math202 : mathématiques pour le numérique
TD 3 : compression par dictionnaire

Pierre Hyvernat
Laboratoire de mathématiques de l'université de Savoie
bâtiment Chablais, bureau 17, poste : 94 22
email : Pierre.Hyvernat@univ-savoie.fr
www : http://www.lama.univ-savoie.fr/~hyvernat/

Partie 1 : algorithme LZ78

Question 1. Rappelez le fonctionnement de l'algorithme de compression par dictionnaire LZ78.

Question 2. Détaillez le fonctionnement de l'algorithme de compression LZ78 sur les chaines
- `lalala`
- `abracadabra`
- `aaaaaaa`, `aaaaaaaa` et `aaaaaaaaa`

Donnez, en séparant bien les deux, le résultat ainsi que le dictionnaire construit.

Donnez, à chaque fois, la taille du résultat compressé (en octets) si on suppose que chaque nombre est codé sur 1 octet.

Question 3. Essayez de donner une approximation de la taille de la compression de `aa...aa` (n fois) si on suppose qu'on stocke toujours les numéros de mots du dictionnaire sur 1 octet.

Question 4. Appliquez l'algorithme de décompression LZ78 sur chacune des chaines compressées suivantes. Dans chaque cas, donnez le dictionnaire construit ainsi que la chaine originale.
- `0-t 0-o 1-o 1-i 4-t 0-u 1-u`
- `0-b 0-o 0-n 1-o 3-n 0-e 4-n 7`

Partie 2 : Représentation binaire d'une chaine codée par LZ78

Question 1. Si on encode la chaine compressée comme étant un texte en UTF-8, on obtient alors une succession de nombres et de lettres. Que problème aurons-nous ?

Question 2. On décide d'encoder la chaine compressée de la manière suivante : les nombres sont encodée sur un octet (entier 8-bits non-signé), les caractères sont encodés en UTF-8. Comme on sait que la chaine commence par un nombre, il n'y a pas de confusion possible.

Pour chaque chaine compressée calculée dans la question précédente, donnez sa représentation hexadécimale.

Rappel : une valeur hexadécimale, de `0` à `F`, est codée sur 4 bits. Un octet est donc codé par deux symboles hexadécimaux que l'on préfxe par `0x`. Par exemple, l'octet `00000010` s'écrit `0x02` et l'octet `00111101` s'écrit `0x3D`. En particulier, le code ASCII du le caractère `a` est 91, soit `0x61`.

Question 3. Utilisez la convention précédente pour décoder les chaines suivantes :
- `0x00 0x62 0x00 0x61 0x01 0x61 0x00 0x63 0x03`,
- `0x00 0x62 0x00 0x61 0x00 0x6c 0x03 0x65 0x01 0x61 0x03 0x70 0x02 0x6C`,
- `0x00 0xC3 0xA9 0x00 0x61 0x02`. Ce dernier exemple contient un caractère qui n'est pas ASCII. Cela pose-t-il un problème au décodage ?

Question 4. Avec cette convention décrite, quel problème peut survenir si on encode une chaine très longue ?

Comment peut-on remédier à ce problème ?

Partie 3 : pour aller plus loin

Question 1. La compression "RLE" (Run Length Encoding) consiste à remplacer `cccc` par `c`$\underline{4}$ où le nombre $\underline{4}$ est codé sur 1 octet.

Plus généralement, on remplace `c...c` (n fois) par `c`$\underline{n}$, où le nombre n est codé sur un octet.

- Comment est compressé `aa...aa`, où il y a exactement 500 `a`?
- Comment est compressé `a`?
- Comment est compressé `abab...abab` où il y a exactement 500 `ab`?

Question 2. On essaie d'améliorer cet algorithme pour ne pas perdre de place sur les caractères seuls : on remplace

- un `c` tout seul par `c`,
- une répétition `c...c` (n fois, avec $n > 1$) par `c`$\underline{n}$, où n est codé sur un octet.

Ainsi, la chaine `abab...ab` est "compressée" par elle même.

Quel problème cela pose-t'il ?

Question 3. Pour corriger le problème de la question précédente, on peut remplacer

- un `c` tout seul par `c`,
- une répétition `c...c` (n fois, avec $n > 1$) par `cc`$\underline{n}$, où n est codé sur un octet.

Donnez un exemple de chaine dont la taille augmentera avec cette méthode de compression.

Question 4. Donnez un exemple de (grande) chaine dont la taille augmente lorsqu'on la compresse avec l'algorithme LZ78.

Question 5. Donnez un exemple de (grande) chaine dont la taille augmente lorsqu'on la compresse avec l'algorithme de Huffman.

Question 6. Justifiez l'affirmation suivante : "Pour tous les programmes de compression existants (`gzip`, `bzip2`, `zip` ou autre), il existe des fichiers dont la taille augmente si on essaie de les compresser."