Variable-Length Maximal Codes*

Véronique Bruyère[1] and Michel Latteux[2]

[1] University of Mons-Hainaut, 15 Avenue Maistriau,
B-7000 Mons, Belgium.
vero@sun1.umh.ac.be
[2] University of Lille, L.I.F.L., Bâtiment M3,
F-59655 Villeneuve d'Ascq, France.
latteux@lifl.fr

Abstract. In this survey, we present some of the main open problems in the theory of variable-length codes, together with the major advancements recently realized to solve them.

1 Introduction

The beginnings of both coding and information theory go back to the pioneer works of C.E. Shannon in 1948. Coding theory has later developed into two independent directions. One is the design of constant-length codes for the problems of error detection and correction. This theory is a nice application of commutative algebra [34, 35, 51]. The other direction, initiated by M.-P. Schützenberger in 1955 [45], has lead to the theory of variable-length codes. This theory is now clearly a part of theoretical computer science, strongly connected to formal languages, combinatorics on words, automata theory and the theory of semigroups [2, 4, 27, 32, 43, 49].

Maximal codes play a central role in the theory of variable-length codes. They reflect extremal properties of codes when they are considered as communication tools. In some sense, maximal codes make use of the whole capacity of the transmission channel. Despite a lot of research activities and important progress, some open problem on maximal codes are still open today, whose cornerstone is the so-called *commutative equivalence conjecture*. This conjecture belongs to the list of six old open problems about regular languages presented by J. Brzozowski at the International Symposium on Formal Language Theory, Santa Barbara, California, 1979 [11]. It states the optimality of prefix codes in a way that reinforces the Kraft-McMillan inequality well known in information theory.

In this paper, we want to survey some of the major developments on regular codes for the last ten years, made around the commutative equivalence conjecture and the related open problems. We particularly emphasize on the variety of the approaches. We sketch the proofs when they are simple; we provide examples in case of more complex arguments.

* This work was partially supported by a cooperation project CGRI-FNRS-CNRS.

The paper is organized as follows. In Sect. 2, we recall the basic notions on codes and the way to characterize them by simple properties of finite automata. For this section, our reference book is mainly [2]. In Sect. 3, we enunciate the commutative equivalence conjecture and its partial results. It has stimulated two other open problems, the problem of completion of finite codes discussed in Sect. 4 and the factorization conjecture studied in Sect. 5. It appears that partial answers to both questions display a strong relation with factorizations of cyclic groups. In Sect. 6, we explain what are the consequences on codes of some properties and open problems on factorizations of cyclic groups. The last section deals with the problem of embedding a code into a maximal one sharing common properties, for which recent important advancements have occurred.

2 Codes

2.1 Definitions

Assume that we are given a *source alphabet* B and a *channel alphabet* A, and we want to code messages v over B into coded messages w over A. The letters of B are put in correspondence with the words of a *code* X over A. Any letter of the source message v is then replaced by the corresponding word of X. We require that the decoding of the resulting coded message $w \in X^*$ does not lead to ambiguities, i.e., any product of words of X is factorized in a unique fashion.

More formally, $X \subseteq A^*$ is a *code* if for any $x_1, \ldots, x_n,\ y_1, \ldots, y_m \in X$, $n, m \geq 1$,

$$x_1 \cdots x_n = y_1 \cdots y_m \quad \Rightarrow \quad n = m \text{ and } x_i = y_i,\ \forall i\ .$$

Example 1. The Morse code X_M is a code over the alphabet $\{., \wedge, -\}$ used to code the letters $a, b, \ldots, z$ (see Fig. 1). The blank symbol $\wedge$ only appears at the end of words of X_M, ensuring that X_M is a code. Another example is the ASCII code X_A over the binary alphabet. Whereas the words of X_M have variable length, all the words of X_A have constant length 8.

It is reasonable to require for a code X that the decoding of a coded message can begin without waiting for the whole reception of the message. Such codes are said to have *bounded decoding delay*, that is, there exists $d \geq 0$ such that for any $x, x' \in X$, $y' \in X^*$ and y prefix of a word of X^* (see Fig. 2),

$$xy \text{ prefix of } x'y' \text{ and } |y| \geq d \quad \Rightarrow \quad x = x'\ . \tag{1}$$

The least integer d satisfying (1) is called the *delay* of X.

