North-Holland

CODES AND LOCAL CONSTRAINTS

Antonio RESTIVO
Dipartimento di Matematica e Applicazioni, Università di Palermo, Italy

1. Introduction

Recently, several papers [1–4, 6, 12] have addressed the problem of encoding digital data from a free source into a constrained set of available sequences. In many transmission or storage devices, physical limitations place some constraints on the sequences of symbols that can be transmitted or stored and it may be necessary to code the information sequence to make it acceptable to the system. The main problems concern the existence and the construction of such codes.

In [1, 2, 12], by recovering some ideas and methods of [9–11], coding algorithms have been proposed which use the formalism of symbolic dynamics. The point of view developed in this paper is very close to those of [3, 4, 6] which make use of methods and results of automata theory.

If A is the alphabet associated with the storage or transmission device, the system of constraints is described by the set T of available words over A. We consider codes X such that the set X^* of coded messages is contained in T. Extremal properties of codes, such as maximality and completeness, are introduced with respect to T and their relationships are investigated. These extremal conditions play a role in the optimization of the encoding process. We then approach the problems of existence and construction, for a given T, of finite codes verifying such extremal properties and other desiderable conditions, such as to be prefix or to be circular. The existence of such codes strongly depends on the combinatorial structure of T.

In this paper we consider local constraints, i.e. constraints such that the set T of available words is defined by specifying a finite set H of forbidden words which do not occur anywhere as factors in the words of T. Local constraints occur in many engineering applications. Moreover, this paper shows their theoretical interest: the locality of T plays indeed an essential role in most of our results.

The paper is organized as follows. Section 2 contains some basic results concerning languages, automata and codes which will be useful in the sequel. Section 3 introduces properties of codes with respect to a system of constraints T. The main result of this section states the equivalence of the notions of maximality and completeness (relative to T). Section 4 considers the problem of existence of finite codes satisfying such extremal properties. Some answers to these problems and algorithms for generating such codes are given in the last section, by introducing

0304-3975/90/$3.50

the notion of T-indecomposable set. Let us remark that our constructions provide an alternative approach to the coding methods of [1, 2, 12].

2. Languages, automata, codes

In this section we briefly introduce the basic definitions concerning languages, automata, codes and report some of their properties which will be useful in the sequel. For more details the reader is referred to [5, 8].

Let A be a finite alphabet and A^* the free monoid generated by A. The elements of A are called letters, the elements of A^* words and the length of a word u is denoted by $|u|$. A word w is *factor* of a word x if there exist words u, v such that $x = uwv$. A word w is a *left factor* (or a *prefix*) of a word x if there exists a word u such that $x = wu$. For any subset X of A^*, denote by X^* the submonoid generated by X and denote by $F(X)$ the set of factors of words in X.

We call *language* any subset of A^*. A language L is *factorial* if $F(L) = L$. L is *transitive* if for all $u, v \in L$ there exists a word $w \in A^*$ such that $uwv \in L$. If L is a factorial language, then it is the complement of an ideal of A^*, i.e. $L = A^* - A^*HA^*$. If H is finite then L is called a *local* language. A local language L is then defined by specifying a finite set H of forbidden words which do not occur anywhere as factors in words of L. A language $L \subseteq A^*$ is *strictly locally testable* if there exist three finite subsets P, S, H of A^* such that $L = (PA^* \cap A^*S) - A^*HA^*$. The maximal length of words in $P \cup S \cup H$ is the *order* of L. A local language is a strictly locally testable language without the restrictive conditions on the prefixes and suffixes of the words in the language, i.e. such that $P = S = \{1\}$, where 1 is the empty word. In particular, if L is strictly locally testable, $F(L)$ is local.

A language is *regular* if it is recognized by a finite automaton. Strictly locally testable languages are in particular regular languages. A factorial regular language can be recognized by a finite automaton such that all co-accessible states are final states. A finite (deterministic) automaton $\mathcal{A}$ is *local* if there exists a positive integer n such that, for all words w of length $|w| \geq n$, $\mathrm{Card}(Q.w) \leq 1$ (Q is the set of states of $\mathcal{A}$). The least integer n satisfying this condition is called the *order* of $\mathcal{A}$. Following Schützenberger [16], a word $w \in A^*$ is a *constant* for $L \subseteq A^*$ if, for all $v_1, v_2, v_3, v_4 \in A^*$, v_1wv_2, $v_3wv_4 \in L$ implies $v_1wv_4 \in L$. The following theorem synthesizes Proposition 2.2 of [7] and Proposition 2.1 of [16].

