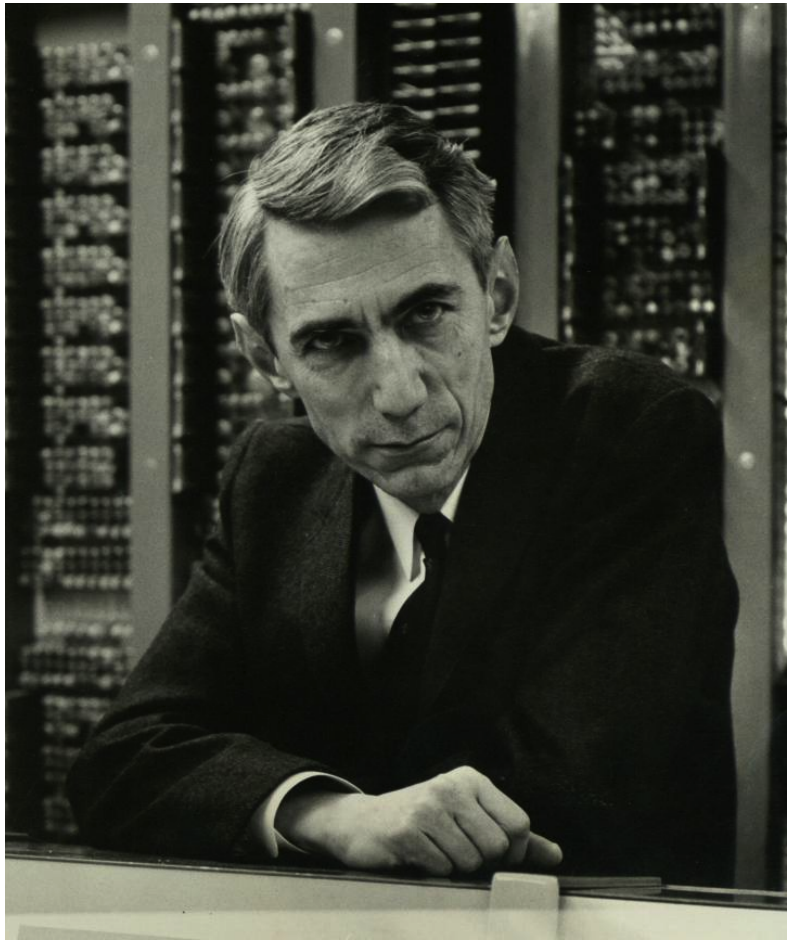
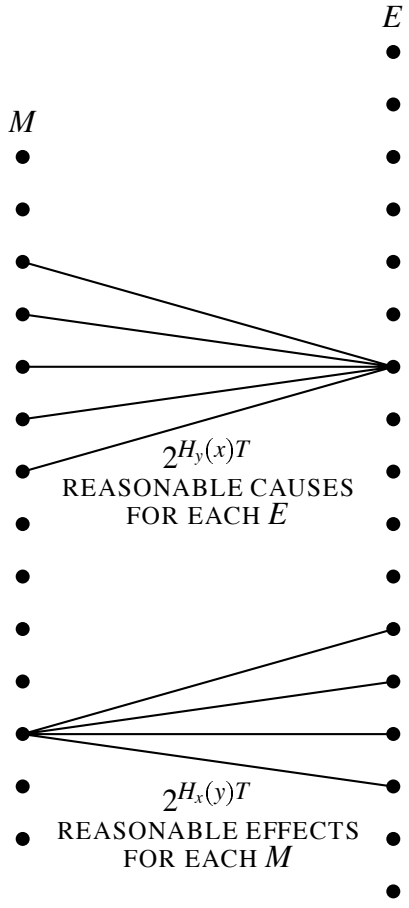
Information Theory
MONDAY

Mathias Winther Madsen
mathias@gmail.com
github.com/mathias-madsen/nasslli2025/

NASSLLI, June 2025

Information Theory





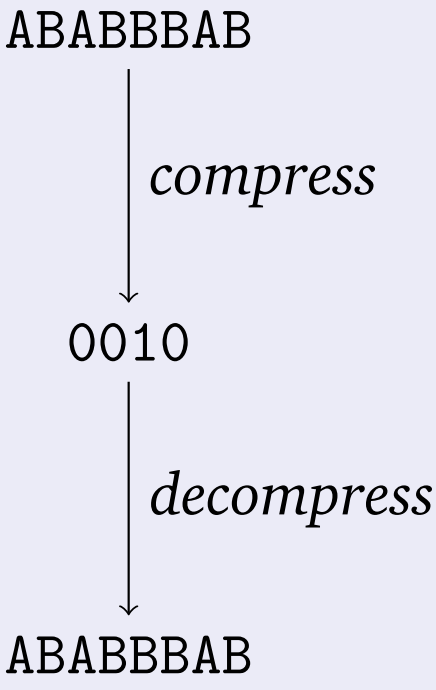
Claude Shannon: "A Mathematical Theory of Communication,"
Bell System Technical Journal, 1948.

