STATISTICAL CODING

: based on frequency of bits / bytes / words, integers

- Huffman
- Arithmetic

• Frequency:
- STATIC MODEL
- SEMISTATIC MODEL
- DYNAMIC MODEL

of frequency → are given

→ SCAN the text, count # of chars

• Σ: alphabet of symbols that we will encode, binary (2), bytes (256), words, integers (∞) (millions of symbols)

• Compressed file:

$C(f)$ = | PREAMBLE | BODY |

PREAMBLE ↓ additional info:
- Σ
- frequency
- ...

BODY ↓ sequence of bits

HUFFMAN

• greedy algorithm → optimal

• the algo repeats these steps:

• tree bottom-up

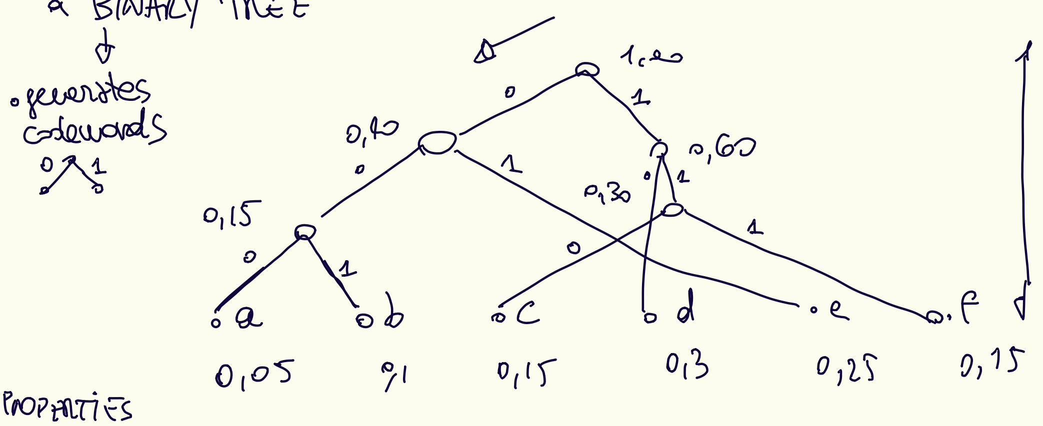
1. pick the 2 lightest (probability) symbols
2. merge them in one new symbol whose prob. is the sum of the probabilities of the merged symbols

• given an alphabet $\Sigma = \{a, b, c, d, e, f\}$ with frequencies:

a: 0,05 b: 0,1 c: 0,15 d: 0,3 e: 0,25 f: 0,15

$T = aaaabb$

• construct a BINARY TREE ↓ • generates codewords



a: 000
b: 001
c: 110
d: 10
e: 01
f: 111

PROPERTIES

1. is a PREFIX-FREE CODE: no codeword is a prefix of the other

2. GOLDEN RULES of data compression: MOST FREQUENT SYMBOLS GET THE SHORTEST CW
   → merges light symbols first

• To evaluate the GOODNESS of a code: AVG CODEWORD LENGTH: $L_c = \sum_{\sigma} f(c) \cdot \ell(cw)$ bits

all the symbols

freq. of the symbol

length of the codeword

L_c: $3 \cdot 0{,}05 + 3 \cdot 0{,}1 + 3 \cdot 0{,}15 + 2 \cdot 0{,}3 +$
$+ 2 \cdot 0{,}25 + 3 \cdot 0{,}15 = 3 \cdot 0{,}45 + 2 \cdot 0{,}55 =$
$= 1{,}35 + 1{,}1 = 2{,}45$ bits

=> AVG CW length of the code

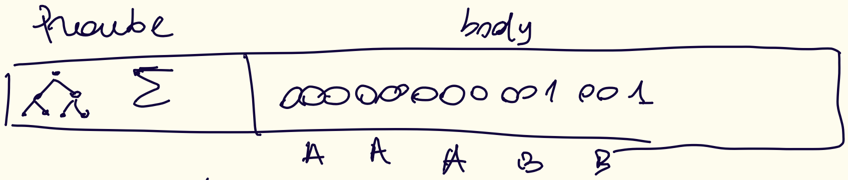
RELATION between:

AVG CW length & height of the tree

=> height of the tree = CW length

=> AVG height of the tree = AVG CW length (L)

Compressed file



$|\text{preamble}| + \underbrace{\quad}_{15 \text{ bits}}$

T = aaabb

5 BYTES

⇓

5·8 bits = 40 bits

vs

|preamble| + 15 bits of body

advantages with big files ⇐

THEO: Huffman code is optimal among prefix free codes for Σ.
(take a symbol, assign a codewords)

avg length of Huffman: $L_H \le L_C$ AVG length of prefix-free code

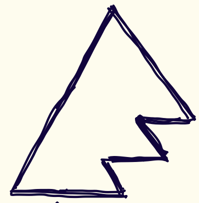
∀ code C prefix-free

Lemma: let F be the set of binary trees corresponding to optimal prefix-free codes

CANONICAL HUFFMAN:

1) Reduce space of Huffman tree
2) Reduce compression time

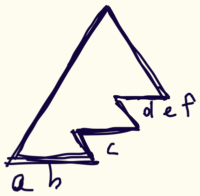
WHAT THAT ⇒ Huffman Tree gets this form:



4 deepest leaves

1. Compute the codeword length $L(\sigma)$ for each symbol $\sigma \in \Sigma$ according to the classical Huffman's algorithm. Say max is the maximum codeword length.
2. Construct the array *num*[1, max] which stores in the entry *num*[ℓ] the number of symbols having Huffman codeword of ℓ-bits.
3. Construct the array *symb* which stores in the entry *symb*[ℓ] the list of symbols having Huffman codeword of ℓ-bits.
4. Construct the array fc[1, max] which stores in the entry fc[ℓ] the first codeword of all symbols encoded with ℓ bits;
5. Assign consecutive codewords to the symbols in *symb*[ℓ], starting from the codeword fc[ℓ].

So
0) Build Huffman tree



1) Compute cw lengths:

	cw length
a	4
b	4
c	3
d	2
e	2
f	2

2) Compute # of symbols per level length
(Num array)

num	1	2	3	4
	0	3	1	2

3) Compute symbol table: for every level we keep the list of the symbols at that level

symbol	0 1 2 3 . . .
1	//
2	d, e, f
3	c
4	a, b

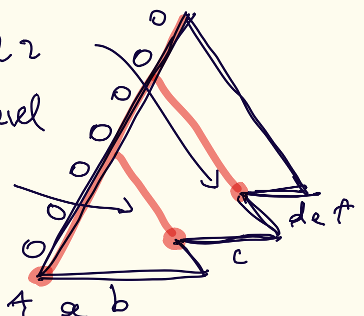
4) Compute the new codewords
(Create the tree here)

- observe that the tree has a special shape: in order to describe the structure of the tree: I need the 1ST codeword of every level

1ST cw level 2

1ST cw level 3

1ST cw of level 4



- if I've fc (1ST codeword) the others are obvious cuz others are just increasing
- and the symbols are in the symbol table

1ST cw: f_c

- FORMULA
- ℓ = level max (4 our case)
- $f_c[\ell_{max}] = 0$

next level num of cw at the next level

$$\underset{\ell < \ell_{max}}{f_c[\ell]} = \frac{f_c[\ell+1] + num[\ell+1]}{2}$$

es. for level 3:

$$fc[3] = \frac{fc[4] + num[4]}{2} = \frac{0+2}{2} = 1$$

=>

fc	1	2	3	4
	2	1	1	0

1ST cw a the lightest level is 0

represent 0 in 4 bits = (0000)

repeat 1 in 3 bits: (001)

