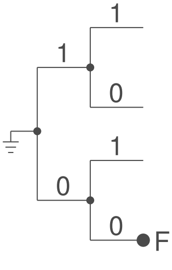
Robert M. Fano



Fano's version of S-F Coding

x	$P_X(x)$	$-\log P_X(x)$	$\lceil -\log P_X(x) \rceil$
A	.05	4.32	5
B	.10	3.32	4
C	.15	2.74	3
D	.20	2.32	3
E	.20	2.32	3
F	.30	1.74	2

Robert M. Fano

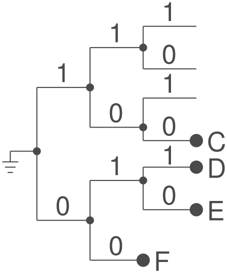


Depth 2

Fano's version of S-F Coding

x	$P_X(x)$	$-\log P_X(x)$	$\lceil -\log P_X(x) \rceil$
A	.05	4.32	5
B	.10	3.32	4
C	.15	2.74	3
D	.20	2.32	3
E	.20	2.32	3
F	.30	1.74	2

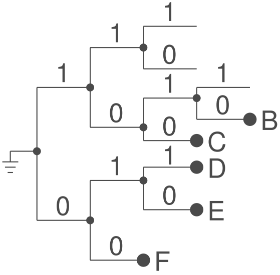
Robert M. Fano



Depth 3

Fano's version of S-F Coding

x	$P_X(x)$	$-\log P_X(x)$	$\lceil -\log P_X(x) \rceil$
A	.05	4.32	5
B	.10	3.32	4
C	.15	2.74	3
D	.20	2.32	3
E	.20	2.32	3
F	.30	1.74	2

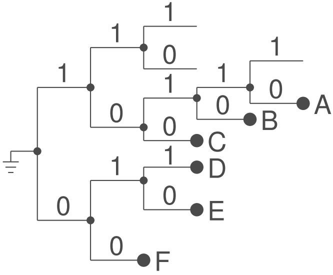


Depth 4

Robert M. Fano

Fano's version of S-F Coding

x	$P_X(x)$	$-\log P_X(x)$	$\lceil -\log P_X(x) \rceil$
A	.05	4.32	5
B	.10	3.32	4
C	.15	2.74	3
D	.20	2.32	3
E	.20	2.32	3
F	.30	1.74	2



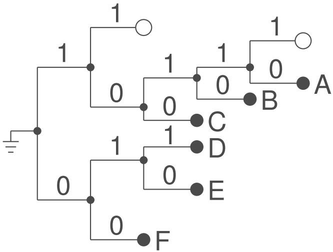
Depth 5

Robert M. Fano

Fano's version of S-F Coding

x	$P_X(x)$	$-\log P_X(x)$	$\lceil -\log P_X(x) \rceil$
A	.05	4.32	5
B	.10	3.32	4
C	.15	2.74	3
D	.20	2.32	3
E	.20	2.32	3
F	.30	1.74	2

Robert M. Fano



x	Codeword
A	10110
B	1010
C	100
D	011
E	010
F	00

Shannon's version of S-F Coding

x	$P_X(x)$	$P(X<x)$	$P(X<x)\vert_b$	$\lceil -\log P_X(x) \rceil$	Codeword
F	.30	0.0	0.00000000	2	00
D	.20	0.3	0.01001101	3	010
E	.20	0.5	0.10000000	3	100
C	.15	0.7	0.10110011	3	101
B	.10	0.85	0.11011010	4	1101
A	.05	0.95	0.11110011	5	11110

- Order the symbols in order of non-increasing probability
- Compute the cumulative probabilities
- Express the cumulative probabilities in D-ary
- The codeword is the fractional part of the cumulative probabilities truncated to length $\lceil -\log P_X(x) \rceil$

Shannon's version of S-F Coding

x	$P_X(x)$	$P(X<x)$	$P(X < x)\|_b$	$\lceil -\log P_X(x) \rceil$	Codeword
F	.30	0.0	0.00000000	2	00
D	.20	0.3	0.01001101	3	010
E	.20	0.5	0.10000000	3	100
C	.15	0.7	0.10110011	3	101
B	.10	0.85	0.11011010	4	1101
A	.05	0.95	0.11110011	5	11110