Information Theory

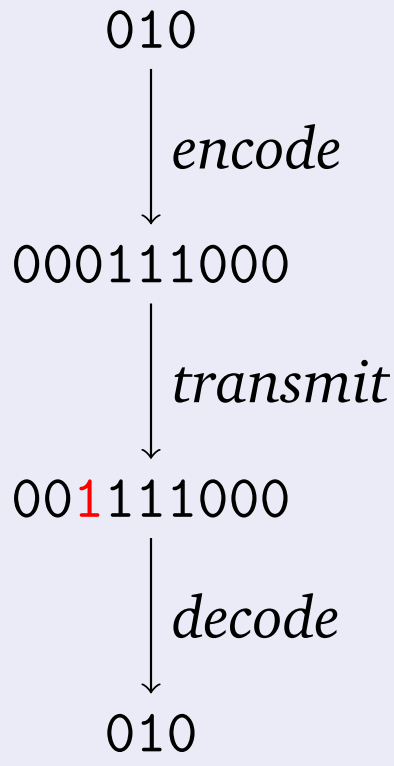
THE CHIEF DIFFICULTY ALICE FOUND AT ▉IRST WAS IN
MANAG▉NG ▉ER FLA▉INGO: SHE SUCCEEDED IN GET▉ING I▉▉▉
BOD▉ TUCKED ▉WAY, C▉MFORTABLY ENOU▉▉▉, UN▉E▉ HER
AR▉, WITH IT▉ LEGS HANG▉NG ▉OWN, BU▉ GENE▉A▉L▉,
▉U▉T ▉▉▉ S▉E H▉▉▉ ▉OT ▉▉S ▉▉CK NI▉ELY
▉T▉AI▉▉▉▉▉NE▉ ▉UT, AND ▉AS G▉ING T▉ G▉VE THE
HEDGEHO▉ ▉ B▉▉▉▉▉ ▉I▉▉▉ I▉▉▉ H▉▉D, ▉T W▉U▉D ▉W▉ST
I▉SEL▉ R▉UND ▉▉▉D L▉▉▉▉▉ ▉▉▉ ▉N H▉R ▉▉▉C▉, WIT▉ S▉CH
▉ ▉▉▉▉Z▉E▉ ▉▉▉PR▉▉▉▉▉▉▉N ▉▉▉▉▉T ▉▉▉▉▉ ▉O▉▉▉D NO▉ ▉▉▉LP
▉U▉▉▉T▉▉▉▉ O▉▉▉ ▉▉▉▉GH▉▉▉▉: ▉▉▉▉▉ ▉▉▉▉▉N ▉▉▉E HA▉ ▉▉▉T
▉▉▉▉▉ ▉E▉▉▉ ▉▉▉W▉, ▉ND ▉▉▉▉▉ ▉▉▉I▉▉▉ ▉▉▉ ▉E▉▉▉N ▉▉▉▉▉▉N,
IT ▉▉▉▉▉ ▉▉▉R▉ ▉R▉▉▉▉▉▉I▉▉▉ ▉▉▉ ▉▉▉N▉ T▉▉▉▉▉ T▉▉▉
▉EDG▉▉▉O▉ ▉▉▉▉▉ ▉▉▉▉▉▉L▉▉▉D ▉▉▉SE▉▉▉, ▉▉▉D ▉A▉ ▉▉▉ ▉▉▉▉▉
▉▉▉▉▉ ▉▉▉ ▉R▉W▉▉▉▉▉▉ ▉▉▉▉▉▉.

Information Theory

Source Coding



Channel Coding



$\longrightarrow$ source **entropy**

$\longrightarrow$ channel **capacity**

Information Theory

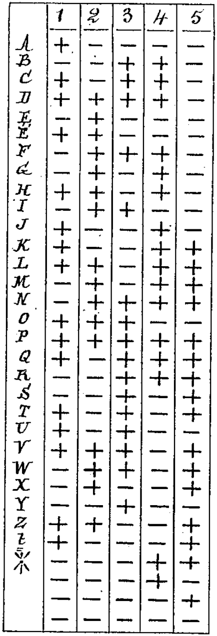
Monday	Data compression
Tuesday	Asymptotic equipartition
Wednesday	Random processes
Thursday	Codebreaking
Friday	Error-correcting codes

AS WE SAT OVER OUR VERMOUTHS HE GLORIFIED THE COMPANY'S BUSINESS, AND BY AND BY I EXPRESSED CASUALLY MY SURPRISE AT HIM NOT GOING OUT THERE.

101001001111101000011110100101011001111000011100
100111011111100000000110111110111001001110111101
011000000001100000101001110001001111101101100101
000110110101101111010100110011111110000010011111
011101000110101011101101001100111111011100000100 1
110000100000001000101011000101000100011101111010 1
001101001111100001001111111101001101001111100000
100111111110101111001101100101101110111011001010
001000011001111110111010100100000010101001010 01
001111111111010101111111010000000110110111011 10
110101010000111110101100111000101011101011101 10
100011001111011011000010101101011011111000000001
100111110000010011101100100001 1

Fixed-Width Codes

Jean Maurice
Emilé Baudot:
“Printing Telegraph,”
US Patent 388244,
1888.



