Math 5251 Huffman Coding (§3.4)

It turns out that given the source word probabilities $(p_1, \dots, p_m)$ for $W = \{w_1, \dots, w_m\}$, we can easily find an n-ary encoding $f: W \to \Sigma^*$ that achieves the minimum for avglength(f), via Huffman coding.

Let's

- describe the binary case first, $(\Sigma = \{0,1\})$

- prove that it achieves the minimum,

- then explain how to modify it for n-ary.

Binary Huffman encoding algorithm:

Assume by re-indexing that

$$p_1 \geq p_2 \geq \dots \geq p_{m-2} \geq p_{m-1} \geq p_m$$

and recursively define $f: W \to \{0,1\}^*$ by induction on m:

If $m=2$, so $W=\{w_1,w_2\}$ encode $f(w_1)=0$

(BASE CASE) probabilities p_1, p_2 $\quad f(w_2)=1$

If $m>2$, build a Huffman encoding for a source $W'=\{w_1', w_2', \dots, w_{m-2}', w_{m-1}'\}$ with probabilities $\{p_1, p_2, \dots, p_{m-2}, p_{m-1}+p_m\}$ and then tack on an extra 0 to $f(w_{m-1}')$ and an extra 1

i.e. $$f(w_i)=\begin{cases} f(w_i') & \text{if } i=1,2,\dots,m-2 \\ f(w_{m-1}')0 & \text{if } i=m-1 \\ f(w_{m-1}')1 & \text{if } i=m \end{cases}$$

Usually this is visualized via binary Huffman trees, reading code words as paths from root to leaves...

EXAMPLES

(1) $W = \{A, B, C, D\}$

probabilities $\frac{1}{2} \geq \frac{1}{5} \geq \frac{1}{5} \geq \frac{1}{10}$

$$p_1 \quad p_2 \quad \underbrace{p_3 \quad p_4}_{\text{add these, giving}}$$

$W' = \{A, CD, B\}$

$$\frac{1}{2} \geq \underbrace{\frac{3}{10} \geq \frac{1}{5}}_{\text{add these, giving}}$$

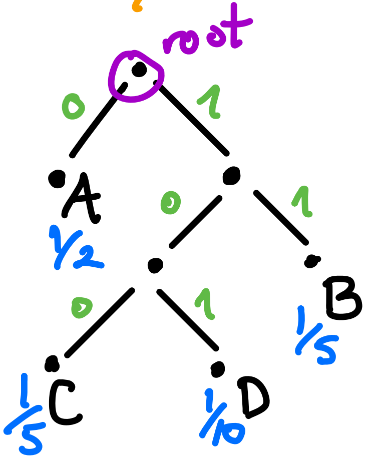
$W'' = \{A, BCD\}$

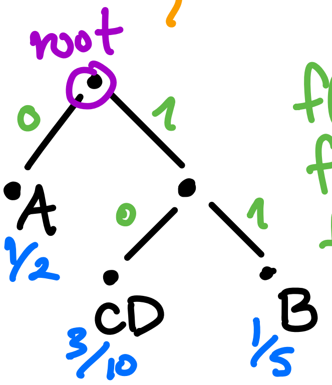
$\frac{1}{2} \geq \frac{1}{2}$

base case n=2

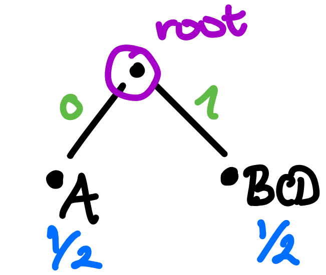
$f(A) = 0$
$f(B) = 11$
$f(C) = 100$
$f(D) = 101$

read paths from root to leaves





$f(A) = 0$
$f(CD) = 10$
$f(B) = 11$



$f(A) = 0$
$f(BCD) = 1$

(2) If some p_i coincide (or their sums coincide), the Huffman encoding may not be unique, e.g.

