

Téchniques de codage

1. Introduction Les techniques de codage de l'information permettent de transformer une source d'alphabets en un message binaire en assurant son décodage instantané et sans ambiguïté.

2. Codage d'Huffman

C'est un codage de source avec des mots de longueur variable. Mis au point en 1952, basé sur les probabilités des caractères de la source. Il a été proposé par David Huffman adapté par Robert Fano. Le code de Huffman est un code préfixe et optimal pour certain modèle. La génération de ce code est inspirée de deux observations:

1. Dans un code optimal, les symboles les plus fréquents (ayant la plus haute probabilité d'occurrence) auront des mots de code de tailles plus petites que ceux qui sont les moins fréquents.
2. Dans un code optimal, les deux symboles qui apparaissent le moins fréquemment auront la même taille.

L'algorithme de Huffman se base sur une construction progressive de l'arbre binaire préfixe. La construction de l'arbre est ascendante c'est-à-dire elle commence des feuilles et monte progressivement pour arriver à la racine.

A chaque phase de construction uniquement les nœuds orphelins sont considérés. Par nœud orphelin on désigne un nœud qui n'a pas de nœud père. La construction de l'arbre s'appuie aussi sur les valeurs des probabilités des alphabets à coder. Ces probabilités sont considérées comme des poids des nœuds.

L'algorithme de Huffman s'annonce comme suit :

Le principe est très simple : les caractères d'un fichier qui **apparaissent le plus souvent** doivent être codés en un **minimum d'espace** possible (caractères les plus fréquents étant représentés par des mots de code courts).

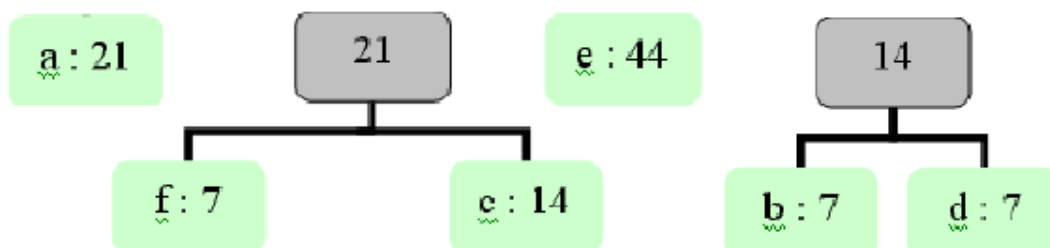
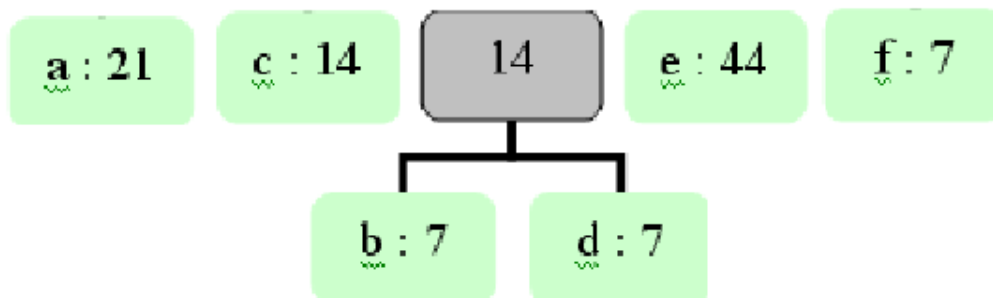
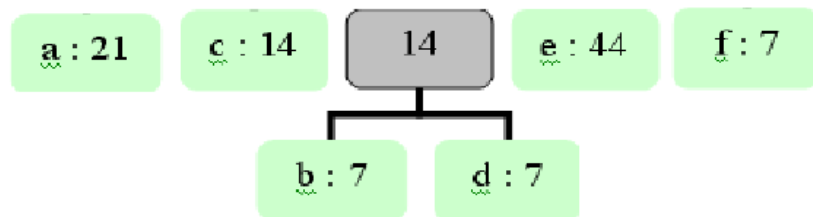
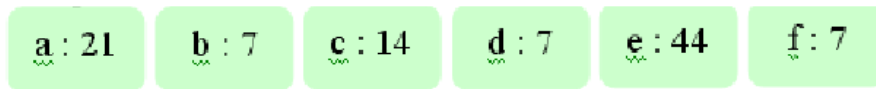
2.1 Principe du codage