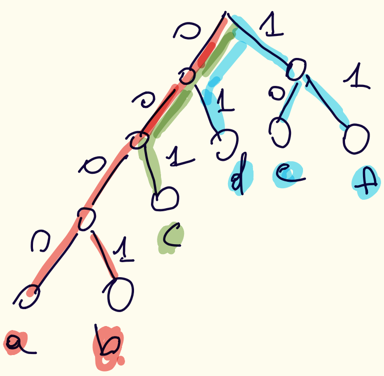
in 2 bits: 01

2 in 1 BIT => not possible but this value is fundamental 4 the algorithm

$$fc[2] = \frac{fc[3] + num[3]}{2} = \frac{1+1}{2} = 1$$

$$fc[1] = \frac{fc[2] + num[2]}{2} = \frac{1+3}{2} = 2$$

=>



=> Canonical Huffman tree

we store $fc[1, max]$

Symb. table

How we code: T = b c f

- b is in position 1 in the symbol table and is related to level 4

=> encode b: $fc[4] + rank(b)$ in its list

$$\underbrace{\overset{\shortparallel}{0} + \overset{\shortparallel}{1}}_{}$$

cw = 1 represented in 4 bits: 0001

- c: $fc[3] + rank(c)$

$$\underbrace{\overset{\shortparallel}{1} + \overset{\shortparallel}{0}}_{}$$

cw 1 in 3 bits: 001

- f: $fc[2] + rank[f]$

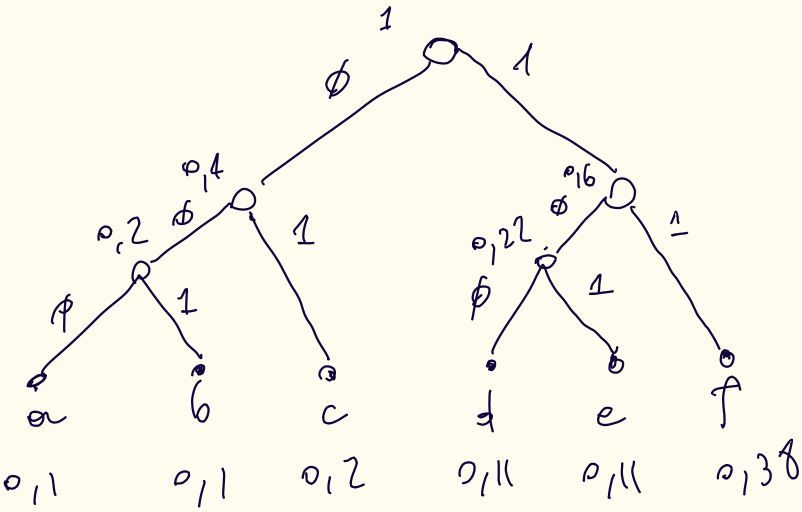
$$2 + 2$$

es: Canonical Huffman:

S: a b c d e f

0,1 0,1 0,2 0,11 0,11 0,38

a encode T = f A D E



0) compute cw length

a =	000	3
b =	001	3
c =	01	2
d =	100	3
e =	101	3
f =	11	2

1) comput num array

num. array

num	1	2	3
	0	2	4

2) compute symbol table

symb table

symb	0 1 2 3
1	
2	c, f
3	a, b, d, e

3) compute 1st cw length

$fc[l_{max}] = 0$

$$fc[l] = \frac{fc[l+1] + num[l+1]}{2} =$$

$$fc[2] = \frac{fc[3] + num[3]}{2} = \frac{0+4}{2} = 2$$

$$fc[1] = \frac{fc[2] + num[2]}{2} = \frac{2+2}{2} = 2$$

l	1	2	3
fc	2	2	0

2 in 2 bits = 10

↳ 0 in 3 bits: 000

F = fc[level] + rank in [symb. table] =

= fc[2] + rank[F] =

2 + 1 = 3 in 2 bits => 11

A = fc[3] + rank[A] =

= 0 + 0 = 0 in 3 bits => 00

$D = fc[3] + rank[d] =$

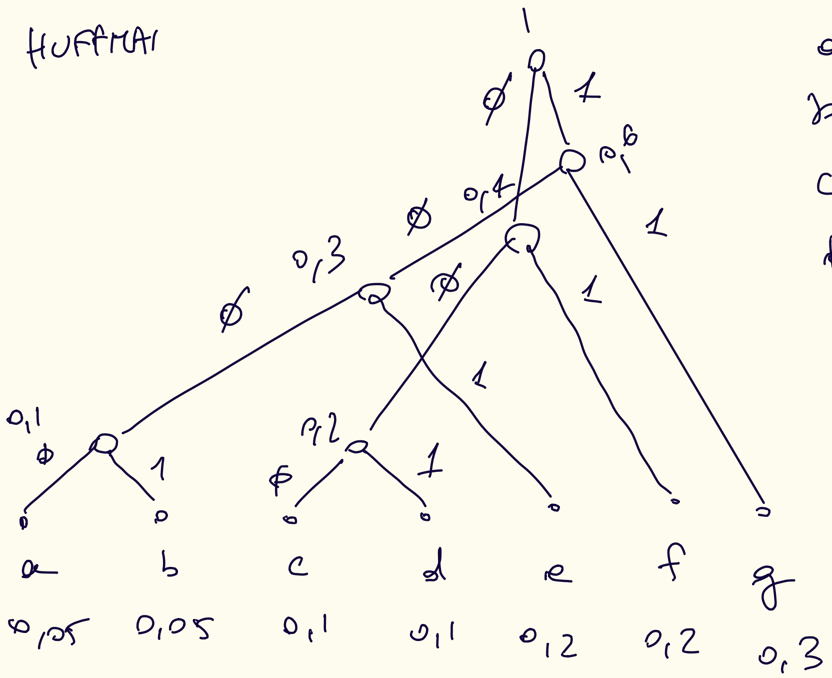
$0 + 2 = 2$ in 3 bits $\Rightarrow$ 010

$E = fc[3] + rank[E] =$

$0 + 3 = 3$ in 3 bits $\Rightarrow$ 011

√

CANONICAL HUFFMAN



		cw length
a =	1000	4
b =	1001	4
c =	000	3
d =	001	3
e =	101	3
f =	01	2
g =	11	2

NUM array

	1	2	3	4
num	0	2	3	2

Symbol table

symb	0	1	2	3
1				
2	f	G		
3	C	D	E	
4	A	B		

compute FC per level

l	1	2	3	4
FC	2	2	1	0

$FC[l_{max}] = 0$

$FC[l] = \frac{FC[l+1] + num[l+1]}{2}$

$FC[3] = \frac{FC[4] + num[4]}{2} = \frac{0 + 2}{2} = 1$

$FC[2] = \frac{1+3}{2} = \frac{4}{2} = 2$

$FC[1] = \frac{2+2}{2} = \frac{4}{2} = 2$

encode $\uparrow$ = G C D

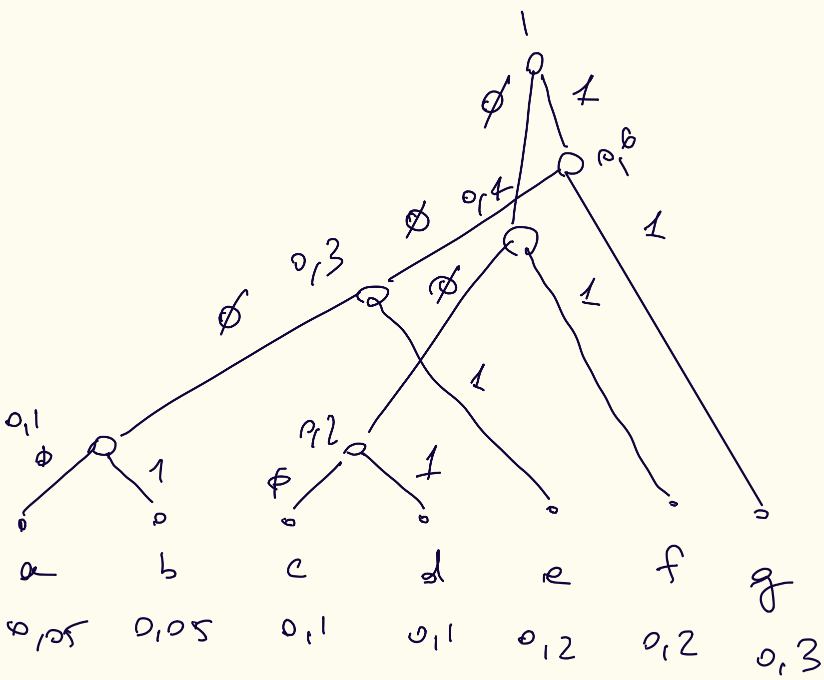
G = FC [level] + rank [G] =

FC [2] + 1 = 2+1 = 3 in 2 bits => 1 1

C = FC [3] + rank [C] = 1 + 0 = 1 in 3 bits => 0 0 1

D = FC [3] + rank [D] = 1 + 1 = 2 in 3 bits =>

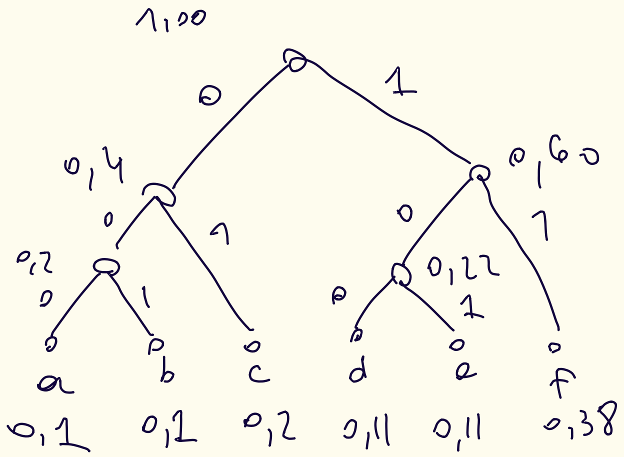
0 1 0 ✓



CANONICAL HUFFMAN

a encode T = f A D E

S:	a	b	c	d	e	f
	0,1	0,1	0,2	0,11	0,11	0,38



A = 000
B = 001
C = 01
D = 100
E = 101
F = 11

NUM	1	2	3
	0	2	4

SYMB	0	1	2	3
1	–			
2	C	F		
3	A	B	D	E

FC	1	2	3
	2	2	0

$\ell_{max} \leftarrow 0$

$$FC[\ell] = \frac{FC[\ell+1] + NUM[\ell+1]}{2}$$

$$FC[2] = \frac{0+4}{2} = \frac{4}{2} = 2$$

$$FC[1] = \frac{2+2}{2} = \frac{4}{2} = 2$$

of the symb

ENCODE : $FC[\ell] + RANK[symb]$

F = FC[2] + RANK[F] = 2+1 = 3 in 2 bits $(3)_{10} = (11)_2$

A : FC[3] + RANK[A] = 0+0 = 0 in 3 bits $(0)_{10} = (000)_2$

D = FC[3] + RANK[D] = 0+2 = 2 in 3 bits $(2)_{10} = (010)_2$

E = FC[3] + RANK[E] = 0+3 = 3 in 3 bits $(3)_{10} = (011)_2$

DECODE : 11000010011

$v = 1 \quad \ell = 1 \qquad v = 1 \overset{?}{<} FC[1]$ yes $\Rightarrow v = 2 \cdot 1 + 1 = 3 \quad \ell{+}{+}$

$v=3 \quad l=2$ $\qquad v=3 \overset{?}{<} FC[2]=2$ no

ret symb$[l; v-FC[l]]=$

$=$ symb$[2; 3-2]=[2,1]$ $=$ F

$v=0 \quad l=1$ $\qquad v=0 \overset{?}{<} FC[1]=2$ yes $\quad v=2\cdot 0+0=0 \quad l{+}{+}$

$v=0 \quad l=2$ $\qquad v=0 \overset{?}{<} FC[2]=2$ yes $\quad v=2\cdot 0+0=0 \quad l{+}{+}$

$v=0 \quad l=3$ $\qquad v=0 \overset{?}{<} FC[3]=0$ no

symb$[l; v-FC[l]]=$ sy$[3; 0-0]=$ A

$v=0 \quad l=\cancel{4}$ $\qquad v=0 \overset{?}{<} fc[1]=2$ yes $\quad v=2\cdot 0+1=1 \quad l{+}{+}$

$v=1 \quad l=2$ $\qquad v=1 \overset{?}{<} FC[2]=1$ yes $\quad v=2\cdot 1+0=2 \quad l{+}{+}$

$v=2 \quad l=3$ $\qquad v=2 \overset{?}{<} FC[3]=0$ no

RET symb$[3; 2-0]=$ D

$v=0 \quad l=1$ $\qquad v=0 \overset{?}{<} FC[1]=2$ no

$v=2\cdot 0+1=1 \quad l{+}{+}$

$v=1 \quad l=2$ $\qquad v=1 \overset{?}{<} FC[2]=1$ no

$v=2\cdot 1+1=3 \quad v{+}{+}$

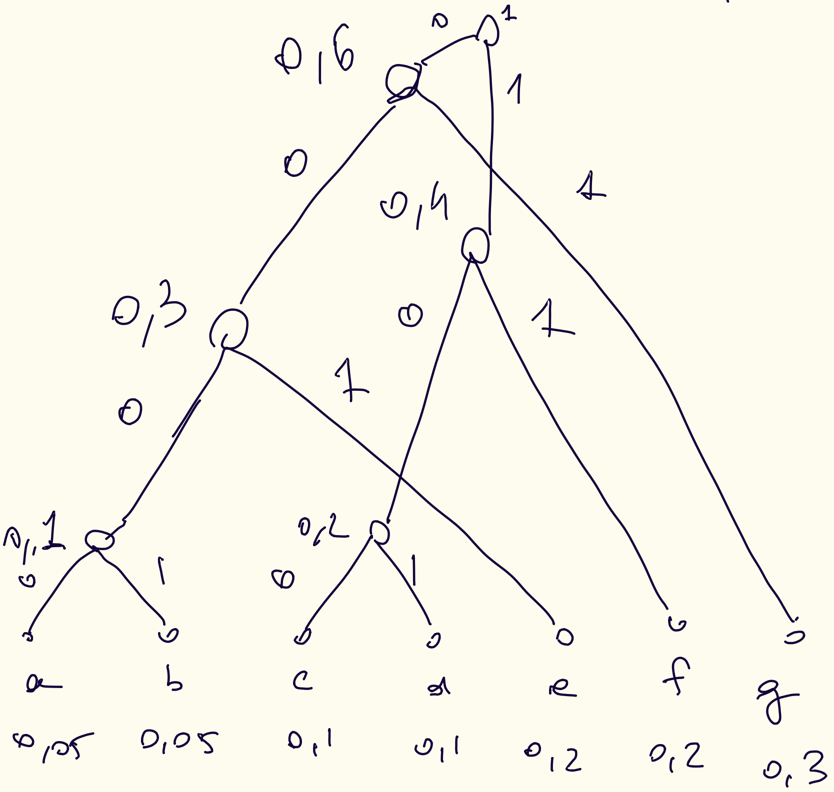
$v=3 \quad l=3$ $\qquad v=3 \overset{?}{<} FC[3]=0$ no

RET sym$[3; 3-0]=$ E

V

Do Canonical Huffman

a	b	c	d	e	f	g
0,05	0,05	0,1	0,1	0,2	0,2	0,3



a = 0000
b = 0001
C = 110
D = 111
E = 001
F = 11
G = 01

NUM	1	2	3	4
	0	2	3	2

symb	0	1	2	3
1	—			
2	F	G		
3	C	D	E	
4	A	B		

Front codeword per level

FC	1	2	3	4
	2	2	1	0

$$FC[\ell] = \frac{fc[\ell+1] + num[\ell+1]}{2}$$

$$FC[3] = \frac{FC[4] + num[4]}{2} = \frac{0+2}{2} = \frac{2}{2} = 1$$

$$FC[2] = \frac{1+3}{2} = \frac{4}{2} = 2$$

$$FC[1] = \frac{2+2}{2} = \frac{4}{2} = 2$$

ENCODE : $FC[\text{level}] + \text{rank}[G^{\text{in symb}}]$ in fc_{level} bits

T: G C D

$G = FC[2] + RANK[G] = 2 + 1 = 3$ in 2 bits

$\Rightarrow 011$ (3 bits)

$C = FC[3] + RANK[C] = 1 + 0 = 1$ in 3 bits

$\Rightarrow 001$

$D = FC[3] + RANK[D] = 1 + 1 = 2$ in 3 bits

010

DECODE : 011 001 010

v = next_bit()

l = 1;

$v \overset{?}{<} FC[l]$ yes => continue

$v = 2v + \text{next\_bit}$

l++

else

decode => return $\text{symb}[l; v - FC[l]]$

$v = 0 \quad l = 1$

$0 < FC[1]$ yes => $v = 2 \cdot 0 + 1 = 1 \quad l = 2$

$1 < FC[2]$ yes $v = 2 \cdot 1 + 1 = 3 \quad l = 3$

$3 < FC[3]^{-1}$ no

return $\text{symb}[l; v - FC[l]]$ — $3;\ 3 - 1 = 2$

$\text{symb}[3; 2]$

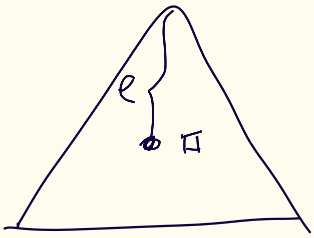
DECODING

fc	1	2	3	4
	2	1	1	0

symbs table

symbs	0	1	2	3
1				
2	c,	f		
3	a,	b,	d,	e

. we need ℓ (→ level), called v



$$\overbrace{10001010}^{v}$$
$$\underbrace{10001010}_{e}$$

if (v < fc[e]) go down
else decode

ALGORITHM

v = next_bit(); (→ fetch 1st bit) e = 1; (→ start with level 1)

while (v < fc[e]) → 1st cw of e

  v = 2·v + next_bit() // go down

  e++;

return symbol[e, v - fc[e]] → pick the right symbol

2·v (add 0 in this pos)

next_bit → if it's 0 remains 0, if 1 get 1

=> Decode S = 11001

v = 1; e = 1;

fc	1	2	3	4
	2	1	1	0

symbs table

symbs	0	1	2	3
1	"			
2	d	e	f	
3	c			
4	a	b		

v = 1 $\overset{?}{<}$ fc[1] = 2 yes continue

V = 2·V + next_bit() = 2·1 + 1 = 3; e = 2;

v = 3 $\overset{?}{<}$ fc[2] = 1 no => DECode

=> return symbol[$\overset{2}{e}$, $\overset{3-1}{v - fc[e]}$]