ABC $\longleftrightarrow$ 100000011010110

Variable-Width Codes

r = *a*	*rlr* = *f*, *v*	*rrlr* = *s*	*lrlr* = 3
l = *e*	*lrr* = *g*	*rlrr* = *t*	*llrr* = 4
rr = *i*	*lll* = *h*	*lrrr* = *w*	*lllr* = 5
rl = *o*	*llr* = *l*	*rrll* = *x*	*llrl* = 6
lr = *u*	*lrl* = *m*	*rlrl* = 0	*lrll* = 7
ll = *b*	*rll* = *n*	*rllr* = 1	*rlll* = 8
rrr = *c*, *k*	*rrrr* = *p*	*lrrl* = 2	*llll* = 9
rrl = *d*	*rrrl* = *r*		

HELLO ⟷ LLL L LLR LLR RL

The Gauss-Weber code; from John Jacob Fahie:
A History of Electric Telegraphy, to the Year 1837, 1884.

Variable-Width Codes

A	B	C
0_	1_	00_

ABACBB $\longleftrightarrow$ 0_1_00_1_1_

A	B	C
0	1	_

ABACBB $\longleftrightarrow$ 010_11

Unique Decodability

Definition

A set of codewords is **uniquely decodable** if no two sequences of codewords produce identical texts.

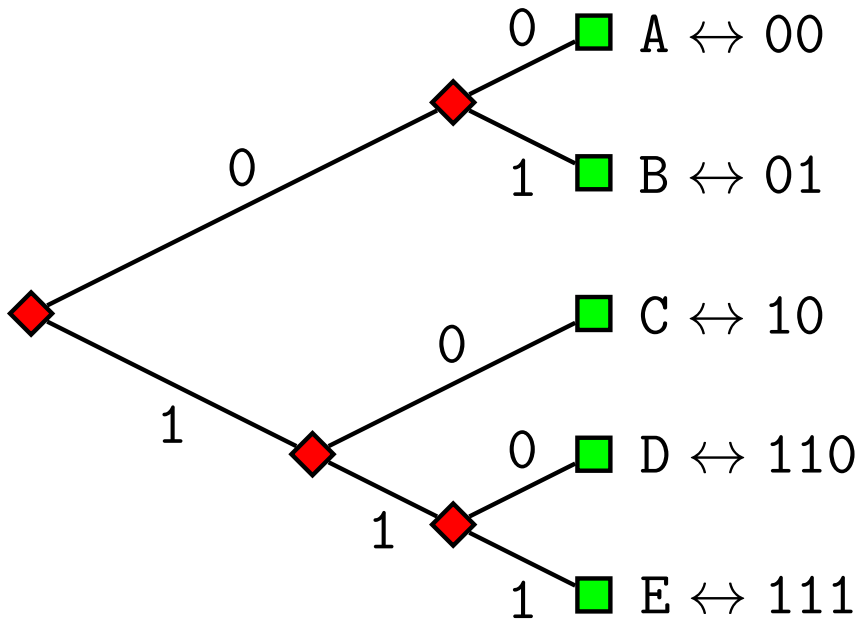
Exercises

1. $\{00, 0, 1\}$
2. $\{0, 10, 11\}$
3. $\{0, 01, 11\}$
4. $\{01, 10, 1001\}$
5. $\{00, 1100, 1111\}$
6. $\{02, 12, 002, 012, 102, 112\}$

Prefix Codes

Definition

A code is a **prefix code** if no codeword is a prefix of another.

