LICENCE INFORMATIQUE
INFO0603 : COMPRESSION ET CRYPTOGRAPHIE
Pascal Mignot

UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Devoir sur table du 22 février 2018

Notes :
- Seul le support de cours est autorisé. Tout autre document est interdit.
- L'utilisation d'une machine à calculer sans fonction communiquante est autorisée.
- Les questions sont indépendantes.

Soit $\Sigma = (\mathtt{A}, \mathtt{B}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{F})$ l'alphabet d'une source S. Cette source produit le message M suivant :

```
A A D C E F A A A D A D C E F A A A C A A D A D C E F C A A C E F A A A A A D A A B A A B A A D A A B D A D
C A A A A A C A A A A D D A D A A D D A D C A A C E F C E F A A B D A D A A A A A B A A A C A A A A D A A B
```

L'espacement entre lettres et le retour à la ligne ne font pas partie du message à coder mais sont là pour améliorer sa lisibilité.

1. Si le message M est stocké sur un ordinateur en utilisant un code `ASCII`, quelle est la taille de M (en bit) ?
2. **Codage à taille fixe :** en ne codant que les symboles effectivement utilisés par le message :
   (a) Combien de bits par symbole sont nécessaires ? On justifiera.
   (b) Proposer un code à taille fixe pour l'alphabet (il n'est pas demandé d'écrire le codage du message M avec ce code).
   (c) Quel serait alors la taille de M en utilisant ce codage à taille fixe (même remarque).
   (d) Quelle est alors le taux de compression obtenu sur M par le codage à taille fixe ? Tous les taux de compression ultérieurs seront calculés **en prenant ce taux comme référence**.
3. **Codage entropique**
   (a) Donner les probabilités d'apparition des symboles dans le message M.
   (b) Calculer l'entropie de la source.
   (c) Calculer alors la taille en bits qu'aurait le message M compressé si l'entropie était atteinte. On donnera également le nombre moyen de bits par symbole.
   (d) Quelle est alors le taux de compression obtenu sur M par un codage qui atteint l'entropie par rapport à un codage à taille fixe ?
   (e) Serait-il possible d'obtenir mieux ? Si non, expliquer pourquoi. Si oui, sous quelles conditions ?
4. **Codage prefixe à taille variable**
   (a) Un codage à taille variable donne-t-il **toujours** une meilleure compression qu'un codage à taille fixe ? On justifiera.

   - (b) Un codage préfixe de taille variable avec des longueurs de code de $\{1, 2, 3, 4, 4, 5\}$ est-il possible ? On justifiera. Si oui, existe-t-il un code plus court ?
   - (c) Donner l'arbre à variance minimale construit par le codage de Huffman. Tout autre codage sera considéré comme faux. En déduire le code de Huffman associé à chaque symbole de l'alphabet.
   - (d) Quel serait alors la taille de M en utilisant ce code de Huffman ? On donnera également le nombre moyen de bits par symbole.
   - (e) Le codage de Huffman obtenu est-il optimal ? On justifiera.

5. **Codage arithmétique :**
   - (a) En utilisant la probabilité d'apparition des symboles de la source calculée à la question 3, appliquer la méthode du codage arithmétique afin de calculer le codage de la chaine $BDAA$.
   - (b) Donner le codage binaire du centre de l'intervalle trouvé (on utilisera la méthode avec les mises à l'échelle).
   - (c) D'après les propriétés du codage arithmétique, combien de bits le codage de cette chaîne aurait-il dû générer au plus ?
   - (d) Donner une estimation du nombre de bits que devrait générer le codage arithmétique du message M complet.

6. **Codage prefixe à taille variable par bloc de taille 3**
   - (a) Effectuer les comptages pour les blocs de taille 3 apparaissant dans le message M.
   - (b) Donner le codage de Huffman pour des blocs de taille 3.
   - (c) Quel serait alors la taille de M en utilisant ce code de Huffman par bloc ? On donnera également le nombre moyen de bits par symbole.
   - (d) Fait-on mieux que l'entropie ? Si oui, pourquoi ? Si non, est-ce prévisible ?

7. **Codage par dictionnaire à taille fixe** On voudrait constituer un dictionnaire à taille fixe constitué des symboles de l'alphabet et d'un ensemble des digrammes le plus courant.
   - (a) Calculer les fréquences des digrammes sur la première ligne du message seulement.
   - (b) Proposer deux dictionnaires à taille fixe : l'un codé sur 3 bits, l'autre codé sur 4 bits.
   - (c) En supposant que les digrammes ne se superposent pas, calculer la performance théorique de chacun des dictionnaires (i.e. supposer que chaque digramme est utilisé autant de fois que son comptage).
   - (d) Utiliser le plus performant des deux pour coder la deuxième ligne du message.
   - (e) Quel serait alors la taille de M en utilisant ce codage par dictionnaire ? On estimera que la performance du dictionnaire sur la première ligne est identique à celle sur la seconde ligne. On donnera également le nombre moyen de bits par symbole.