$$W = \{A, B, C, D, E\}$$

$$\frac{1}{5} \geq \frac{1}{5} \geq \frac{1}{5} \geq \underbrace{\frac{1}{5} \geq \frac{1}{5}}$$

$$W' = \{DE, A, B, C\}$$

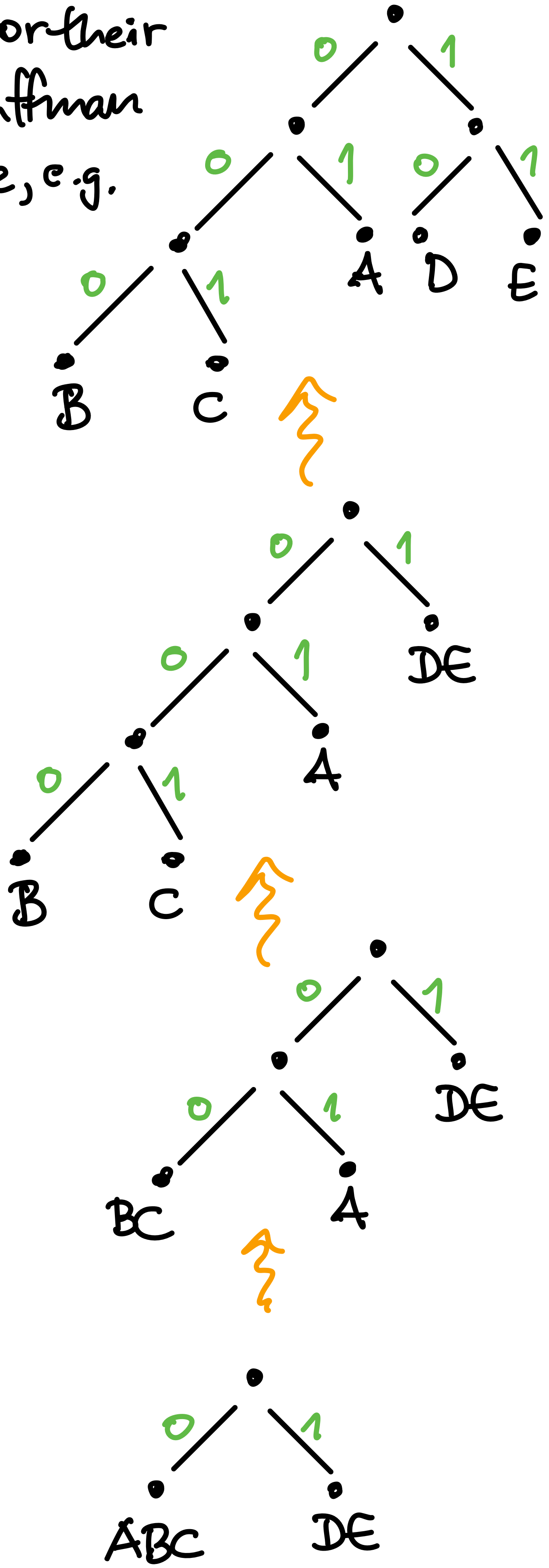
$$\frac{2}{5} \geq \frac{1}{5} \geq \underbrace{\frac{1}{5} \geq \frac{1}{5}}$$

$$W'' = \{DE, BC, A\}$$

$$\frac{2}{5} \geq \underbrace{\frac{2}{5} \geq \frac{1}{5}}$$

$$W''' = \{ABC, DE\}$$

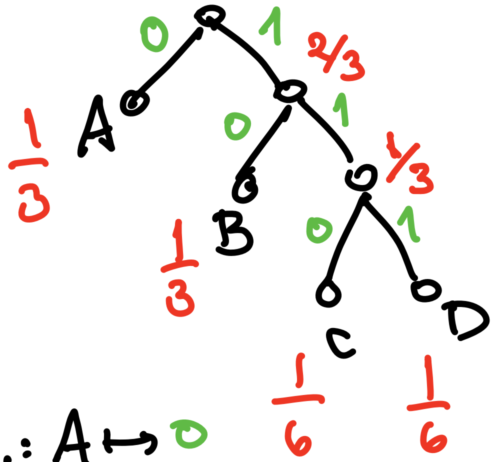
$$\frac{3}{5} \geq \frac{2}{5}$$



BETTER EXAMPLE of non-uniqueness.

