Codage de l'information
Codage de Huffman
ITU 2021
Rakotoarimalala Tsinjo

1 Théorie de l'information

1.1 Source d'information

Une ***source d'information*** est caractérisée par :

- un alphabet de source $\mathcal{S} = \{s_1, s_2, \ldots, s_m\}$ décrivant quels sont les symboles s_i que cette source peut émettre ;
- une distribution de probabilité $P = \{p_1, p_2 \ldots, p_m\}$, chaque p_i étant un réel compris entre 0 et 1 donnant la probabilité que la source délivre le symbole $s_i \in \mathcal{S}$.

Plus formellement, une source est une variable aléatoire S prenant ses valeurs dans un ensemble fini $\mathcal{S}$ et dont la loi de probabilité est donnée par P . Ainsi pour tout symbole $s_i \in \mathcal{S}$

$$Pr(S = s_i) = p_i$$

Dans la suite on noterons $(\mathcal{S}, P)$ les sources d'informations

1.2 Quantité d'information

La quantité d'information contenue dans un symbole $s \in \mathcal{S}$ d'une source $S = (\mathcal{S}, P)$, de probabilité non nulle est

$$I(s) = -\log_2 Pr(S = s).$$

Par extension, pour un symbole s de probabilité nulle, on pose

$$I(s) = +\infty.$$

L'unité de quantité d'information est le ***bit***.

1.3 Entropie d'une source

L' entropie d'une source S est la valeur moyenne, ou espérance mathématique, de la quantité d'information contenue dans chacun de ces symboles. On la note $H(S)$.

$$H(S) = \sum_{s \in \mathcal{S}} Pr(S = s) I(s) = \sum_{i=1}^{m} p_i I(s_i) = -\sum_{i=1}^{m} p_i \log_2 p_i$$

On peut poser par prolongement que $0 \log_2(0) = 0$

L'unité de l'entropie d'une source est le bit.

On a alors

$$0 \leqslant H(S) \leqslant \log_2 m.$$

2 Codages optimaux

2.1 Longueur moyenne d'un codage de source

La longueur moyenne d'un codage c est la moyenne des longueurs des mots utilisés dans le codage, longueurs coefficientées par la probabilité des symboles de source correspondante. Elle s'exprime par

$$\bar{n}_c = \sum_{i=1}^{m} p_i |c(s_i)| = \sum_{s \in \mathcal{S}} Pr(S = s)|c(s_i)|$$

2.2 Codage optimal d'une source

Un codage c d'une source $S = (\mathcal{S}, P)$ dans un alphabet cible $\mathcal{A}$ est dit ***optimal***, si pour tout autre codage $c^{'}$ de la même source sur le même alphabet cible, on a

$$\bar{n}_c \leqslant \bar{n}_{c'}$$

3 Théorème du codage sans bruit

Le théorème du codage sans bruit est dû à Claude Shannon (1948). Il donne une minoration de la longueur moyenne de tout codage d'une source, et une majoration de celle d'un codage optimal.

3.1 Conventions

Si un symbole s d'une source S a une probabilité nulle, la longueur du mot qui lui est associé dans un codage n'a aucune influence sur la longueur moyenne de ce codage. D'ailleurs, ce symbole n'étant jamais émis par la source, on peut toujours considérer qu'il n'est pas nécessaire de le coder.

Convention 1.
Désormais, toutes les sources considérées ne comprendront aucun symbole de probabilité nulle.

Convention 2.
Toutes les sources considérées par la suite comprendront au moins deux symboles ($m \geqslant 2$).

Théorème 1 *La longueur moyenne d'un codage optimal d'une source S vers un alphabet cible de taille q satisfait l'encadrement :*

$$\frac{H(s)}{\log_2 q} \leqslant \bar{n}_c < \frac{H(s)}{\log_2 q} + 1 \tag{1}$$

4 Codage de Huffman

Le codage de ***Huffman*** utilise un code à longueur variable pour représenter un symbole de la source (par exemple un caractère dans un fichier).

Le code est déterminé à partir d'une estimation des probabilités d'apparition des symboles de source, un code court étant associé aux symboles de source les plus fréquents.

Il permet de construire un codage préfixe binaire optimal à partir de toute source.

Le principe du codage de Huffman repose sur la création d'une structure d'arbre composée de nœuds.

4.1 Construction de l'arbre du codage de Huffman

La construction d'un codage de Huffman consiste à associer à chaque symbole des sources successives un arbre binaire.