Theorem 2.1. *Let L be a regular language and let $\mathcal{A}$ be the minimal deterministic automaton recognizing L. The following three conditions are equivalent:*

(i) *L is strictly locally testable of order k;*

(ii) *$\mathcal{A}$ is local of order k;*

(iii) *all words of A^* of length $\geq k$ are constants for L.*

A set of words $X \subseteq A^*$ is a *code* if it is the minimal generating set of a free submonoid of A^*. In other words X is a code if every word of X^* admits a unique

factorization in words of X. The following characterization of free submonoids, due to Schützenberger, will be useful in the sequel.

Theorem 2.2. *A submonoid M of A^* is free if and only if, for all $u \in A^*$, $uM \cap M \neq \emptyset$ and $Mu \cap M \neq \emptyset$ imply $u \in M$.*

A code $X \subseteq A^*$ is *maximal* if it is no proper subset of any other code over the same alphabet. A set of words $X \subseteq A^*$ is *complete* if $F(X^*) = A^*$. In other words X is complete if any word of A^* is factor of some word of X^*. In this sense a complete code uses the whole capacity of the channel. The following fundamental theorem of Schützenberger states the equivalence, in particular for regular codes, of these two notions.

Theorem 2.3. *Let X be a regular code. X is maximal if and only if it is complete.*

A subset X of A^* is *prefix* if no word of X is prefix of another word of X. It is easy to verify that any prefix set is a code, which is called *prefix code.* A submonoid M of A^* is *very pure* if, for all u, $v \in A^*$, $uv \in M$ and $vu \in M$ imply $u \in M$ and $v \in M$. As a consequence of Theorem 2.2, a very pure submonoid is free. A set X is a *circular code* if it is the minimal generating set of a very pure submonoid of A^*. We are interested in circular codes since they have some remarkable synchronization properties [5, 14] and then limited error propagation in decoding. The following result [13] relates the notions of circular code and locally testable language.

Theorem 2.4. *Let X be a finite code. X is a circular code if and only if X^* is a strictly locally testable language.*

As a consequence of this theorem, if X is a finite circular code, then $F(X^*)$ is a local language.

In the sequel of the paper, by the word "language" we mean a factorial and transitive language, and by the word "automaton" we mean a finite deterministic (non-complete) automaton such that all states are initial and final states. It is obtained from the complete automaton by deleting the "sink" state. Moreover, the state graph of this automaton is strongly connected.

3. Codes with constraints

Let us consider an encoding process into a "channel" (a storage or transmission device) satisfying a given system of constraints. If A is the channel alphabet, the system of constraints can be described by the set $T \subseteq A^*$ of available words over A. Time invariance and ergodicity of the encoding process imposes that T is a

factorial and transitive language. According to a previous remark, T is here called simply a language. In most applications, T is a regular language, i.e. the system of constraints is described by a finite automaton: this corresponds to sofic systems of symbolic dynamics. In particular we are interested here in the case of local constraints, i.e. a system of constraints described by a local language T: this corresponds to a system of finite type of symbolic dynamics.

Any subset $X \subseteq T$ is a *T-set* if $X^* \subseteq T$. A *T-code* is a T-set which is a code. T-codes may be used for coding messages from a source to a channel satisfying a system of constraints described by T. We now introduce the notions of maximality and completeness with respect to T. A T-code is *T-maximal* if it is no proper subset of any other T-code. A T-set X is *T-complete* if $F(X^*) = T$. Best encoding usually requires T-complete codes: in fact they use the whole capacity of the transmission channel in the sense that every available word of T occurs as a factor in coded messages. Let us now investigate the relations between the notions of T-maximality and T-completeness. We first give the following definition. A T-set X is *thin* if there exists at least a word $w \in T$ such that $TwT \cap X = \emptyset$. A regular T-set is in particular thin.