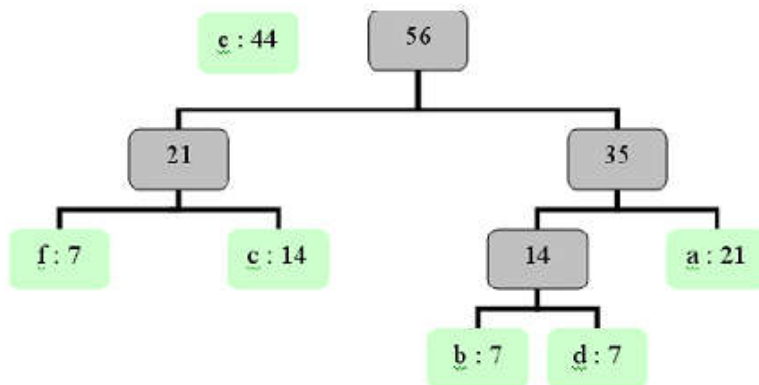
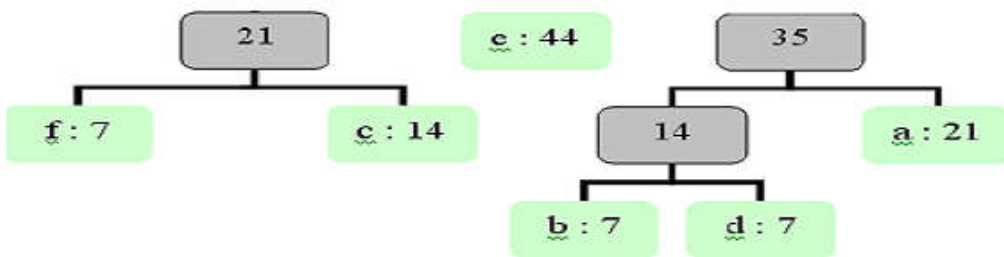
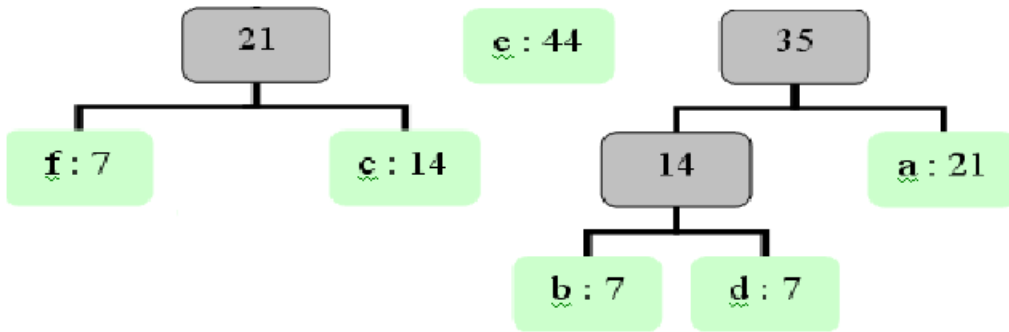
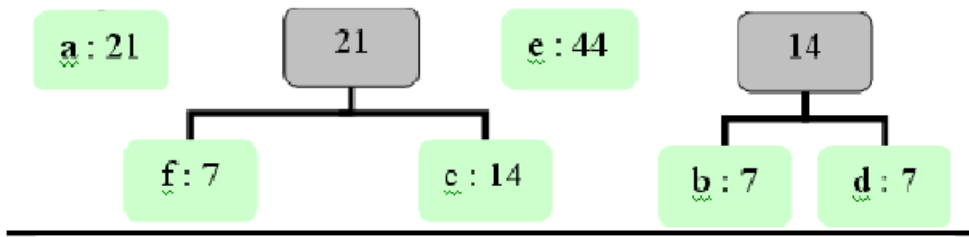
- On commence par choisir les deux symboles qui ont le moins d'occurrences
- on leur donne les codes 0 et 1
- on les regroupe dans un arbre binaire auquel on attribue un nombre d'occurrences = la somme des deux symboles qu'il regroupe
- on choisit deux symboles et/ou nœuds d'arbres qui ont le moins d'occurrences et on les regroupe
- on regroupe ainsi les symboles en fonction du nombre d'occurrences qu'ils représentent.

