TD 7 : Entropie et codage adaptatif

Exercice 1 (Découpage d'un flux infini) On veut coder un mot infini $w \in \Sigma^\omega$ par une suite de mots dans $\nu \subset \Sigma^+$. Afin de coder tous les mots possibles, sans ambiguïté, on exige de ν les deux propriétés suivantes :

(A) Complétude : tout mot infini $w \in \Sigma^\omega$ admet un préfixe $v \in \nu$

(B) ν est "ω-uniquement déchiffrable" : Si $(u_i)_i$ et $(v_i)_i$ sont deux suites de mots de ν telles que $u_1 u_2 \ldots = v_1 v_2 \ldots$, alors $\forall i \; u_i = v_i$.

Montrer que ν est un code préfixe.

On s'intéresse au codage d'un flux de données, décrit par une séquence $(U_i)_{i>0}$ de lettres aléatoires ($U_i \in \Sigma$). La séquence est supposée infinie, et nous souhaitons pouvoir coder, puis décoder, le message au fur et à mesure.

Exercice 2 (Codes de Elias)

1. On $B_0(n) \in \{0,1\}^+$ l'écriture en base 2 de l'entier $n \geq 1$. Exprimer $|B_0(n)|$ et donner un équivalent (quand n tend vers $+\infty$.) Est-ce que B_0 est un code préfixe ?
2. Mêmes questions pour $B_1(n) = 0^{|B_0(n)|-1} \cdot B_0(n)$ (ie, le mot formé de $k = |B_0(n)| - 1$ chiffres 0 suivis de $B_0(n)$.)
3. Mêmes questions pour $B_2(n) = B_1(|B_0(n)|) \cdot B_0(n)[2, -]$

Exercice 3 (Codage par rang) On suppose dans cet exercice ne pas disposer des fréquences d'apparition des lettres données en entrée. Nous allons concevoir un algorithme de compression en ligne, dont le codage change au cours du temps, afin de s'adapter aux fréquences de lettres effectivement constatées.
Pour cela, on conserve à chaque instant, la liste des lettres ordonnée par date de dernière apparition. On note $W_k = x_1 \ldots x_{|\Sigma|} \cdot U_1 \ldots U_k$ le mot lu en entrée à l'instant k, concaténé à toutes les lettres de l'alphabet d'entrées, ainsi toute lettre est apparue au moins une fois.

1. Soit $x \in \Sigma$ une lettre quelconque, et un indice $k > 0$. On note
$$N_k[x] = 1 + \min\{|P| \mid P \subseteq \Sigma \quad \wedge \quad W_k \in \Sigma^* \cdot x \cdot P^*\}$$
Expliquer ce que représente le tableau N_k, et donner un algorithme calculant N_{k+1} à partir de N_k et de U_{k+1} en temps linéaire ($O(|\Sigma|)$ opérations).
2. On se donne une fonction $C : [1, n] \to \{0,1\}^+$ injective. On code la k-ième lettre lue U_k grâce au mot $V_k = C(N_{k-1}[U_k])$. Donner un algorithme *en ligne* lisant en entrée les mots $V_1, V_2 \ldots$ et écrivant en sortie les lettres $U_1, U_2 \ldots$.
3. On suppose désormais que l'algorithme lit le mot infini "$V_1 V_2 \ldots$" lettre par lettre. Quelle hypothèse supplémentaire doit-on faire sur la fonction C ?
4. Soit $\Delta_k = \min\{i \geq 1 \mid W_k[|W_k| - i] = U_k\}$. Que représente Δ_k ? Montrer que $N_{k-1}[U_k] \leq \Delta_k$.
5. On suppose les U_i indépendants et identiquement distribués. On fixe une lettre $u \in \Sigma$. Calculer la limite de $\mathbb{E}(\Delta_k \mid U_k = u)$ (hint : il s'agit d'une loi "presque" géométrique.) Que se passe-t-il si on suppose désormais que les U_i forment une chaîne de Markov irréductible et apériodique ?
6. On suppose $C = B_1$. Montrer que la longueur d'une lettre codée vérifie :
$$\limsup_{k\to\infty} \mathbb{E}(|V_k|) \leq 2H(U) + 1$$
7. On suppose cette fois $C = B_2$. Montrer que :
$$\limsup_{k\to\infty} \mathbb{E}(|V_k|) \leq H(U) + 2\log(H(U) + 1) + 1$$

8. Comment améliorer ce taux de compression ? À quel coût ?

Exercice 4 (Entropie dans les arbres) Un arbre probabiliste est un arbre fini d'arité D dont les nœuds internes et les feuilles sont étiquettés par des probabilités, telles que

(a) La probabilité de la racine vaut 1

(b) La probabilité d'un nœud interne est la somme des probabilités de ses enfants

On note :

— $P_1 \dots P_N$ les probabilités des N nœuds internes

— $p_1 \dots p_n$ les probabilités des n feuilles

— $q_{l,j}$ la probabilité du j-ième fils du l-ième nœud interne (ainsi $\forall l \; P_l = \sum_j q_{l,j}$)

1. Soit $F \in \{1 \dots n\}$ une variable aléatoire suivant la loi définie par les $p_1 \dots p_n$ et L la profondeur de la F-ième feuille. Montrer que
$$\mathbb{E}(L) = \sum_{i=1}^{N} P_i$$
2. Soit $W \in \{1 \dots N\}^*$ la suite de nœuds internes aboutissant à la feuille F et $B \in \{1 \dots D\}^*$ la suite de branchements correspondantes (ainsi $W[i+1]$ est le $B[i]$−ème enfant du nœud $W[i]$.) Soit l un nœud interne de profondeur i. Exprimer $H_l = H(B[i] \mid W[i] = l)$ en fonction des $q_{l,j}$ et de P_l.
3. Montrer que $H(F) = \sum_{l=1}^{N} P_l \cdot H_l$
4. Soit un algorithme probabiliste effectuant L tirages indépendants de même distribution μ, et dont le résultat est la variable aléatoire Y. Montrer que $H(Y) \leq H(\mu) \times \mathbb{E}(L)$.