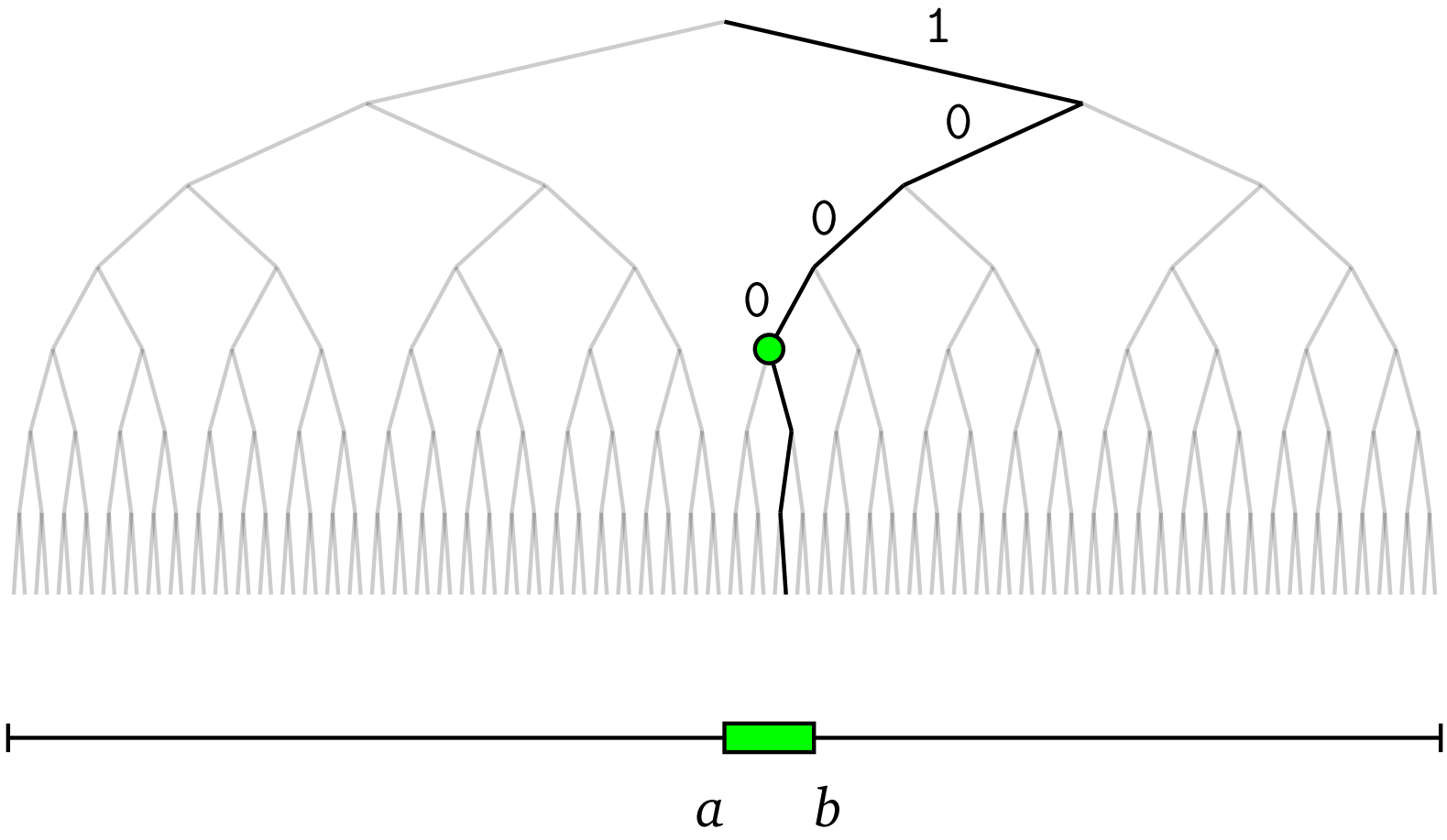


Exercises

1. $\{0, 10, 11\}$
2. $\{0, 1, 11\}$
3. $\{0, 1\}$
4. $\{02, 12, 012\}$
5. $\{0, 01\}$

$$\texttt{ABEADCB} \longleftrightarrow \texttt{0001111001101001}$$

Strings $\sim$ Paths $\sim$ Intervals



$$
\begin{aligned}
a &= s_1/2 + s_2/4 + s_3/8 + \cdots + s_w/2^w \\
b &= a + 1/2^w
\end{aligned}
$$

Kraft-McMillan

Theorem: Kraft-McMillan Theorems

There is a uniquely decodable binary code with codeword lengths $w_1, w_2, \ldots, w_k$ if and only if

$$\underbrace{\left(\frac{1}{2}\right)^{w_1} + \left(\frac{1}{2}\right)^{w_2} + \cdots + \left(\frac{1}{2}\right)^{w_k}}_{\text{code “footprint”}} \leq 1$$

Exercises

1. $\{0, 1\}$
2. $\{0, 01\}$
3. $\{0, 1, 11\}$
4. $\{0, 10, 11\}$
5. $\{0, 0\}$

Leon G. Kraft: *A Device for Quantizing, Grouping, and Coding Amplitude Modulated Pulses*, MIT master’s thesis, 1949.

Brockway McMillan: “Two Inequalities Implied by Unique Decipherability,” *IEEE Transactions on Information Theory*, 1956.