1. La première étape consiste à associer à chaque symbole un arbre réduit à un seul nœud (étiqueté par la lettre à de l'alphabet). Ainsi on obtient une forêt initiale qui est constituée d'arbres (contenant qu'un seul nœud).
2. On réunit les deux arbres de probabilité les plus petites (ou d'occurrences plus faibles) en un unique arbre, sa probabilité étant égale à la somme des deux probabilités des arbres réunis. Le nœud qui les relie sera étiqueté par la concaténation des deux étiquettes des deux nœuds reliés.
3. On refait 2 jusqu'à obtenir un seul arbre dont la racine est le nœud qui rejoint tous les arbres.
4. Pour reconstruire les mots de code pour chacun des alphabet initial, on parcourt l'arbre final à partir de la racine, à chaque fois qu'on prend à gauche on ajoute un 0 au code et 1 sinon, jusqu'à atteindre le nœud représentant la lettre à coder.

4.2 Exemple de codage de Huffman

Supposons qu'on a une source $S = (\mathcal{S}, P)$ avec

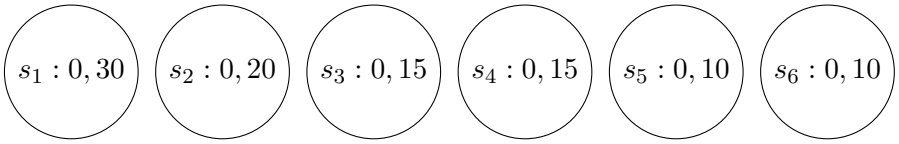
s	s_1	s_2	s_3	s_4	s_5	s_6
$Pr(S = s)$	$0,3$	$0,2$	$0,15$	$0,15$	$0,1$	$0,1$

L'entropie de cette source vaut

$$H(S) \sim 2,471.$$

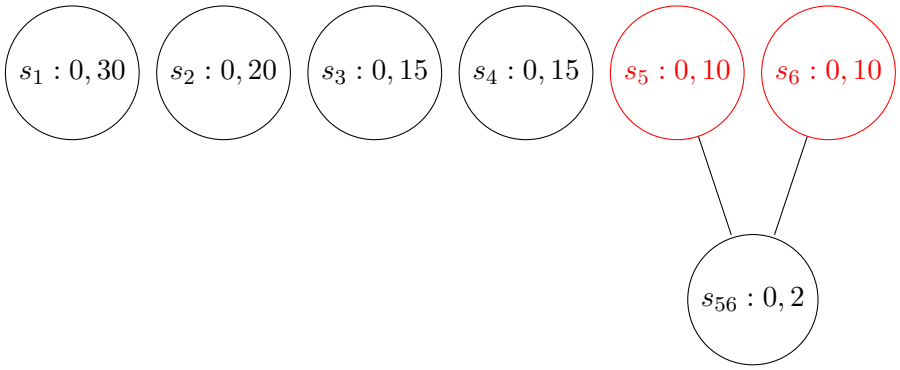
Déroulons la construction de l'arbre de code de Huffman