=> symbol [2, 2] = f ✓

RESTART

v = 0; e = 1;

v = 0 $\overset{?}{<}$ fc[1] (2) yes continue

V = 2·v + next_bit() = 0 + 0 = 0 e = 2

$v = 0; \ \ell = 2$

$v = 0 \overset{?}{<} fc[2] \overset{=1}{}$ yes continue

$v = 2 \cdot 0 + \text{next\_bit}() = 0 + 1 = 1; \ \ell = 3$

$v = 1 \overset{?}{<} fc[3] \overset{=1}{}$ NO $\Rightarrow$ Decode

$\text{symbol}[\ell, v - fc[\ell]]$

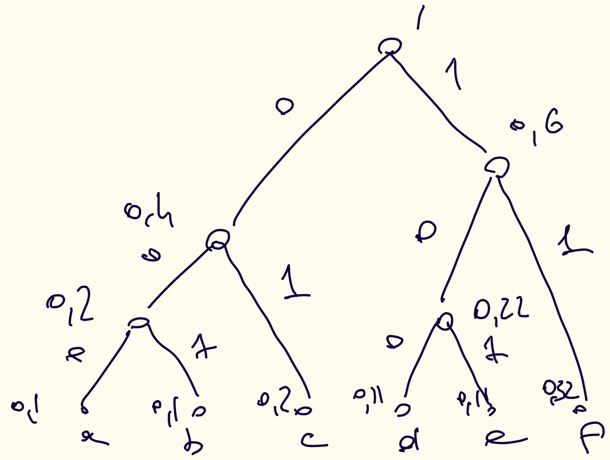
$\Rightarrow \text{Symbol}[3, 1 - 1] = [3, 0] = c$

```
v = next_bit();
l = 1;
while( v < fc[l] )
    v = 2v + next_bit();
    l++;
return symb[ l, v-fc[l] ];
```

Canonical Huffman

encode

symb	a	b	c	d	e	f
prob.	0,1	0,1	0,2	0,11	0,11	0,32



a = 000
b = 001
c = 01
d = 100
e = 101
f = 11

num	1	2	3
	0	2	4

symb	0	1	2	3
1				
2	C, F			
3	A, B, D, E			

FC	1	2	3
	2	2	0

1

$FC[l_{max}] = 0$

$FC[l] = \frac{fc[l+1] + num[l+1]}{2} =$

$FC[2] = \frac{FC[3] + num[3]}{2} = \frac{0+4}{2} = 2$

$FC[1] = \frac{FC[2] + num[2]}{2} = \frac{2+2}{2} = 2$

ENCODE

symb = FC[symb] + RANK[symb] in FC bits (level of)

T = FADE

F = FC[2] + RANK[F] = 2 + 1 = 3 in 2 bits => 11

A = FC[3] + RANK[A] = 0 + 0 = 0 in 3 bits => 000

D = FC[3] + RANK[D] = 0 + 2 = 2 in 3 bits => 010

E = FC[3] + RANK[E] = 0 + 3 = 3 in 3 bits => 011

DECODE

```
v = next_bit();
l = 1
 while (v < FC[l])
        v = 2·v + next_bit();
        l++;
    return symb[ l; v - FC[l]];
```

— o —

Decode first 3 symbols of 1000101101 1100

$v = 1; \quad l = 1$

$(v=1 < FC[\overset{2}{l}])$ si

$FC[l]$ with $l = 1$

$v = 2 \cdot 1 + 0;$

$l = 2;$

FC	1	2	3
	2	2	0

symb	0	1	2	3
1	G			
2	C	F		
3	A	B	D	E

$v = 2; \quad l = 2;$

$(2 < FC[2])$ NO

=> return symb[l; v - FC[l]]

$\Downarrow$

symb[2; 0] => E

— - —

$v = 0; \quad l = 1$

$(0 < FC[1])$ yes

$v = 2 \cdot 0 + \text{next\_bit}() = 0 + 0$

$l = 2$

$v = 0 \quad l = 2$

$(0 < FC[2])$ yes

$v = 2 \cdot 0 + \text{next\_bit} = 1$

$l = 3$

$v = 1 \quad l = 3$

$(1 < FC[3])$ no

return symb[l ; v - FC[l]] =>

(with $l = 3$)

symb[3; 1-0=1] => B

$v = \text{next\_bit} = 0 \qquad \ell = 1$

$(0 < FC[\ell])$ yes ($FC[\ell] = 2$)

$v = 2 \cdot 0 + \text{nextb} = 0 + 1 = 1$

$\ell{+}{+};$ // $\ell = 2$

$v = 1 \quad \ell = 2$

$(1 < FC[2])$ yes ($FC[2] = 2$)

$v = 2 \cdot 1 + nb() = 2 + 1 = 3$

$\ell{+}{+};$

$v = 3 \quad \ell = 3$

$(3 < FC[3])$ no ($FC[3] = 0$)

$\Rightarrow$ return $\text{symb}[\ell; v - FC[\ell]] \Rightarrow$

$\Rightarrow \text{Symb}[3; 3] \Rightarrow E$

Decode first 3 symbols of 1000101101 1100

FC	1	2	3
	2	2	0

symb	0	1	2	3
1	C			
2	C	F		
3	A	B	D	E

$v = 1 \quad l = 1$

$1 = v \overset{?}{<} FC[\overset{1}{l}] = 2$

yes =>

$v = 2 + 0 = 2 \quad l{+}{+}$

$V = 2 \quad l = 1$

$2 \overset{?}{<} FC[2] = 2$ no

decode => return $\text{sym}[\overset{2}{l}; \ v - FC[l]] = [2; 0]$

return C

$\sim 0 -$

$v = 0 \quad l = 1$

$v = 0 \overset{?}{<} FC[1] \overset{2}{} \Rightarrow v = 2 \cdot 0 + 0 \quad l{+}{+} \Rightarrow l = 2$

$v = 0 \quad l = 2$

$0 \overset{?}{<} FC[2] \overset{2}{} \Rightarrow v = 2 \cdot 0 + 1 \quad l = \cancel{3}{+}$

$l = 3 \quad v = 1$

$1 < FC[3] \overset{0}{}$ no => decode

ret $\text{symb}[l; \ 1 - FC[3]] = [3, 1]$

=> B

$V = 0 \quad l = 1$

$0 \overset{?}{<} FC[1] \overset{2}{} \Rightarrow v = 2 \cdot 0 + 1 = 1 \quad l{+}{+}$

$v = 1 \quad l = 2$

$1 < FC[2] \overset{2}{} \quad v = 2 \cdot 1 + 1 = 3 \quad l{+}{+}$

$V = 3 \quad l = 2$

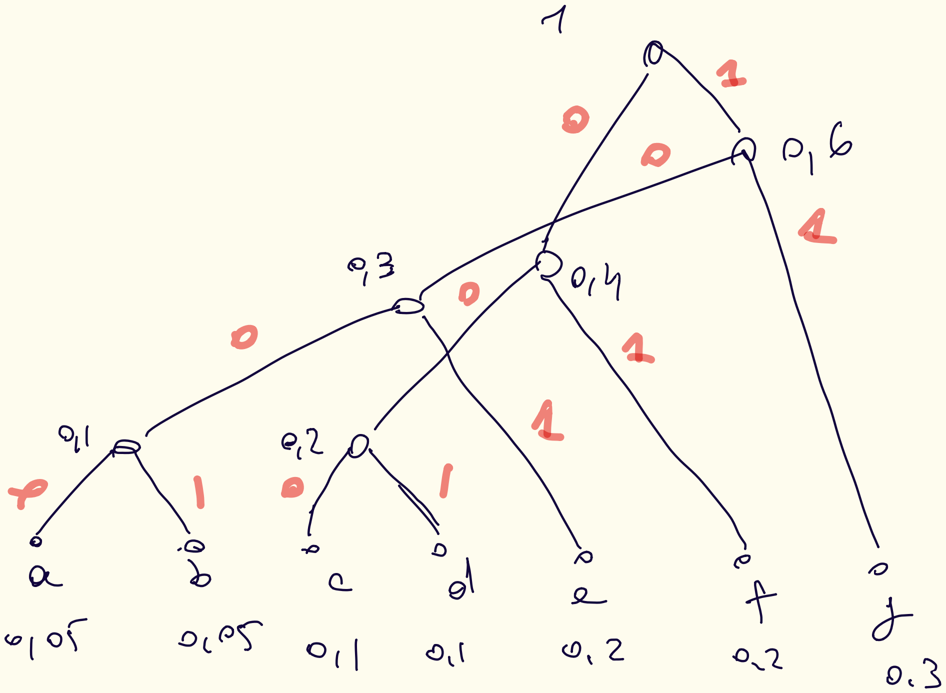
$3 \overset{?}{<} FC[2] \overset{2}{}$ no