Kraft's Inequality

Kraft Codes

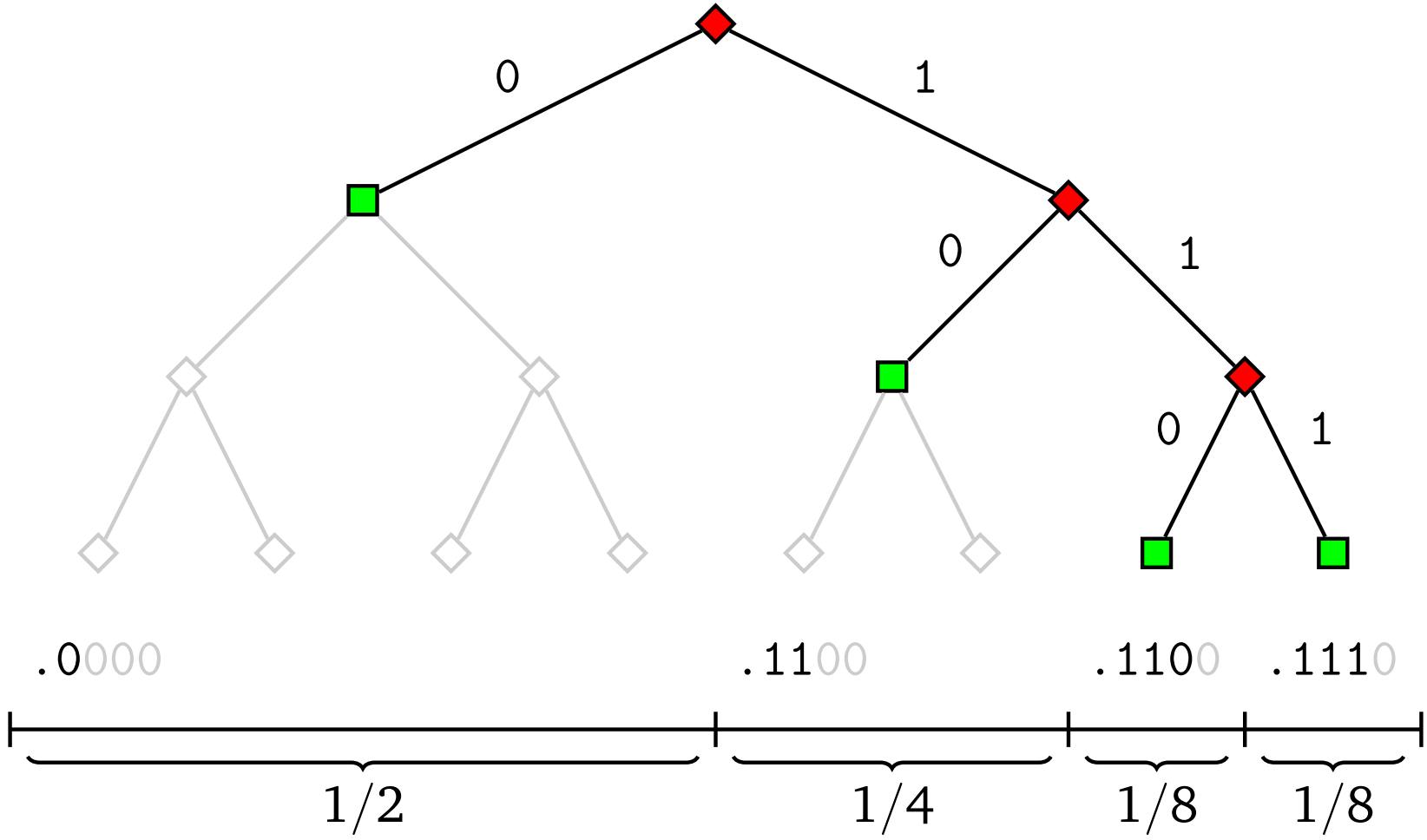
Pick codewords of length $w_1 \geq w_2 \geq \cdots \geq w_k$, always using the alphabetically first codeword not blocked by a previous codeword.

Exercises

1. $1 \geq 2 \geq 2$
2. $2 \geq 2 \geq 3 \geq 3 \geq 3 \geq 4 \geq 4$
3. $2 \geq 2 \geq 3 \geq 3 \geq 3 \geq 3$

Kraft's Inequality

First available codewords of lengths $1 \leq 2 \leq 3 \leq 3$:



McMillan's Theorem: Small Footprints

Multiword sentences

Let $C^{\otimes n}$ be the set of all sentences of n codewords from C.

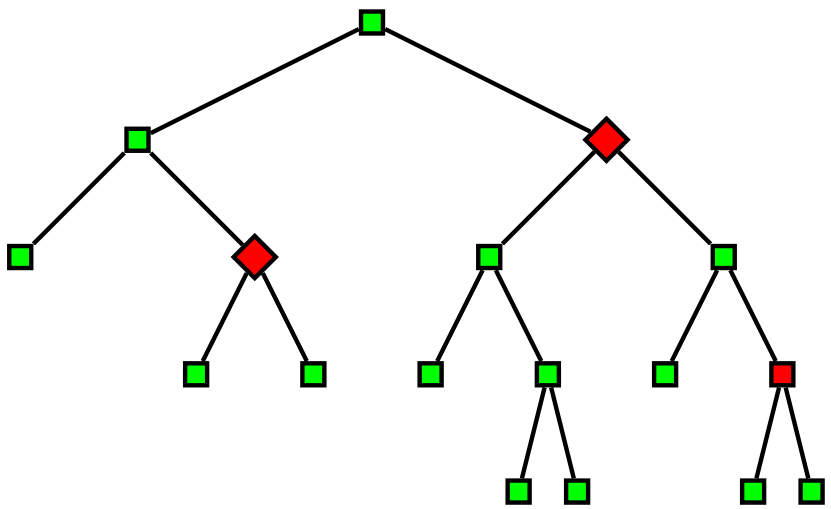
$$
\begin{aligned}
\{0, 10\}^{\otimes 1} &= \{0, 01, 10\} \\
\{0, 10\}^{\otimes 2} &= \{00, 010, 100, 1010\} \\
\{0, 10\}^{\otimes 3} &= \{000, 0010, 0100, 01010, \\
&\quad\ 1000, 10010, 10100, 101010, \\
&\quad\ 1100, 11010, 11100, 111010\} \\
&\ \ \vdots
\end{aligned}
$$