$W = \{A, B, C, D\}$ has two possible binary Huffman tree structures, having different codeword lengths (but necessarily same avg length):

probs $\frac{1}{3}$ $\frac{1}{3}$ $\frac{1}{6}$ $\frac{1}{6}$

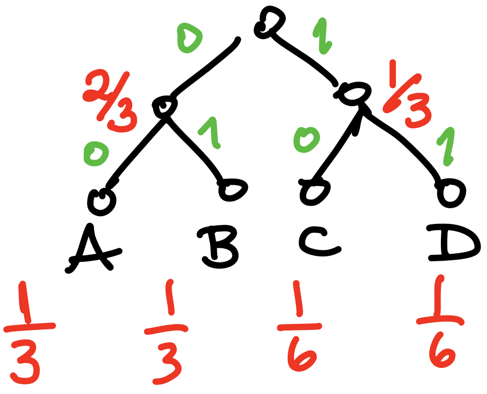


$$(l_1, l_2, l_3, l_4) = (1,2,3,3)$$

$$\text{avglength}(h_1) = \frac{1}{3}\cdot 1 + \frac{1}{3}\cdot 2 + \frac{1}{6}\cdot 3 + \frac{1}{6}\cdot 3$$

$$= \frac{2+4+3+3}{6} = 2$$

h_1: $A \mapsto 0$
$B \mapsto 10$
$C \mapsto 100$
$D \mapsto 101$



$$(l_1, l_2, l_3, l_4) = (2,2,2,2)$$

$$\text{avglength}(h_2) = \frac{1}{3}\cdot 2 + \frac{1}{3}\cdot 2 + \frac{1}{6}\cdot 2 + \frac{1}{6}\cdot 2$$

$$= 2$$

h_2: $A \mapsto 00$
$B \mapsto 01$
$C \mapsto 10$
$D \mapsto 11$

THEOREM Let $W = \{w_1, \dots, w_m\}$ have probabilities $\{p_1, \dots, p_m\}$ and $h : W \to \{0,1\}^*$ any Huffman encoding.

Then (a) h is prefix, so u.d., and

(b) for any u.d. encoding $f : W \to \{0,1\}^*$

$$\text{avg length}(h) \leq \text{avg length}(f)$$

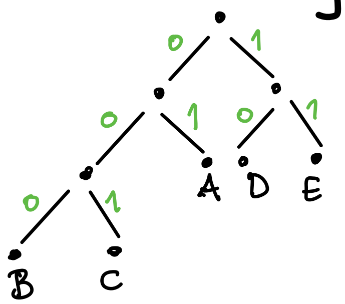
(so h achieves the minimum bounded in Shannon's Thm.)

EXAMPLE

$W = \{A, B, C, D, E\}$

$\frac{1}{5} \geq \frac{1}{5} \geq \frac{1}{5} \geq \frac{1}{5} \geq \frac{1}{5}$

This Huffman encoding



has

$A \overset{h}{\longmapsto} 01$

$B \longmapsto 000$

$C \longmapsto 001$

$D \longmapsto 10$

$E \longmapsto 11$

with lengths $(l_1, l_2, l_3, l_4, l_5) = (2, 2, 2, 3, 3)$.

Why can't we find something shorter, like $(2, 2, 2, 2, 3)$?

ACTIVE LEARNING

Explain why a binary code with lengths $(2, 2, 2, 2, 3)$ is never u.d.

Can you find two very different arguments?

Proof of THEOREM:

For (a), note that each Huffman codeword $f(\omega)$ is the labels on a path from root to a leaf in the tree. So $f(\omega)$ can't be a prefix of another $f(\omega')$, else the path from the root continues lower, so it wasn't stopping at a leaf to read $f(\omega)$.

For (b), assume that $f: W \to \{0,1\}^*$ is a u.d. encoding achieving the minimum of avglength(f) among all u.d. encoding
We'll show avglength$(h) \leq$ avglength(f) in several steps

STEP 1: We can assume f is prefix, not just u.d., because of the Kraft-McMillan Theorems: the lengths $(\ell_1, \dots, \ell_m)$ for $f(\omega_1), \dots, f(\omega_m)$ satisfy $\sum_{i=1}^{m} \frac{1}{n^{\ell_i}} \leq 1$ and hence $\exists$ a prefix code with the same lengths.

STEP 2: We can assume after re-indexing that if $p_1 \geq p_2 \geq \cdots \geq p_{m-2} \geq p_{m-1} \geq p_m$ then f has $l_1 \leq l_2 \leq \cdots \leq l_{m-2} \leq l_{m-1} \leq l_m$.

Otherwise, if $l_i > l_{i+1}$, swap images $f(w_i), f(w_{i+1})$ of w_i, w_{i+1} creating a new u.d. f with smaller $\text{avglength}(f) = \sum_{i=1}^{n} p_i l_i$.

STEP 3: We can assume $l_{m-1} = l_m$, otherwise if $l_{m-1} < l_m$ then we can drop the last letter of $f(w_m)$ without ruining the prefix property (Why?), and making avglength(f) smaller.

