Science de l'information : Série 2

Exercice 2.1 :

1. Pour tous les codes : L'alphabet du code est {0, 1} et l'alphabet de la source est {a, b, c, d, e}. Les dictionnaires des codes sont : pour A {10, 100, 1000, 10000, 100000}, pour B {101, 0, 111, 110, 100}, pour C {11, 10, 001, 110, 01} et pour D {01, 1, 00, 110, 111}

2. (a) A et B sont à décodage unique, C et D ne le sont pas. Pour vérifier cela, il faut s'assurer qu'aucun symbole n'est le préfixe d'un autre. Cela se vérifie par inspection. A commence toujours par un 1, ce qui rend son décodage unique. Pour B, on vérifie le premier terme. S'il s'agit d'un 0, le décodage est terminé, et s'il s'agit d'un 1, on décode en prenant par triplets. Pour C, le code de (110 01) est le même que ac (11 001), et finalement pour D, bbb (1 1 1) et e (111) sont identiques, donc pas à décodage unique.

   (b) Seul B est à décodage instantané. Pour vérifier cela, on peut exclure C e D (qui ne sont pas à décodage unique), et pour A il y a énormément de préfixes, donc pas instantané.

3. Pour vérifier cela, nous appliquons le théorème de Kraft-McMillan. Nous pouvons affirmer que si l'inégalité est fausse, alors le code (D-aire avec les longueurs fixes) n'est pas à décodage unique. En d'autres mots, si l'inégalité est fausse, on ne peut pas rendre le code à décodage unique. Les exemples de codes se trouvent par inspection des arbres.

   Pour A : l'inégalité donne ≈ 0.484, donc il est théoriquement possible de l'améliorer. Un exemple de code A' amélioré serait : a{101}, b{10001}, c{100001}, d{101}, e{11} Par ce procédé (remplacer chaque dernier 0 par un 1) permet d'instaurer une ''borne'' au début et à la fin, donc rendre le code instantané.

   Pour C : l'inégalité de Kraft donne 1, donc est améliorable. Par inspection de l'arbre, on se rend compte que seul d (qui a comme préfixe a) pose problème. En changeant le code de d par {000}, il ne possède plus de préfixe, donc le code est rendu instantané.

   Pour D, l'inégalité de Kraft donne 1.25, donc le code, dans ces conditions, n'est pas améliorable.

4. Pour A, il est facile de réduire la taille, en enlevant un 0 à chaque code (a devient {10}, b {1000},...). Pour B en revanche, comme l'inégalité de Kraft donne 1, il est impossible de réduire la taille sans lui enlever la propriété du décodage unique.

Exercice 2.2

1. Note : j'utilise les probabilités vues dans la série 1.

   (a) En annexe : l'arbre du code de Shannon-Fanno pour C_0. Pour calculer la longueur des mots, il faut mettre les lettres qui composent le mot ENTROPIE sur un arbre, et ainsi décider d'un code pour toutes les lettres. Ma version donne le dictionnaire suivant : E{00} N{010}, T{011}, R{100}, O{101}, P{110}, I{111}. On applique ensuite le théorème de la longueur, qui donne $L(\Gamma)^* \Gamma \log_2(1/p_i) \urcorner$. De cela, on trouve une longueur moyenne de 2*1/4*2+6*1/8*3 = 3.25. L'entropie, vue dans la série 1, est de 2.75. Donc l'entropie de L1 est plus petite que sa longueur moyenne (ce qui est toujours vrai, car on ne peut pas faire mieux que l'entropie).

   (b) Pour calculer cela, il faut déterminer 3 cas. Comme les longueurs de E est de 2 bits, et que celle des autres lettres est de 3, il faut différencier lorsque le mot : (1) se compose de 2 E. (2) se compose d'1 E. (3) ne contient pas de E. 64 mots étant possibles, le cas (1) se retrouvera 4 fois, le cas (2) 24 fois et le cas (3) 36 fois. Le nombre de bits qui composent (1) est 4 (2x 2 bits), celui de (2) est 5, et (3) est 6. En compilant tout cela, on trouve que la longueur moyenne est de 4*4/64 + 5 * 24/64 + 6 * 36/64 = 352/64 = **5.5 bits** de longueur moyenne.

   (c) Dans la série 1, il a été démontré que l'entropie de cette séquence est de 5.5. Or, il a été montré qu'il est impossible de faire mieux que l'entropie (qu'elle représente une borne inférieure). Vu que notre longueur moyenne est égale à l'entropie (donc le meilleur encodage possible), Bart ne pourra pas trouver de longueur inférieure.