McMillan's Theorem: Small Footprints

Lemma: Linear Upper Bound

If C is uniquely decodable with $w_1, w_2, \ldots, w_k \leq w^*$, then

$$F(C^{\otimes n}) \;\leq\; 1 + nw^*$$



$F_0 \leq 1$

$F_1 \leq 1$

$F_2 \leq 1$

$\vdots$

$F_{nw^*} \leq 1$

McMillan's Theorem: Large Footprints

Lemma: Sentence Footprint Exponential in Code Footprint

If C is uniquely decodable, then

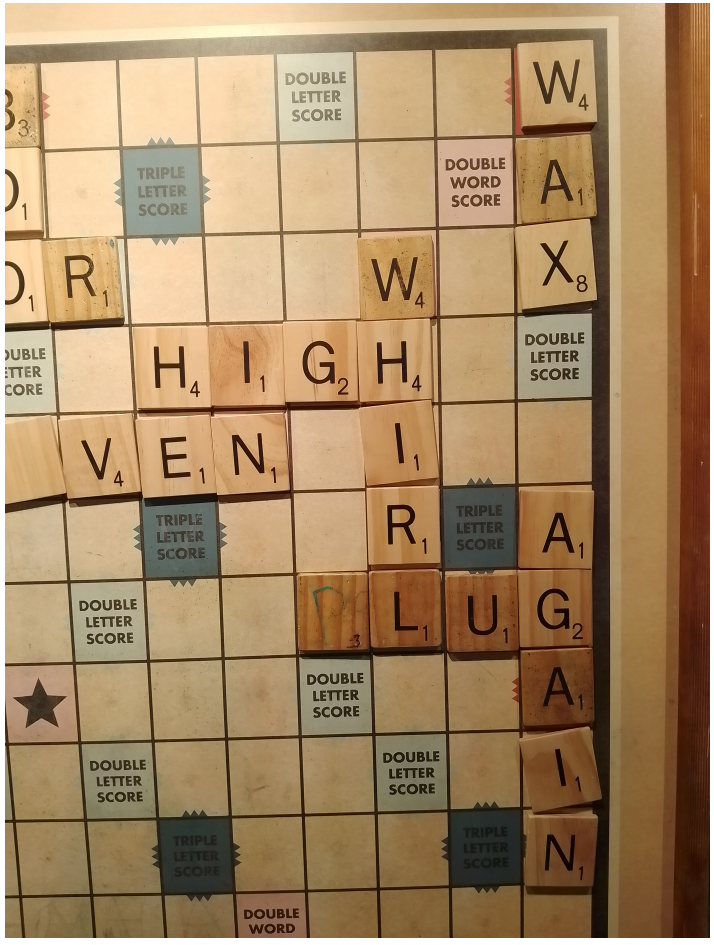
$$F(C^{\otimes n}) \;=\; F(C)^n$$

Proof: Listing sentences $\sim$ combining codewords

$$\sum_{i=1}^{k}\sum_{j=1}^{k} 2^{-(w_i+w_j)} \;=\; \left(\sum_{i=1}^{k} 2^{-w_i}\right) \times \left(\sum_{j=1}^{k} 2^{-w_j}\right)$$

Least Average Codeword Length

With letter probabilities $p_1, p_2, \ldots, p_k$:

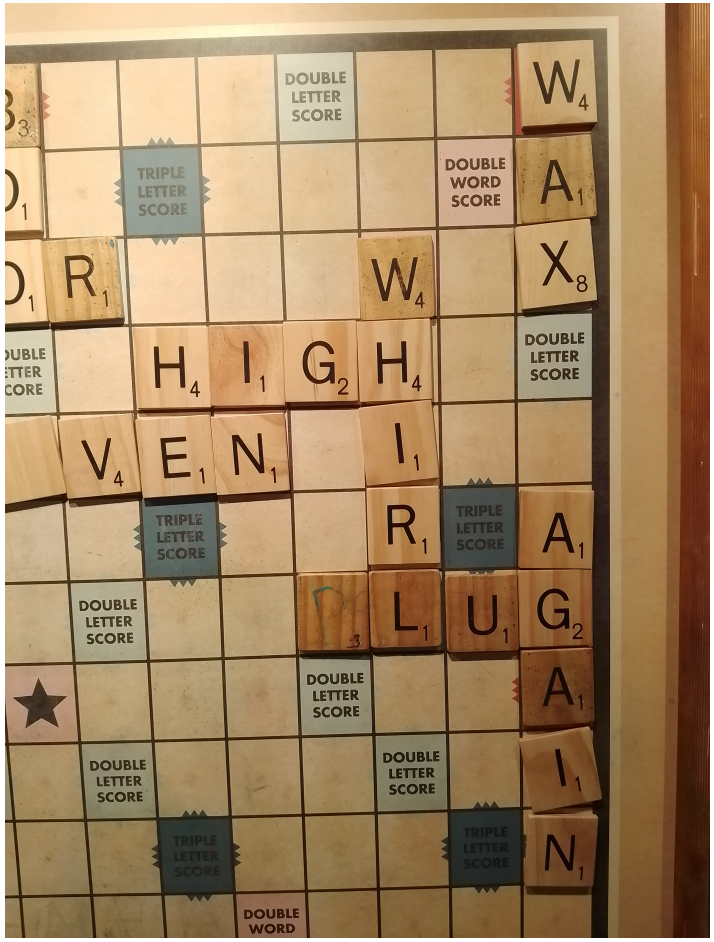


$$\text{minimize} \sum_{i=1}^{k} p_i w_i$$

$$\text{subject to} \sum_{i=1}^{k} 2^{-w_i} \leq 1$$

Least Average Codeword Length

With letter probabilities $p_1, p_2, \ldots, p_k$:



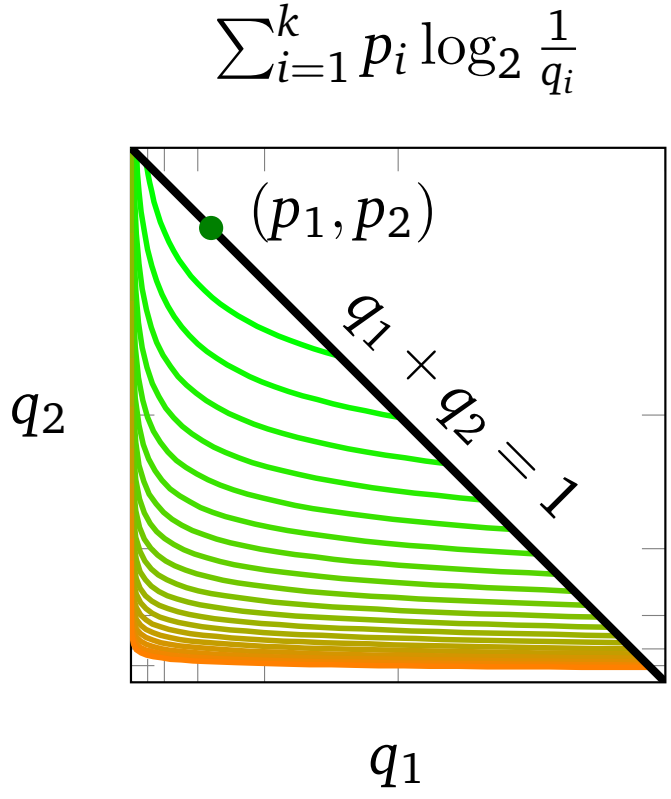
$$\text{minimize} \sum_{i=1}^{k} p_i \log_2 \frac{1}{q_i}$$

$$\text{subject to} \sum_{i=1}^{k} q_i \leq 1$$

Least Average Codeword Length

Solution: $q_i^* = p_i$,
with value

$$H = \sum_{i=1}^{k} p_i \log_2 \frac{1}{q_i}$$



Entropy

Definition

$$H = \sum_{i=1}^{k} p_i \log_2 \frac{1}{p_i}$$

Claude Shannon: "A Mathematical Theory of Communication," *Bell System Technical Journal*, 1948.

Entropy

Definition

$$H = \sum_{i=1}^{k} p_i \log_2 \frac{1}{p_i}.$$

1.

x	A	B
$p(x)$	$1/2$	$1/2$

2.

x	A	B	C	D
$p(x)$	$1/2$	$1/4$	$1/8$	$1/8$

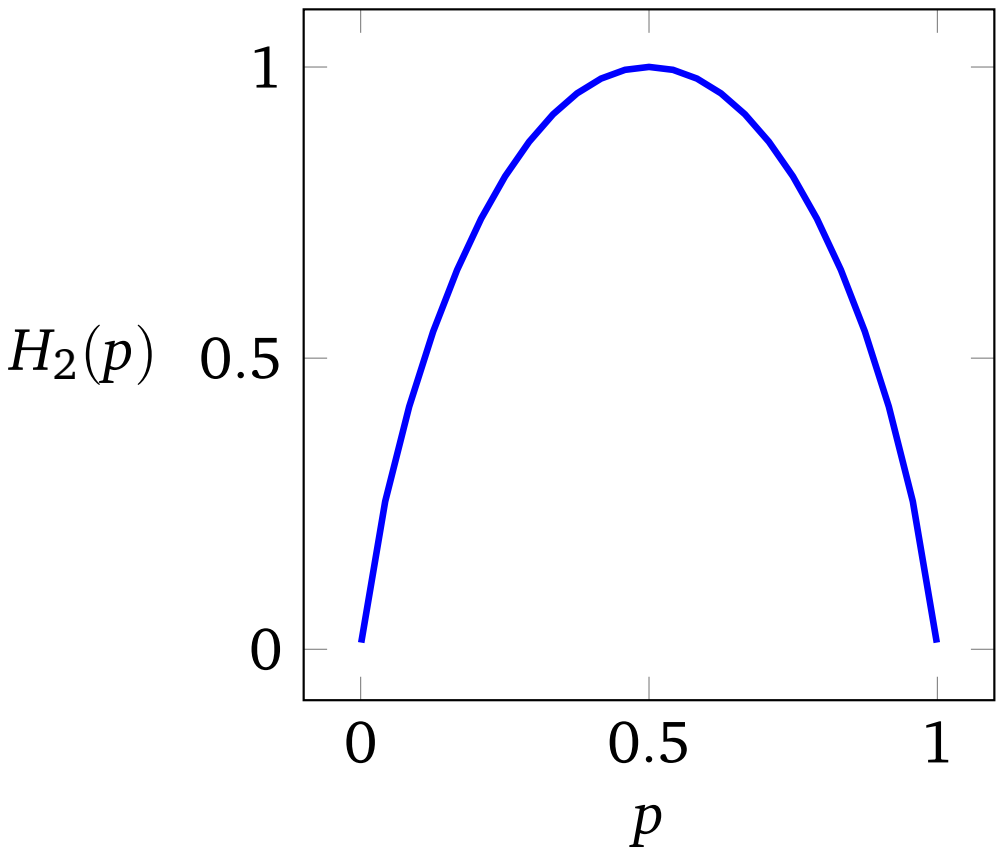
3.

x	A	B	C
$p(x)$	$1/3$	$1/3$	$1/3$

4.

