1. DÉFINITIONS :

1.1. Longueur moyenne d'un code pour une source X:

$$L = \sum_{x \in X} p(x) l_x \geq H(x)$$

l_x est la longueur de chaque symbole x, L s'exprime en bits/symbole.

1.2. Efficacité d'un codage est donnée par :

$$E = \frac{H(X)}{L} \leq 1$$

1.3. Redondance d'un codage est donnée par :

$$R = 1 - E$$

Remarque : Un code est dit optimal pour une source discrète sans mémoire s'il minimise la longueur moyenne du code d'une lettre parmi tous les codes possibles pour cette source.

2. CODAGE DE HUFFMAN :

Un code de Huffman est un code préfixe binaire associé à une source discrète sans mémoire (X, p) de cardinal M. On le construit par récurrence sur L.

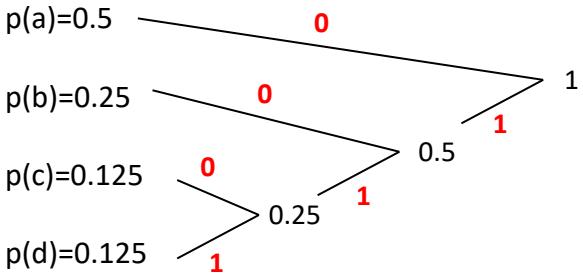
2.1. Algorithme de Huffman :

1. Ordonnancer les symboles de la source dans l'ordre des probabilités décroissantes ;
2. Isoler les deux symboles les moins probables et les distinguer avec 1 pour le plus faible et 0 pour le plus fort ;
3. Regrouper ces deux symboles en un seul nouveau en additionnant leurs probabilités ;
4. Recommencer jusqu'à ce que tous les symboles soient regroupés.

Attention : Matlab utilise ce standard, on désigne 1 pour le plus faible et 0 pour l'autre.

Exemple :

X={a, b, c , d} où p(a)=0.5, p(b)=0.25, p(a)= p(a)=0.125.



a	0
b	10
c	110
d	111

H(X)=1.75 bits/symboles, L=1.75 bits/symboles, E=100%, R=0%

Exercice:

X={A,B,C,D,E,F,G,H} où p(A)=0.2, p(B)=0.18, p(C)=0.3, p(D)=0.05, p(E)=0.06, p(F)=0.1, p(G)=0.07, p(H)=0.04,

1- Calculer H(X)

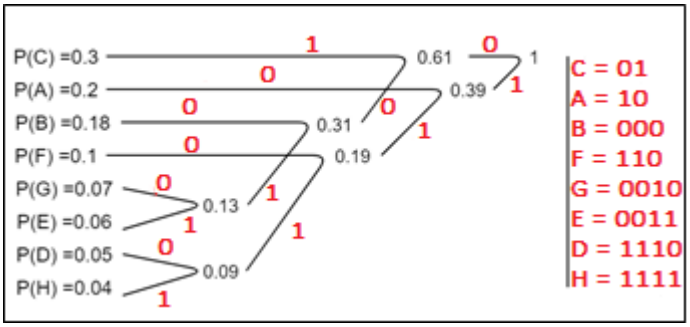
2- Donner le code de Huffman correspondant.

3- Calculer L, E et R.

Solution:

1- H(X) = 2.55 bits/symboles

2- Codage de Huffman:



3- L=2.72, E=93.75%, R=6.25%

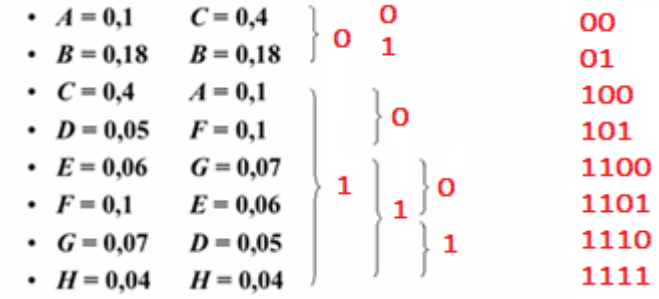
3. CODAGE DE SHANNON-FANO :

Fano construit un arbre binaire de manière descendante. Diviser de manière récursive l'ensemble des symboles en deux sous-ensembles de probabilités cumulées peut être vu comme une dichotomie récursive pour obtenir des subdivisions de l'intervalle [0,1[**:**

3.1. Algorithme de Shannon-Fano :

1. Ordonner les symboles de l'alphabet sur une colonne par probabilité décroissante ;
2. diviser l'ensemble des symboles en deux sous-ensembles de probabilités cumulées presque égales $\geq \frac{1}{2}$, $\leq \frac{1}{2}$;
3. coder avec "0" les éléments du premier sous-ensemble ;
4. continuer de manière récursive jusqu'au moment où tous les sous-ensembles ne comportent plus qu'un élément.