Theorem 3.1. *Let T be a local language and let X be a thin T-code. X is T-maximal if and only if it is T-complete.*

Proof. The proof of the "if" part of the theorem follows that of the analogous Theorem 2.3 of Schützenberger. It uses Theorem 4.5 in [5]. Let μ be the morphism of A^* in the syntactic monoid of X^*. Let $x \in X^*$ such that $e = \mu(x)$ is an idempotent of the 0-minimal ideal J of $\mu(A^*)$ (such an idempotent exists by Theorem 4.5 of [5], Claim 3). If X is T-complete, then $t \in T$ if and only if $\mu(t) \neq 0$. Let $y \in T - X$. If $e\mu(y)e = 0$, then $xyx \notin T$ and so $X \cup \{y\}$ is not a T-code. If $e\mu(y)e \neq 0$ then it is in the $\mathcal{H}$-class of e, which is a group. There exists then an integer $n \geq 1$ such that $(\mu(xyx))^n = e$. Consequently $(xyx)^n = x \in X^*$. This shows that $X \cup \{y\}$ is not a code.

In order to prove the "only if" part of the theorem, let us first consider the case in which T is of the form $T = F(w^*)$ for some primitive word $w \in A^*$. If X is a (non-empty) T-code, then, as a consequence of elementary facts concerning free monoids (see [5]), $\mathrm{Card}(X) = 1$ and $X = \{g\}$, where g is a conjugate of w. In this case X is a T-maximal and T-complete code. Consider now the case $T \neq F(w^*)$. The proof is by contradiction. We prove that if X is not T-complete, then there exists $z \in T - X$ such that $X \cup \{z\}$ is a T-code. Since T is local, then $T = A^* - A^*HA^*$, with H a finite set. Let k be the maximal length of words in H. Since X is not T-complete, there exists a word $t \in T - F(X^*)$ such that $|t| > k$. Consider two words $u, v \in X^*$ such that $|u|, |v| > k$. Since T is transitive, there exist $x, y \in T$ such that $z = uxtyv \in T$. We show that it is always possible to chose u, v, x, y, such that t is a *simple* factor of z, i.e. t appears once as a factor of z. Indeed, if t is not a simple factor of z, consider the first and the last occurrence of t in z given by the factorizations $z = z_1tz_2 = z_3tz_4$. Since $|t| > k$ and T is local, by Theorem 2.1, t is a

constant for T. It follows that $z_1 t z_4 \in T$ and t is, by construction, a simple factor of this word. The fact that t is a simple factor of z implies that z is *unbordered*, i.e. no proper prefix of z is a suffix of z itself. We prove that

(i) $(X \cup \{z\})^* \subseteq T$.

It suffices to consider the factors of length k of the words zz, wz and zw for $w \in X^*$. Since, by construction, the prefix and the suffix of length k of z belong to X^*, the factors of length k of previous words do not belong to H.

(ii) $X \cup \{z\}$ is a code.

This is a consequence of the fact that $z \notin F(X^*)$ and z is unbordered (see [5]). This concludes the proof of the theorem. □

Remark 3.2. The "only if" part of previous theorem does not hold if T is not local, even in the case of finite codes, as shown by the following example. Let T be the set of all words over the alphabet $A = \{a, b\}$ such that consecutive "a"s appear in blocks of *even* length between two "b"s. T is a (non-local) regular language. Consider the T-code $X = \{a\}$. X is a T-maximal code. Suppose indeed, by contradiction, that $\{a\} \cup \{x\}$ is a T-code for some $x \in T$. $x \notin a^*$, otherwise $\{a, x\}$ is not a code. x is then of the form $x = a^i u a^j$, with $i, j \geq 0$ and $u \in bA^*b \cup \{b\}$. It follows that $\{a, x\}^* \not\subseteq T$. However, X is not T-complete: in fact $F(X^*) = a^* \subsetneq T$.

We consider now the problem of existence and construction of T-maximal codes (any other T-code can be obtained, by definition, as a subset of a T-maximal code). In the case of local constraints, by the previous theorem, this problem leads to the study of T-complete codes. The following theorem, which synthesizes Proposition 1.3 of [4] and Corollary II-2 of [6], gives a preliminary answer to our problem.