Codes with decoding delay 0 are the most interesting ones. They are also called *prefix* codes, or *instantaneous* codes. In this case, no word of the code is prefix of another one. A prefix code can then be represented as a tree with arity $|A|$,– the size of the alphabet. The leaves correspond to the words of the code (see Fig. 1). This shows that prefix codes are easy to construct.

a	. − ∧	j	. − − − ∧	s	...∧
b	−...∧	k	−. − ∧	t	−∧
c	−. − .∧	l	. − ..∧	u	.. − ∧
d	−..∧	m	− − ∧	v	... − ∧
e	.∧	n	−.∧	w	. − −∧
f	.. − .∧	o	− − −∧	x	−.. − ∧
g	− − .∧	p	. − −.∧	y	−. − −∧
h	....∧	q	− − . − ∧	z	− − ..∧
i	..∧	r	. − .∧		

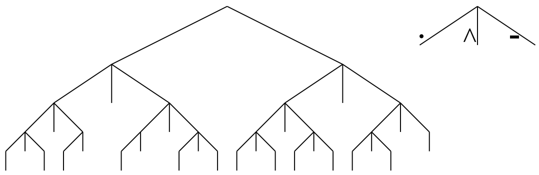

Fig. 1. Morse code.

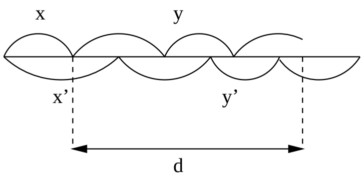

Fig. 2. Code with bounded decoding delay d

Example 2. The Morse code and the ASCII code are prefix codes. The code $X = \{a, aab\}$ is not prefix, it has a decoding delay 2. The code $X = \{a, ab, bb\}$ has no bounded decoding delay, because the coded message ab^n can be decoded only after its complete reception.

Notice that the latter example is a prefix code if one looks at its words from right to left. Such codes are called *suffix*, and a code which is both prefix and suffix is a *bifix* code.

Another interesting class is the one of codes with bounded synchronization delay. A code $X \subseteq A^*$ has a *synchronization delay* s if

$$uxyv \in X^*, \ \text{ with } \ x, y \in X^s, \ u, v \in A^* \quad \Rightarrow \quad ux, yv \in X^* \ .$$

In practice, the presence inside a coded message $uxyv$ of the synchronization pair xy forces the decoding to pass between x and y. In this way, an error occurring before x does not affect the decoding of yv.

Example 3. The Morse code X_M has synchronization delay 1, since the end of each word of X_M is "marked" by the blank symbol ∧ (see Fig. 1). The bifix code

$X = ab^*c \cup \{b\}$ has no bounded synchronization delay, because for all $s \in \mathbb{N}$, one has $b^s \in X^s$ and $ab^s c \in X$.

Many problems related to the optimization of the encoding process deal with extremal properties of codes, such as the concepts of maximality and completeness. A code $X \subseteq A^*$ is *maximal* if it is properly included in no other code over the same alphabet A. It is not difficult to see that any code is included in a maximal one. A code X is *complete* if for any word $w \in A^*$,

$$A^* w A^* \cap X^* \neq \emptyset \ .$$

Roughly, any word of A^* occurs as part of a coded message. In this sense, a complete code makes use of the whole capacity of the transmission channel.

A fundamental result of M.-P. Schützenberger [45] states the equivalence of these two notions for finite codes and more generally for the class of regular codes. We recall that a set of words is *regular* or *rational* if it is recognizable by a finite automaton [22, 26, 38].

Theorem 1. *A regular code is complete if and only if it is maximal.*

Complete prefix codes $X \subseteq A^*$ are those corresponding to trees of which all nodes have either 0 sons (the leaves) or $|A|$ sons. In this way, any word $w \in A^*$ verifies $wA^* \cap X^* \neq \emptyset$. Figure 1 shows that the Morse code is not complete.

2.2 Automata

In this paper, we focus on maximal codes which are finite or regular. The automata that we consider are strongly connected directed graphs $\mathcal{A} = (S, 1, T)$ with a set S of vertices or *states*, a particular state $1 \in S$ and a set T of edges or *transitions* labeled by an alphabet A. Notation $(r, a, s) \in T$ means that a labels an *edge* going from state r to state s. It is extended to paths: $(r, w, s) \in T^*$ means that w is the label of a *path* from r to s. The set $X_{\mathcal{A}} \subseteq A^*$ of *first returns* is defined as the set of labels of all paths going from state 1 to state 1, without passing through 1 (see Fig. 3). If the automaton $\mathcal{A}$ is finite, this set $X_{\mathcal{A}}$ is regular.

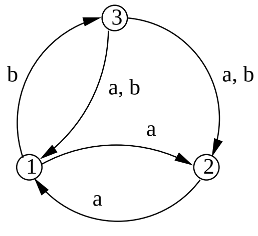

Fig. 3. Automaton $\mathcal{A}$ and set $X_{\mathcal{A}} = \{aa, ba, baa, bba, bb\}$.

Code properties are easily checked on automata. We recall that an automaton is *deterministic* if

$$(r, a, s_1), (r, a, s_2) \in T \quad \Rightarrow \quad s_1 = s_2 \ .$$

More generally, an automaton is *unambiguous* if for all $r, s \in S$ and $w \in A^*$, there is at most one path from r to s labeled by w (see Fig. 3).

