Théorie de l'information

M1 Informatique 2012-2013

Fiche de TD 1

Exercice 1. (Compression et théorie)

Soit $C : \{0,1\}^* \to \{0,1\}^*$ un algorithme de compression.

1. Donner l'unique propriété que doit nécessairement vérifier C pour être un algorithme de compression.
2. Démontrer qu'il existe un mot $u \in \{0,1\}^*$ tel que $|C(u)| \geq |u|$.
3. Démontrer que s'il existe un mot $u \in \{0,1\}^*$ tel que $|C(u)| < |u|$, alors il existe un mot $v \in \{0,1\}^*$ tel que $|C(v)| > |v|$.
4. En déduire qu'il n'existe pas d'algorithme de compression parfait.
5. Décrire un algorithme de compression C' qui utilise C et qui vérifie, pour tout mot $u \in \{0,1\}$,
$$|C'(u)| \leq |u| + 1.$$

Exercice 2. (Algorithme de Huffman)

1. Calculer, en donnant l'arbre de Huffman correspondant, les codes de Huffman des mots
$$\texttt{baacabaab}, \quad \texttt{aaaaabbbc}, \quad \texttt{baaabbbaccaaadbaaeebafaa}.$$
2. Décrire un moyen pour faire en sorte que l'algorithme de construction de l'arbre de Huffman d'un mot se fasse sans aucun choix (et donc de manière unique).
3. Donner la propriété principale des codes de Huffman.
4. Pour les trois codes précédents, calculer l'espérance de la longueur du codage obtenu.
5. Encoder les trois mots précédents à l'aides de leurs codages respectifs.
6. Soit $A := \{\mathtt{a}_0, \ldots, \mathtt{a}_{k-1}\}$ un alphabet à k lettres. Décrire la forme de l'arbre du Huffman du mot $\mathtt{a}_0 \ldots \mathtt{a}_{k-1}$.
7. Décrire la forme de l'arbre du Huffman du mot $\mathtt{a}_0^{2^0} \ldots \mathtt{a}_{k-1}^{2^{k-1}}$.
8. Encoder $u := \texttt{baacabaab}$ à l'aide de l'algorithme de Huffman dynamique.
9. Décompresser le résultat de l'encodage de u par l'algorithme de Huffman dynamique.

Exercice 3. (Lempel-Ziv)

1. Encoder, en donnant le trie correspondant, le mot `aabcbcbcbacbabcbabccbabb` par l'algorithme LZ78.

2. Encoder, en donnant le trie correspondant, le mot `abababababababababab` par l'algorithme LZ78.

3. Le résultat de l'encodage d'un mot u sur l'alphabet $\{\mathtt{a}, \mathtt{b}\}$ par l'algorithme LZ78 est le mot `0b1a2a0a1b`. Calculer le mot original u.

4. Décrire les mots sur l'alphabet $\{\mathtt{a}, \mathtt{b}\}$ qui nécessitent — en comparaison avec les autres mots de même longueur — le moins d'étapes lors de l'exécution de l'algorithme LZ78. Décrire la forme des trie correspondants. Déterminer ce nombre d'étapes en fonction de la longueur des mots.

5. Décrire les mots sur l'alphabet $\{\mathtt{a}, \mathtt{b}\}$ qui nécessitent — en comparaison avec les autres mots de même longueur — le plus d'étapes lors de l'exécution de l'algorithme LZ78. Décrire la forme des trie correspondants. Déterminer ce nombre d'étapes en fonction de la longueur des mots.

Exercice 4. (Move to Front)

Soit $(A, \leq)$ un alphabet totalement ordonné. L'algorithme MTF revient à associer à un mot u sur A un mot d'entiers positifs obtenu de la manière suivante.

(1) Créer une liste ℓ qui contient les lettres de A de l'unique façon que si une lettre b suit une lettre a dans ℓ alors $a \leq b$;

(2) pour chaque lettre a de u lue de gauche à droite :

   (a) $v := v \cdot i$ où i est la position de a dans ℓ ;

   (b) supprimer a de ℓ ;

   (c) placer a en tête de ℓ.

1. Encoder le mot `baaabacaabacab` par l'algorithme MTF.

2. Soit $A := \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ un alphabet ordonné par $\mathtt{a} \leq \mathtt{b} \leq \mathtt{c}$. Le résultat de l'encodage d'un mot u sur A par l'algorithme MTF est le mot 2121000. Calculer le mot original u.

3. Discuter à propos de la question de savoir si MTF est un algorithme de compression.

4. Déterminer l'intérêt de cet algorithme.

5. Encoder le mot $u :=$ `abbbbbbaccccbb` à l'aide de l'algorithme LZ78 puis encoder u en appliquant d'abord MTF et ensuite LZ78. Comparer les résultats obtenus.