Science de l'information : Série 5

Exercice 5.1 :

1. Le bloc a 5 manières différentes de se présenter : (1111), (1110), (1101), (1011), (1010)

   Les probabilités de chaque bloc se calculent en partant d'une lettre, et en multipliant par la probabilité d'avoir le chiffre suivant. Par exemple, (1101) a une probabilité de 1*0.1*0.1*0.99 = 0.000099. Nous trouvons donc des probabilités respectivement de : 0.0000001, 0.000099, 0.0099, 0.099, 0.9801. En plaçant les codes sur un arbre de Huffman, on trouve un code optimal, avec le code par bloc suivant (respectivement) : 1111, 1110, 110, 10, 0. Le calcul de longueur moyenne (somme des produits des longueurs de la longueur et de la probabilité d'apparition) nous donne un longueur moyenne d'environ **1.03 bits**

2. L'entropie conditionnelle se trouve facilement grâce a la formule vue dans le cours :

   $1/100*\log_2(100) + 99/100*\log_2(100/99) + 1*\log_2(1)$ = **0.0807 environ**

3.

   a. Nous devons ici appliquer la lois des probabilités totales :

   $P(A|B) = \Sigma P(A|B_1)\ P(B_1|B)$,

   Ce qui, appliqué ici donne : $x_n = 1/00*x_{n-1} + 1 + x_{n-1}$ Une résolution rapide nous donne que x_n vaut **$1 - 99/100\ x_{n-1}$.** Nous trouvons plus tard que la probabilité que x_{n-1} soit 1 est de 100/199.

   b. L'entropie **d'un symbole** est triviale, **elle est de 1**, un symbole donné ne peut être que 0 ou 1.

   L'entropie par symbole en revanche, elle se trouve de la manière suivante : on calcule l'entropie d'un symbole selon son précédent, sommant toutes les possibilités pour x_{n-1} et en multipliant par la probabilité d'un tel x_{n-1}.

   $(1/100*\log_2(100) + 99/100*\log_2(100/99))*\ x_{n-1}$. Or nous trouverons plus tard que la probabilité de x_{n-1} est de 100/199. Nous devrions aussi ajouter le cas ou x_{n-1} est 0, mais dans ce cas l'entropie est de 0. Nous trouvons donc **une entropie par symbole de 0.0405**

4.

a. La question ici est pour le moins floue. Étant donné les probabilités de suite (0 puis 1, ou 1 suivi de 0 a 99%), on trouve que 1 apparaît 101/200 fois, contre 99/200 pour 0, ce qui donne un 1 50.25 % du temps, et un 0 49.75 % du temps. A partir de la, sur 10'000 symboles, on aura 50250 "1" contre 49750 "0", soit une différence de 0.5 %. On aura donc « a peu près » le même nombre, mais pas exactement. Bart a donc raison.

b. Elle a totalement tort ! En effet, pour s'en convaincre il suffit de regarder l'exercice 1, ou nous avons légèrement comprimé la source.

c. Marge a raison. En effet, comme la suite la suite 10 arrivera, nous l'avons vu au premier exercice, à 99 chances sur 100. Nous pouvons donc coder ainsi : en choisissant un bloc de 30 caractères : s'il est tout du long égal à "101010...." nous le substituons par (1). En revanche, si deux 1 se suivent dans la séquence, nous codons alors individuellement en remplaçant les "1" par (01) et les "0" par (00). Mais le cas de deux "1" se suivant est tellement rare, que la compression reste efficace. Pour preuve le rapport de compression se calcule comme tel :

0.86*30 + 0.14*15 = 25.87 fois

0.86 est 0.99^15 et est la probabilité que "10" arrive 15 fois à la suite sur notre bloc de 30 caractères.

Tout ça ne fait qu'enfoncer notre pauvre Lisa, qui avait bien tort

5. :D

Exercice 5.2

1. Au lieu de coder avec comme alphabet les cases du monopoly, nous allons utiliser les lancers de dés de Lisa. Nous prenons donc les possibilités de total de ses dés (1/36 pour 2 et 12, 2/36 pour 3 et 11...), et les plaçons sur un arbre de Huffman. Nous obtenons donc les codes suivants pour les tirages :

2 : 10001	3 : 0111	4 : 0000	5 : 010	6 : 101	7 : 111
8 : 110	9 : 001	10 : 1001	11 : 0110	12 : 10000	

En calculant la somme des produits des longueurs de code et des probabilités d'apparition du tirage associé, nous trouvons une longueur moyenne de **3.2439 bits.** Cela est donc inférieur à la longueur de Lisa de 3.4.

2. L'essentiel est dans le programme java. Lancez et... enjoy:D

   Pour le reste, il manque juste de comparer les deux longueurs. Nous voyons que la longueur théorique est plus courte que la longueur pratique. Cela est du au manque d'aléatoire des lancers, et le manque de longueur de la séquence. Mille lancers ne sont pas assez pour tester, bien que la différence, vous le voyez, ne soit pas a vouvoyer mais bien a déplorer. J'eus pu me fourvoyer, dans l'antre de mon foyer, mais je vous serai gré de ne pas me criticailler.

   Signé : votre tout dévoué exploité.

Exercice 5.3

Mon code initial est

[HVJLHODLQW?RSCSZZOLEVC.H?L.TTLEVSLPODYSEPOZZLESO MLDVSLCO?LOHOJLTC. LEVSLPOZZN]

J'ai procédé ici par inspection : La différence de position entre les deux lettres codées est la même qu'entre les deux lettre décodées, et ainsi de suite. J'ai donc regardé les cas les plus probables (dans la langue anglaise) avec les deux premières lettres. PA, TE, KY, UF et WH ont retenu mon attention, avec des décalages de respectivement 9, 12, 3, 13 et 15. L'inspection des deux lettres suivantes a tout de suite éliminé tous les candidats, a l'exception du dernier, qui s'est révélé être WHY_ , avec un espace de 15. La suite est ainsi triviale, et donne

WHY WAS CINDERELLA THROWN OFF THE BASKETBALL TEAM? SHE RAN AWAY FROM THE BALL.

Voilà quoi :)