Proposition 2. *A regular set X is a code if and only if $X = X_{\mathcal{A}}$ for some unambiguous finite automaton $\mathcal{A}$. Moreover X is prefix if and only if $\mathcal{A}$ can be chosen deterministic.*

Completeness of a code $X = X_{\mathcal{A}}$ is translated by the following property of the automaton $\mathcal{A}$: any word $w \in A^*$ labels some path of $\mathcal{A}$. In the case of a complete prefix code, $\mathcal{A}$ is deterministic and complete, i.e.,

$$\forall r \in S, \forall a \in A, \exists! s \in S, \quad (r, a, s) \in T.$$

Finally, a regular code $X = X_{\mathcal{A}}$ is finite if and only if all cycles of $\mathcal{A}$ go through the particular state 1.

2.3 Composition of Codes

We now introduce the operation of composition of codes for constructing more complicated codes from simpler. Beginning with prefix or suffix codes, this operation can lead to codes neither prefix nor suffix.

Let $Z \subseteq A^*$ and $Y \subseteq B^*$ be two codes together with a bijection $\varphi : B \to Z$. Then the set $X = \varphi(Y)$ obtained by coding the letters of Y by the corresponding words of Z is again a code over the alphabet A. We say that X is obtained by *composition* of Y and Z and we denote $X = Y \circ Z$.

Example 4. Let $Z = \{a, ba, bb\}$ over $A = \{a, b\}$ and $Y = \{\alpha\alpha, \beta, \beta\alpha, \gamma\alpha, \gamma\}$ over $B = \{\alpha, \beta, \gamma\}$, then $X = Y \circ Z = \{aa, ba, baa, bba, bb\}$ with φ a bijection such that $\varphi(\alpha) = a$, $\varphi(\beta) = ba$ and $\varphi(\gamma) = bb$.

Notice that in Example 4, Z is prefix, Y is suffix, but X is neither prefix nor suffix. However, if the codes used in the composition are all prefix (resp. suffix), then the resulting code is still prefix (resp. suffix). This is still true for complete codes: the composition of complete codes always gives a complete code (see Example 4).

A code which is constructed by a finite number of compositions of prefix and/or suffix codes only, is called a *prefix-suffix composed code*. Example 4 is such a code. Until the seventies, people thought that the composition of codes was a powerful tool, in the sense that any finite maximal code was prefix-suffix composed. In 1974, Y. Césari [13] found the first counterexample with a total number of 73 words over a 4-letter alphabet. J.-M. Boë [5] later proposed smaller counterexamples of size 11, such as the following code

$$X = \{b, aba, ab^2, a^4, ba^2b, aba^3, ab^2a^2, ba^3b, ba^2ba^2, ba^3ba^2, ba^6\}$$

which is not prefix, not suffix and not decomposable into simpler codes.

These results show that finite maximal codes are complex objects. In the next sections, we will enunciate some of the major conjectures on codes. These conjectures would all be true if every finite maximal code was prefix-suffix composed.

3 Optimality of Prefix Codes

In information theory, Shannon's first theorem relates the average length of a code X to the entropy of the alphabet B encoded by X [25]. More precisely, assume that X is a code over the channel alphabet A, and let $p : B \to [0,1]$ be a probability distribution of the source alphabet B. The *entropy* of B is then defined as

$$H(B) = -\sum_{b \in B} p(b) \log_{|A|} p(b) \ ,$$

and the *average length* of the code X is equal to

$$L(X) = \sum_{b \in B} p(b)|x_b|$$

such that x_b is the word of X encoding the letter b. One has the following inequality

$$H(B) \leq L(X)$$

which can be proved thanks to the Kraft-McMillan inequality:

Theorem 3. *Let $X \subseteq A^*$ be a code with length-distribution $(f_n)_{n \in \mathbb{N}}$, i.e., f_n is the number of words in X with length n. Then*

$$\sum_{n \in \mathbb{N}} \frac{f_n}{|A|^n} \leq 1 \ . \tag{2}$$

Moreover, there exists a prefix code $Y \subseteq A^*$ with the same length-distribution.

The first part of Theorem 3 shows that if a set has too many too short words, then it cannot be a code. For regular codes, one can prove that equality with 1 in (2) holds exactly when the code is maximal. The second part of the theorem shows the optimality of prefix codes. This remarkable property is used by the well-known Huffman algorithm which constructs a finite maximal prefix code X with minimal average length $L(X)$.

One central conjecture in the theory of codes states the optimality of prefix codes in a stronger way. Let us give an example.

Example 4 (continued). Consider the code $X = \{aa, ba, baa, bba, bb\}$ with length-distribution $f_2 = 3, f_3 = 2$. By Theorem 3, there exists a prefix code with the same length-distribution, namely $Y = \{aa, ab, baa, bab, bb\}$. This set Y satisfies a stronger property: it has been obtained by a rearrangement of the letters inside each word of X.