Theorem 3.3. *For any regular language T, there exists a regular T-complete prefix code. For any local language T, there exists a regular T-complete prefix circular code.*

The proof of this theorem also gives a construction of such T-complete codes. However, this construction generates only prefix codes and there exist T-complete codes that cannot be obtained in this way. In particular the theorem does not give any criterium to decide whether there exist *finite* T-complete codes. The finiteness of the codes is indeed the interesting feature in many applications.

4. Finite T-complete codes

In this section we consider the following problems: given a local language T, does there exist a *finite* T-complete set (resp. code, prefix code, circular code)? The answer to these questions depends on the structure of T.

Proposition 4.1. *There exist local languages T which do not admit finite T-complete sets, i.e. such that $T \neq F(X^*)$ for any finite set X.*

Proof. Let T be a local language defined by a set of forbidden factors H such that, for some $a, b \in A$, $a^+b^+ \cap H \neq \emptyset$, $a^+ \cap H = b^+ \cap H = \emptyset$. Let us suppose, by contradiction, that there exists a finite set X such that $T = F(X^*)$ and let k be the maximal length of words in X. By the hypothesis on H, $a^n, b^n \in T$ for $n > k$, and then there exist integers $p, q > 1$ such that $a^p, b^q \in X$. One derives $a^+b^+ \subseteq T$ against the hypothesis. The simplest example of language T satisfying the conditions of the proposition is the local language over the alphabet $\{a, b, c\}$ defined by the set of forbidden factors $H = \{ab\}$. □

Concerning the existence of finite T-complete codes, Ashley (private communication), answering a conjecture in a preliminary version of this paper, has found an example of a local language T which admits a finite T-complete set, but does not admit finite T-complete codes. Further examples have also been found by Blanchard (private communication).

Example 4.2 (*Ashley*). Let T be the local language over the alphabet $A = \{a, b, c\}$ defined by the set of forbidden factors $H = \{bba, bbb, cba, cbb\}$. A finite T-complete set (which is not a code) is $X = \{a, bc, ab, c\}$. Ashley proves, using techniques from symbolic dynamics, that no finite T-complete code exists. We then have the following proposition.

Proposition 4.3. *There exist local languages T which admit finite T-complete sets, but do not admit finite T-complete codes.*

Example 4.4. Let T be the local language over the alphabet $A = \{a, b\}$ defined by the set of forbidden factors, $H = \{aba, bab\}$. A finite T-complete set (which is not a code) is $X = \{a^2, a^3, b^2, b^3\}$. A finite T-complete code is $Y = \{a^2, b^2, a^3b^3, a^2b^3, a^3b^2\}$. Y is not a circular code: indeed $(a)(a) \in Y^*$ but $a \notin Y^*$. We can prove that no finite T-complete circular code exists. In fact, if Z is such a code, let k be the maximal length of words in Z. Since $a^n, b^n \in T$ for $n > k$, there exist integers $p, q > 1$ such that $a^p, b^q \in Z$ and this contradicts the hypothesis that Z is circular. We have then proved the following proposition.

Proposition 4.5. *There exist local languages T which admit finite T-complete codes, but do not admit finite T-complete circular codes.*

Remark 4.6. The problem of existence of finite T-complete circular codes can be posed only in the case where T is a local language. In fact, as a consequence of Theorem 2.4, if T is not local, no finite T-complete circular code exists.

Proposition 4.7. *There exist local languages T which admit finite T-complete codes, but do not admit finite T-complete prefix codes.*

Proof. The proof is given by the following example. Let T be local language over the alphabet $A=\{a, b, c\}$ defined by the set of forbidden factors $H=\{ac, cc\}$. A T-complete (non-prefix) code is $X=\{a, b, bc\}$. One can verify that no finite T-complete prefix code exists. □

Remark 4.8. The previous example also shows that, for some local language T, there exist prefix T-codes which are maximal as prefix T-codes, but are not maximal as T-codes. The set $Y=\{a, b\}$ verifies these conditions in the previous example.