STEP 4: We can assume $\exists$ some $i \leq m-1$ such that $f(w_i)$ and $f(w_m)$ have same length $l_i = l_m$ and differ only in their last digit:

$$f(w_i) = a_1 a_2 \cdots a_{l-1} 0$$

$$f(w_m) = a_1 a_2 \cdots a_{l-1} 1$$

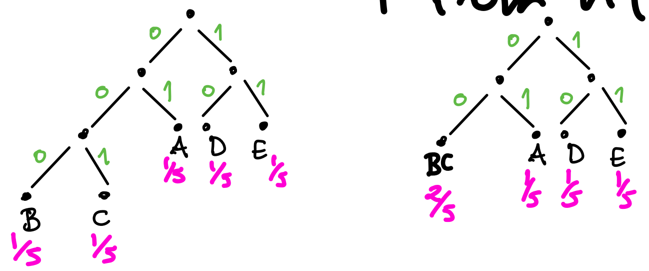
(In which case, re-index so that $i = m-1$).

This is because otherwise, we could again drop the last letter of $f(w_m)$ without ruining the prefix property (Why?), but reducing avglength(f).

LAST (INDUCTIVE) STEP:
on m

Create the smaller Huffman code $h' : W' \to \{0,1\}^*$
for the source with probabilities $p_1, p_2, \ldots, p_{m-2}, p_{m-1} + p_m$
by removing the final 0 from $h(\omega_{m-1})$
1 from $h(\omega_m)$

$W = \{A, B, C, D, E\}$
$\frac{1}{5} \geq \frac{1}{5} \geq \frac{1}{5} \geq \frac{1}{5} \geq \frac{1}{5}$



Similarly create the smaller prefix code $f' : W' \to \{0,1\}^*$
for that same source W'
by removing the final 0 from $f(\omega_{m-1})$
1 from $f(\omega_m)$.

Note how avg length for h and h' relate:
if the Huffman codewords have lengths $\hat{\ell}_1 \geq \ldots \geq \hat{\ell}_{m-2} \geq \hat{\ell}_{m-1} = \hat{\ell}_m$,

$$\operatorname{avglength}(h) = p_1 \hat{\ell}_1 + \ldots + p_{m-2} \hat{\ell}_{m-2} + \underbrace{p_{m-1} \hat{\ell}_{m-1} + p_m \hat{\ell}_m}_{= (p_{m-1} + p_m) \hat{\ell}_m}$$

$$\operatorname{avglength}(h') = p_1 \hat{\ell}_1 + \ldots + p_{m-2} \hat{\ell}_{m-2} + (p_{m-1} + p_m)(\hat{\ell}_m - 1)$$

$$\Rightarrow \boxed{\operatorname{avglength}(h) = \operatorname{avglength}(h') + p_{m-1} + p_m}$$

Similarly,

$$\boxed{\text{avglength}(f) = \text{avglength}(f') + p_{m-1} + p_m}$$

This lets us prove $\text{avglength}(h) \leq \text{avglength}(f)$ by induction on $m = |W|$, since it's easy to check in the base case where $m=2$ $\left(\text{so } \begin{array}{l} h(A)=0 \\ h(B)=1 \end{array}\right)$

and then in the inductive step, use

$\text{avglength}(h') \leq \text{avglength}(f')$

together with the two boxed facts above. ∎

It's easy to modify Huffman coding for an n-ary alphabet $\Sigma = \{0,1,2,\dots,n-1\}$:

the Huffman trees are n-ary and built by grouping $p_1 \geq p_2 \geq \dots \geq p_{n-m} \geq \underbrace{p_{n-m+1} \geq \dots \geq p_{m-1} \geq p_m}$

$$p_1 \geq p_2 \geq \dots \geq p_{n-m} \geq \sum_{i=n-m+1}^{m} p_i \quad \text{in } W'.$$

The only issue is that n-ary trees have their number of leaves $\equiv 1 \mod n-1$

i.e. remainder of 1 on division by $n-1$.

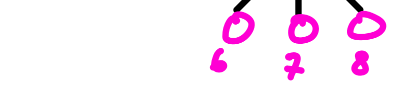
So one pads $p_1 \geq \dots \geq p_m \rightsquigarrow p_1 \geq \dots \geq p_m \geq 0 \geq \dots \geq \underset{p_M}{\underset{\shortparallel}{0}}$

with zeroes to make $M \equiv 1 \mod n-1$.