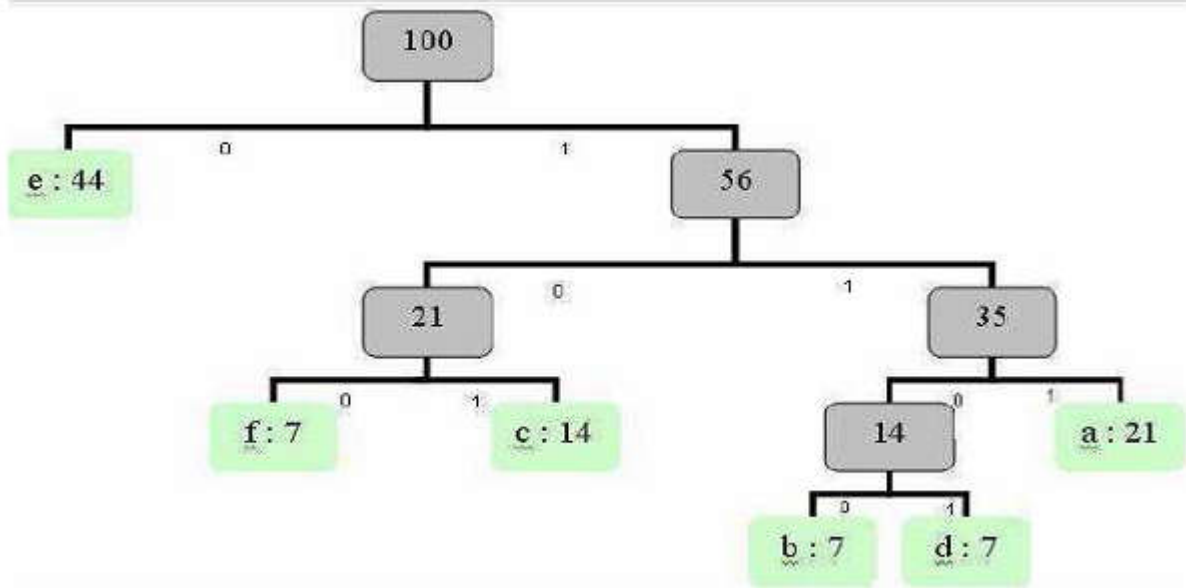
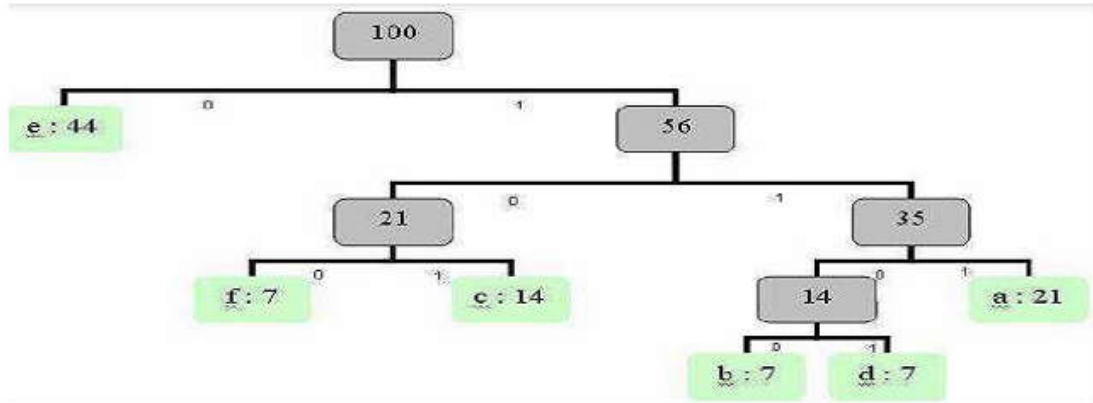
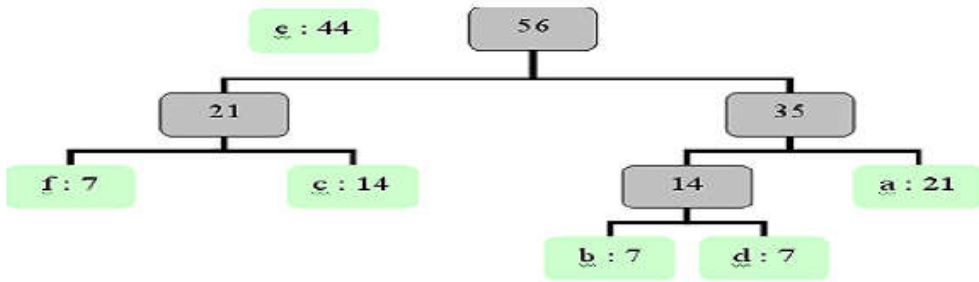
Exemple

