Colloques Internationaux du C.N.R.S.
N° 276. -- THEORIE DE L'INFORMATION

UN PROBLEME ELEMENTAIRE DE LA THEORIE DE L'INFORMATION

D. PERRIN et M.P. SCHÜTZENBERGER

Le problème évoqué se situe dans l'hypothèse où la source est définie par une distribution de Bernouilli π sur un alphabet B et où la transmission est effectuée par une ligne sans bruit d'alphabet A. Le codage est donc simplement un morphisme injectif α du monoïde libre B^* dans le monoïde libre A^*. Si toutes les lettres de A ont le même coût, le coût moyen de la transmission, $L(\pi, \alpha)$ est simplement la longueur moyenne (par rapport à π) des mots du code $X = B\alpha$. Les inégalités classiques permettent alors de le borner inférieurement (en fonction de l'entropie de la source) et on sait que le minimum peut être atteint à l'intérieur de la sous-famille des codes préfixes (ou encore des codes X ayant un délai borné de décodage). Nous conjecturons que cette propriété remarquable reste vraie dans le cas général où les coûts des lettres de l'alphabet de transmission A sont quelconques. Cette conjecture admet une série de reformulations en langage algébrique. Nous établissons sa validité pour la sous-famille des codes X qui jouissent de la propriété d'avoir un délai de synchronisation borné.

On an elementary problem in information theory

One considers a source of information defined by an alphabet B and a Bernouilli distribution π on the the sequences over the source alphabet B; and A is the alphabet of the noiseless channel through which a transmission is realised by an encoding (with the property of unique decipherability).

If all letters in A have the same cost, the average cost of the transmission, $L(\pi, \alpha)$ is precisely the average length (with respect to π) of the code formed of the words on the alphabet A which encode the symbols from B. The classical inequalities allow an inferior bound on $L(\pi, \alpha)$, which is the entropy of the source; and it is well known that the minimum may be reached within the class of prefix codes.

We conjecture that this remarkable property remains valid in the general case, whatever be the costs of the letters from the transmission alphabet A; there is always a prefix code which does as well as any code. This conjecture admits several formulations in algebraic terms and we establish its validity for the subfamily of codes X having bounded synchronization delay.

1. INTRODUCTION

Plaçons-nous dans une situation très simple de transmission de l'information : une source émet des symboles appartenant à un ensemble B qui sont successivement codés par des mots sur un alphabet A, c'est-à-dire que l'on dispose d'une application :

$$\alpha : B \to X$$

qui à un symbole b de B associe un mot x sur l'alphabet A pris dans un ensemble X ; on suppose que l'application α réalise un codage, ce qui signifie que tout mot sur l'alphabet A s'écrit d'au plus une façon comme un produit d'éléments de X (c'est "l'unique déchiffrage") ; on dit alors que l'ensemble X est un code.

On supposera de plus, dans un souci de simplicité, que la source n'a pas de mémoire, c'est-à-dire que les apparitions successives des symboles sont indépendantes et que leur probabilité est donnée par une distribution fixe π sur l'ensemble B :

$$\pi : B \to]0,1]$$

On sait bien que, dans cette situation, le coût de la transmission est mesuré par :

$$L(\pi, \alpha) = \sum_{b \in B} |b\alpha| \; b\pi$$

qui est la longueur moyenne des mots du code $X = B\alpha$ (on note $|\ |$ la longueur d'un mot). On sait aussi que, π étant donnée, il existe un codage α rendant $L(\pi, \alpha)$ minimal et ayant la propriété que $B\alpha$ soit un code *préfixe* ; en effet la valeur de $L(\pi, \alpha)$ ne dépend que de la suite $(x_n)_{n \in N}$ donnant la distribution du nombre de mots de X de longueur n et l'on démontre facilement que toute distribution d'un code est aussi celle d'un code préfixe.