2. Dans cet exercice je trouve la longueur de chaque caractère en les plaçant sur un arbre. Il est à noter qu'en utilisant la formule de Shannon-Fano $\Gamma \log_2(1/p_i) \urcorner$, le même nombre de bits est trouvé.

   (a) Nous avons vu dans la série 1 qu'il y a 13 mots possibles. En partant du code précédent (E{00} N{010}, T{011}, R{100},

O{101}, P{110}, I{111}), l'encodage des 5 mots avec 1 E est sur 5 bits : EN, ET, NE, RE, TE. Les mots restants (IN, ON, OR, NI, RI, PI, TO, NO) ne comportent pas de E, et sont donc sur 6 bits. Chaque mot à une probabilité d'apparition de 1/13. La longueur moyenne est donc de 5*1/13*5 + 8*1/13*6 = 73/13 ≈ **5.615 bits**.

(b) Pour B_1, toutes les lettres sont possibles. En les plaçant sur un arbre, on trouve l'encodage et les quantités d'apparition suivantes : N{00}x3 E{010}x2, T{011}x2, R{100}x2, O{101}x2, P{110}x1, I{111}x1. La longueur moyenne est donc de 3/13*2 + 4*2/13*3 + 2*1/13*3 = **2.7692 bits** contre 2.7192 d'entropie pour B_1

(c) Il n'y a que 6 lettres possibles pour B2 : E, N, I, O, T, R. En les plaçant sur un arbre, l'encodage et la quantité d'apparitions à la seconde lettre (sur 13 mots) : T{000}x1, R{001}x1, I{010}x3, O{011}x2, E{10}x3, N{11}x3. Donc la longueur moyenne est de 2*1/13*3 + 2/13*3 + 3/13*3 + 2*3/13*2 = 33/13 ≈ **2.538 bits en moyenne.** L'entropie de B_2 est de 2.44 environ.

(d) Le code C1 est le suivant : E{00} N{010}, T{011}, R{100}, O{101}, P{110}, I{111} et le C2 est T{000}, R{001}, I{010}, O{011}, E{10}, N{11}. NE a ainsi une longueur de 4 bits, les mots NE, TE, RE, ON, IN et ET une longueur de 5, et les mots OR, NO, TO, RI, RT, NT une longueur de 6, chacun avec une probabilité d'apparition de 1/13. La longueur moyenne ainsi obtenue est de 1/13*4 + 5*1/13*5 + 6*1/13*6 = **5**.

(e) L'entropie de B, comme vu dans la série 1, est d'environ 3.7. Or, La longueur moyenne selon Lisa est de 5, et celle d'Homer est de 5.6 (c.f. Repectivement (d) et (a)). Comme nous avons dit que la méthode de Shannon-Fano est généralement bonne, mais pas la plus optimale, et que la plus optimale est celle de l'entropie, nous pouvons affirmer que Bart à raison, et qu'il existe un encodage dont la longueur moyenne est plus proche de l'entropie.

Exercice 2.3

1. Le message épelle **AMOUR** (preuve : bon, voilà quoi…).

   Le message de la figure 2 épelle **COMMUNICATIONS** (preuve similaire au premier message)

2. SOS s'épelle ''--- … ---'' et HELP s'épelle ''…. . .-.. .--.'' (preuve par bon sens et une grosse observation du glossaire)

3. **Le code est sans préfixe**. En effet, si au premier coup d'œil, la lettre e est préfixe de i par exemple, il ne faut pas considérer le code comme binaire mais ternaire (ti, ta et espace). Ainsi, l'espace marquant la fin de chaque symbole, il est totalement sans préfixe. Etant sans préfixe, il est donc instantané et ainsi **aussi à décodage unique** (également pour les même raisons que pour les préfixes : l'espace positionné toujours à la fin e la séquence permet de délimiter chaque caractère. Un symbole n'est pas terminé tant qu'il n'y a d'espace).

4. L'inégalité de Kraft ici est 3^-3 + 3^-5 + 3^-5 + 3^-4 + 3^-2 + 3^-5 + 3^-4 + 3^-5 + 3^-3 + 3^-5 + 3^-4 + 3^-5 + 3^-3 + 3^-3 + 3^-4 + 3^-5 + 3^-5 + 3^-4 + 3^-4 + 3^-2 + 3^-4 + 3^-5 + 3^-4 + 3^-5 + 3^-5 + 3^-5 + 10*(3^-6) ce qui vaut précisément 388/729, soit **environ 0.5322, donc l'inégalité de Kraft est respectée.**