Exemple1: Soit un texte possédant 100 lettres, dans lesquelles il y a 44 fois la lettre 'e', 21 fois la lettre 'a', 14 fois la lettre 'c', et 7 fois les lettres 'b' 'd' et 'f'.

On applique l'algorithme et on aura :



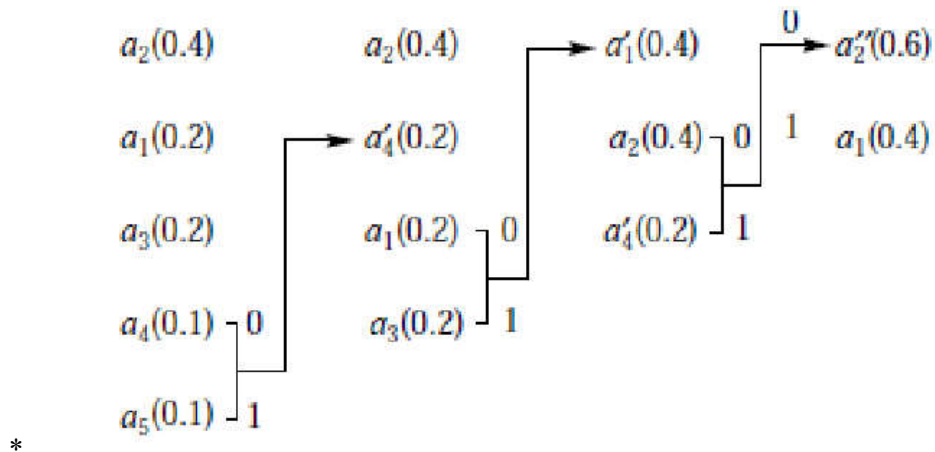




Lettre	a	b	c	d	e	f
Mot de code de Huffman	111	1100	101	1101	0	100

Exercice

Soit la source $S=\{a_1, a_2, a_3, a_4, a_5\}$ avec $P(a_1)=0.2$, $P(a_2)=0.4$, $P(a_3)=0.2$, $P(a_4)=0.1$ et $P(a_5)=0.1$. Appliquer l'algorithme de Huffman et donner les mots de code correspondant.