Plaçons-nous maintenant dans une situation un peu plus générale en supposant que les symboles de A ont chacun un *coût* qui intervient dans le coût de la transmission ; on dispose donc d'une application :

$$\gamma : A \to \mathbf{R}_+$$

que l'on étend à l'ensemble A^* des mots sur l'alphabet A par additivité :

$$\gamma(a_1 a_2 \cdots a_n) = \gamma(a_1) + \gamma(a_2) + \cdots + \gamma(a_n) .$$

Avec ces hypothèses, le coût de la transmission devient :

$$L(\pi, \alpha, \gamma) = \sum_{b \in B} \gamma(b\alpha) \; b\pi$$

et l'on voit que le cas précédent correspond à un coût $\gamma(a)$ égal à 1 pour tout symbole a de A.

PROBLEME – Est-il possible de rendre L *minimal en choisissant un codage* α *tel que* $X = B\alpha$ *soit un code préfixe ?*

En d'autres termes, la classe des codes préfixes est-elle encore optimale du point de vue de la longueur moyenne, parmi la classe de tous les codes.

Exemple – Soit $B = \{1,2,3,4,5\}$ et $A = \{a, b\}$; si π prend la valeur 1/5 pour chaque élément de B et si

$$\gamma(a) = 3 \qquad \gamma(b) = 1$$

la valeur minimale du coût de transmission se trouve égale à 22/5 ; elle est réalisée par chacun des trois codes suivants :

$$X_0 = \{a, ba, bba, bbba, bbbb\}$$
$$X_1 = \{a, abb, ba, babb, bbb\}$$
$$X_2 = \{aa, ab, ba, bba, bbb\}$$

Les codes X_0 et X_2 sont préfixes, et on répond donc, dans ce cas, par l'affirmative au problème posé.

On peut donner au problème précédent une forme plus générale et plus algébrique que nous exposons plus loin (§ 2) ; le but de cet article est de démontrer que l'on peut répondre de façon affirmative à ce dernier problème en se restreignant à une classe de codes particulière, définie au § 3.

2. CODES

De façon plus formelle que ci-dessus, un *code* X sur l'alphabet A est un ensemble de mots sur l'alphabet A tels que toute égalité de la forme :

$$x_1 x_2 \cdots x_n = y_1 y_2 \cdots y_m \quad ; \quad x_i, y_j \in X \tag{1}$$

implique $n = m$ et $x_i = y_i$ pour $i = 1, \ldots, n$. Ainsi, par exemple, l'ensemble $\{a, ab, ba\}$ n'est pas un code sur l'alphabet $A = \{a, b\}$ puisque l'on a :

$$a(ba) = (ab)a$$

On dit qu'un code X est *maximal* si et seulement si pour tout mot $z \notin X$ l'ensemble $X \cup \{z\}$ n'est plus un code. On sait que si X est un code maximal, alors pour tout mot z, il existe des mots u, v tels que uzv soit un message :

$$uzv = x_1 x_2 \cdots x_n .$$

Réciproquement si cette propriété est vraie pour tout z, et si X est fini, alors X est maximal (cf. [3] p. 94).

Les codes *préfixes* forment une classe bien conue de codes : ce sont les X tels qu'aucun élément de X ne soit un préfixe d'un autre élément de X. Dans l'exemple que nous avons donné ci-dessus, X_0 et X_2 sont des codes préfixes, mais pas X_1 ; on peut vérifier que X_1 est un code de la manière suivante : soient

$$u = a \quad ; \quad v = ba \quad ; \quad w = bb \quad ;$$

l'ensemble $Y = \{u, v, w\}$ est un code préfixe ; et les mots de X_0 peuvent s'écrire comme produits de mots de Y :

$$X_0 = \{u, v, wu, wv, ww\}$$

Si l'on considère Y comme nouvel alphabet, X_0 est un code suffixe : c'est-à-dire que ses mots lus de droite à gauche forment un code préfixe. On voit donc que le codage réalisé par X_0 est la composition d'un codage par un code préfixe suivie d'un codage par un code suffixe(1).