ret $\text{symb}[l; \ v - FC[l]] =$

$2; \ 3 - 2 = 1$

$\text{sy}[2; 1] =$ F

Canonical Huffman



a = 1 0 0 0
b = 1 0 0 1
c = 0 0 0
d = 0 0 1
e = 1 0 1
f = 0 1
g = 1 1

	1	2	3	4
NUM	//	2	3	2

symb	0	1	2
1	//		
2	F	G	
3	C	D	E
4	A	B	

ℓ	1	2	3	4
Fc	2	2	1	0

$$FC[\ell_{max}] := 0$$

$$FC[\ell] = \frac{FC[\ell+1] + NUM[\ell+1]}{2}$$

$$FC[3] = \frac{FC[4] + NUM[4]}{2} = \frac{0+2}{2} = 1$$

$$FC[2] = \frac{FC[3] + NUM[3]}{2} = \frac{1+3}{2} = 2$$

$$FC[1] = \frac{FC[2] + NUM[2]}{2} = \frac{2+2}{2} = 2$$

T = G C D

ENCODE:

$SYMB = FC[\ell \text{ of symb}] + rank[symb]$

$G = FC[2] + rank[G] = 2 + 1 = 3$ in 2 bits

11

C = FC[3] + rank[C] = 1 + 0 = 1 in 3 bits => 001

D = FC[3] + rank[D] = 1 + 1 = 2 in 3 bits => 010

Decode with canonical Huffman, 1st 3 symb

000 10 11 01 00 10

ACEB

l	1	2	3	4
Fc	2	2	1	0

symb	0	1	2
1	//		
2	F	G	
3	C	D	E
4	A	B	

```
v = next_bit; l = 1;
while (v < FC[l])
        v = 2·v + next_bit();
        l++;
return symbol[ l ; v - FC[l] ]
```

— o —

v = 0   l = 1

$0 = v < FC[1]$ (= 2) yes $v = 2\cdot 0 + 0 = 0$

l++

v = 0 l = 2

$0 < FC[2]$ (= 2) yes => $v = 2\cdot 0 + 0 = 0$

l++

v = 0 l = 3

$0 < FC[3]$ (= 1) yes $v = 2\cdot 0 + 1 = 1$

l++

v = 1 l = 4

$1 < FC[4]$ (= 0) no => RETURN symb[4 ; 1 - 0]

=> B

— o —

v = 0   l = 1

$0 < FC[1]$ (= 2) yes $v = 2\cdot 0 + 1 = 1$

l++;

v = 1   l = 2

$1 < FC[2]$ (= 2) yes => $v = 2\cdot 1 + 1 = 3$

v = 3   l = 3   l++

$3 < FC[3]$ (= 1) no   RETURN symb[3 ; 3 - 1] = s[3, 2]

=> E

— o —

v = 0 l = 1 $0 < FC[1]$ yes $v = 2\cdot 0 + 1 = 1$ l++

v = 1 l = 2 $1 < FC[2]$ (= 2) yes $v = 2\cdot 1 + 0 = 2$ l++

v = 2 l = 3 $2 < FC[3]$ (= 1) NO => RETURN symb[3 ; 2 - 1] => D

ARITHMETIC CODING

- given a number $x \in [0,1)$

=> • positional representation:

1ST BIT

$\cdot b_1 \; b_2 \; b_3 \; \dots \; b_k \; \dots$ (possible ∞ representation) $b_i \in [0,1)$

$\downarrow$ $1/2 = 2^{-1}$, $1/4 = 2^{-2}$, $1/8 = 2^{-3}$, $1/2^k = 2^{-k}$

if representation is: $.101 = \frac{1}{2} + 0 + \frac{1}{8} = \frac{5}{8}$ => DYADIC FRACTION

VALUE $\frac{v}{2^k}$

COME BACK?

▷ binary representation of v

$\cdot \underbrace{0 \dots 0 \, bin(v)}_{k}$

$\frac{5}{8} = 2^3$

$\cancel{8}$ represent 5 in 3 bits

$.101$

$\frac{5}{16} = \frac{5}{2^4}$ => 5 in 4 bits

$.0101$ → $1/4$, $1/16$

$\frac{1}{4} + \frac{1}{16} = \frac{4+1}{16} = \frac{5}{16}$

- if the number is fractional, the dyadic fraction & binary representation:

CONVERTER (x, k) $x \in [0,1)$

x: number; k: precision/accuracy (how many bits I want to use)

- consider $2x$

if $2x < 1$ => output 0; repeat; (consider twice the value you get)

if $2x \geq 1$ => output 1; $2x - 1$; repeat;

$x = \frac{1}{3}$

$2x = \frac{2}{3} < 1$ => OUTPUT 0; repeat

$2 \cdot \frac{2}{3} = \frac{4}{3} \geq 1$ => OUTPUT 1; $2x - 1 = \frac{4}{3} - 1 = \frac{1}{3}$

∞ process

=> representation is $.\overline{01}$

es. encode $\frac{5}{8}$ $\quad .101$

$2x = \frac{10}{8} \overset{?}{<} 1 \Rightarrow$ no OUTPUT 1

$\frac{10}{8} - 1 = \frac{2}{8} = \frac{1}{4}$

$2 \cdot \frac{1}{4} = \frac{2}{4} = \frac{1}{2} \overset{?}{<} 1$ yes OUTPUT 0 $\Rightarrow$ repeat

$2 \cdot \frac{1}{2} = 1 \overset{?}{<}$ no $\Rightarrow$ OUTPUT 1

$1 - 1 = 0$

LEMMA 1: converter (x,k) MAKES an error 2^{-k}

- given an alphabet of symbols with frequency

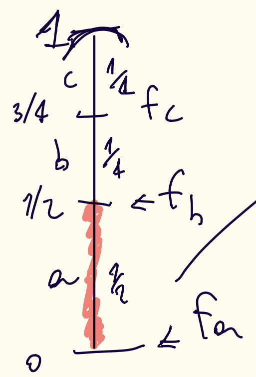
$p(a) = \frac{1}{2} \quad p(b) = p(c) = \frac{1}{4} \Rightarrow \underbrace{f_a = 0 \quad f_b = \frac{1}{2} \quad f_c = \frac{3}{4}}_{\text{cumulative frequency of previous symbols}}$

$T = a\ b\ c\ a$

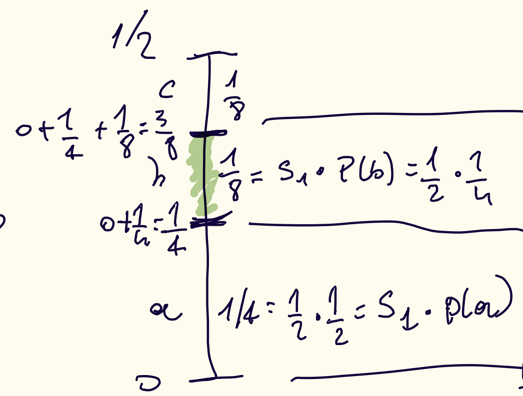
size current interval

$\boxed{S_m = S_{m-1} \cdot P(\sigma) \text{ considered}}$

PARTITION ACCORDING $P(a), P(b), P(c)$



$S_0 = 1$

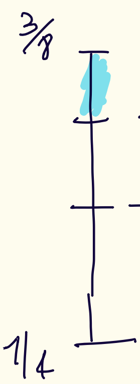


$\rightarrow$ STARTING $+ f_c \cdot S_1$
$0 + \frac{3}{4} \cdot \frac{1}{2} = \frac{3}{8}$

$\rightarrow$ STARTING $+ f_b \cdot S_1 = \frac{1}{2}$
$0 + \frac{1}{2} \cdot \frac{1}{2}$

$\rightarrow$ STARTING $+ f_a \cdot S_1 = 0$
$0 + 0 \cdot \frac{1}{2}$

$S_1 = \frac{1}{2} = 1 \cdot \frac{1}{2} = \frac{1}{2}$



$\rightarrow \frac{1}{4} + f_c \cdot S_2 = \frac{1}{4} + \frac{3}{4} \cdot \frac{1}{8} = \frac{1}{4} + \frac{3}{32} = \frac{8+3}{32} = \frac{11}{32}$

$\rightarrow \frac{1}{4} + f_b \cdot S_2 = \frac{1}{4} + \frac{1}{2} \cdot \frac{1}{8} = \frac{1}{4} + \frac{1}{16} = \frac{5}{16}$