We say that two codes X and Y are *commutatively equivalent* if there exists a bijection between them such that a word and its image only differ by the ordering of their letters.

Conjecture 1. Let $X \subseteq A^*$ be a finite maximal code. Then X is commutatively equivalent to a finite maximal prefix code $Y \subseteq A^*$.

Clearly every suffix code is commutatively equivalent to a prefix code. More generally, any prefix-suffix composed code is commutatively equivalent to a prefix code (just replace each suffix code used in the composition by a commutatively equivalent prefix code,– see Sect. 2.3). See Example 4 above.

Conjecture 1 can be translated into a divisibility property of polynomials. Given a set $X \subseteq A^*$, we denote by $\underline{X}$ (resp. $\underline{\underline{X}}$) its characteristic series in non commutative (resp. commutative) variables $a \in A$. For example, if $X = \{ab, ba\}$, then $\underline{X} = ab + ba$ and $\underline{\underline{X}} = 2ab$. Given a finite maximal prefix code $Y \subseteq A^*$, we can write

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

where P is the set of proper prefixes of words of Y. Hence, if a code X is commutatively equivalent to Y, then $\underline{\underline{X}} = \underline{\underline{Y}}$ and $\underline{\underline{X}} - 1 = \underline{\underline{P}}(\underline{\underline{A}} - 1)$. More generally, one can prove the following proposition.

Proposition 4. *A finite maximal code $X \subseteq A^*$ is commutatively equivalent to a prefix code if and only if the quotient $T = (\underline{\underline{X}} - 1)/(\underline{\underline{A}} - 1)$ is a polynomial with coefficients in $\mathbb{N}$.*

Example 4 (continued). For $X = \{aa, ba, baa, bba, bb\}$, T is the polynomial $1 + a + b + ab$.

A first step in this direction is proved in [46]: if $X \subseteq A^*$ is a finite maximal code, then the quotient T is a polynomial with integer coefficients; moreover if T is irreducible, then X is prefix or suffix.

Two interesting results shed different light on Conjecture 1. The first one states that any code with bounded synchronization delay (see Sect. 2.1) is commutatively equivalent to a prefix code [39]. The second one is due to P. Shor [48]:

Theorem 5. *The code*

$$X_S = b\{1, a, a^7, a^{13}, a^{14}\} \cup \{a^3, a^8\}b\{1, a^2, a^4, a^6\} \cup a^{11}b\{1, a, a^2\} \tag{3}$$

is not commutatively equivalent to a prefix code.

Of course, code (3) is not maximal. It is an open problem whether it can be embedded into a finite maximal code (see Sect. 4). If yes, Conjecture 1 would be false.

Up to now, Conjecture 1 is widely open. It has stimulated other open problems that we discuss in the next two sections: the problem of completion of finite codes and the factorization conjecture.

4 Finite Codes

4.1 A Decision Problem for Finite Codes

Another main open problem in the theory of variable-length codes deals with the completion of finite codes (see Sect. 7 for the general problem of completion). A code X is said to have a *finite completion* if it can be embedded into a finite complete code.

Problem 2. Is it decidable whether a finite code has a finite completion?

As mentioned in Sect. 3, if it exists, such a decision procedure could be applied to Shor's code X_S (see (3)). In the case of a positive answer, Conjecture 1 would be false. There exists another code for which we do not known whether it has a finite completion. It is the 4-word code [41]

$$X_4 = \{a^5b, a^2b, ba, b\} \ . \tag{4}$$

Some classes of finite codes always have a finite completion. For instance, a finite prefix (resp. suffix) code X can be embedded into a finite complete prefix (resp. suffix) code. Just add to the tree associated with X the lacking leaves (see Sect. 2.1). More generally, any finite prefix-suffix composed code has a finite completion: any prefix or suffix code used in the composition is replaced by a complete prefix or suffix code (see Sect. 2.3).

Example 5. The code $X = \{aa, ba\} = \{\alpha\alpha, \beta\} \circ \{a, ba\}$ has the finite completion $\{\alpha\alpha, \beta, \beta\alpha, \gamma\alpha, \gamma\} \circ \{a, ba, bb\}$.

However it is known [33, 40] that some finite codes have *no* finite completion, such as the small code

$$\{a^5, a^2b, ba, b\} \ . \tag{5}$$

Notice that this code and code X_4 above only differ by one word.

Of course, code (5) is not prefix-suffix composed, otherwise it would have a finite completion. However each code with 1 or 2 words has a finite completion because it is always prefix-suffix composed [41]. To fill the gap between 2-word codes and the 4-word code (5), the authors of [41] have conjectured that any code with 3 elements was prefix-suffix composed. D. Derencourt [15] has recently found a family of counterexamples whose smallest one is the not prefix-suffix composed code

$$\{b, bab, abab^2a\} \ .$$

However, the following problem remains open, since all these counterexamples have a finite completion [16].

Problem 3. Is there a 3-word code which has no finite completion?