De plus les trois codes X_0, X_1, X_2 sont maximaux. Pour X_0 ou X_2, cela peut se démontrer en observant que pour tout mot z il existe un mot u tel que zu ait un facteur gauche dans X_0 (ce sont des "full prefix codes").

Sur ces notions de la théorie générale des codes, on pourra se reporter à [8] ou [11], ou encore [3] chap. IV (où les codes sont nommés "bases").

Maintenant nous disons que deux mots f et g sur l'alphabet A sont *commutativement équivalents* s'ils ne diffèrent que par l'ordre de leurs lettres. Formellement, on écrira

$$f \sim g$$

si et seulement si pour chaque lettre a de A, le nombre des occurrences de a dans f et g sont égaux :

$$|f|_a = |g|_a .$$

Pour deux ensembles de mots X et Y, on dira qu'ils sont commutativement équivalents s'il existe une bijection de l'un sur l'autre qui échange des mots commutativement équivalents ; formellement, on écrira :

$$X \sim Y$$

si $$X = \{x_1, x_2, \ldots, x_i, \ldots\}$$

et $$Y = \{y_1, y_2, \ldots, y_i, \ldots\} \quad \text{avec} \quad x_i \sim y_i .$$

Si X et Y sont deux codes commutativement équivalents, ils ont les mêmes paramètres numériques ; c'est-à-dire en particulier que si :

$$\gamma : A \to \mathbf{R}_+$$

est étendue à l'ensemble des mots par additivité, on aura :

$$\sum_{x \in X} \gamma(x) = \sum_{y \in Y} \gamma(y)$$

La conjecture suivante implique donc une réponse affirmative au problème posé dans l'introduction :

CONJECTURE – Tout code est commutativement équivalent à un code préfixe.

On observera par exemple que le code X_1 de l'exemple 1 est commutativement équivalent au code X_0 qui est préfixe ; en effet la bijection suivante de X_1 sur X_0 échange des mots commutativement équivalents :

(1) Il est faux que tout code puisse être obtenu par un tel procédé de surcodage utilisant seulement des codes préfixes ou suffixes. Cela est même faux des codes maximaux, comme l'a démontré Césari [1].

X_0	*a*	*ba*	*bba*	*bbba*	*bbbb*
X_1	*a*	*ba*	*abb*	*babb*	*bbbb*

Nous développons maintenant des notations et quelques résultats relatifs à l'équivalence de commutativité qui nous seront utiles dans la suite.

Notons L une partie de A*, c'est-à-dire un ensemble de mots sur l'alphabet A. Pour tout mot f de A*, on notera :

$$\lambda_L(f) = \text{Card}\ \{\ell \in L \mid f \in \ell A^*\}$$

qui est donc le nombre de facteurs gauches de f qui sont dans L. Pour un ensemble C de mots on écrira encore :

$$\lambda_L(C) = \sum_{f \in C} \lambda_L(f)$$

On voit que L est un code préfixe si et seulement si la fonction λ_L ne prend sur A* que les valeurs 0 ou 1.

PROPOSITION 2.1 – Une condition nécessaire et suffisante pour que L *soit commutativement équivalent à un code préfixe est que la moyenne arithmétique des valeurs de* λ_L *sur une quelconque classe* C *de l'équivalence de commutativité soit au plus égale à* 1.

Démonstration – Tout d'abord, si L est commutativement équivalent à un code préfixe X :

$$L \sim X$$

alors toute classe de commutativité contient autant de mots de L que de X. Or :

$$\lambda_L(C) = \sum_{DE \subset C} \text{Card}\,(L \cap D)\ \text{Card}\,(E)$$

où la sommation porte sur toutes les paires de classes de commutativité D, E telles que $DE \subset C$, c'est-à-dire encore sur toutes les classes D, E de mots d, e tels que :

$$de \in C\ .$$

On en déduit que

$$\lambda_L(C) = \lambda_X(C)$$

d'où la propriété.

