INFORMATION AND CONTROL 25, 93–101 (1974)

On a Question of McNaughton and Papert

ANTONIO RESTIVO

Laboratorio di Cibernetica del Consiglio Nazionale delle Ricerche,
Arco Felice, Napoli, Italy

In a recent book, McNaughton and Papert asked under what conditions a free submonoid of a free monoid is locally testable. The answer to this question is given here. The solution relates the concept of local testability with that of synchronization in a code and the algebraic notion of conjugacy in a monoid. The finiteness of the basis (or code) which generates the free submonoid plays an essential role in our result.

INTRODUCTION

Answering a question of McNaughton and Papert (1971), we provide conditions under which a free submonoid of a free monoid is strictly locally testable. The solution is found connecting three independent notions: conjugacy, synchronization, and local testability, which we introduce in Sections 2, 3, and 4, respectively. The notion of conjugacy in a free monoid is the natural extension of the analogous notion in group theory; a wider investigation of this concept can be found in Lentin and Schützenberger (1967).

Synchronization is a familiar notion in coding theory. It was first investigated by Golomb and Gordon (1965) and in a slightly different form is also considered in McNaughton and Papert (1971) in connection with our problem.

The approach we follow goes along lines developed in the works of Schützenberger (1956 a and b) dealing with the application of algebraic methods to the theory of codes.

Another result, closely related with the main theorem of the present paper, is reported in Restivo (1973).

1. PRELIMINARIES

Let X be a finite, nonempty set, and let X^* be the free monoid generated by X.

We call *letters* the elements of X, *words* the elements of X^*, and denote by $|f|$ the length of the word $f \in X^*$.

Let A be a subset of X^*. A is a *code* iff A^* has a unique factorization in terms of elements of A; A is also called a *basis* of A^*. Equivalently, we can say that A is a code iff A^* is a free submonoid of X^* and has the basis A.

The following theorem, due to Schützenberger (1956b), will be used in the sequel.

THEOREM 1.1. *A necessary and sufficient condition for A^* to be free and of basis A is that*

$$\forall f \in X^* \qquad fA^* \cap A^* \cap A^*f \neq \varnothing \Rightarrow f \in A^*.$$

The family of codes considered in this paper is a special subfamily of the *pure* codes. In order to introduce this last notion, let us give some other definitions.

An element $f \in X^*$ is primitive iff any relation $f = g^p$ implies $f = g$. Every word f of a free monoid is in a unique matter a power of a primitive element which we call the *root* of f and which we designate by $\sqrt{f}$.

A code A is *pure* iff $\forall f \in X^*, f \in A^* \Rightarrow \sqrt{f} \in A^*$.

2. CONJUGACY

DEFINITION 2.1. Let A be a code and A^* the free submonoid of X^* generated by A. If f and g are two elements of X^*, we define the relation of *A-conjugacy* as follows:

$$f\ A\text{-conj. } g \Leftrightarrow \exists h, h' \in A^* \text{ such that } f = hh' \text{ and } g = h'h.$$

If neither h nor h' is the empty word, then f and g are *strictly A-conjugate*. Two words X-conjugate (strictly X-conjugate) are simply called *conjugate* (*strictly conjugate*).

It is clear that if f and g are A-conjugate, they are also conjugate. The converse is not generally true.

DEFINITION 2.2. Let A be a code. A is *very pure* iff

$$\forall f, g \in A^* \qquad f \text{ strictly conj. } g \Rightarrow f \text{ strictly } A\text{-conj. } g.$$

Equivalently, we can say that a code A is very pure iff

$$\forall h, h' \in X^* \qquad hh', h'h \in A^* \Rightarrow h, h' \in A^*.$$

An immediate consequence of the Theorem 1.1 is that, in a code A,

hh', $h'h$, $h \in A^*$ implies $h' \in A^*$. We have then the following useful statement: If a code A is *not* very pure, there exist h, $h' \in X^*$ such that hh', $h'h \in A^*$, and *neither* h *nor* h' belongs to A^*.

We now give a necessary and sufficient condition for a code A to be very pure, which will be used in the sequel.