Exemple :



L= 2.64, E=96.6%, R= 3.4 %

Exercice 01 :

Coder cette séquence d'information en utilisant l'algorithme de Shannon-Fano.

x_i	$P(x_i)$	Etape1	Etape2	Etape3	Etape4	Code
x_1	0.30	0	0			00
x_2	0.25	0	1			01
x_3	0.20	1	0			10
x_4	0.12	1	1	0		110
x_5	0.08	1	1	1	0	1110
x_6	0.05	1	1	1	1	1111

H(x)=2.36 bits/symbole L=2.38 bits/symbole E= H(X)/L = 0.99

Exercice 02 : Coder la source suivante en utilisant l'algorithme de Shannon-Fano.

Message	Prob (%)	Etape1	Etape2	Etape3	Etape4	Etape5	Code
AAA	51.20	0					0
AAB	12.80	1	0	0			100
ABA	12.80	1	0	1			101
BAA	12.80	1	1	0			110
ABB	3.20	1	1	1	0	0	11100
BAB	3.20	1	1	1	0	1	11101
BBA	3.20	1	1	1	1	0	11110
BBB	0.80	1	1	1	1	1	11111

4. CODAGE ARITHMÉTIQUE :

Le codage arithmétique est un codage statistique, c'est-à-dire que plus un caractère est représenté, moins il faudra de bits pour le coder.

Contrairement au codage de Huffman, le codeur arithmétique traite un bloc de symbole en lui associant un unique nombre décimal rationnel.

Ce codage n'est que très peu utilisé en pratique mais elle reste présente, notamment dans la norme JPEG2000.

4.1. Méthode de codage arithmétique :

Chaque symbole est entré avec sa probabilité d'occurrence comprise entre 0 et 1, et le codage se traduit par l'affectation d'un nombre unique, à virgule flottante, compris entre 0 et 1, à un ensemble de symboles.

Exemple :

Considérons le codage du message "BALLE" :

Caractère	Probabilité
A	1/5
B	1/5
E	1/5
L	2/5

Dans l'intervalle général de probabilité [0,1], chaque symbole se voit affecter un intervalle de probabilité.

Caractère	Probabilité	intervalle
A	1/5	[0 - 0.2[
B	1/5	[0.2 - 0.4[
E	1/5	[0.4 - 0.6[
L	2/5	[0.6 – 1]

Le codage s'effectue progressivement à partir de la première lettre du message "BALLE" selon les opérations suivantes :

- LI = LI + DL * (LI du symbole)
- LS = LI + DL * (LS du symbole)
- DL = (LS - LI) de l'intervalle précédente.

En continuant, on arrive au tableau suivant :

Caractère successif	Limite inférieure (LI)	Limite supérieure (LS)	DL
B	0.2	0.4	0.4-0.2=0.2
BA	0.2+0.2*0=0.2	0.2+0.2*0.2=0.24	0.24-0.2=0.04
BAL	0.2+0.04*0.6=0.224	0.2+0.04*1=0.24	0.24-0.224=0.016
BALL	0.224+0.016*0.6=0.2336	0.224+0.016*1=0.24	0.24-0.2336=0.0064
BALLE	0.2336+0.0064*0.4= **0,23616**	0.2336+0.0064*0.6= 0.23744	

Le codage du message est constitué par la dernière limite inférieure = $\mathbf{0,23616}$.

4.2. Décodage arithmétique :

Prenons l'exemple précédent où le nombre $\mathbf{0,23616}$ qui code le mot "BALLE". Le principe de la décompression (décodage) est très simple. Celle-ci suit les deux étapes suivantes qui se répète jusqu'à l'obtention du mot :

- La prochaine lettre du mot est celle dont l'intervalle contient le nombre du mot actuel (Ex : $\mathbf{0,23616}$ est dans l'intervalle de "B" donc la première lettre est B).
- On modifie le nombre représentant le mot comme suit :

  (Code du mot – borne inférieure de la lettre) / Probabilité de la lettre

  Ex : nouveau code = (0.23616 – 0.2) / 0.2 = 0.1808

Le tableau suivant montre les différentes étapes de la décompression :

Symbole	Nouveau code
B	0.1808
A	0.904
L	0.76
L	0.4
E	

Ainsi, nous avons récupéré notre premier message "**BALLE**".

Exercice : Soit ce message 'COMM' à coder selon l'algorithme Arithmétique.

Solution :