Réciproquement, si l'inégalité

$$\lambda_L(C) \leqslant \text{Card}\,(C)$$

est vérifiée pour toute classe de commutativité C, établissons que L est commutativement équivalent à un code préfixe X. Il nous suffit de définir X par son intersection X_C avec chaque classe C ; pour la classe $C = \{1\}$ qui est

réduite au mot au vide, on prendra $X_{\{1\}} = \emptyset$ à moins que $1 \in L$ (ce qui signifie que $L = \{1\}$ puisque $\lambda_{\{1\}}(C) = \mathrm{Card}(C)$). Si l'on suppose maintenant par récurrence que les X_C sont définis pour toute classe C de mots de longueur au plus $n-1$ et que leur union est un code préfixe, on choisit une classe C de mots de longueur n et on écrit :

$$\lambda_L(C) = \sum_{DE \subset C}{}' \mathrm{Card}(L \cap D)\,\mathrm{Card}(E) + \mathrm{Card}(L \cap C)$$

où la somme Σ' porte sur tous les D, E tels que $DE \subset C$ et $C \neq D$. D'après l'hypothèse, $\lambda_L(C) \leqslant \mathrm{Card}(C)$ et donc :

$$\mathrm{Card}(L \cap C) \leqslant \mathrm{Card}(C) - \Sigma' \mathrm{Card}(X_D)\,\mathrm{Card}(E)\,.$$

On choisit alors dans C un ensemble de $\mathrm{Card}(L \cap C)$ mots pris hors de l'union des $X_D E$, que l'on définit comme X_C. On obtient ainsi un ensemble de mots de longueur au plus n qui est un code préfixe équivalent commutativement à l'ensemble des mots de L de longueur au plus n ; ceci démontre la propriété par récurrence sur n ■

REMARQUE 2.2 — On observera que la condition sur la valeur moyenne de λ_L sur une classe C peut être modifiée de façon à apparaître comme une forme plus précise de l'inégalité attribuée à Kraft et Mac Millan.

En effet, nous avons vu que :

$$\lambda_L(C) = \sum_{DE \subset C} \mathrm{Card}(L \cap D)\,\mathrm{Card}(E)\,.$$

Et l'inégalité $\lambda_L(C) \leqslant \mathrm{Card}(C)$ s'écrit donc encore :

$$\sum_{DE \subset C} \mathrm{Card}(L \cap D)\,\mathrm{Card}(E) \leqslant \mathrm{Card}(C) \tag{1}$$

Si l'on additionne membre à membre ces inégalité pour toutes les classes C de mots de longueur n, on obtient, avec $k = \mathrm{Card}(A)$:

$$\sum_{|C|=n} \sum_{DE \subset C} \mathrm{Card}(L \cap D)\,\mathrm{Card}(E) \leqslant k^n$$

d'où

$$\sum_{|D| \leqslant n} \mathrm{Card}(L \cap D)\, k^{n-|D|} \leqslant k^n$$

ou encore

$$\sum_{i=0}^{n} \mathrm{Card}(L \cap A^i)\, k^{-i} \leqslant 1 \tag{2}$$

Cette dernière inégalité est en fait un cas particulier d'une autre relation selon laquelle, si π est un morphisme multiplicatif de A^* dans l'intervalle $]0,1]$ des nombres réels, de somme 1 sur A :

$$\pi(A) = 1$$

alors, pour tout code X, on a l'inégalité :

$$\pi(X) \leqslant 1 \tag{3}$$

on démontre de plus que, pour un code X qui est fini, $\pi(X) = 1$ si et seulement si X est maximal (cf. [3], p. 231).