Le parcours de l'arbre aboutit au code ci-dessous

Symbole	probabilité	Mot de code
a_1	0.2	10
a_2	0.4	00
a_3	0.2	11
a_4	0.1	010
a_5	0.1	011

La taille moyenne de ce code est

$$l = 0.4 \times 1 + 0.2 \times 2 + 0.2 \times 3 + 0.1 \times 4 + 0.1 \times 4 = 2.2 \text{ bits/symbole .}$$

3. Code Tunstall

Exemple

- Soit le code de Tunstall à 2 bits suivant:
- Encodage de AAABAABAABAAA

Séquence	Mot-codes
AAA	00
AAB	01
AB	10
B	11

AAA B AAB AAB AAA
 00 11 01 01 00

Toute séquence de sources doit pouvoir être représentée par une séquence de symbole apparaissant dans le code. AAA B AAB AAB AAA 00 11 01 01 00 Toute séquence de sources doit pouvoir être représentée par une séquence de symbole apparaissant dans le code.

3.1. Principe

Algorithme

n : nombre de bits du code ;

N = taille de l'alphabet ;

$S = \{x_i : i = 1 \dots N\}$ // les alphabets de la source

$Pr = \{p(s_i)\}$ // probabilité de l'alphabet x_i

$F = S$ // feuilles de l'arbre.

$k = 0$;

Créer un arbre avec un nœud racine et les feuilles F .

Répéter

Calculer la probabilité p_j de chaque feuille $L \ M \in N$;

Choisir la feuille f_{max} dont la probabilité est maximale ;

f_{max} = nœud interne ;

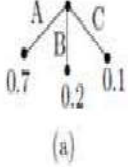
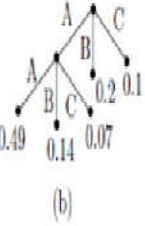
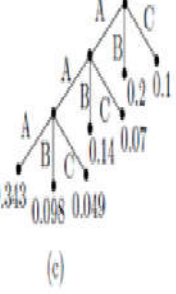
Eclater f_{max} en $N-1$ feuilles ;

$k++$;

Jusqu'à $O + (P + 1) O - 1 > 2$

3.2. Exemple

Construire un code de Tunstall à 3 bits pour une source sans mémoire de l'alphabet $A = \{A, B, C\}$. $P(A) = 0.7$, $P(B) = 0.2$ et $P(C) = 0.1$

<i>k</i>	<i>Valeurs obtenues</i>	<i>Action à élaborer</i>	<i>Vue de l'arbre</i>
0	$N=3$ $F=A=\{A, B, C\}$ $Pr=\{0.7, 0.2, 0.1\}$ $f_{max}=A$ $n = 3, 2^n = 8$	On prend la feuille A qui a la plus grande probabilité et on la concatène avec les autres lettres de l'alphabet.	 <p>(a)</p>
1	$F=\{AA, AB, AC, B, C\}$ $Pr=\{0.49, 0.14, 0.07, 0.2, 0.1\}$ <i>Taille de F</i> = $N+1*(N-1)=3+2=5$: $f_{max}=AA$ $N + 2(N - 1) = 7 < 2^3$;	On prend la feuille AA qui a la plus grande probabilité et on la concatène avec les autres lettres de l'alphabet.	 <p>(b)</p>
2	$F=\{AAA, AAB, AAC, AB, AC, B, C\}$ <i>Pr</i> = $\{0.343, 0.098, 0.049, 0.49, 0.14, 0.07, 0.2, 0.1\}$ <i>Taille de F</i> = $N+2*(N-1)=3+4=7$: $f_{max}=AAA$ $N + 3(N - 1) = 9 > 2^3$;	Arrêt	 <p>(c)</p>

Le code généré par la méthode de Tunstall est :

Alphabet	Code
B	000
C	001
AB	010
AC	011
AAA	100
AAB	101
AAC	110
---	111

On remarque que le code 111 est non utilisé.