- Order the symbols in order of non-increasing probability
- Compute the cumulative probabilities
- Express the cumulative probabilities in D-ary
- The codeword is the fractional part of the cumulative probabilities truncated to length $\lceil -\log P_X(x) \rceil$

Shannon's version of S-F Coding

x	$P_X(x)$	$P(X<x)$	$P(X < x)\|_b$	$\lceil -\log P_X(x) \rceil$	Codeword
F	.30	0.0	0.00000000	2	00
D	.20	0.3	0.01001101	3	010
E	.20	0.5	0.10000000	3	100
C	.15	0.7	0.10110011	3	101
B	.10	0.85	0.11011010	4	1101
A	.05	0.95	0.11110011	5	11110

- Order the symbols in order of non-increasing probability
- Compute the cumulative probabilities
- Express the cumulative probabilities in D-ary
- The codeword is the fractional part of the cumulative probabilities truncated to length $\lceil -\log P_X(x) \rceil$

Shannon's version of S-F Coding

x	$P_X(x)$	$P(X<x)$	$P(X<x)\|_b$	$\lceil -\log P_X(x) \rceil$	Codeword
F	.30	0.0	0.00000000	2	00
D	.20	0.3	0.01001101	3	010
E	.20	0.5	0.10000000	3	100
C	.15	0.7	0.10110011	3	101
B	.10	0.85	0.11011010	4	1101
A	.05	0.95	0.11110011	5	11110

- Order the symbols in order of non-increasing probability
- Compute the cumulative probabilities
- Express the cumulative probabilities in D-ary
- The codeword is the fractional part of the cumulative probabilities truncated to length $\lceil -\log P_X(x) \rceil$

The Huffman Story (Scientific American, Sept. 1991)

In 1951 David A. Huffman and his classmates in an electrical engineering graduate course on information theory were given the choice of a term paper or a final exam. For the term paper, Huffman's professor, Robert M. Fano, had assigned what at first appeared to be a simple problem. Students were asked to find the most efficient method of representing numbers, letters or other symbols using a binary code. [. . .] Huffman worked on the problem for months, developing a number of approaches, but none that he could prove to be the most efficient. Finally, he despaired of ever reaching a solution and decided to start studying for the final. Just as he was throwing his notes in the garbage, the solution came to him. "It was the most singular moment of my life," Huffman says. "There was the absolute lightning of sudden realization."[. . .] Huffman says he might never have tried his hand at the problem — much less solved it at the age of 25 — if he had known that Fano, his professor, and Claude E. Shannon, the creator of information theory, had struggled with it. "It was my luck to be there at the right time and also not have my professor discourage me by telling me that other good people had struggled with this problem," he says.

Properties of the optimal binary code

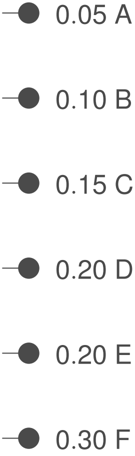
Unused Leaves

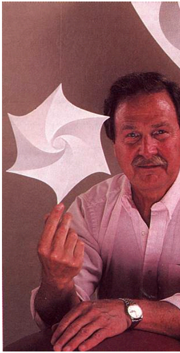
The optimal binary prefix tree has no unused leaves.

Least probable symbols

The two least probable symbols are on neighbouring leaves in the optimal binary tree.