4.2 Codes with no Finite Completion

There exists a simple method to construct finite codes with no finite completion, which is related to factorizations of cyclic groups.

Let $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$ be the cyclic group of order $n \geq 1$. We say [23] that (R, S), with R, S two finite *subsets* of $\mathbb{N}$, is a *factorization* of $\mathbb{Z}_n$ if and only if each element of $\mathbb{Z}_n$ can be expressed uniquely as the sum, modulo n, of an element of R and an element of S:

$$\mathbb{Z}_n = R \oplus S \bmod n \ .$$

More generally [31], given $m \geq 1$, a pair (R, S) of finite *multisets* of $\mathbb{N}$ is called a *m-factorization* of $\mathbb{Z}_n$ if

$$\underbrace{\mathbb{Z}_n \cup \ldots \cup \mathbb{Z}_n}_{m \text{ times}} = R \oplus S \bmod n \ .$$

Example 6. If n is a prime number, then $\mathbb{Z}_n$ only admits the trivial factorization (R, S) (up to a translation) with $R = \{0\}$ and $R = \mathbb{Z}_n$ (since $|R| \cdot |S| = n$). The cyclic group $\mathbb{Z}_6$ admits several factorizations, such as $(\{0,1\}, \{0,2,4\})$ or $(\{0,5\}, \{0,2,4\})$. It also admits the 2-factorization $(\{0,0,1,5\}, \{0,2,4\})$ since

$$\{0,0,1,5\} \oplus \{0,2,4\} = \{0,0,1,1,2,2,3,3,4,4,5,5\} \bmod 6 \ .$$

The next proposition shows how m-factorizations of cyclic groups appear in the framework of finite complete codes. It uses the property that in a finite complete code X over A, X contains a unique power of each letter $a \in A$.

Proposition 6. *Let X be a finite complete code over the binary alphabet $\{a, b\}$. Let $n, m \geq 1$ such that $a^n, b^m \in X$. Define the multisets*

$$\begin{aligned} R_X &= \{p \in \mathbb{N} \mid a^p b^i \in X, \ \text{with } i \geq 1\} \ , \\ S_X &= \{q \in \mathbb{N} \mid b^i a^q \in X, \ \text{with } i \geq 1\} \ . \end{aligned} \tag{6}$$

Then (R_X, S_X) is a m-factorization of $\mathbb{Z}_n$.

Example 7. For $X = \{a^6, aba^4, ba^4, aba^2, ba^2, a^5b, aba^3b, ba^3b, abab, bab, b\}$, we obtain the factorization $(\{0,5\}, \{0,2,4\})$ of $\mathbb{Z}_6$. For the code $\{aa, ba, baa, bba, bb\}$ of Example 4, one gets the 2-factorization $(\{0\}, \{0,1,1,2\})$ of $\mathbb{Z}_2$.

Proof. We just sketch the proof of Proposition 6 when $m = 1$ [41]. The general case is solved in [31]. The sum $R \oplus S$ is unambiguous, otherwise using the hypothesis that $b \in X$, one can find a word which ambiguously decomposes over the code X. To show that $\mathbb{Z}_n$ is contained in $R \oplus S \bmod n$, we use the completeness of the code: any word $b^{i'} a^m b^{j'}$ is factor of some word of X^*. If i', j', m are chosen large enough, as the code is finite, we obtain a decomposition $b^i a^q (a^n)^t a^p b^j$ with $m = p + q \bmod n$ and $i \leq i'$, $j \leq j'$ (see Fig. 4). □

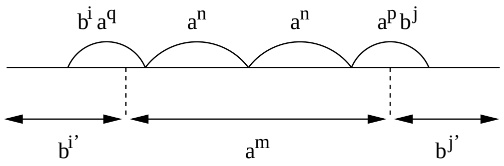

Fig. 4. Decomposition over X.

We are now ready to describe an infinite family of finite codes which cannot be embedded into a finite complete code [40]. The smallest example is the code $\{a^5, a^2b, ba, b\}$ given above (see (5)).

Theorem 7. *Let* $X = \{a^n\} \cup a^{R_X} b \cup ba^{S_X}$ *be a finite code, where sets* R_X, S_X *both contain 0 (that is* $b \in X$*). If* $|R_X| \geq 2$, $|S_X| \geq 2$ *and if* n *is a prime number, then* X *has no finite completion.*

Proof. Assume that X can be embedded into a finite complete code Y. By Proposition 6, the pair (R_X, S_X) is included into the factorization (R_Y, S_Y) of the cyclic group $\mathbb{Z}_n$. But n is prime, and neither R_Y nor S_Y is equal to $\{0\}$. $\square$

Codes of Theorem 7 are all subsets of $a^* \cup a^* b \cup ba^*$ containing both a^n and b. A collection of results show that inside the class of codes $X \subseteq a^* \cup a^* b \cup ba^*$ such that $a^n, b \in X$, there exist codes with no finite completion if and only if $n \neq 1, 2, 3, 4, 6$ [18, 19, 30, 40, 41]. Very recently, N.H. Lam [31] has used m-factorizations to get examples of codes with no finite completion, containing b^m instead of b. Among these examples, one finds the code $\{a^8, b^4, ba^8, b^4a^3, ba^3, a^8b, ab^4, ab, a^2b^4, a^2b\}$.