EXAMPLE

$n=3$ Ternary trees have number of leaves $\equiv 1 \mod 2$

i.e. odd



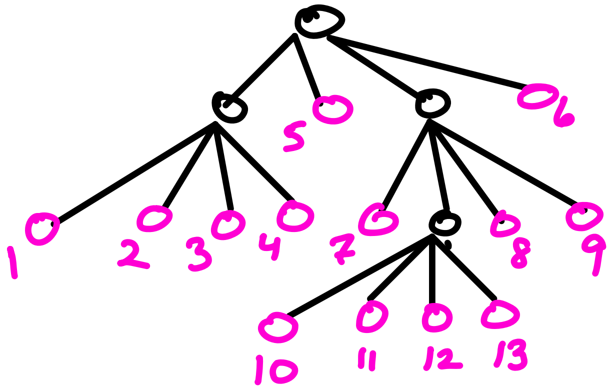
$9 \equiv 1 \mod 2$

odd

EXAMPLE

$n=4$

4-ary trees have number of leaves $\equiv 1 \bmod 3$



$13 \equiv 1 \bmod 3$

EXAMPLE

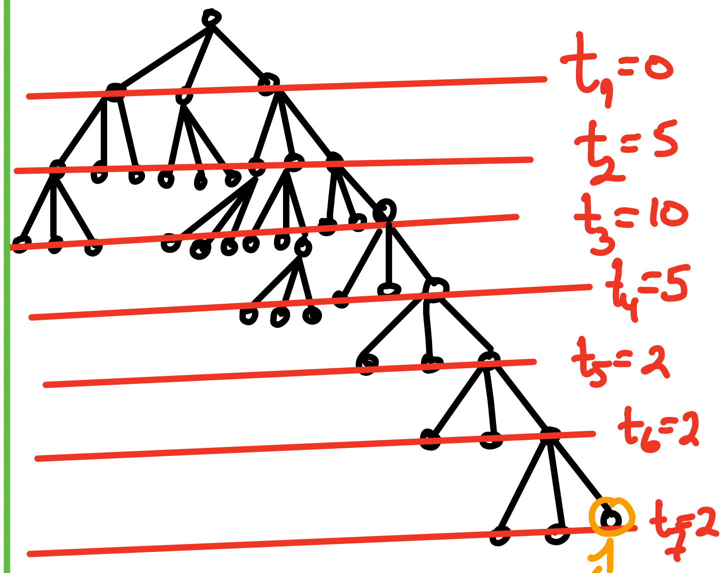
Morse code is a $\overset{n=3}{\text{ternary}}$ and prefix code $f: W = \underbrace{\{A, B, C, --, Z\}}_{m=26} \longrightarrow \{\circ, -, \text{space}\}^* = \Sigma^*$

How well does a ternary Huffman code $h: W \to \{0,1,2\}$ beat its avg length?

Since $n = 26 \not\equiv 1 \bmod 2$, need to add an extra fake 27th letter with probability $p_{27} = 0$, then use a computer to build a ternary Huffman tree...

Letter	English Probability	Morse code lengths (with space)	Ternary Huffman code lengths
E	0.12702	2	2
T	0.09056	2	2
A	0.08167	3	2
O	0.07507	3	2
I	0.06966	3	2
N	0.06749	3	3
S	0.06327	4	3
H	0.06094	4	3
R	0.05987	4	3
D	0.04253	4	3
L	0.04025	4	3
C	0.02782	4	3
U	0.02758	4	3
M	0.02406	4	3
W	0.0236	5	3
F	0.02228	5	4
G	0.02015	5	4
Y	0.01974	5	4
P	0.01929	5	4
B	0.01492	5	4
V	0.00978	5	5
K	0.00772	5	5
J	0.00153	5	6
X	0.0015	5	6
Q	0.00095	5	7
Z	0.00074	5	7

Ternary Huffman code tree structure:



fake 27th letter with probability 0

avg length (h) = 2.7

Morse code (with final space) has length tallies

$(t_1, t_2, t_3, t_4, t_5, t_6, t_7)$
$= (0, 2, 4, 8, 12, 0, 0)$

avg length (f) = 3.41

REMARK

Although a Huffman encoding achieves the minimum for avglength(f) among u.d. codes, it may not get as low as Shannon's $\frac{H(W)}{\log_2(n)}$ lower bound. But one way to improve it is is by grouping source words $W=\{w_1,\ldots,w_n\}$ into sequences $W^{(\ell)}=\{(w_{i_1},w_{i_2},\ldots,w_{i_\ell}) : w_i\in W\}$ sent ℓ at a time, called the ℓ^{th} extension of W, with $P(w_{i_1},w_{i_2},\ldots,w_{i_\ell})=p_{i_1}\cdot p_{i_2}\cdots p_{i_\ell}$