$\rightarrow \frac{1}{4} \cdot f_a \cdot S_2 = \frac{1}{4} + 0 \cdot \frac{1}{8} = \frac{1}{4}$ ✓

$S_2 = \frac{1}{2} \cdot P(b) = \frac{1}{2} \cdot \frac{1}{4} = \frac{1}{8}$

$3/8$

$\longrightarrow \frac{11}{32} + f_c \cdot S_3 = \frac{11}{32} + \frac{3}{4} \cdot \frac{1}{32} = \frac{11}{32} + \frac{3}{128} = \frac{44+3}{128} = \frac{47}{128}$

$\longrightarrow \frac{11}{32} + f_b \cdot S_3 = \frac{11}{32} + \frac{1}{2} \cdot \frac{1}{32} = \frac{11}{32} + \frac{1}{64} = \frac{22+1}{64} = \frac{23}{64}$

$11/32 \longrightarrow \frac{11}{32} + f_a \cdot S_3 = \frac{11}{32} + 0 \cdot \frac{1}{32} = \frac{11}{32}$

$S_3 = S_2 \cdot P(c) = \frac{1}{8} \cdot \frac{1}{4} = \frac{1}{32}$

$23/64$

$\longrightarrow \frac{11}{32} + f_c \cdot S_4 = \frac{11}{32} + \frac{3}{4} \cdot \frac{11}{64} = \frac{11}{32} + \frac{33}{256} = \frac{44+33}{256} = \frac{88}{256}$

$\longrightarrow \frac{11}{32} + f_b \cdot S_4 = \frac{11}{32} + \frac{1}{2} \cdot \frac{11}{64} = \frac{11}{32} + \frac{11}{128} = \frac{33+11}{128} = \frac{44}{128}$

$11/32 \longrightarrow \frac{11}{32} + \underset{0}{\underset{\shortparallel}{f_a}} \cdot \underset{11/64}{\underset{\shortparallel}{S_4}} = \frac{11}{32}$

$S_4 = S_3 \cdot p(a) = \frac{11}{32} \cdot \frac{1}{2} = \frac{11}{64}$

- ARITHMETIC CODE BRINGS 2 INTEGERS => left END-POINT of the interval & l_n + size

- $\boxed{[\, l_n \,,\, l_n + S_n \,]}$

beginning: $l_0 = 0$, $S_n = 1$

in every STEP:

character in pos. n => to be encoded

- $\begin{cases} S_{n+1} = S_n \cdot P[T[n+1]] \\ l_{n+1} = l_n + f_{T[n+1]} \cdot S_n \end{cases}$

$\longrightarrow S_{n+1} = P(T[n+1]) \cdot S_n = P(T[n+1]) \cdot p(T[n]) \cdot S_{n-1} =$

$= P(T[n+1]) \cdot P(T[n]) \cdot P(T[n-1]) \cdots P(T[1])$

=> since I'm looking for $T = a\ b\ c\ a$

=> final size => $\frac{1}{2} \cdot \frac{1}{4} \cdot \frac{1}{4} \cdot \frac{1}{2} =$

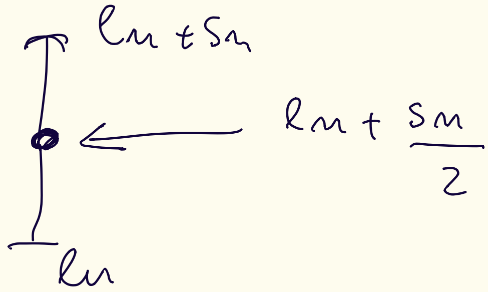
if $T = \{a, b\}$

$\rightarrow \left[\frac{5}{16}, \frac{11}{32} \right)$

$$\left[\underbrace{\frac{5}{16}}_{=\,l_2}, \underbrace{\frac{5}{16}}_{=\,l_2} + \underbrace{\frac{1}{8} \cdot \frac{1}{4}}_{s_2} \right)$$

size of the interval

ARITMETIC CODING SENDS THE middle element



ENCODE WITH ARITHMETIC CODING THE TEXT $T = B\ B\ A\ C$

(prob are not given)

$P(a) = \frac{1}{4}$ $\quad P(b) = \frac{2}{4} = \frac{1}{2}$ $\quad P(c) = \frac{1}{4}$

$F_a = 0$ $\quad F_b = F_a + P(a) = 0 + \frac{1}{4} = \boxed{\frac{1}{4}}$

$F_c = F_b + P(b) = \frac{1}{2} + \frac{1}{4} = \boxed{\frac{3}{4}}$

$T = B\ B\ A\ C$

1

c $1/4$ $\rightarrow F_c$

b $1/2$ $\rightarrow F_b$

a $1/4$ $\rightarrow F_a$

0

$F_a = 0$ $\quad F_b = \frac{1}{4}$ $\quad F_c = \frac{3}{4}$

$S_0 = 1$

$3/4$

c $\rightarrow$ STARTING $+ F_c \cdot S_1 = \frac{1}{4} + \frac{3}{4} \cdot \frac{1}{2} = \frac{1}{4} + \frac{3}{8} = \frac{5}{8}$

b $\rightarrow$ STARTING $+ F_b \cdot S_1 = \frac{1}{4} + \frac{1}{4} \cdot \frac{1}{2} = \frac{1}{4} + \frac{1}{8} = \frac{3}{8}$

a

$1/4$ $\rightarrow$ STARTING $+ F_a \cdot S_1 = \frac{1}{4} + 0 \cdot \frac{1}{2} = \frac{1}{4}$

$S_1 = S_0 \cdot P(b) = 1 \cdot \frac{1}{2} = \frac{1}{2}$

$16 \cdot 4 = 64$

$5/8$

c $\rightarrow \frac{3}{8} + \frac{3}{4} \cdot \frac{1}{4} = \frac{3}{8} + \frac{3}{16} = \frac{9}{16}$

b $\rightarrow \frac{3}{8} + \frac{1}{4} \cdot \frac{1}{4} = \frac{3}{8} + \frac{1}{16} = \frac{7}{16}$

a $\rightarrow \frac{3}{8} + 0 \cdot \frac{1}{4} = \frac{3}{8}$

$\frac{3}{8}$

$\frac{1}{16}$

c $\quad \frac{3}{8} + \frac{3}{4} \cdot \frac{1}{16} = \frac{3}{8} + \frac{3}{64} = \frac{27}{64}$

b $\rightarrow \frac{3}{8} + \frac{1}{4} \cdot \frac{1}{16} = \frac{3}{8} + \frac{1}{64} = \frac{25}{64}$

a $\rightarrow \frac{3}{8} + 0 \cdot \frac{1}{16} = \frac{3}{8}$

$\frac{3}{8}$

$S_2 = S_1 \cdot P(b) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$

$S_3 = S_2 \cdot P(a) = \frac{1}{4} \cdot \frac{1}{4} = \frac{1}{16}$

$\frac{7}{16}$

c $\rightarrow \frac{27}{64} + \frac{3}{4} \cdot \frac{1}{64} = \frac{27}{64} + \frac{3}{256} = \frac{108+3}{256} = \frac{111}{256}$

b $\rightarrow \frac{27}{64} + \frac{1}{4} \cdot \frac{1}{64} = \frac{27}{64} + \frac{1}{256} = \frac{109}{256}$

a $\rightarrow \frac{27}{64} + 0 \cdot \frac{1}{64} = \frac{27}{64}$

$\frac{27}{64}$

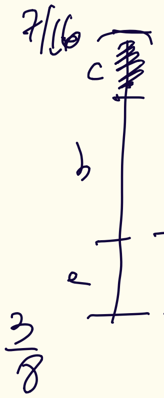
$S_4 = S_3 \cdot P(c) = \frac{1}{16} \cdot \frac{1}{4} = \frac{1}{64}$

$\rightarrow$ NON VA FATTO, focus

$|T|$ STEPS

$64 \cdot 4 = 256$

$27 \cdot 4 = 108$



$\frac{3}{8}+\frac{3}{4}\cdot\frac{1}{16}=\frac{3}{8}+\frac{3}{64}=\frac{27}{64}$

$\to \frac{3}{8}+\frac{1}{4}\cdot\frac{1}{16}=\frac{3}{8}+\frac{1}{64}=\frac{25}{64}$

$\to \frac{3}{8}+0\cdot\frac{1}{16}=\frac{3}{8}$

$S_3 = S_2\cdot P(a) = \frac{1}{4}\cdot\frac{1}{4}=\frac{1}{16}$

FINAL RANGE

$P(\alpha)\ \forall\alpha$ (symbols)