Methods to construct codes with no finite completion all use properties of m-factorizations of cyclic groups. We do not know other methods. Concerning Problem 3, we think that m-factorizations are not helpful. Indeed, a pair (R_X, S_X) associated with a code X gives information only if $|R_X| \geq 2$ and $|S_X| \geq 2$. These lower bounds are not achieved by 3-word codes.

4.3 Codes with Unknown Finite Completion

We now come back to finite codes of Sect. 4.1, namely Shor's code X_S and code $X_4 = \{a^5b, a^2b, ba, b\}$ (see (3) (4)) .

The latter code is very close to the code $\{a^5, a^2b, ba, b\}$ having no finite completion (see (5)). The only difference is about word a^5b replacing a^5, in a way that hypotheses of Theorem 7 are no longer satisfied. In other words, Theorem 7 is helpful only if one knows the order n of the cyclic group, i.e., the exponent n such that a^n belongs to the code.

People think (or hope) that codes X_S, X_4 have no finite completion. In this spirit, the next conjecture is formulated in [41] in relation with results of Sect. 4.2.

Conjecture 4. Let $p, q \geq 2$ such that $\gcd(p,q) = 1$. Then the pair (R, S) with $R = \{0, p, q\}$ and $S = \{0, 1\}$ is not embeddable into a factorization of a cyclic group.

Example 8. The pair (R_X, S_X) (see (6)) associated with Shor's code X_S is equal to $(\{0,3,8,11\}, \{0,1,7,13,14\})$. If Conjecture 4 is true, then X_S cannot be embedded into a finite complete code Y. Otherwise $\{0,3,8\} \subseteq R_Y$, $\{0,1\} \subseteq S_Y$ where (R_Y, S_Y) is the factorization associated with Y by Proposition 6. A similar argument holds for the pair $(\{0,2,5\}, \{0,1\})$ of the code X_4.

5 Factorization Conjecture

5.1 Factorizing Codes

A finite code $X \subseteq A^*$ is called *factorizing* if there exist two finite sets $P, Q \subseteq A^*$ such that

$$\underline{A}^* = \underline{Q}\underline{X}^*\underline{P} \ , \tag{7}$$

i.e., any word over A uniquely factorizes as qxp, with $q \in Q$, $x \in X^*$ and $p \in P$. Equivalently, X is factorizing if and only if

$$\underline{X} - 1 = \underline{P}(\underline{A} - 1)\underline{Q} \ . \tag{8}$$

Notice that (7) implies that the code X is complete. Definition (8) shows that if X is factorizing, then X is commutatively equivalent to a prefix code (see Proposition 4). So Conjecture 1 is a particular case of the following conjecture.

Conjecture 5. Any finite maximal code is factorizing.

Again, finite maximal codes which are prefix-suffix composed, are factorizing. Indeed, any finite maximal prefix code $X \subseteq A^*$ is factorizing, since $\underline{X} - 1 = \underline{P}(\underline{A} - 1)$ with P the set of proper prefixes of words of X. Symmetrically for suffix codes, we get $\underline{X} - 1 = (\underline{A} - 1)\underline{Q}$. Finally, a composition of factorizing codes is still a factorizing code.

Example 4 (continued). Let $X = Y \circ Z$ such that $X = \{aa, ba, baa, bba, bb\}$, $Y = \{\alpha\alpha, \beta, \beta\alpha, \gamma\alpha, \gamma\}$ and $Z = \{a, ba, bb\}$. Then $\underline{Y} - 1 = (\alpha + \beta + \gamma - 1)(1 + \alpha)$, $\underline{Z} - 1 = (1 + b)(a + b - 1)$ and $\underline{X} - 1 = (1 + b)(a + b - 1)(1 + a)$.

Definition (7) is "dual" to the notion of complete code, since a code X is complete if any word over A is factor of a word of X^* (see Sect. 2.1). The next results make this relationship more precise [4, p. 136][42].

Theorem 8. *Let $X \subseteq A^*$ be a regular maximal code. Then, there exist an integer $d \geq 1$ and $d + 1$ words y, $z_1, \ldots, z_d \in A^*$ such that*

$$\forall w \in A^*, \exists! i \in \{1, \ldots, d\},\ ywz_i \in X^* \ .$$

The integer d in Theorem 8 is unique; it is the *degree* of the code X. Codes with degree 1 are called *synchronous*. See [12] for interesting properties on unambiguous finite automata $\mathcal{A}$ where code $X_{\mathcal{A}}$ is synchronous.