Huffman Coding





David A. Huf

Huffman Coding

0.05 A

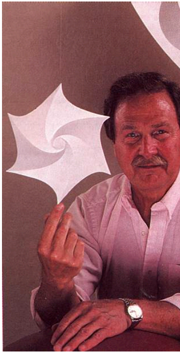
0.10 B

0.15 C

0.20 D

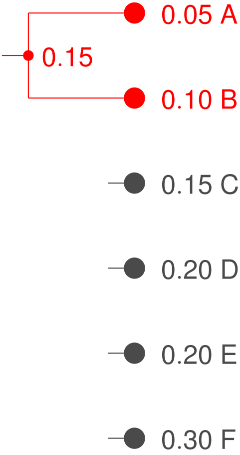
0.20 E

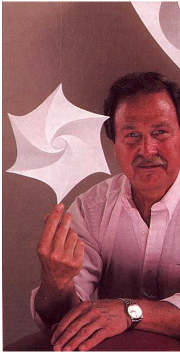
0.30 F



David A. Huff

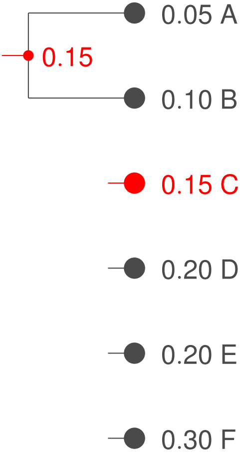
Huffman Coding

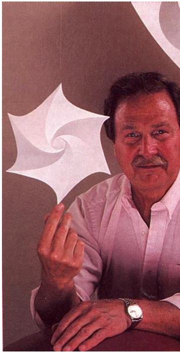




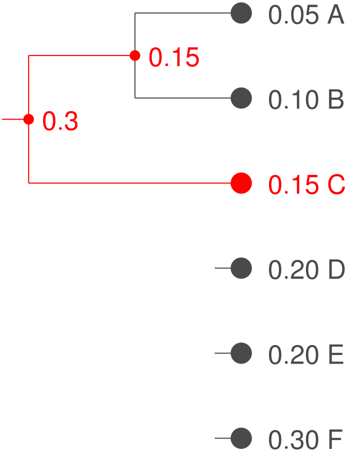
David A. Huf

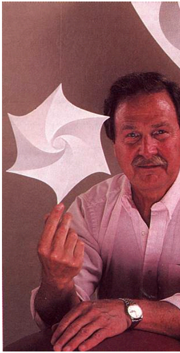
Huffman Coding




David A. Huf

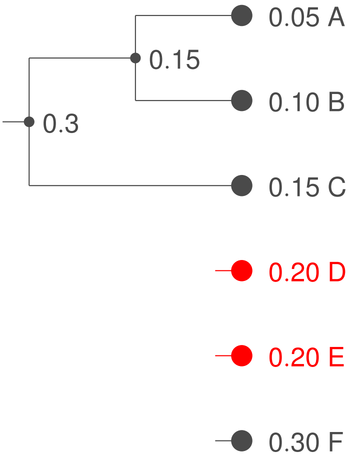
Huffman Coding

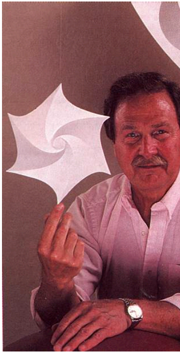




David A. Huf

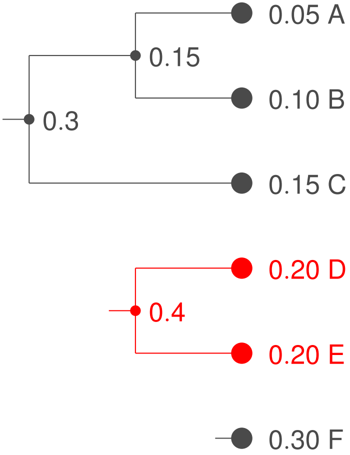
Huffman Coding