THEOREM 2.1. *A code A is very pure iff*

$$\forall h, h' \in X^* \qquad hh', h'h \in A^* \Rightarrow (hh')^*h \cap A^* \neq \varnothing.$$

Proof. *Necessity*. Straightforward. Indeed if A is very pure,

$$hh', h'h \in A^* \Rightarrow (hh')^*h \subset A^*.$$

Sufficiency. If A is not very pure, there exist h, $h' \notin A^*$ such that hh', $h'h \in A^*$. We prove, proceeding by induction, that

$$\forall n \geqslant 0 \qquad (hh')^n h \notin A^*.$$

This is trivially true for $n = 0$. In order to show that if it is true for n it is also true for $n + 1$, it is sufficient to prove the following implication

$$(hh')^{n+1}h \in A^* \Rightarrow (hh')^n h \in A^*.$$

Put $f = (hh')^n h$. One has

$$(hh')^{n+1}h = f(h'h) = (hh')f.$$

If $(hh')^{n+1}h \in A^*$, then, by Theorem 1.1, also $f \in A^*$.

3. SYNCHRONIZATION

Let us recall some definitions from coding theory dealing with the concept of synchronization.

DEFINITION 3.1. Let A be a code. A pair (a, a') of elements of A^* is *synchronizing* iff

$$\forall f, f' \in X^* \qquad faa'f' \in A^* \Rightarrow fa, a'f' \in A^*.$$

A has *synchronization delay* q iff every pair (a, a') of elements of A^q is *synchronizing*. A has *finite synchronization delay* iff it has synchronization delay q for some q.

The notion of finite synchronization delay is related to the possibility of decoding a message, that is placing parsing lines between code words, without knowing its beginning and its end. Given indeed a message written in a code A with synchronization delay q, for any space between letters in the message, the occurrence of q elements of A on its left and q elements of A on its right implies the existence, in this space, of a parsing line which separates an element of A from the next. We now give a stronger definition.

DEFINITION 3.2 (McNaughton and Papert, 1971). Let A be a code. A is *m-parsable* iff for any space between letters in a word of A^*, we can decide whether to place a parsing line there by looking only at the m letters on the left and the m letters on the right of that space. A is *locally-parsable* iff it is m–parsable for some m.

It is clear from the definitions that every locally parsable code has a finite synchronization delay. The converse is not always true, because in an infinite code the length of the code words is not bounded, and the possibility of decoding considering only a finite number of code words does not coincide with the possibility of decoding considering only a finite number of letters. However, the following proposition, whose proof is straightforward, holds.

PROPOSITION 3.1. *Let A be a finite code. If A has finite synchronization delay then it is locally parsable.*

We now prove a sufficient condition for a code A to have finite synchronization delay. Let us first give a definition. Let f be a word in AA^*. An *A-decomposition* of f is a pair (f_1, f_2) of elements of AA^* such that $f_1f_2 = f$.

THEOREM 3.1. *A code A has synchronization delay q if for any $g \in A^{q+1}$ and for any $f, f' \in X^*$ such that $fgf' \in A^*$, there exists an A-decomposition (g_1, g_2) of g such that (fg_1, g_2f') is an A-decomposition of fgf'.*

Proof. Assume that A has *not* synchronization delay q. Then there exist $a, a' \in A^q$ and $f, f' \in X^*$ such that $faa'f' \in A^*$, but either $fa \notin A^*$ or $a'f' \notin A^*$, or both.

Put

$$\begin{aligned} a &= a_1a_2 \cdots a_q \\ a' &= a_{q+1}a_{q+2} \cdots a_{2q} \qquad a_i \in A \\ faa'f' &= b_1b_2 \cdots b_n \qquad b_i \in A. \end{aligned}$$

Let us consider the relation

$$fa_1a_2 \cdots a_i = b_1b_2 \cdots b_j .$$

If it is satisfied for $i < q$, then one has $fa \in A^*$. Indeed,

$$fa_1a_2 \cdots a_i \in A^* \quad \text{and} \quad a_{i+1}a_{i+2} \cdots a_q \in A^* \quad \text{imply} \quad fa \in A^*.$$

If the relation is satisfied for $i > q$, then ones has $a'f' \in A^*$. Indeed,

$$a_{q+1}a_{q+2} \cdots a_i \in A^*$$

$$\text{and} \quad a_{i+1}a_{i+2} \cdots a_{2q}f' = b_{j+1}b_{j+2} \cdots b_n \in A^* \quad \text{imply} \quad a'f' \in A^*.$$

Since the code A has not synchronization delay q, then the above relation can be satisfied either for $i < q$, or for $i > q$, but cannot be satisfied in both the cases.