EXAMPLE

$$W=\{\overset{3/4}{A},\overset{1/4}{B}\}$$

has $H(W)=\frac{3}{4}\log_2\left(\frac{4}{3}\right)+\frac{1}{4}\log_2(4)\approx 0.811278$

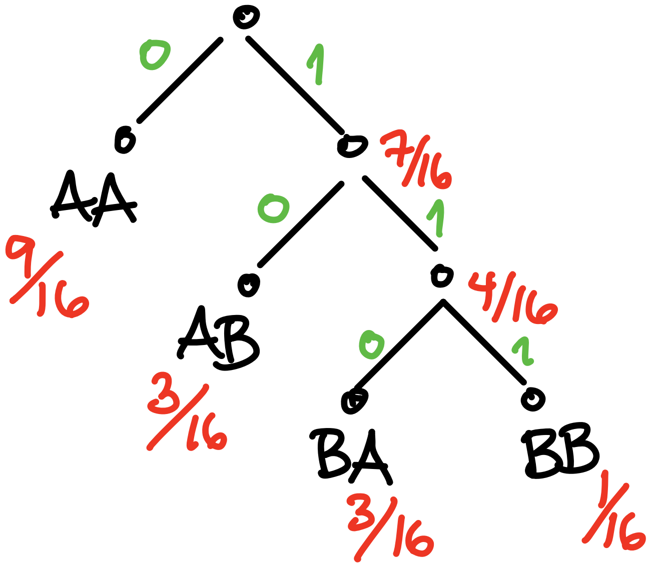
and binary Huffman encoding $\begin{matrix} f(A)=0 \\ f(B)=1\end{matrix}$

with avglength$(f)=\frac{3}{4}\cdot 1+\frac{1}{4}\cdot 1=1$ $\left(> \underset{=H(W)}{0.811278}\right)$

But its 2nd extension

$$W^{(2)} = \{ \underset{\frac{3}{4}\cdot\frac{3}{4} = \frac{9}{16}}{AA}, \underset{\frac{3}{4}\cdot\frac{1}{4} = \frac{3}{16}}{AB}, \underset{\frac{1}{4}\cdot\frac{3}{4} = \frac{3}{16}}{BA}, \underset{\frac{1}{4}\cdot\frac{1}{4} = \frac{1}{16}}{BB} \}$$

has binary Huffman encoding as shown:



$$\text{so avglength}(f) = \frac{9}{16}\cdot\underbrace{1}_{\ell(0)} + \frac{3}{16}\cdot\underbrace{2}_{\ell(10)} + \frac{3}{16}\cdot\underbrace{3}_{\ell(110)} + \frac{1}{16}\cdot\underbrace{3}_{\ell(111)}$$

$$= \frac{27}{16} = 1.6875$$

But it makes sense to divide this by 2, since we're sending 2 words at a time:

$$\frac{\text{avglength}(f)}{2} = \frac{27}{32} = 0.84375,$$ much closer to $H(W) \approx 0.811278$

In fact, its 3rd extension

$W^{(3)} = \{$	AAA,	AAB,	ABA,	BAA,	ABB,	BAB,	BBA,	BBB$\}$
probs	$\frac{27}{64}$	$\frac{9}{64}$	$\frac{9}{64}$	$\frac{9}{64}$	$\frac{3}{64}$	$\frac{3}{64}$	$\frac{3}{64}$	$\frac{1}{64}$

gets amazingly close: $\frac{\text{avglength}(f)}{3} = 0.811278$ matching to 6 digits!

It's not hard to show this version of Shannon's Noiseless Coding Thm:

(Roman Thm 2.3.4)
THEOREM: The ℓth extension $W^{(\ell)}$ of a source W has entropy $\frac{H(W^{(\ell)})}{\ell} = H(W)$, and among all n-ary u.d. encodings $f: W^{(\ell)} \to \Sigma^*$, the ones achieving minimum avglength(f) have

$$\frac{H(W)}{\log_2(n)} \leq \frac{\text{avglength}(f)}{\ell} \leq \frac{1}{\ell} + \frac{H(W)}{\log_2(n)}$$

can be made smaller by picking ℓ larger