1. La première étape est la construction de la forêt

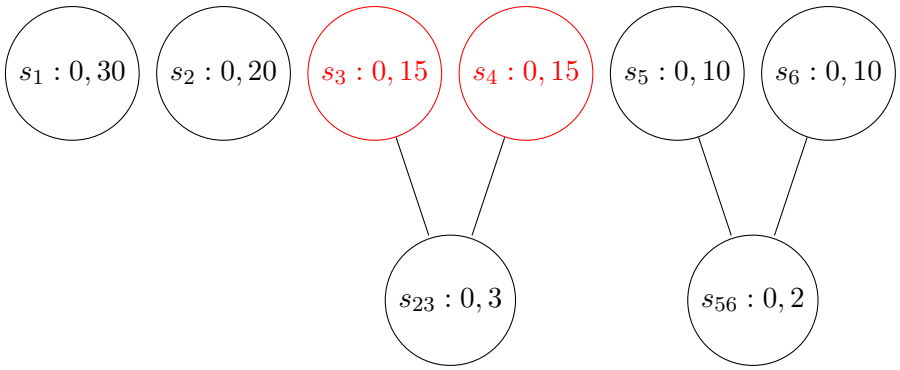


2. On fusionne les arbres ayant les plus faibles probabilités tant qu'on n'a pas un seul arbre.

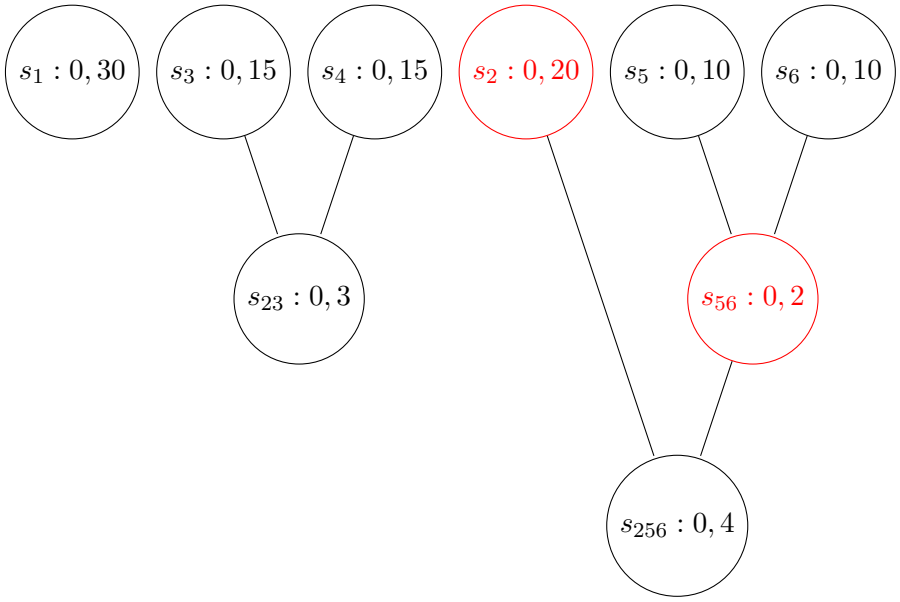
   (a) :



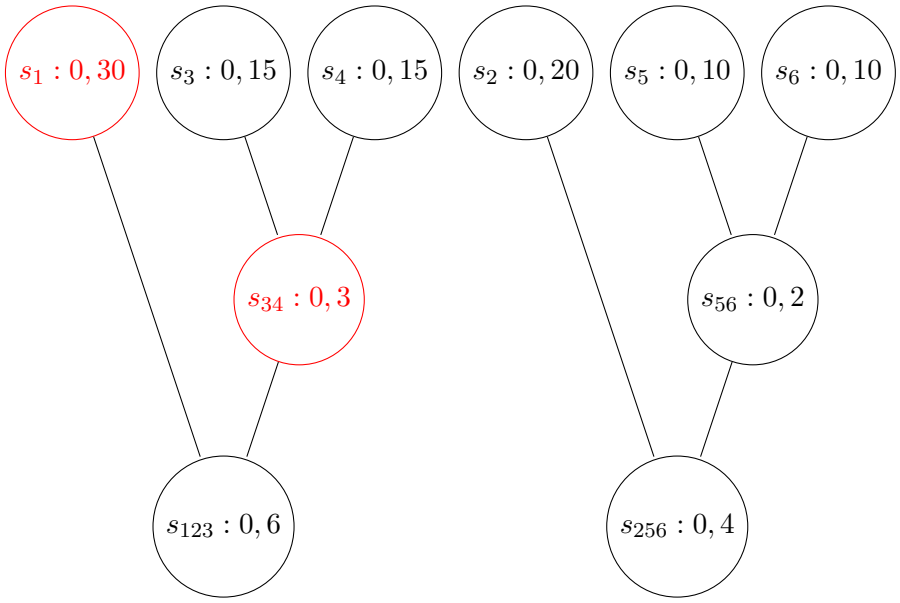
   (b) :



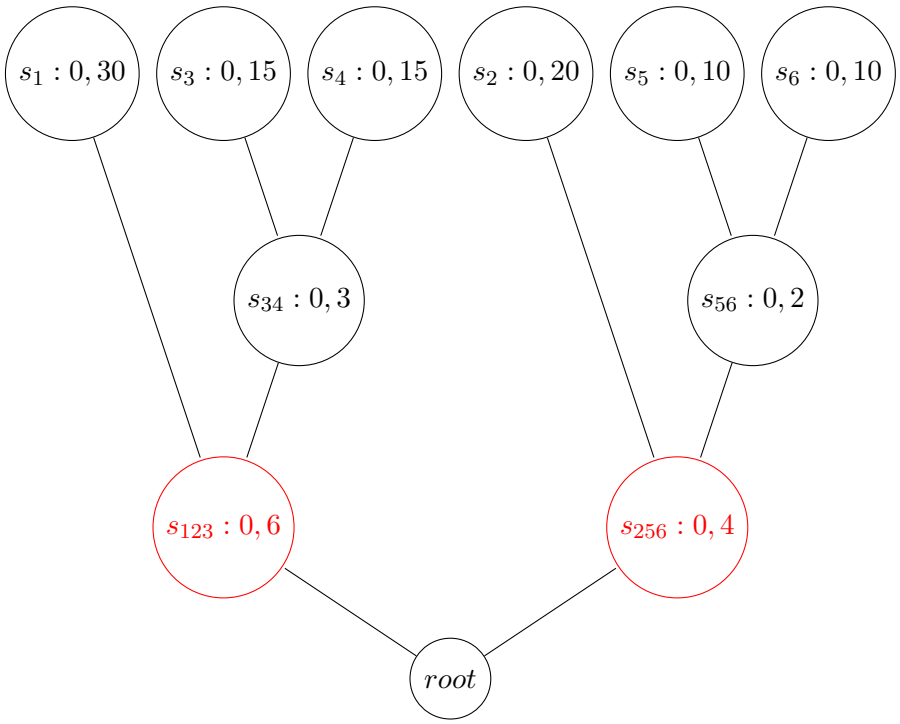
(c) :