5. T-Indecomposable sets

In order to give an answer to the questions of the previous section and to construct, for a given T, T-complete sets and T-complete codes, we introduce the notion of a T-indecomposable set. Let $M \subseteq T$ be a submonoid of A^*. M is a *T-maximal* submonoid if it is no proper subset of any other submonoid of A^* contained in T. A set $X \subseteq T$ is *T-indecomposable* if it is the minimal generating set of a T-maximal submonoid. A T-indecomposable set works like an "alphabet for T". Indeed any T-set is, by definition, included in some X^*, where X is T-indecomposable: $Y \subseteq X^*$. In this sense the words of Y are "written" over the "alphabet" X.

Example 5.1. Let T be the local language over the alphabet $A=\{a, b, c\}$ defined by the set of forbidden factors $H=\{aa, ab\}$. The set $X=\{ac, b, c\}$ is the *unique* finite T-indecomposable set. In fact, let Y be a finite T-complete set. We have to prove that $Y^* \subseteq X^*$. Since $X^* = T \cap A^*\{b, c\}$, it suffices to prove that no word of Y ends with the letter "a". Indeed, since Y is T-complete and finite, there exists an integer k such that $b^k \in Y$. If $ua \in Y$, for some $u \in A^*$, then $uab^k \in T$, against the definition of T.

The problem to produce T-complete sets for a given T leads to the problem of constructing T-indecomposable sets. For this we introduce the notion of a stabilizer of a set of states of an automaton. Let T be a regular language and let $\mathcal{A}=(A, Q, q_0)$ be the minimal deterministic automaton recognizing T. For any subset S of the set of states Q, the *stabilizer of S* is defined as follows:

$$\mathrm{Stab}(S)=\{v \in A^* \mid S.v \subseteq S\}.$$

The following properties of $\mathrm{Stab}(S)$ can be easily derived:

(i) $\mathrm{Stab}(S)$ is a submonoid of A^*;

(ii) if $\mathrm{Card}(S)=1$, then $\mathrm{Stab}(S)=X^*$ where X is a prefix code (see [5]); if $\mathrm{Card}(S)>1$, then $\mathrm{Stab}(S)$ is not in general a free submonoid of A^*;

(iii) if $q_0 \in S$, then $\mathrm{Stab}(S) \subseteq T$.

Given $\mathcal{A} = (A, Q, q_0)$ and $S \subseteq Q$, a finite automaton $\mathcal{A}(S)$ recognizing Stab(S) can be constructed as follows. The set of states of $\mathcal{A}(S)$ is the set Q_S of subsets of Q accessible starting from the state $S \in Q_S$. If $V \in Q_S$ and $a \in A$, the transition function $V.a$ is defined as follows: $V.a = \{q.a \mid q \in V\}$ if $q.a$ is defined for all $q \in V$, $V.a$ is not defined otherwise. The initial state of $\mathcal{A}(S)$ is S and $W \in Q_S$ is a final state if $W \subseteq S$. We prove the following theorem.

Theorem 5.2. *Let T be a regular language recognized by the automaton $\mathcal{A} = (A, Q, q_0)$. For any T-indecomposable set X, there exists a set $S \subseteq Q$ such that $X^* = \text{Stab}(S)$.*

Proof. It suffices to prove that, if $M \subseteq T$ is a submonoid of A^*, there exists then a set $S \subseteq Q$ such that $M \subseteq \text{Stab}(S)$. Set $S = \{q = q_0.v \mid v \in M\}$. Let $u \in M$. We have to prove that, for any $q \in S$, $q.u \in S$. Indeed, $q \in S$ implies that there exists $v \in M$ such that $q = q_0.v$. Then $q.u = q_0.vu \in S$, since $vu \in M$. □

Corollary 5.3. *For any regular language T, there exist a finite number of T-indecomposable sets which can be effectively constructed.*

We use this result to give an answer to a question from a previous section.