OUTPUT: $x \in [l_n ; l_n + S_n]$

$n=4$

- for instance choose middle element of range of C

$\Rightarrow \left(\frac{27}{64}+\frac{7}{16}\right):2 =$

$=\left(\frac{27+28}{64}\right):2 =$

$=\frac{55}{64}\cdot\frac{1}{2}=\frac{55}{128}$

$\Rightarrow$ encode $\frac{55}{128}$ in # bits $\left\lceil \log_2 \frac{2}{S}\right\rceil =$ $\left\lceil \log_2 \frac{2}{\frac{1}{16}}\right\rceil = \log_2$

CONVERTER(x, k)

$\alpha = 2x$

if $2x < 1$ $\Rightarrow$ OUTPUT 1; REPEAT;

else $(2x \geq 1)$ $\Rightarrow$ OUTPUT 0; $\alpha - 1$; repeat

converter$\left(\frac{55}{128}\right.$

ARITHMETIC CODING - LAST PART: EMITS

$|T| = 4$

$T = a\ b\ a\ c$

$P(a) = \frac{1}{2}$ $\quad P(b) = \frac{1}{4}$ $\quad P(c) = \frac{1}{4}$

$f(a) = 0$ $\quad f(b) = \frac{1}{2}$ $\quad f(c) = \frac{3}{4}$

1

c $\rightarrow \frac{3}{4}$

b $\rightarrow \frac{1}{2}$

a

0 $\rightarrow 0$

$S_0 = 1$

$\frac{1}{2}$

c $\rightarrow 0 + \frac{1}{2} \cdot \frac{3}{4} = \frac{3}{8}$

b $\rightarrow 0 + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$

a

0 $\rightarrow 0 + 0 \cdot \frac{1}{2} = \frac{1}{2}$

$S_1 = 1 \cdot \frac{1}{2} = \frac{1}{2}$

$\frac{3}{8}$

c $\rightarrow \cancel{\frac{3}{8}}\frac{1}{4} + \frac{3}{4} \cdot \frac{1}{8} = \cancel{\frac{3}{8}}\frac{1}{4} + \frac{3}{32} = \frac{11}{32}$

b $\rightarrow \cancel{\frac{3}{8}}\frac{1}{4} + \frac{1}{2} \cdot \frac{1}{8} = \cancel{\frac{3}{8}}\frac{1}{4} + \frac{1}{16} = \frac{5}{16}$

a

$1/4 \xrightarrow{P(b)} \cancel{\frac{3}{8}}\frac{1}{4} + 0 \cdot \frac{1}{8} = \cancel{\frac{3}{8}}\frac{1}{4}$

$S_2 = \frac{1}{2} \cdot \frac{1}{4} = \frac{1}{8}$

$5/16$

c $\rightarrow \frac{1}{4} + \frac{3}{4} \cdot \frac{1}{16} = \frac{1}{4} + \frac{3}{64} = \frac{19}{64}$

b $\rightarrow \frac{1}{4} + \frac{1}{2} \cdot \frac{1}{16} = \frac{1}{4} + \frac{1}{32} = \frac{9}{32}$

a

$1/4 \rightarrow \frac{1}{4}$

$S_3 = \frac{1}{8} \cdot \frac{1}{2} = \frac{1}{16}$

$\Rightarrow S_4 = \frac{1}{16} \cdot P(c) = \frac{1}{16} \cdot \frac{1}{4} = \frac{1}{64}$

$16 \cdot 4 = 64$

OUTPUT $\left[\overset{\ell}{\frac{19}{64}} ; \frac{5}{16}\right) \rightarrow$ # in the middle of the interval

$S =$

$\left(\frac{5}{16} + \frac{19}{64}\right)\frac{1}{2} = \frac{20+19}{64} \cdot \frac{1}{2} =$

$= \frac{39}{128} = \ell + \frac{S}{2} =$

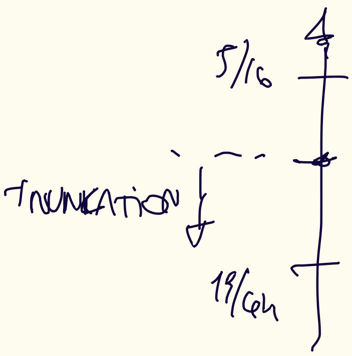
$= \frac{19}{64} + \frac{1}{128} = \frac{39}{128}$ ✓

16	1
32	2
48	3
64	4

- Need to represent $\frac{39}{128}$ with some bits

$\frac{v}{2^k} = \frac{bin(v)}{k \text{ \# of bits}}$ (full representation)

- we want use minimizing bits



- THEO: truncate to a # of bits $d = \lceil \log_2 \frac{2}{S} \rceil$ bits

$$d = \lceil \log_2 \frac{2}{1/64} \rceil = \lceil \log_2 128 \rceil = 7 \text{ bits}$$

CONVERTER $\left(l + \frac{S}{2} ; d \right)$

CONVERTER $\left(\frac{39}{128} ; 7 \right)$

$\alpha = 2x$

if $(\alpha < 1)$ output 1; repeat

if $(\alpha \geq 1)$ output 0; $\alpha - 1$; repeat

...

$\Rightarrow$ 39 in 7 bits $= 0.0100111$

$\frac{1}{4}$ $\frac{1}{32}$ $\frac{1}{64}$ $\frac{1}{128}$

- Decode the compressed sequence $\langle 2, 1101 \rangle$ via arithmetic coding

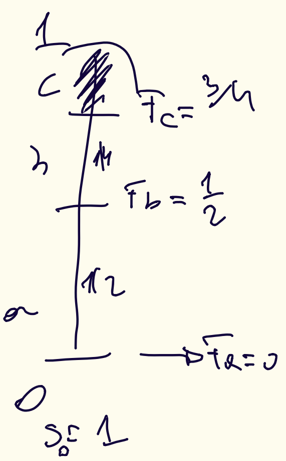
with $p(a) = \frac{1}{2}$ $p(b) = \frac{1}{4}$ $p(c) = \frac{1}{4}$

1 1 0 1

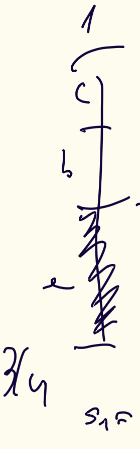
$\frac{1}{2^1}$ $\frac{1}{2^2}$ $\frac{1}{2^4}$

$$\frac{1}{2} + \frac{1}{4} + \frac{1}{16} = \frac{8+4+1}{16} = \frac{13}{16} \Rightarrow \text{search this}$$

START $[0, 1]$



$\longrightarrow$ p



$\rightarrow \frac{3}{4} + \frac{3}{4} \cdot \frac{1}{4} = \frac{3}{4} + \frac{3}{16} = \frac{15}{16}$

$\rightarrow \frac{3}{4} + \frac{1}{2} \cdot \frac{1}{4} = \frac{3}{4} + \frac{1}{8} = \frac{6+1}{8} = \frac{7}{8} = \frac{14}{16}$

$\rightarrow \frac{3}{4} + 0 \cdot \frac{1}{4} = 3/4 = \frac{12}{16}$

$\Rightarrow T = C, A$

ARITMETIC CODING

$P(a) = \frac{1}{8} = \frac{1}{2^3}$ $P(b) = \frac{1}{4} = \frac{1}{2^2}$ $P(c) = \frac{5}{8} = \frac{5}{2^3}$

specify which is the length in bits of $T = aabbaa$

$\hookrightarrow \left\lceil \log_2 \frac{2}{s} \right\rceil \to$ last $\overset{\text{interval}}{\frac{UP + DOW}{2}}$

The length in bits is $b = \left\lceil \log_2 \frac{2}{s} \right\rceil$ where $s = \prod p[T(i)]$

$\prod p[T(i)] = (P(a))^4 \cdot (p(b))^2 = \left(\frac{1}{8}\right)^4 \cdot \left(\frac{1}{4}\right)^2 =$

$= \left[\frac{1}{2^3}\right]^4 \cdot \left[\frac{1}{2^2}\right]^2 = \frac{1}{2^{12}} \cdot \frac{1}{2^4} = \frac{1}{2^{16}}$

$d = \left\lceil \log_2 \frac{2}{s} \right\rceil = \left\lceil \log_2 2 : \frac{1}{2^{16}} \right\rceil =$

$= \left\lceil \log_2 2 \cdot 2^{16} \right\rceil = \left\lceil \log_2 2^{17} \right\rceil = 17$ bits

• Given $T = AAB$ arithmetic coding

$P(a) = \frac{2}{3}$ $\quad P(b) = \frac{1}{3}$

1

b $\frac{1}{3}$ $\quad F_b = 2/3$

a $\frac{2}{3}$ $\quad \to F_a = 0$

0

$S_0 = 1$

$2/3$

b $\to$ STARTING $+ S_1 \cdot F_b = 0 + \frac{2}{3} \cdot \frac{2}{3} = \frac{4}{9}$

a

0 $\to$ STARTING $+ S_1 \cdot F_a = 0 + \frac{2}{3} \cdot 0 = 0$

$S_1 = 1 \cdot \frac{2}{3} = \frac{2}{3}$

$4/9$

b $\quad 0 + \frac{4}{9} \cdot F_b = \frac{4}{9} \cdot \frac{2}{3} = \frac{8}{27}$