4. Code unaire

Le code unaire est un code qui a été conçu pour les nombres. Son principe est défini par :

Définition Un nombre positif n est codé par n bits 1 suivi d'un bit 0.

Le code unaire est le code le plus simple pour représenter les entiers. Exemple Par exemple l'entier 5 est codé par 111110. Le code unaire est identique au code de Huffman pour les entiers équiprobables 1,2,...etc.

5.Code de Golomb

Code paramétré par un entier strictement positif $m > 0$. Pour un entier n on associe et on encode deux entier q et r tel que:

$$q = \left\lfloor \frac{n}{m} \right\rfloor, \quad r = n - qm$$

$$c = \lceil \log_2 m \rceil$$

5.1 Codage du quotient q

Le quotient q peut prendre des valeurs de $\{0,1,2,\dots\}$. Le quotient q est codé avec un code unaire.

Codage du reste r Le reste r peut prendre m valeurs possibles appartenant à l'ensemble suivant: $\{0,1,2,\dots,m-1\}$.

Codage du reste r – solution 1

r est divisé en deux parties selon la valeur de c

- Si $r < c$ alors r est codé par c bits qui commence par 0
- Sinon r est codé par $c+1$ bit qui commence par 1

Exemple 1 : si $m = 3$ donc $c = 1$ alors r peut prendre les valeurs 0, 1 ou 2.

1/ Si $r=0$ code de r est 0.

2/ Si $r=1$ code de r 10 ; Si $r=2$ code de r 11

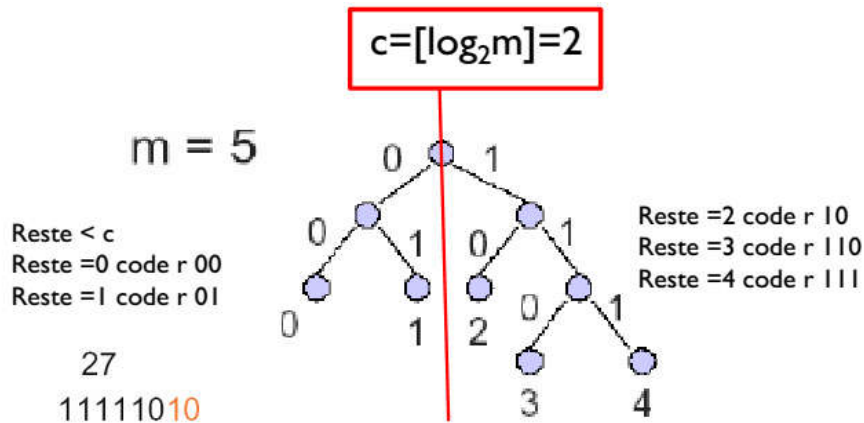
Exemple 2 : si $m = 5$ donc $c = 2$ alors r peut prendre les valeurs 0, 1,2,3 ou 4.

1/ Si $r=0$ code de r est 00; Si $r=1$ code de r 01

2/ Si $r=2$ code de r 100; Si $r=3$ code de r 101; Si $r=4$ code de r 110.

Codage du reste r – solution 2

Tracer l'arbre préfix séparé en axe des 0 et en axe des 1 selon la valeur de c



5. Conclusion

Des techniques de codage source ont été présentées dans ce chapitre. Ces techniques permettent de transformer une information en forme binaire avec la contrainte de minimiser la taille de celle-ci en vue de la transmettre ou de la stocker. Le code unaire et le code de Golomb permettent le codage de valeurs entières. Le code de Huffman assure un codage de longueur variable et le code de Tunstall permet un codage de longueur fixe.