L'inégalité (2) correspond au cas où π prend la même valeur $1/k$ sur toutes les lettres de A. Nous ignorons s'il existe une formulation unique qui donnerait (1) et (3) comme cas particuliers.

Exemple 2.3 – Considérons un ensemble fini L de mots sur un alphabet $A = \{a, b\}$ ayant deux symboles. On suppose que tout mot dans L contient exactement une occurrence de la lettre a :

$$L \subset b^* a b^*$$

La proposition 2.1 dit, dans ce cas particulier, que L est commutativement équivalent à un code préfixe si et seulement si pour tout entier k,

$$\text{Card}\,\{b^i a b^j \in L \mid i + j \leqslant k - 1\} \leqslant k$$

cette dernière inégalité est en effet l'application de la formule (1) ci-dessus au cas où C est la classe du mot ab^{k-1}.

Ainsi la conjecture que nous avons énoncé implique en particulier qu'un tel ensemble L ne soit plus un code dès qu'il a $d + 1$ mots ou plus, où :

$$d = \max\{|\ell| \mid \ell \in L\}$$

Signalons que l'on peut montrer par une méthode d'énumération (cf. [10], théorème 3) que si $\text{Card}(L) \geqslant d + 1$, alors $L \cup \{b^d\}$ n'est pas un code.

3. SYNCHRONISATION

L'étude des algorithmes de décodage amène à poser la définition suivante : soit X un code sur l'alphabet A, on dit qu'un paire (f, g) de mots de A^* est *synchronisante* si pour tous mots p, q, la relation :

$$pfgq = x_1 x_2 \ldots x_n \quad \text{avec} \quad x_i \in X$$

implique l'existence d'un indice j, $1 \leqslant j \leqslant n$, tel que :

$$\begin{cases} pf = x_1 \ldots x_j \\ gq = x_{j+1} \ldots x_n \end{cases}$$

Intuitivement, l'apparition de la paire (f, g) au milieu d'un message permet de le couper en deux messages, indépendamment de la partie du message antérieure à f ou postérieure à g.

Exemple 3.1 – Avec le code X_2 de l'exemple du §1, la paire (ab, ab) n'est pas synchronisante ; en effet si $p = b$, $q = a$, alors

$$pfgq = (ba)(ba)(ba)$$

qui est dans X_2^* bien que ni pf, ni gq ne soient dans X_2^*. Par contre, pour tout mot g, la paire $(abba, g)$ est synchronisante car on peut vérifier que l'on a l'inclusion

$$A^* abba \subset X_2^*$$

Sur l'existence des paires synchronisantes, on pourra consulter [9]. On démontre en [15] que tout code a la même distribution de longueurs qu'un code ayant des paires synchronisantes (et on peut même choisir ce dernier préfixe).

Maintenant on dira qu'un code X a un *délai de synchronisation borné* $\leqslant s$ s'il existe un entier s tel que toute paire f, g de mots de la forme :

$$\begin{cases} f = x_1 \ldots x_s \\ g = x_{s+1} \ldots x_{2s}, \quad x_i \in X \end{cases}$$

est une paire synchronisante.

Les codes à délai de synchronisation borné ont été étudiés en [2], [4] et [14] comme une généralisation de la notion de code "comma-free". Plus récemment, ils ont été réintroduits en [7] sous le nom de "locally parsable codes" en liaison avec la théorie des langages apériodiques ; on pourra consulter à ce sujet [13] et [5].

Enfin ces codes interviennent dans la théorie des factorisations du monoïde libre (cf. [16], [17]) qui trouve des applications à la construction des bases des algèbres de Lie libres.