$s_1 : 0, 30$
$s_3 : 0, 15$
$s_4 : 0, 15$
$s_2 : 0, 20$
$s_5 : 0, 10$
$s_6 : 0, 10$
$s_{23} : 0, 3$
$s_{56} : 0, 2$
$s_{256} : 0, 4$

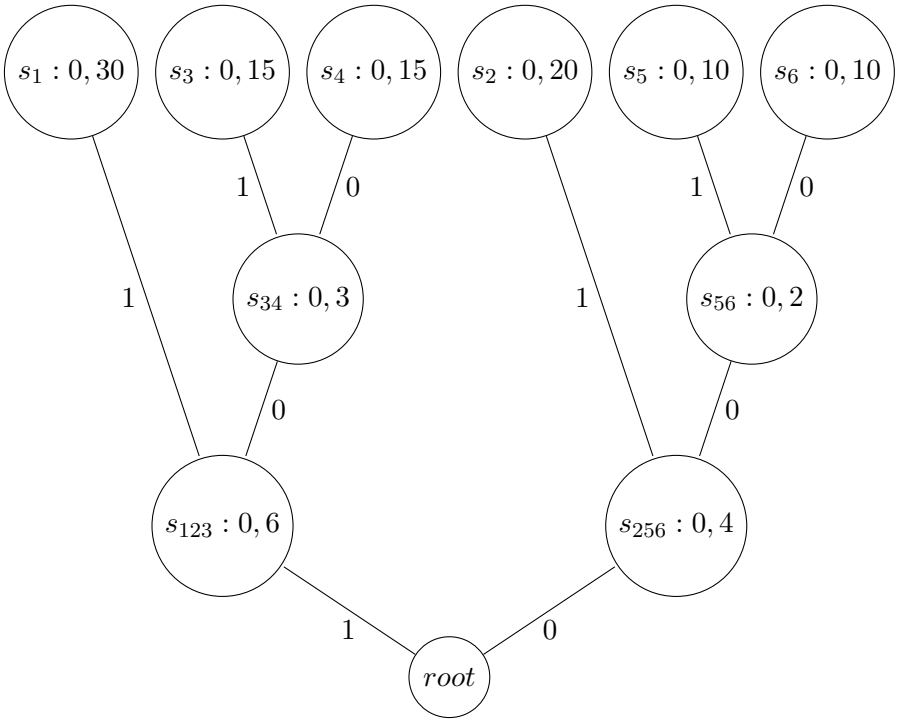
(d) :


$s_1 : 0, 30$
$s_3 : 0, 15$
$s_4 : 0, 15$
$s_2 : 0, 20$
$s_5 : 0, 10$
$s_6 : 0, 10$
$s_{34} : 0, 3$
$s_{56} : 0, 2$
$s_{123} : 0, 6$
$s_{256} : 0, 4$

(e) :



3. Pour obtenir le code de chaque s_i on ajoute les étiquettes des arêtes.



Donc on obtient finalement en concaténant les étiquettes des arêtes jusqu'aux nœuds s_i

s	s_1	s_2	s_3	s_4	s_5	s_6
code	11	01	101	100	001	000

La longueur moyenne de ce codage est

$$n_c = 2, 5$$

5 TP

1. (a) Soit $S = (\mathcal{S}, P)$ une source d'information. Écrire un programme permettant d'avoir les codes de chaque s_i de $\mathcal{S}$ selon le codage de Huffman

   (b) Écrire un programme qui prend en paramètre un mot/texte de $\mathcal{S}$ et renvoie le mot/texte codé par le codage de Huffman précédent

   (c) Écrire un programme qui prend en paramètre un mot/texte codé selon le codage de Huffman de la question 1.(a) et renvoie le mot/texte initial.

2. (a) Soit t un texte en anglais dans un fichier *texte.txt*. Proposez une façon de compresser le texte en utilisant le codage de Huffman. Implanter votre proposition.

   (b) Écrire un algorithme permettant de décompresser le fichier obtenu dans la question précédente. Implanter votre algorithme.

   Indication: N'oublier pas de mettre le codage dans le fichier compressé pour facilement décompresser.