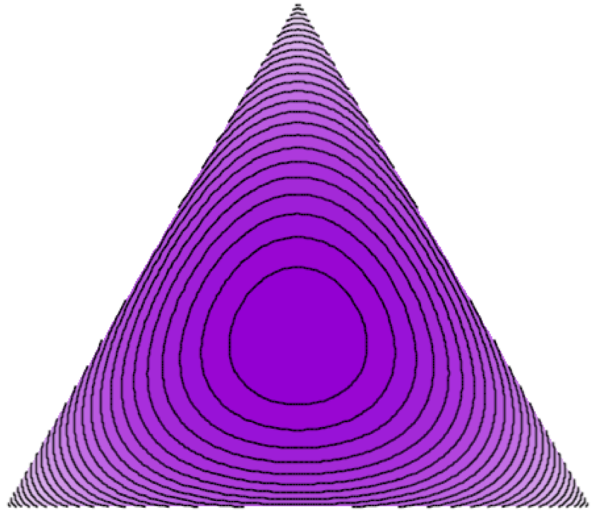
x	A	B	C
$p(x)$	$1/2$	$1/3$	$1/6$

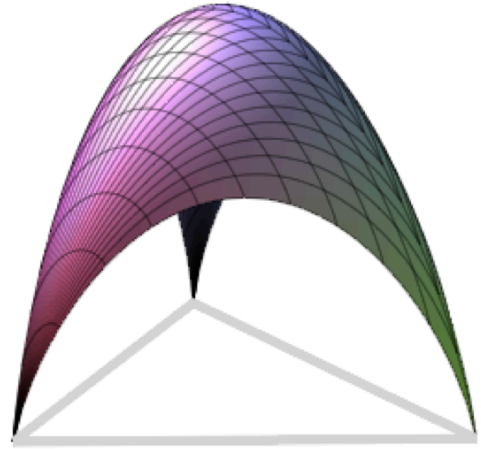
Binary Entropy



p_1	p_2	H
0.0	1.0	0.00
0.1	0.9	0.47
0.2	0.8	0.72
0.3	0.7	0.88
0.4	0.6	0.97
0.5	0.5	1.00

Tertiary Entropy





p_1	p_2	p_3	H
$1/3$	$1/3$	$1/3$	1.58
$1/2$	$1/2$	0	1.00
$1/2$	$1/4$	$1/4$	1.50

Lower and Upper bounds

Theorem

The least average codeword length L^* for a random letter with entropy H satisfies

$$H \leq L^* \leq H + 1$$

Proof. The codeword lengths

$$w_i = \left\lceil \log \frac{1}{p_i} \right\rceil \leq \log_2 \frac{1}{p_i} + 1$$

satisfy Kraft's inequality.

Huffman Coding

A	B	C	D	E
.35	.25	.20	.15	.05

Huffman Coding

x	$p(x)$	$-\log_2 p(x)$	w	codeword
A	.0634	3.98	4	1001
B	.0135	6.21	6	011101
C	.0242	5.37	5	00011
D	.0321	4.96	5	10100
E	.0980	3.35	3	001
F	.0174	5.84	6	101111
G	.0165	5.92	6	101011
H	.0438	4.51	5	11011
I	.0552	4.18	4	0110
J	.0009	10.17	9	011100000
K	.0061	7.35	7	0111001
L	.0336	4.89	5	10110
M	.0174	5.85	6	101110
N	.0551	4.18	4	0101
O	.0622	4.01	4	1000
P	.0180	5.80	6	110100

x	$p(x)$	$-\log_2 p(x)$	w	codeword
Q	.0008	10.33	10	0111000100
R	.0470	4.41	4	0000
S	.0502	4.32	4	0100
T	.0729	3.78	4	1100
U	.0234	5.42	5	00010
V	.0075	7.06	7	0111110
W	.0156	6.00	6	011110
X	.0014	9.46	9	011100001
Y	.0160	5.97	6	101010
Z	.0005	11.04	11	01110001011
¶	.0084	6.89	7	0111111
_	.1741	2.52	3	111
’	.0019	9.06	9	011100011
,	.0117	6.42	7	1101011
.	.0109	6.52	7	1101010
?	.0003	11.56	11	01110001010

Exercises

1. construct a Huffman code for

A	B	C	D
.1	.2	.3	.4

2. encode the message `DCDDAB` according to your code
3. decode the message `0010100001` according to your code
4. compute the expected codeword length of your code
5. compute the entropy of the original distribution