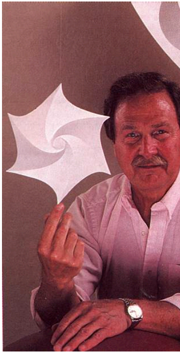




David A. Huf

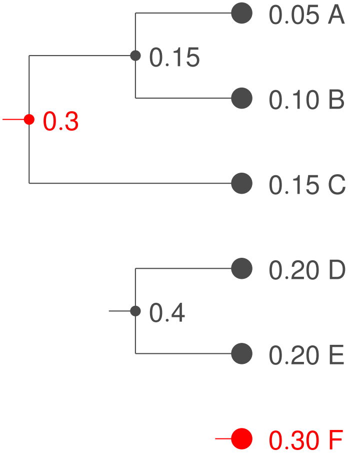
Huffman Coding

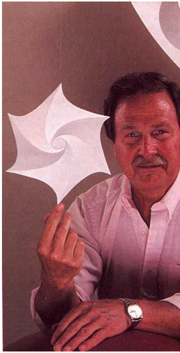




David A. Huff

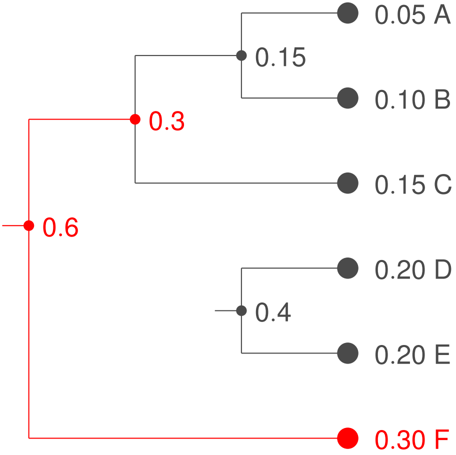
Huffman Coding

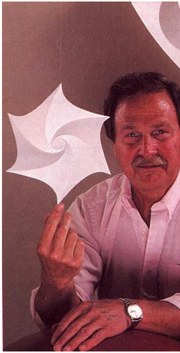




David A. Huf

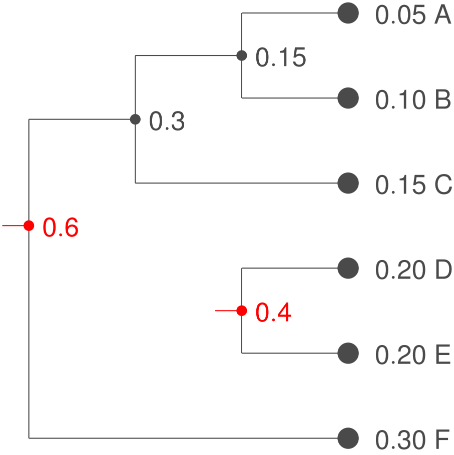
Huffman Coding

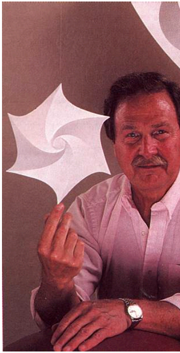




David A. Huf

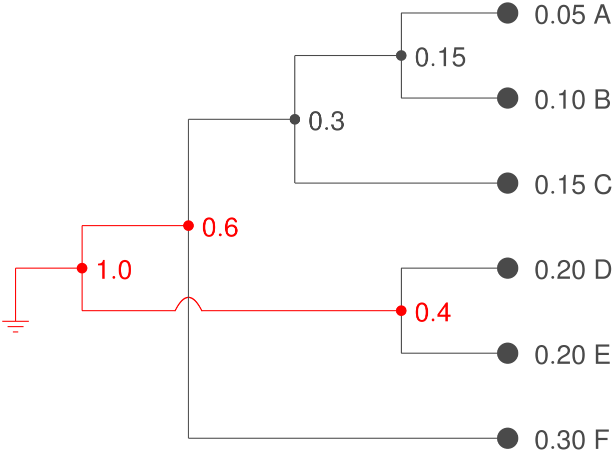
Huffman Coding

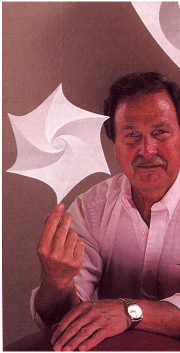




David A. Huf

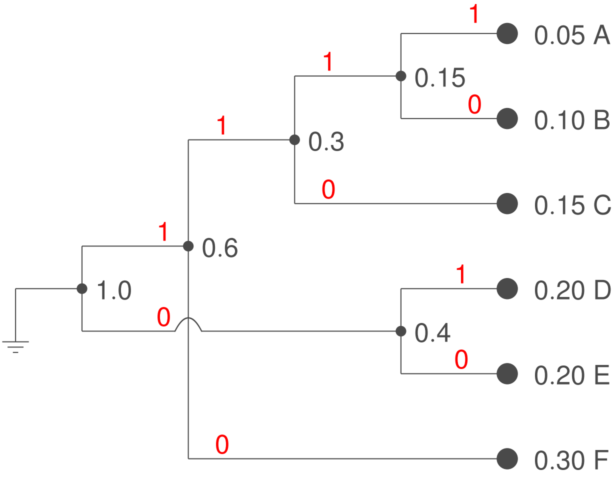
Huffman Coding

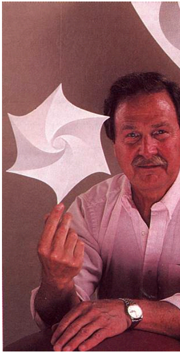




David A. Huf

Huffman Coding





David A. Huf