PROJ. 3 SEANCE 1

1

Dimostrare che C1 è non decif. e C2 sì, mediante prova entropia

C1

C_1 n'est pas préfixé.

'10' et aussi le debut du code '1001'

C_1 n'est pas uniquement dechiffrable

Ex. 10010

↓

$S_5 - S_1$ oppure $S_2 - S_1 - S_2$

C2

C_2 est préfixé

↓

Puisqu'il est préfixé il est donc UNIQUEMENT DECHIFFRABLE

- Calcul longueur moyenne de chaque code

$$\bar{N} = \sum_{i=1}^{N} P_i N_i$$

N_i = n. bit parola
P_i = probabilità

$$N_1 \geq \sum_i p_i N_i = (0.5 \cdot 1) + (0.18 + 0.14) \cdot 2 + (0.12 \cdot 3) + 0.06 \cdot 4 =$$
$$= 1.74$$

$$N_2 \geq \sum_i p_i N_i = (0.5 + 0.18 + 0.14) \cdot 2 + (0.12 + 0.06) \cdot 3 =$$
$$= 2.18$$

- Determinions $\langle N_{min} \rangle$, la longueur moyenne minimum de mots de tout codage binaire → ENTROPIE

$$H = -\sum p_i \log_2 p_i$$
$$= -(0.5 \log_2 0.5 + 0.18 \log_2 0.18 + 0.14 \log_2 0.14 +$$
$$0.12 \log_2 0.12 + 0.06 \log_2 0.06)$$
$$= 1.953$$

Em general

$N < H$ CODE PAS UNIQUEMENT DECHIFFRABLE

$N > H$ CODE UNIQUEMENT DECHIFFRABLE (codice ottimo)

$N = H$ CODE ABSOLUMENT OTTIMO (condizione ideale)

→ $N_1 < H$ code pas dec.

$N_2 > H$ code dec (on peux l'optimiser)

(2)

Choisir code avec 5 charactèr
- 2 avec longeur 2
- 2 " " 3
- 1 " " 4

Ex. code N:

$S_1 = 00$

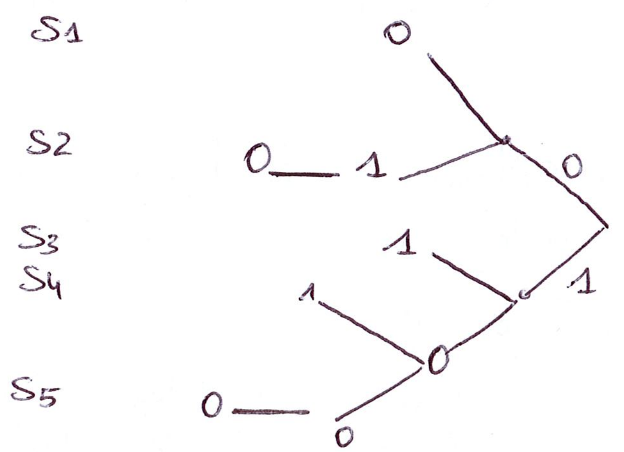
$S_2 = 11$

$S_3 = 101$

$S_4 = 010$

$S_5 = 1000$

→ code bien dechiffrable puisqu'on l'a choisi préfixé



→ On a de pont qu'elles ne sont pas nécessaries pour que le code soit déchiffrable, on peut avoir un code plus courte →

On a le code N_{cor}.

$S_1 = 00$
$S_2 = 11$
$S_3 = 101$
$S_4 = 01$
$S_5 = 100$

$$\langle N \rangle = (p_1 + p_2) \cdot 2 + (p_3 + p_4) \cdot 3 + p_5 \cdot 4)$$
$$\langle N_{cor.} \rangle = (p_1 + p_2 + p_3) \cdot 2 + (p_4 + p_5) \cdot 3$$

$\longrightarrow$ N_{cor} plus court !

(2)

$p_1 = 0.45$
$p_2 = 0.23$
$p_3 = 0.16$
$p_4 = 0.14$
$p_5 = 0.05$

$$\langle N_{cor} \rangle = (p_1 + p_2 + p_3) 2 + (p_4 + p_5) 3 = 2.19$$

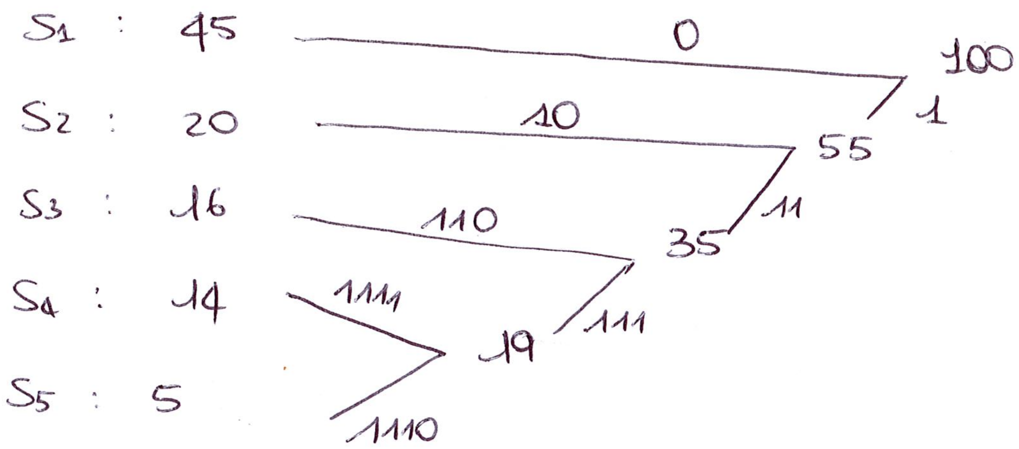
$$H = - \sum_i p_i \log p_i$$
$$= - (0.45 \log_2 0.45 + 0.23 \log_2 0.23 + 0.16 \log 0.16 + 0.14 \log_2 0.14 + 0.05 \log_2 0.05)$$
$$= 2.019$$

