TD 7 : Codage adaptatif

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 décoder le message au fur et à mesure.

Exercice 1 (Découpage d'un flux infini) On veut coder la suite $(U_i)_i$ par un ensemble fini $\nu \subset \Sigma^+$ de messages de taille variable. Afin de décoder tous les flux possibles, on exige de ν les deux propriétés suivantes :

A Complétude : tout flux $U \in \Sigma^\omega$ admet un préfixe $V \in \nu$

B ν est "ω-uniquement déchiffrable" : Si $(u_i)_i$ et $(v_i)_i$ sont deux suites de ν telles que $u_1u_2\ldots = v_1v_2\ldots$, alors $u \equiv v$.

1. Montrer que ν est un code préfixe. On notera M la variable aléatoire telle que $V_1 = U_1 \ldots U_M \in \nu$.
2. Pour $k > 0$, montrer que $H(U_1 \ldots U_M \mid M = k) = H(U_1 \ldots U_k)$
3. En déduire que $H(U_1) \cdot \mathbb{E}(M) = H(V_1)$

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 de l'information sur la distribution exacte des données en entrées : nous ferons certaines hypothèses sur la distribution des lettres lues, mais jamais nous n'obtiendrons explicitement la liste des fréquences d'apparition de chaque lettre. 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.

Nous allons maintenir la liste des rangs de chaque lettre (dernière apparition.) Pour alléger les notations, notons $W_k = U_k \ldots U_1 \cdot x_1 \ldots x_{|\Sigma|}$ le mot lu en entrée à l'instant k, sous forme miroir, concaténé à toutes les lettres de l'alphabet d'entrées (ainsi le rang d'une lettre est toujours bien défini, même à l'instant initial.)

1. Soit $x \in \Sigma$ une lettre quelconque, et un indice $k > 0$. On note $N_k[x] = \min\{|P| \mid P \subseteq \Sigma \wedge W_k \in P^* x \Sigma^*\} + 1$. Expliquer comment calculer le tableau N_{k+1}, connaissant N_k et la lettre lue U_{k+1}, en temps linéaire.
2. On code la k-ième lettre lue U_k par le code $B_2(N_{k-1}[U_k])$. Expliquer comment retrouver le flux de données $U_1 \cdot U_2 \ldots$ à partir des données codées.
3. Soit $\Delta_k = \min\{i \geq 1 \mid W_{k-1}[i] = U_k\}$. Que représente Δ_k ? Montrer que $N_{k-1}[U_k] \leq \Delta_k$.
4. 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 ?
5. En déduire que la longueur d'une lettre codée, L_k, vérifie :

$$\lim \mathbb{E}(L_k) \leq H(U) + 2\log(H(U) + 1) + 1$$

6. Comment améliorer le taux de compression ?