If the relation is satisfied only for $i < q$, one has that for the element $g = a_qa_{q+1} \cdots a_{2q} \in A^{q+1}$ the hypothesis of the theorem is not true. If the relation is satisfied only for $i > q$, one has that for the element $g = a_1a_2 \cdots a_qa_{q+1} \in A^{q+1}$ the hypothesis of the theorem is not true.

4. Locally Testable Languages

We call language any subset of X^*. In this section we deal with the notion of locally testable language, whose strings are characterized by the simple presence and absence of segments of a certain length. In order to give the formal definitions, let us introduce some notation. If $f \in X^*$ and $|f| \geqslant k$, let $P_k(f)$ be the length-k prefix of f, let $S_k(f)$ be the length-k suffix of f, and let $I_k(f)$ be the *set* of all interior length-k segments of f.

Definition 4.1. Let k be a positive integer. A language $L \subset X^*$ is *strictly k-testable* iff there exist subsets P, S, and I of X^k such that for all $f \in X^*$, of length k or more, $f \in L$ iff $P_k(f) \in P$, $S_k(f) \in S$, and $I_k(f) \subset I$. L is *strictly locally testable* iff it is strictly k-testable for some k.

In other words, a language L is strictly k-testable iff it is possible to decide if a word $f \in X^*$, of length at least k, is or is not an element of L by only making on each length-k segment of f independent tests to verify if these segments do or do not belong to a prescribed set.

The family of strictly locally testable languages is not closed under the Boolean operations (i.e., union, intersection, and complementation). Its closure is the class of *locally testable* languages.

Our theory is concerned only with strict local testability, which moreover seems to us a more natural concept than local testability. (For a more detailed discussion on this two concepts, see MacNaughton, 1971).

5. Main Result

Theorem 5.1. *Let A be a finite code. The following three conditions are equivalent*:

(a) *A is very pure.*

(b) *A has finite synchronization delay.*

(c) *A^* is strictly locally testable.*

Proof. (a) $\Rightarrow$ (b). Let $\ell(A)$ be the length of the longest element of A and card(A) the cardinality of A. Let $\nu = \text{card}(A)[\ell(A)]^2$. We prove that if A has synchronization delay $q \geqslant \nu$, then A is not very pure. By Theorem 3.1, there exist $g \in A^\nu$ and $f, f' \in X^*$ such that $fgf' \in A^*$ and for every A-decomposition (g_1, g_2) of g, (fg_1, g_2f') is not an A-decomposition of fgf'.

Put

$$g = a_1a_2 \cdots a_\nu \qquad a_i \in A,$$
$$fgf' = b_1b_2 \cdots b_n \qquad b_i \in A.$$

Then for every α and β one has

$$fa_1a_2 \cdots a_\alpha \neq b_1b_2 \cdots b_\beta .$$

We introduce a relation between the indices α and β. We say that α *precedes* β, or that β *follows* α, iff $fa_1a_2 \cdots a_\alpha$ is a prefix, or left factor, of $b_1b_2 \cdots b_\beta$. To every pair (α, β) such that α *precedes* β, there corresponds uniquely an element $h_{\alpha\beta} \in X^*$ such that

$$fa_1a_2 \cdots a_\alpha h_{\alpha\beta} = b_1b_2 \cdots b_\beta .$$

We say that α and β are *associated* iff α is the greatest index which *precedes* β and β is the smallest index which *follows* α.

We enumerate the pairs of *associated* indices following the increasing value:

$$(\alpha_1\beta_1), (\alpha_2\beta_2), \ldots, (\alpha_N\beta_N), \qquad N \leqslant \nu.$$

Let $(\alpha_i\beta_i)$ be a pair of *associated* indices. The smallest index β which *follows* $\alpha_i + 1$ is, as a consequence of the definition, β_{i+1} ; $h_{\alpha_i+1\beta_{i+1}}$ is then a suffix (or right factor) of $b_{\beta_{i+1}}$. Considering also the possibility of the occurrence of elements of A of unit length, one has

$$\alpha_{i+1} - (\alpha_i + 1) \leqslant | b_{\beta_{i+1}} | - 2$$

and then

$$\alpha_{i+1} - \alpha_i \leqslant | b_{\beta_{i+1}} | - 1 \leqslant \ell(A) - 1.$$

From the above relation, it follows by simple manipulations that

$$N \geqslant ((\nu - 1)/(\ell(A) - 1)) - 1 > \operatorname{card}(A)\, \ell(A).$$

Let $h_i = h_{\alpha_i\beta_i}$. h_i is, by construction, a prefix of an element of A. The cardinality of the set of all the prefixes of A is

clearly less than card(A)[$\ell(A)$.