$\to 0$

a

0 $\to 0 + \frac{4}{9} \cdot F_a = 0 + \frac{4}{9} \cdot 0 = 0$

$S_2 = \frac{2}{3} \cdot p(a) = \frac{2}{3} \cdot \frac{2}{3} = \frac{4}{9}$

$$\text{size} = \left(\frac{4}{9} + \frac{8}{27}\right) : 2 = \left(\frac{12+8}{27}\right) : 2 = \frac{20}{27} \cdot \frac{1}{2} = \frac{10}{27}$$

Din d bits

last

$$S_3 = \frac{4}{9} \cdot p(b) = \frac{4}{9} \cdot \frac{1}{3} = \frac{4}{27}$$

$$d = \left\lceil \log_2 \frac{2}{S_3} \right\rceil = \left\lceil \log_2 \cancel{2} \cdot \frac{27}{\cancel{4}\,2} \right\rceil = \left\lceil \log_2 13{,}5 \right\rceil = 4 \text{ bits}$$

$\frac{10}{27}$ in 4 Bits

converter(x, k) $\quad x \to \frac{10}{27}$, $\quad k \to 4$

if $2x < 1 \Rightarrow$ repeat emit 0

if $2x \geq 1$ = emit 1 $\quad 2x - 1$

$27 \cdot 2 = 54$

$\frac{10}{27} \cdot 2 = \frac{20}{27} < 1 \Rightarrow$ emit 0

$\frac{20}{27} \cdot 2 = \frac{40}{27} \geq 1 \Rightarrow$ emit 1 $\qquad \frac{40}{27} - 1 = \frac{40-27}{27} = \frac{13}{27}$

$\frac{13}{27} \cdot 2 = \frac{26}{27} < 1 \Rightarrow$ emit 0

$\frac{26}{27} \cdot 2 = \frac{52}{27} \geq 1 \Rightarrow$ emit 1

OUTPUT $\langle .0101, 3 \rangle$

$l_n + s_n = \frac{4}{27} + \frac{8}{27} = \frac{12}{27}$ $\quad (n = 3)$

• Decompress arithmetic coding

$p(a) = \frac{1}{2}$ $p(b) = \frac{1}{4}$ $p(c) = \frac{1}{4}$

1st 2 symbols

$T = 1\ 1\ 0\ 1$

$\frac{1}{2^1}$ $\frac{1}{2^2}$, $\frac{1}{2^4}$

$\frac{1}{2} + \frac{1}{4} + 1/16 = \frac{8+4+1}{16} = \frac{13}{16} = \frac{13}{2^4}$

1

c 1/4 $\rightarrow F_c = \frac{1}{4} + \frac{1}{2} = \frac{1+2}{4} = \frac{3}{4}$

b 1/4 $F_b = \frac{1}{2}$

a 1/2

0 $\rightarrow 0 = F_a$

$s_0 = 1$

C

1

c $\rightarrow \frac{3}{4} + \frac{1}{4} \cdot \frac{3}{4} = \frac{3}{16} + \frac{3}{4} = \frac{3+12}{16} = \frac{15}{16}$

b

$\rightarrow \frac{3}{4} + \frac{1}{4} \cdot \frac{1}{2} = \frac{3}{4} + \frac{1}{8} = \frac{7}{8} = \frac{14}{16}$

a $\rightarrow$ A

13/16

3/4 $F_a = \text{start} + S_1 \cdot F_a = \frac{3}{4} = \frac{12}{16}$

$S_1 = 1 \cdot p(c) = 1 \cdot \frac{1}{4}$

$\frac{7}{8} \cdot \frac{2}{2} = \frac{14}{16}$

=> 2 symbols

=> CA

Arithmetic coding, decompress 3 bits = 1 1 1

$p(a) = \frac{1}{2}$ $p(b) = \frac{1}{4}$ $p(c) = \frac{1}{4}$

$1\ 1\ 1 \to \frac{1}{2^1}\ \frac{1}{2^2}\ \frac{1}{2^3}$

$\frac{1}{2} + \frac{1}{4} + \frac{1}{8} = \frac{4+2+1}{8} = \frac{7}{8}$

1

c 1/4 $\to F_c = \frac{3}{4} = \frac{6}{8}$

b 1/4 $\to F_b = \frac{1}{2} = \frac{4}{8}$

a 1/2 $\to F_a = 0$

0

$S_0 = 1$

1

c $\to \frac{3}{4} + \frac{3}{4} \cdot \frac{1}{4} = \frac{3}{4} + \frac{3}{16} = \frac{15}{16}$

b $\to \frac{3}{4} + \frac{1}{2} \cdot \frac{1}{4} = \frac{3}{4} + \frac{1}{8} = \frac{7}{8}$

a

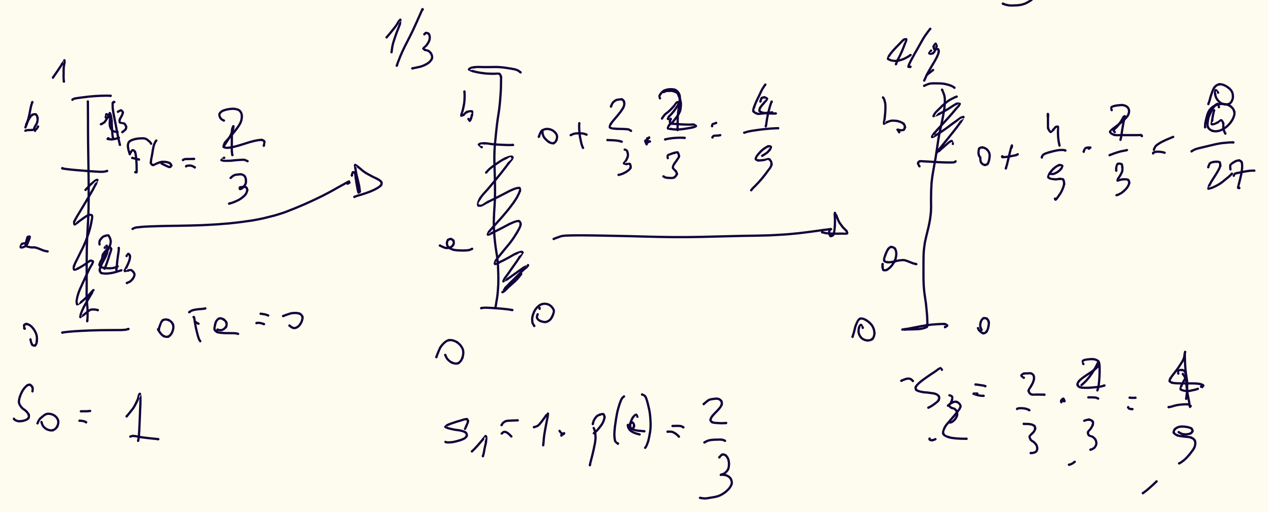
$\frac{3}{4} \to \frac{3}{4}$

$S_1 = 1 \cdot p(c) = 1 \cdot \frac{1}{4} = \frac{1}{4}$

$T = c\ B$

ARITMETIC CODING

$T = \underline{a}\,\underline{a}\,b \qquad p(a) = \frac{2}{3} \qquad p(b) = \frac{1}{3}$



$b = \lceil \log_2 \frac{2}{S_3} \rceil$

encode the middle # in range $= \left(\frac{8}{27} + \frac{4}{9}\right)$

$\frac{4}{27} + \frac{2}{9}$

$S_3 = \frac{4}{9}\cdot\frac{1}{3} = \frac{4}{27}$

$= \left(\frac{8+12}{27}\right)$

$b = \lceil \log_2 2^1 \cdot \frac{27}{4_2} \rceil = \lceil \log_2 13{,}5 \rceil = 4$ bits

$= \frac{20}{27} \cdot \frac{1}{2} = \frac{10}{27}$

in 10/27

converter$\left(\frac{10}{27}, 4\right)$

$2 \cdot \frac{10}{27} = \frac{20}{27} < 1 \Rightarrow$ OUTPUT $\underline{0}$

$2 \cdot \frac{20}{27} = \frac{40}{27} \geq 1 \Rightarrow$ OUPUT $\underline{1} \qquad \frac{40}{27} - 1 = \frac{13}{27}$

$\frac{13}{27} \cdot 2 = \frac{26}{27} < 1 \Rightarrow$ OUTPUT $\underline{0}$

$\Rightarrow \langle .0101, 3 \rangle$

$\frac{26}{27} \cdot 2 = \frac{54}{27} \geq 1 \Rightarrow$ OUTPUT $\underline{1}$