Exemple 3.2 – Considérons maintenant le code X_1 de l'exemple du §1 ; il n'est pas à délai de synchronisationborné car la paire (b^{4n}, b^{4n}) n'est synchronisante pour aucun entier n. Mais le code :

$$Y_1 = X_1 \setminus \{b^4\} = \{a, abb, ba, babb\}$$

a un délai de synchronisation égal à 1. En effet, si (f, g) sont deux mots de Y_1, posons :

$$\begin{cases} f = b^i a b^j \\ g = b^k a b^\ell \end{cases}$$

on a alors $0 \leqslant j + k \leqslant 3$ et on peut vérifier que la valeur de $j + k$ détermine j et k suivant le tableau :

$j + k$	0	1	2	3
j	0	0	2	2
k	0	1	0	1

Cela implique que si $ufgv$ est dans X^*, alors uf et gv aussi. Restivo a démontré [12] que cette situation se produit dès que l'on a un code maximal fini $Y \subset \{a, b\}^*$ et que chaque mot comporte au plus une occurrence de la lettre a.

4. RESULTAT PRINCIPAL

Nous établissons maintenant le résultat suivant :

THEOREME 4.1 – Un code ayant un délai de synchronisation borné est commutativement équivalent à un code préfixe.

Démonstration : La démonstration utilise un comptage et nécessite l'usage des séries formelles en variables non-commutatives que nous avons évité jusqu'ici. Etant donnée une partie L de A*, on notera $\underline{L}$ la série formelle à coefficients entiers :

$$\underline{L} = \sum_{f \in L} f$$

en d'autres termes $\underline{L}$ est la fonction caractéristique de L. On notera $\mathbf{Z} \ll A \gg$ l'anneau des séries à coefficients entiers dont les variables (non commutatives) sont les éléments de A ; étant donnée $S \in \mathbf{Z} \ll A \gg$, on note

$$\langle S, f \rangle$$

le coefficient du mot $f \in A^*$; c'est un élément de **Z**. On écrira donc :

$$S = \sum_{f \in A^*} \langle S, f \rangle f$$

Maintenant, si S est une série dont le terme constant est nul, on pose :

$$\mathrm{Log}(1 - S) = S + S^2/2 + S^3/3 + \cdots + S^n/n + \cdots$$

Soit alors X un code dont le délai de synchronisation est borné et posons :

$$T = \underline{A}^* - \underline{X}\,\underline{A}^* = (1 - \underline{X})\,\underline{A}^*$$

On aura pour tout mot $f \in A^*$: $\langle T, f \rangle = 1 - \lambda_X(f)$ où $\lambda_X(f)$ est le nombre de facteurs gauches de f dans X (cf. §2).

Ainsi, si C est une classe de commutativité, on aura :

$$\sum_{f \in C} \langle T, f \rangle = \mathrm{Card}(C) - \lambda_X(C)$$

et il nous faut donc prouver que pour toute classe C le coefficient

$$\langle T, C \rangle = \sum_{f \in C} \langle T, f \rangle$$

est positif. Calculons pour cela le logarithme de T :

$$\mathrm{Log}\, T = \mathrm{Log}\,[(1 - X)\,\underline{A}^*] ;$$

et pour toute classe de commutativité, on a (1) :

$$\langle \mathrm{Log}\, T, C \rangle = \langle \mathrm{Log}(1 - X), C \rangle + \langle \mathrm{Log}\, \underline{A}^*, C \rangle .$$

Rappelons maintenant que l'on dit que deux mots f, g de A* sont conjugués s'il existe u, v dans A* tels que :

$$f = uv ; \quad g = vu$$

cette relation est une relation d'équivalence plus fine que l'équivalence de commutativité ; considérons une classe D de conjugués comprise dans C et notons p son

(1) Quand on l'évalue sur toute une classe de commutativité, le logarithme d'un produit est la somme des logarithmes. Il suffit pour s'en convaincre de remplacer les symboles de A par des variables réelles.

