$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad ad - bc \not\equiv 0 \pmod{29}$$

if $ad \equiv 0 \pmod{29}$

(must be that $b \cdot c \not\equiv 0 \pmod{29}$
OR $b \not\equiv 0$ & $c \not\equiv 0$)

$a \equiv 0$ & $d \not\equiv 0$ $\quad$ $a \not\equiv 0$ & $d \equiv 0$ $\quad$ $a \equiv 0$ & $d \equiv 0$

$(28 + 28 + 1) \cdot$

($c \not\equiv 0$: 28; $d \not\equiv 0$: 28)

and $28 \cdot 28$

if $ad \not\equiv 0 \pmod{29}$

must be that $a \not\equiv 0$ & $d \not\equiv 0$

$28 \cdot 28$ ways of choosing a & d

{ if $bc \equiv 0$ then ~~there~~ there are 57 ways of choosing b & c $\quad bc \not\equiv ad \pmod{29}$ $c \not\equiv b^{-1}ad$

OR $bc \not\equiv 0$ then $ad - bc \not\equiv 0$ we must have that $b \cdot c \equiv$ anything except ad

$b \not\equiv 0$ so there are 28 choices for b

$c \not\equiv 0$ OR $b^{-1}ad$ so there are 27 choices for c

$28^2(57 + 28 \cdot 27)$

total = $57 \cdot 28^2 + 28^2(57 + 28 \cdot 27) = 492{,}080$

Better estimate for unicity distance of Hill (mod 29)

$$H(C) = \log_2(29^N) = 4.9N$$

$$H(M) \cong 3.2N$$ (estimate based on ~~chart~~ table

$$H(K) = \log_2(682{,}080)$$

N comes from setting

$$H(C) = H(K) + H(M)$$

$$4.9N = \log_2(682{,}080) + 3.2N$$

$$N = \frac{\log_2(682080)}{1.7} \cong 11.4$$

Vernam $p = 11$ $q = 18 \pmod{26}$

# of keys $26^{11} \cdot 26^{18} = 26^{29}$

$H(K) = \log_2(26^{29})$

N = length of the cypertext ~~plaintext~~

$H(M) = 1.2N$

$H(C) = \log_2(26^N)$

when $H(K|C) = 0$

$H(C) = H(k) + H(M)$

$4.7N = 29(4.7) + 1.2N$

$N = \frac{29(4.7)}{3.5} \approx 39$

N.B. Quiz 4 for Tuesday March 6.

Homophonic of key length 4

# of keys $= 25^4$

$H(K) = \log_2 25^4$

$H(M) \approx 2 \cdot N$ (guess that ~~[illegible]~~ $15 \leq N \leq 25$)

$H(C) = \log_2(100^N) = N \cdot \log_2(100) =$ ~~[illegible]~~

$H(C) = H(K) + H(M)$

~~$4N\log_2(25)$~~ $= 4 \cdot \log_2 25 + 2 \cdot N$

$N \cdot \log_2(100)$

$$N = \frac{4 \cdot \log_2 25}{(\log_2(100) - 2)} \approx 4$$

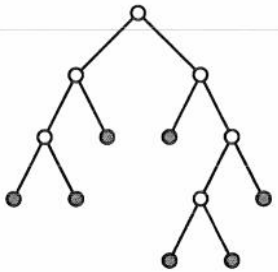
Means we took a bad guess for $H(M)$

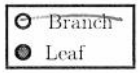
$H(M) = 4.16N$

$$N = \frac{4 \cdot \log_2 25}{(\log_2(100) - 4.16)} \approx 7.5$$

Binary Trees

Complete





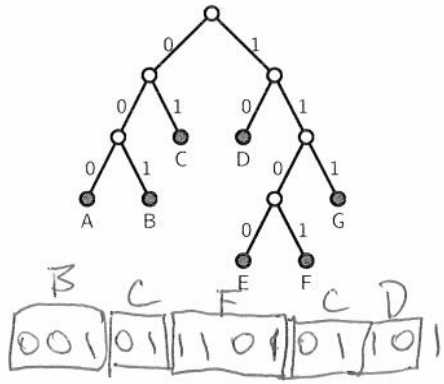
0 node

1 Branch

Incomplete

A Comma-Free Binary Code

Definition: A binary code is *comma-free* if no prefix of the code of a letter is the code of another letter.