Theorem 5.4. *Given a local language T, it is decidable whether there exists a finite T-complete set.*

Proof. We prove that, if there exists a finite T-complete set, then there exists also a finite T-indecomposable set. The decision procedure can then be obtained by the constructions of the previous theorem. Let X be a finite T-complete set, i.e. $F(X^*) = T$. By definition, X^* is contained in some Z^*, where Z is a T-indecomposable set. Let us suppose, by contradiction, that Z is infinite. Since X is finite, there exists a finite subset $Y \subseteq Z$ such that $X^* \subseteq Y^*$. Since X is T-complete, $F(Y^*) = F(Z^*) = T$. T is local and then defined by a finite set H of forbidden factors. Let k be an integer greater than the maximal length of words in $H \cup Y$, and consider a word $z \in Z$ of length $|z| > 3k$. Since $F(Y^*) = F(Z^*)$, there exist words $s, t \in A^*$ such that

$$szt = y_1 y_2 \ldots y_n, \quad y_i \in Y.$$

By a length argument, there exists a factorization $z = z_1 z_2$ with $|z_1|, |z_2| > k$ and such that z_1 has a suffix of length greater than k in Y^* and z_2 has a prefix of length greater than k in Y^*. One derives that the factors of length k of the words $z_2 z_1$, wz_1, $z_1 w$, wz_2, $z_2 w$, with $w \in Z^*$, do not belong to H. It follows that no word in the submonoid $M = (Z \cup \{z_1\} \cup \{z_2\})^*$ has a factor in H. Thus $Z^* \subsetneq M \subseteq T$, against the hypothesis that Z is T-indecomposable. This concludes the proof. □

The next theorem gives a method to construct a T-complete code Y from a T-indecomposable set X. However, if X is finite, the code Y obtained by this

construction may be infinite. Let T be a local language defined by the finite set H of forbidden factors and let k be the maximal length of words in H. Let X be a T-indecomposable set, m an integer $\geq k$ and let Y be the minimal generating set of the submonoid $X^* \cap (A^m)^*$.

Theorem 5.5. *Y is a T-complete code.*

Proof. It is easy to verify that Y is T-complete. In order to prove that Y^* is a free submonoid, consider a word $u \in A^*$ such that (i) $uY^* \cap Y^* \neq \emptyset$ and (ii) $Y^*u \cap Y^* \neq \emptyset$. We show that $u \in Y^*$ and then, by Theorem 2.2, Y^* is free. By previous conditions (i) and (ii), $|u| = dm$ for some integer d, i.e. $u \in (A^m)^*$. It remains to prove that $u \in X^*$. Let $\mathcal{A} = (A, Q, q_0)$ be the minimal deterministic automaton recognizing T. By Theorem 2.1, $\mathcal{A}$ is a local automaton such that, for all $w \in A^*$ with $|w| \geq k$, $\mathrm{Card}(q.w) \leq 1$. Since X is a T-indecomposable set, by Theorem 5.2, there exists a set $S \subseteq Q$ such that $X^* = \mathrm{Stab}(S)$. Condition (i) and the fact that $m \geq k$ imply that $S.u = t \in Q$. Condition (ii) implies that $t \in S$. By recalling the construction of the automaton recognizing $\mathrm{Stab}(S)$, it follows that $u \in \mathrm{Stab}(S) = X^*$. Thus $u \in Y^*$. This concludes the proof. □

Previous results lead to the following coding scheme. Given a system of local constraints described by the (local) automaton $\mathcal{A}$ recognizing the set T of available words, construct the T-indecomposable sets by computing $\mathrm{Stab}(S)$, where S is a set of states of $\mathcal{A}$. For each T-indecomposable set X, if it is a code, one obtains a T-complete code; if it is not a code, construct the base of the submonoid $X^* \cap (A^i)^*$ for $i = 1, 2, \ldots$, until one obtains a code. The previous theorem guarantees that one obtains a code for $i \leq k$, where k is the maximal length of words in the set H of forbidden words.

Example 5.6. A local constraint frequently encountered in engineering applications consider words over the alphabet $A = \{0, 1\}$ with forbidden factors specified by lower and upper bounds on the number of consecutive "0"s separating the "1"s. It is called a $[d, k]$ *run-length constraint*, where d is the lower and k is the upper bound. We apply our methods for the $[2, 7]$ run-length constraint and in particular we derive a code obtained by Franaszek by different methods [9]. The set T of available words is described by the set $H = \{11, 101, 00000000\}$ of forbidden factors. An automaton recognizing T is shown in Fig. 1. The automaton recognizing $\mathrm{Stab}(\{2, 3\})$ obtained by our construction and by minimization is shown in Fig. 2.