exposant, c'est-à-dire l'entier p tel que tout mot de D s'écrive :

$$f = u^p$$

avec u un mot primitif c'est-à-dire qui n'est pas puissance d'un autre mot v (cf. [6] sur toutes ces questions). On sait que la classe D a exactement n/p éléments. De plus, on peut écrire :

$$\mathrm{Log}\,\underline{A}^* = -\,\mathrm{Log}(1 - \underline{A})$$

et on a donc :

$$\langle \mathrm{Log}\,\underline{A}^*, D\rangle = -\langle \mathrm{Log}(1-\underline{A}), D\rangle = -\sum_{f\in D}\langle -\frac{1}{n}f, f\rangle = \frac{1}{p}\cdot$$

Maintenant, soit X* ne rencontre pas la classe de conjugaison D et donc :

$$\langle \mathrm{Log}(1-\underline{X}), D\rangle + \langle \mathrm{Log}\,\underline{A}^*, D\rangle = \frac{1}{p}\,.$$

Soit X* rencontre D ; posons $E = D \cap X^*$ et montrons que E est une classe de X-conjugaison, c'est-à-dire que si $f, g \in E$, il existe $u, v \in X^*$ tels que $f = uv$, $g = vu$. Mais si $f, g \in D$, il existe $u, v \in A^*$ tels que $f = uv$, $g = vu$ et si l'on n'avait pas $u, v \in X^*$, aucune des paires $((u, v)^n, (vu)^n)$ ne serait synchronisante. On en déduit que tous les mots de E ont même longueur m sur l'alphabet X et que E a précisément m/p éléments, d'où :

$$\langle \mathrm{Log}(1-\underline{X}), D\rangle + \langle \mathrm{Log}\,\underline{A}^*, D\rangle = 0$$

et enfin :

$$\langle \mathrm{Log}\,T, C\rangle = \sum_{D\subset C}(\langle \mathrm{Log}(1-\underline{X}), D\rangle + \langle \mathrm{Log}\,\underline{A}^*, D\rangle) \geqslant 0$$

Cela implique que $\langle T, C\rangle$ soit lui aussi non négatif car $T = \exp \mathrm{Log}\,T$ et que l'exponentielle n'a que des coefficients positifs ; ceci achève la preuve ■

Exemple 4.2 – Soit $A = \{a, b\}$ un alphabet à deux lettres et $X \subset A^*$ un code maximal fini tel que chaque mot comporte au plus une occurence de la lettre a :

$$X \subset b^* \cup b^* a\, b^* .$$

Nous avons cité (cf. exemple 3.2) le résultat de Restivo suivant lequel si n est l'entier tel que $b^n \in X$ il existe deux parties P, Q de l'ensemble $\{0, 1, \ldots, n-1\}$ telles que :

i) $X = \{b^n\} \cup \{b^p a b^q \mid p \in P, q \in Q\}$

ii) tout entier $i \in \{0, 1, \ldots, n-1\}$ s'écrit d'une façon et d'une seule comme :

$$i = p + q \quad \text{avec} \quad p \in P, q \in Q\,.$$

Césari(1) a donné une preuve très élégante de ce résultat qui utilise les séries formelles : soient :

$$P = \{p \in N \mid b^p a \in X\}$$
$$Q = \{q \in N \mid a b^q \in X\}$$

(1) Communication personnelle.

et notons b^P la série :

$$b^P = \sum_{p \in P} b^p$$

on établit alors l'égalité entre séries suivante :

$$\underline{A}^* = b^Q \underline{X}^* b^P \tag{1}$$

en effet pour tout mot $f \in A^*$, il existe u et v dans A^* tels que

$$uaafaav \in X^* \tag{2}$$

puisque X est un code maximal (cf. § 2). On en déduit que $f = b^q x b^p$, avec $p \in P$, $x \in X$ et $q \in Q$; et cette écriture est unique sans quoi (2) aurait plusieurs factorisations en mots de X. Prenons maintenant les inverses des deux membres de (1) et multiplions-les ensuite par b^P à gauche et par b^Q à droite :

$$1 - \underline{X} = b^P (1 - \underline{A}) b^Q$$

si l'on ne considère que les mots de b^*, on obtient donc :

$$1 - b^n = b^P (1 - b) b^Q$$

$$1 + \cdots + b^{n-1} = b^P b^Q$$

ce qui est équivalent à la condition (ii) ci-dessus. On en déduit alors :

$$\underline{X} = b^n + b^P a b^Q$$

ce qui est la condition (i).