Corollary 9. *Let $X \subseteq A^*$ be a finite maximal code. Then, there exist finite subsets P, Q, R of A^* such that*

$$\underline{A}^* = \underline{Q}\underline{X}^*\underline{P} + \underline{R} \ .$$

Proof. Apply Theorem 8 to each word w of A^*. Either the decomposition over X of the word ywz_i "cuts" w, or w is strictly inside a word $x \in X$ of the decomposition. All words w verifying the second case are put in a set R. This set is finite because X is finite. In the first case, w is cut and factorized as

$$w = qxp$$

with q (resp. p) a proper suffix (resp. prefix) of some word of X, and x a word of X^* (see Fig. 5). All such possible words q define set Q, similarly with words p and set P. Finally, any $w \in A^* \setminus R$ uniquely factorizes as qxp with $q \in Q$, $x \in X^*$ and $p \in P$ because of the unicity of word z_i in Theorem 8. $\square$

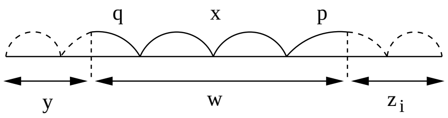

Fig. 5. Decomposition of ywz_i over X.

So, one has to erase set R in Corollary 9 to prove Conjecture 5. In [7], another interesting property is proved for finite maximal codes with degree 1: there exist finite sets P, Q, R such that

$$\underline{A}^* = \underline{Q}\underline{X}^*\underline{P} - \underline{R} \ .$$

Example 9. Code $X = \{aa, ba, baa, bba, bb\}$ of Example 4 is synchronous, because

$$bwaa \in X^* \tag{9}$$

for all $w \in A^*$. A simple method to verify (9) is to make deterministic the automaton $\mathcal{A}$ of Fig. 3 by beginning with state 1. The new automaton is depicted on Fig. 6. We see that feeding b at state 1 leads to the strongly connected component $\{(3), (1, 2)\}$, and that inside it all paths labeled by aa lead to $(1, 2)$ which contains the particular state 1. Hence, for any word w, there is a path $(1, bwaa, 1)$ in $\mathcal{A}$, which means that $bwaa \in X^*$.

Now, as in the proof of Corollary 9, we get $\underline{A}^* = \underline{Q}\underline{X}^*\underline{P} + \underline{R}$ with $Q = \{a, aa, b, ba\}$, $P = \{1, b\}$ and $R = \{1\}$.

Notice that we have also $bbwaa \in X^*$ for all $w \in A^*$. This choice is better because R becomes empty and thus $\underline{A}^* = \{1, a\}\underline{X}^*\{1, b\}$.

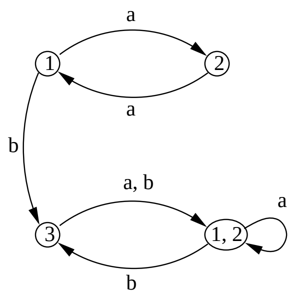

Fig. 6. Determinization of $\mathcal{A}$.

5.2 Reutenauer's Theorem

Up to now, the best result approaching Conjecture 5 is due to C. Reutenauer [4, 42]. It uses polynomials of $\mathbb{Z}[A]$ with coefficients in $\mathbb{Z}$ and non commutative variables in A.

Theorem 10. *Let $X \subseteq A^*$ be a finite maximal code. Then there exist polynomials $P, Q \in \mathbb{Z}[A]$ such that*

$$\underline{X} - 1 = P(\underline{A} - 1)Q \ . \tag{10}$$

More precisely, if d is the degree of X, there exist polynomials $P, Q, R \in \mathbb{Z}[A]$ such that

$$\underline{X} - 1 = P\left(d(\underline{A} - 1) + (\underline{A} - 1)R(\underline{A} - 1)\right)Q \ .$$

Therefore, to get Conjecture 5, one has to prove that polynomials P and Q can be chosen with coefficients $0, 1$ in (10). This idea has been developed for codes over $A = \{a, b\}$ having a bounded number of b inside their words [17, 20, 40]:

Theorem 11. *Let $X \subseteq \{a, b\}^*$ be a finite maximal code with at most 3 occurrences of letter b inside its elements. Then, any polynomials P, Q as in (10) have necessarily coefficients $0, 1$. In particular, X is a factorizing code.*

Proof. We only give the proof for codes with at most one b in each word. The general case is rather elaborated. By (10), $\underline{X} - 1 = PaQ + PbQ - PQ$. By hypothesis, P and Q are polynomials in $\mathbb{Z}[a]$ instead of $\mathbb{Z}[a, b]$. Hence PbQ is the only term dealing with words of X containing letter b, and P, Q only have coefficients $0, 1$. $\square$

Theorem 11 cannot be extended. For example, the maximal code $\{b^6\}$ verifies (10) with $P = 1 - b + b^2$, $Q = 1 + 2b + 2b^2 + b^3$ [20]. However, it is factorizing since other polynomials exist which have coefficients $0, 1$ only: $P = 1$ and $Q = 1 + b + b^2 + b^3 + b^4 + b^5$.