In the above construction, we have defined a set of N prefixes with N greater than the number of all the prefixes; then at least two of them, for instance h_r and h_s , with $r < s$, must coincide.

Let $h_r = h_s = h$. Let h' be the element of X^* that is both prefix of $b_{\beta_r+1}b_{\beta_r+2} \cdots b_{\beta_s}$ and suffix of $a_{\alpha_r+1}a_{\alpha_r+2} \cdots a_{\alpha_s}$. One has

$$hh' = a_{\alpha_r+1}a_{\alpha_r+2} \cdots a_{\alpha_s} \in A^*,$$
$$h'h = b_{\beta_r+1}b_{\beta_r+2} \cdots b_{\beta_s} \in A^*.$$

We can readily verify that h is not an element of A^*; otherwise $hh'h$ would have two different factorizations in terms of elements of A in contradiction with the freedom of A^*. Then A is not very pure.

(b) ⇒ (c). If A has finite synchronization delay and is finite, then by Proposition 3.1, it is k-parsable for some k. We prove that A^* is k'-strictly testable, where $k' = 2k + 2\ell(A) - 1$; it is sufficient to show that one can decide for each string $f \in X^*$ if it does or does not belong to A^* by independent tests on each length-k' segment of f. Indeed, if $f \in A^*$, by looking each time only at k' consecutive letters of the string f, we are able to place at least two consecutive parsing lines between code words and then verify that the word delimited by these lines is an element of A. If $f \notin A^*$, the described procedure fails. (See Exercise 6, p. 31, McNaughton and Papert, 1971).

(c) ⇒ (a). Let A^* be strictly k-testable for some k; then there exist three subsets P, S, and I of X^k such that all and only the words of A^* have prefix, suffix, and interior segments of length k in P, S, and I, respectively. If A is not very pure, there is in A^*, by Theorem 2.1, a pair of conjugate elements $(hh', h'h)$ such that $(hh')^*h \cap A^* = \varnothing$.

By hypothesis, we have that the length-k prefixes, suffixes, and interior segments of $(hh')^k$ and $(h'h)^k$ belong to P, S, and I, respectively. Then, for every element $f \in (hh')^*h$, of length of at least k, we have that $P_k(f) \in P$, $S_k(f) \in S$, and $I_k(f) \subset I$. But $f \notin A^*$, and this is in contradiction with the hypothesis that A^* is strictly locally testable. Then A is very pure.

The arguments used in the proof of Theorem 5.1 show also that in a finite code the synchronization delay, if it is finite, is bounded by the size of the code. Indeed, proving the implication (a) ⇒ (b), we show that if A has synchronization delay $q \geqslant \text{card}(A)[\ell(A)]^2$, then A is not very pure; by the implications (b) ⇒ (c) ⇒ (a), A has not, then, finite synchronization delay. We have the following

COROLLARY 5.1. *Let A be a finite code. If A has synchronization delay q (finite), then* $q \leqslant \text{card}(A)[\ell(A)]^2$.

ACKNOWLEDGMENTS

I wish to express my deep gratitude to Professor M. P. Schützenberger without whose encouragement and invaluable technical assistance this work would never have been written.

It is also a pleasure to thank Professor A. De Luca and Dr. S. Termini for many helpful discussions.

Particular thanks are finally due to Dr. J. F. Perrot for his critical reading of the manuscript.

RECEIVED: September 13, 1973

REFERENCES

GOLOMB, S. W., AND GORDON, B. (1965), Codes with bounded synchronization delay, *Information and Control* 8, 355.

LENTIN, A. AND SCHÜTZENBERGER, M. P. (1967), A combinatorial problem in the theory of free monoids, Proceedings of the Conference held at the University of North Carolina at Chapel Hill on Combinatorial Mathematics and its Applications, April 10–14, pp. 128–144.

McNaughton, R. (1971), Algebraic decision procedures for local testability, preprint.

McNaughton, R., and Papert, S. (1971), "Counter-Free Automata," MIT Press.

Restivo, A. (1973), Codes and aperiodic languages, Conference on Automata Theory and Formal Languages, Bonn University, July 9–12.

Schützenberger, M. P. (1956a), On an application of semigroup methods to some problems in coding, *IEEE Trans. Information Theory* 2, 47–60.

Schützenberger, M. P. (1956b), Une theorie algébrique du codage, *C.R. Acad. Sci. Paris* 242, 862–864.