Il est intéressant de remarquer que si X est un tel code, alors

$$Y = X \cap b^* a b^*$$

est à délai de synchronisation 1, d'après le théorème de Restivo ; et qu'il est d'autre part (évidemment) équivalent commutativement au code préfixe :

$$Z = \{b^i a \mid 0 \leqslant i \leqslant n - 1\}$$

qui a lui aussi un délai de synchronisation égal à 1. Nous ne savons pas s'il est toujours possible de trouver un code préfixe commutativement équivalent à un code à délai de synchronisation borné qui ait lui aussi un délai borné. Signalons seulement que cela est vrai en ce qui concerne la distribution des longueurs (cf. [14]) : pour tout code à délai de synchronisation borné, il existe un code préfixe ayant la même propriété qui a même distribution de longueurs.

REFERENCES

[1] Y. CESARI – Sur l'application du théorème de Suschkevitch à l'étude des codes rationnels complets, in *Automata, Languages and Programming*, (J. Loeckx ed.). Lecture Notes in Computer Science, Springer Verlag (1974), 342-350.

[2] W.L. EASTMAN – On the construction of comma-free codes, *IEEE Trans. on Inf. Th.*, IT-11, 2 (1965), 263-367.

[3] S. EILENBERG – *Automata, Languages and Machines*, Vol. A, Academic Press (1974).

[4] S.W. GOLOMB and B. GORDON – Codes with bounded synchronization delay, *Information and Control, 8* (1965), 355-372.

[5] K. HASHIGUCHI and N. HONDA – Properties of code events and homomorphisms over regular events, *J. Computer Syst. Sci. 12* (1976), 352-367.

[6] A. LENTIN – *Equations dans les Monoïdes Libres,* Gauthier-Villars, Paris (1972).

[7] R. Mc NAUGHTON and S. PAPERT – *Counter Free Automata,* MIT Press, Cambridge, Mass. (1971).

[8] M. NIVAT – Elements de la théorie générale des codes, in *Automata Theory* (E.R. Caianiella ed.) Academic Press (1966), 278-294.

[9] D. PERRIN – Codes asynchrones, *Bull, Soc. Math. de France,* 105, 1977, p. 385-404.

[10] D. PERRIN et M.P. SCHUTZENBERGER – Codes et sous-monoïdes possédant des mots neutres, in *Theoretical Computer Science* (H. Walter ed.) Lecture Notes in Comput. Sci. *48,* Springer Verlag (1977), 270-281.

[11] J.F. PERROT – La théorie des codes à longueur variable, in *Theoretical Computer Science,* Lecture Notes in Comput. Sci. *48,* Springer Verlag, 27-44.

[12] A. RESTIVO – On a family of codes, in *Automata, Languages and Programming* (S. Michaelson ed.) Edinburth University Press (1976), 38-44.

[13] A. RESTIVO – A combinatorial property of codes having finite synchronisation delay, *Theoretical Comput. Sci. 1* (1975), 95-101.

[14] M.P. SCHUTZENBERGER – Sur une question concernant certains sous-monoïdes libres, *C.R. Acad. Sci. Paris, 261* (1965), 2419-2420.

[15] M.P. SCHUTZENBERGER – On synchronizing prefix codes, *Information and Control 11* (1967), 396-401.

[16] M.P. SCHUTZENBERGER – On a factorization of free monoïds, *Proc. Amer. Math. Soc. 16* (1965), 21-24.

[17] G. VIENNOT – *Algèbres de Lie libres et monoïdes libres,* Thèse, Paris (1974).