8. **Codage LZ78 :** on veut maintenant effectuer le codage avec la méthode LZ78 **avec une taille maximale de dictionnaire de 16** (attention, les réponses seront comptées comme fausse s'il n'est pas tenu compte de cette contrainte).
   - (a) Donner le codage `LZ78` de la première ligne du message.
   - (b) A partir du moment où le dictionnaire atteint sa taille maximale, quel est le nombre moyen de bits par symbole produit par le codage ?
   - (c) En déduire l'estimation de la taille du codage produit par le codage (on pourra diviser ce calcul en deux parties : celle où le dictionnaire grandit et celle où le dictionnaire

devient fixe). **Attention :** le codage de la sortie devra être optimisé en faisant en sorte que le codage de l'indice dans le dictionnaire dépendant de la taille du dictionnaire au moment du codage (*i.e.* comme vu en TD).

9. **Codage PPM d'ordre 2 :** après la lecture des 100 premiers caractères, on a généré l'ensemble des contextes d'ordre 2 suivants :

ordre 0	ordre 1	ordre 2
A=54	A/ A=31 B=5 C=5 D=13 Δ=4	AA/ A=14 B=5 C=5 D=7 Δ=4
B=5	B/ A=3 D=2 Δ=2	AB/ A=3 D=2 Δ=2
C=12	C/ A=5 E=6 Δ=2	AC/ A=2 E=2 Δ=2
D=17	D/ A=10 C=5 D=2 Δ=3	AD/ A=6 C=5 D=2 Δ=3
E=6	E/ F=6 Δ=1	BA/ A=3 Δ=1
F=6	F/ A=4 C=2 Δ=2	BD/ A=2 Δ=1
Δ=6		CA/ A=5 Δ=1
		CE/ F=6 Δ=1
		DA/ A=4 D=6 Δ=2
		DC/ A=2 E=3 Δ=2
		DD/ A=2 Δ=1
		EF/ A=4 C=2 Δ=2
		FA/ A=4 Δ=1
		FC/ A=1 E=1 Δ=2

   (a) Appliquer l'algorithme PPM d'ordre 2 des 8 derniers caractères du texte en utilisant les ordres et les contextes donnés ci-dessus. On écrira la totalité des mises à jour des contextes (seules les mise-à-jour seront données).

   (b) Quel est le nombre de bits engendré par le codage de ces derniers caractères ? On calculera explicitement la probabilité conditionnelles dans ces contexte.
   A partir de la question suivante, on supprimera le symbole Δ dans les calculs de probabilité contextuelle (par exemple, dans le contexte AC, $\Pr[A|EF] = 4/6 = 2/3$ et $\Pr[C|EF] = 2/6 = 1/3$).

   (c) A quoi correspond un codage PPM d'ordre 0 ?

   (d) En supposant que l'ensemble des contextes ont déjà été créé, et que les probabilités des différents contextes ne changent plus, comment calculer la taille du message M avec un codage PPM d'ordre 1 (il n'est pas demandé de faire l'application numérique).

   (e) calculer l'entropie conditionnelle $H(X_i|X_{i-2}X_{i-1})$.

   (f) pourrait-on utiliser cette entropie conditionnelle pour estimer la taille du message M compressé avec un codage PPM d'ordre 2 ? Si oui, on expliquera comment, et on effectuera le calcul.

10. **Codage RLE :** on veut utiliser le codage RLE sans augmenter la taille de l'alphabet à taille fixe donné à la question 2.

   (a) Proposer un codage vérifiant ces contraintes, en faisant en sorte de maximiser la longueur des runs qu'il est possible de représenter.

   (b) Effectuer le codage RLE la première moitié de la deuxième ligne : on écrira la compression en utilisant @ comme caractère spécial et en écrivant les longueurs des runs avec nombres (@4a = run de 4 a).

(c) Déduire des deux questions précédentes la taille complète du codage RLE du message en supposant que le reste du message produit le même nombre de bits par symbole.

(d) Y-a-t-il un gain si l'on utilise un codage à taille variable pour coder le résultat du codage RLE ?

Copies supplémentaires du message M :

A A D C E F A A A D A D C E F A A A C A A D A D C E F C A A C E F A A A A A D A A B A A B A A D A A B D A D

C A A A A A C A A A A D D A D A A D D A D C A A C E F C E F A A B D A D A A A A A B A A A C A A A A D A A B

A A D C E F A A A D A D C E F A A A C A A D A D C E F C A A C E F A A A A A D A A B A A B A A D A A B D A D

C A A A A A C A A A A D D A D A A D D A D C A A C E F C E F A A B D A D A A A A A B A A A C A A A A D A A B

A A D C E F A A A D A D C E F A A A C A A D A D C E F C A A C E F A A A A A D A A B A A B A A D A A B D A D

C A A A A A C A A A A D D A D A A D D A D C A A C E F C E F A A B D A D A A A A A B A A A C A A A A D A A B

A A D C E F A A A D A D C E F A A A C A A D A D C E F C A A C E F A A A A A D A A B A A B A A D A A B D A D

C A A A A A C A A A A D D A D A A D D A D C A A C E F C E F A A B D A D A A A A A B A A A C A A A A D A A B

A A D C E F A A A D A D C E F A A A C A A D A D C E F C A A C E F A A A A A D A A B A A B A A D A A B D A D

C A A A A A C A A A A D D A D A A D D A D C A A C E F C E F A A B D A D A A A A A B A A A C A A A A D A A B

A A D C E F A A A D A D C E F A A A C A A D A D C E F C A A C E F A A A A A D A A B A A B A A D A A B D A D

C A A A A A C A A A A D D A D A A D D A D C A A C E F C E F A A B D A D A A A A A B A A A C A A A A D A A B