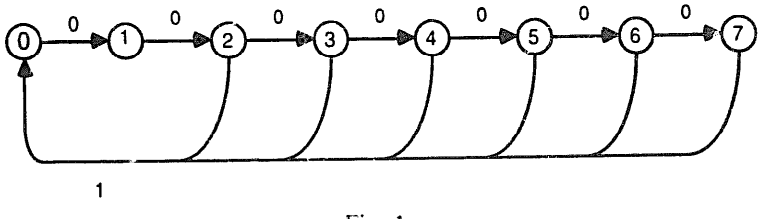

Fig. 1.

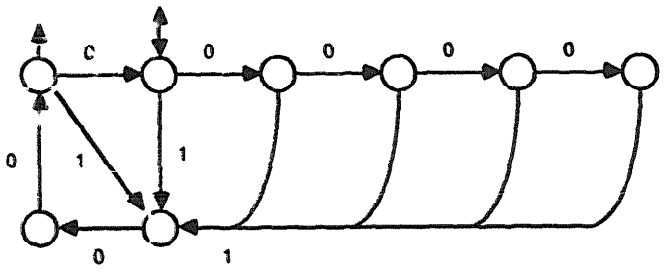

Fig. 2.

We derive the following T-indecomposable set:

$$X = \{1, 01, 001, 0001, 00001\}.\{00, 000\}.$$

The set X is not a code as one can easily verify. We construct then the base Y of the submonoid $X^* \cap (A^2)^*$. Y is a code and the set of its words of length less than or equal to 8 is exactly the Franaszek code

$$\{0100, 1000, 000100, 001000, 100100, 00100100, 00001000\}.$$

Acknowledgment

I would like to thank J. Ashley, F. Blanchard, D. Perrin and the anonymous referee for their helpful suggestions and comments.

References

[1] R. Adler, D. Coppersmith and M. Hassner, Algorithms for sliding block codes, *IEEE Trans. Inform. Theory* **29** (1983) 5-22.
[2] R. Adler, J. Friedman, B. Kitchens and B. Marcus, State splitting for variable-length graphs, *IEEE Trans. Inform. Theory* **32** (1986) 108-112.
[3] M.P. Béal, Codes circulaires, automates locaux et entropie, *Theoret. Comput. Sci.* **57** (1988) 283-302.
[4] M.P. Béal and D. Perrin, Une caractérisation des ensembles sofiques, *C. R. Acad. Sci. Paris* **303** (1986) 255-257.
[5] J. Berstel and D. Perrin, *Theory of Codes* (Academic Press, New York, 1985).
[6] F. Blanchard and G. Hansel, Systèmes codés, *Theoret. Comput. Sci.* **44** (1986) 17-49.
[7] A. de Luca and A. Restivo, A characterization of strictly locally testable languages and its application to subsemigroups of a free semigroup, *Inform. and Control* **44** (1980) 300-319.
[8] S. Eilenberg, *Automata, Languages and Machines* (Academic Press, New York, 1974).
[9] P.A. Franaszek, On synchronous variable length coding for discrete noiseless channel, *Inform. and Control* **1** (1969) 155-164.
[10] P.A. Franaszek, A general method for channel coding, *IBM J. Res. Dev.* **24** (1980) 638-641.
[11] P.A. Franaszek, Construction of bounded delay codes for discrete noiseless channel, *IBM J. Res Dev.* **26** (1982) 506-514.
[12] B. Marcus, Sofic systems and encoding data, *IEEE Trans. Inform. Theory* **31** (1985) 366-377.
[13] A. Restivo, On a question of McNaughton and Papert, *Inform. and Control* **25** (1974) 93-101.
[14] A. Restivo, A combinatorial property of codes having finite synchronization delay, *Theoret. Comput. Sci.* **1** (1975) 95-101.
[15] A. Restivo, Codes with constraints, Report Dip. Mat. Palermo, 1986.
[16] M.P. Schützenberger, Sur certaines opérations de fermature dans les langages rationnels, *Sympos. Math.* **15** (1975) 245-256.