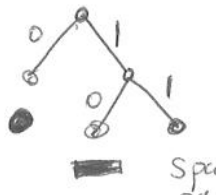
A=000 B=001 C=01
D=10 E=1100 F=1101
G=111

File length $= 3N_A + 3N_B + 2N_C + 2N_D + 4N_E + 4N_F + 3N_G$

Morse Code

A	•—	J	•———	S	•••
B	—•••	K	—•—	T	—
C	—•—•	L	•—••	U	••—
D	—••	M	——	V	•••—
E	•	N	—•	W	•——
F	••—•	O	———	X	—••—
G	——•	P	•——•	Y	—•——
H	••••	Q	——•—	Z	——••
I	••	R	•—•		

• ↔ 0 — ↔ 10 *Comma* ↔ 11



Space
pause
Comma

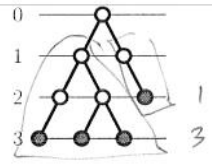
Expected code length

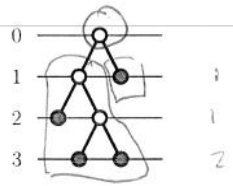
$$5N_A + 7N_B + 8N_C + \cdots + 9N_Y + 8N_Z$$

Using single letter english frequencies, the average number of bits per letter is

$$\frac{5 \cdot 73 + 7 \cdot 9 + 8 \cdot 30 + \cdots + 9 \cdot 19 + 8 \cdot 1}{1000} = 5.738$$

Leaf Heights





$$\frac{1}{2^3} + \frac{1}{2^3} + \frac{1}{2^3} + \frac{1}{2^2} = 5/8$$

$$\frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^3} + \frac{1}{2^2} = 1$$

$\frac{1}{2^2} + 3 \cdot \frac{1}{2^3} = \frac{5}{8} < 1$ sum < 1 indicates tree is not complete

Sum will be 1 if tree is complete

Theorem 1 *The sequence of integers $h_1, h_2, \ldots, h_n$ are leaf heights of a binary tree if and only if*

$$\sum_{i=1}^{n} \frac{1}{2^{h_i}} \leq 1$$

with equality only if the tree is complete.

Tree from heights

Note that given probabilities $p_A, p_B, \ldots, p_Z$, if we set

$$h_\alpha = \left\lceil \log_2\left(\frac{1}{p_\alpha}\right) \right\rceil$$

then since we know from Theorem 4 that $\sum_{\alpha=A}^{Z} h_\alpha \leq 1$ then by Theorem 1 these values must correspond to heights of a (possibly incomplete) binary tree.

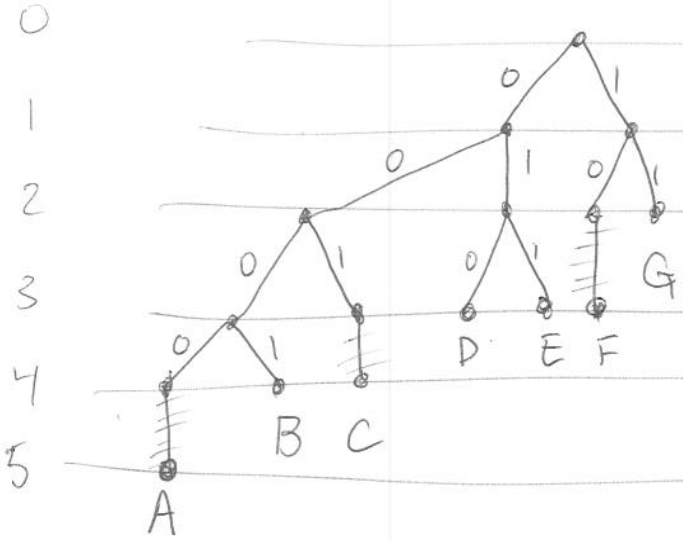
By the same proof as in theorem 4, this code will also have an expected code length less than or equal to $H + 1$.

Expected code length = $4 \cdot 2 + 4 \cdot 4 + 3 \cdot 6 + 3 \cdot 10 + 3 \cdot 13 + 2 \cdot 13 + 2 \cdot 16$

$= 169$ per letter

$= 169/64$

≈ 2.64



Tree from heights

Begin with a text file with the following frequencies

letter	A	B	C	D	E	F	G
frequency	2	4	6	10	13	13	16
	64	64	64				64

The goal is to encode each letter in such a way that minimizes the average number of bits used to store the file.

$H\left(\begin{matrix}\text{R.V. which} \\ \text{produces this} \\ \text{file}\end{matrix}\right) = 2.57$

$h_G = \lceil \log_2(64/16) \rceil = \lceil \log_2(4) \rceil = 2$

$h_E = h_F = \lceil \log_2(64/13) \rceil = 2$

$h_B = \lceil \log_2(64/4) \rceil = \log_2 16 = 4$

$h_D = \lceil \log_2(64/10) \rceil = 3$

$h_C = \lceil \log_2(64/6) \rceil = 4$

$h_A = \lceil \log_2(64/2) \rceil = 5$