$\Longrightarrow \langle N_{cor} \rangle > H$ codice ottimo

$\downarrow$

mais on peux l'optimiser pour avoir $N = H$

③ On cherche un code plus optimisé avec le MÉTHODE DE HUFFMAN



↓

S1 : 0
S2 : 10
S3 : 110
S4 : 1111
S5 : 1110

$\longrightarrow \langle N_{Huffman} \rangle = (p_1 \cdot 1) + (p_2 \cdot 2) + (p_4 + p_5) \cdot 4 + p_3 \cdot 3 =$

$= 2.09 \quad < \langle N_{cor} \rangle$

↓

~~(N_{Huff} migliore perché più vicino a H)~~

$N_{Huff.}$ MEILLEUR CODE PARCEQUE PLUS PROCHE À H.

③

CODAGE RLE et QUANTIFICATION

① On a la seq. X

↓

Un codage RLE est

(2,6) (5,3) (7,8) (8,6) (1,6) avec (X,m) pour m répétitions de X

↓

8 bit per lettera (x codé sur 1 octet)
5 bit per codificare le repetioni (m)

↓

per coppia (X, m) → 13 bits (8+5)

↓

le champ m peut coder jusqu'au $2^5 - 1 = 31$ mais puisqu'il n'y peut pas avoir 0 repetitions, on peut décaler le le code et "00000" équivaut a 1 répétition
"00010" équivaut a 3 répétition

↓

m jusqu'au 32

Un couple codé sur 13 bits peut remplacer 32 entiers consecutives identiques donc $32 \cdot 8 = 256$ bits

↓

TAUX de COMPRESSION $\tau = \frac{13}{256} = 5\%$

Dans le cas idéal où on veut coder une séquence de n octet, on peut coder la seq avec $\left(\frac{n}{32}\right) \rightarrow ent_{sup}\left(\frac{n}{32}\right)$ (*)

↓

Il suffira donc $ent_{sup}\left(\frac{n}{32}\right) \cdot \frac{13}{8} \longrightarrow ent_{sup}\left(\frac{n}{32}\right)\frac{13}{8}$

Ex.
- une suite de 64 '2' normalement il est codé en 64 octets
  ↓
  avec RLE : $(2 \cdot 13) = 26$ bits $\longrightarrow$ 4 octets

- une suite de 10 chiffres tous différents on va avoir $(10 \cdot 13) = 130$ bits $\longrightarrow$ 16 octets
  ↓
  Invece in codage naturel on a 10
  ↓

En général : le CODAGE RLE est performant dans le cas où il y a beaucoup d'octets consécutifs identiques, mais il va donner un codage plus volumineux dans le cas où les octets consécutifs sont toujours différentes.

(*) $ent_{sup}(x) = x$ si x est entier, $ent(x)+1$ sinon

3

On va quantifier la sequence a de 29 character

- $Q = 4$ on obtient

  0 0001 0101 1111 1121 2021 1000 0000

- $Q = 8$ on obtient

  0 0000 0000 0000 0010 1010 0000 0000.

4

Pour reconstruire la sequence initial a partir de la quantif.
on peut multiplier par Q

$Q = 4$ 0 0004 0404 4444 4484 8084 4000 0000 $= y_1$

$Q = 8$ 0 0000 0000 0000 0080 8080 0000 0000 $= y_2$

On va calculer l'ERREUR QUADRATIQUE MOYENNE pour $Q = 4$ et $Q = 8$ avec

$$\varepsilon = \frac{1}{m} \sum (y - x)^2$$

m = n'carac. x
x = seq. initiale
y = seq. reconstr.

On a :

- pour $Q = 4$: $\varepsilon_4^2 = 4.10 \longrightarrow \varepsilon = 2.02$
- pour $Q = 8$: $\varepsilon_8^2 = 17.34 \longrightarrow \varepsilon = 4.16$

$\longrightarrow \varepsilon_4^2 = (7)(2)^2 + (9) \times (3)^2 + 10 / 29 = 4.10$

$\longrightarrow \varepsilon_8^2 = (7)(2)^2 + (3) \times (3)^2 + 7^2 * 6 + 6 \times 5^2 + 4 / 29$

$\Longrightarrow$ Plus la quantification est faible plus l'erreur est faible

⑤ On applique le CODE RLE sur la séquence quantifiée reconstruite naïvement :

ⓐ Pour $Q = 4$

0 0004 0404 4444 4484 8084 4000 0000

Devient (0,4) (4,1) (0,1) (4,1) (0,1) (4,7) (8,1) (4,1) (8,1) (0,1) (8,1) (4,2) (0,7)

↳ $13 \cdot 13 = 169$ bits (contre 232 (29 concec. x 8 bits) à la base)

ⓑ Pour $Q = 8$

0 0000 0000 0000 0080 8080 0000 0000

Devient (0,15) (8,1) (0,1) (8,1) (0,1) (8,1) (0,9)

↳ $7 \times 13 = 91$ bits (contre 232 à la base)

⟹ En general

- la QUANTIFICATION va a augmenter le nombre de répétition qui permet de diminuer la taille de la séquence en RLE
- Avec la quantification plus grand on reduit plus la taille mais on a un erreur plus grand

↓

Il faut trouver un compromis entre QUALITÉ et QUANTITÉ si on va compresser sans trop de pert avec ces méthods.