As observed in Sect. 4.2, finite maximal codes and factorizations of cyclic groups cannot be separated. Factorizations of cyclic groups also appear in the context of codes X described in Theorem 11:

• If words in X have at most one b:

$$\underline{X} - 1 = a^I(a + b - 1)a^J \tag{11}$$

where (I, J) is a factorization of $\mathbb{Z}_n$ (assuming that $a^n \in X$). This factorization is particular since $I \oplus J = \mathbb{Z}_n$ *without* mod n (see Sect. 4.2). Indeed, (11) restricted to a gives the relation $a^n - 1 = a^I(a-1)a^J$. Such factorizations are called *Krasner's factorizations*. They are all constructed as follows. Given a decomposition

$$\frac{a^n - 1}{a - 1} = \frac{a^n - 1}{a^{n_1} - 1}\frac{a^{n_1} - 1}{a^{n_2} - 1}\cdots\frac{a^{n_k} - 1}{a - 1} \ ,$$

separate factors $(a^{n_l}-1)/(a^{n_{l-1}}-1)$ into two groups, in a way to get $\frac{a^n-1}{a-1} = a^I a^J$ [29].

• If words in X have at most two b:

$$\underline{X} - 1 = a^I(a + b - 1)(a^J + \sum_{j \in J} a^{M_j} b a^j)$$

where (I, J) is a Krasner's factorization of $\mathbb{Z}_n$ and for all $j \in J$, (I, J_j) is a factorization of $\mathbb{Z}_n$ such that

$$a^{J_j} = a^J(1 + a^{M_j}(a - 1)) \ . \tag{12}$$

We will come back on these factorizations in Sect. 6.

5.3 Separating Boxes

Another attempt in solving Conjecture 5 uses combinatorics on words instead of polynomials. The starting point is Theorem 8 and its Corollary 9. In [6], the concept of *separating* synchronous code X is introduced, i.e., there exist $y, z \in A^*$ such that any $w \in A^*$ admits a decomposition

$$w = w'w'' \ \text{ such that } \ yw', w''z \in X^* \ . \tag{13}$$

Looking at the proof of Corollary 9, one sees that set R is empty. It follows that any finite maximal code which is separating synchronous is factorizing (see end of Example 9). In [6], J.-M. Boë asked whether all synchronous codes were separating. Y. Césari [2, p. 419] gave a negative answer (see Fig. 7). This remarkable finite maximal code X over $A = \{a, b\}$ is a synchronous code such that $X^* = Y^* \cap Z^*$ where $Y = (A^2 \setminus \{b^2\}) \cup b^2A$ is a maximal prefix code and $Z = A^2a \cup \{b\}$ is a maximal suffix code (see [9, 28] for codes constructed in this way). Nevertheless, Césari's code is factorizing because

$$\underline{X} - 1 = (1 + \underline{A} + b^2)(\underline{A} - 1)(1 + ab + \underline{A}a + a\underline{A}^2a) \ . \tag{14}$$

Recently, L. Zhang [52] has proposed the notions of box and separating box for codes. A code $X \subseteq A^*$ has a *box* (U, V), with U, V two finite subsets of A^*, if for any $w \in A^*$, there exist a unique $u \in U$ and a unique $v \in V$ such that

$$uwv \in X^* \ .$$

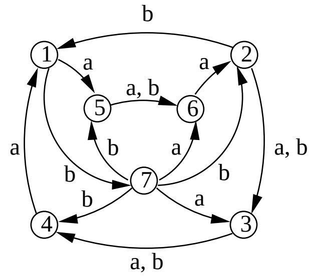

Fig. 7. A synchronous code which is not separating.

This box is *separating*, if

$$uwv \in X^* \text{ with } w = w'w'' \text{ such that } uw', w''v \in X^* .$$

By Theorem 8, any regular maximal code has a box $(\{y\}, \{z_1 \ldots z_d\})$. Separating synchronous codes all have a separating box $(\{y\}, \{z\})$ (see (13)). As for separating synchronous codes, one can easily verify that finite maximal codes having a separating box are factorizing.

An interesting fact is that Césari's code has a separating box $(\{b^3\}, \{1, a^5\})$ leading to (14). Such a box can be read in the automaton of Fig. 7, in the manner briefly outlined in Example 9. Another separating box is $(\{1, a^4, a^4b\}, \{1, aba\})$. There is also a box which is not separating like $(\{a, a^2, a^3\}, \{1, a^5\})$.

Hence, an open question is the following.

Problem 6. Has any finite maximal code a box which is separating?

A first step in this direction appears in [53] for finite maximal code $X \subseteq \{a, b\}^*$ such that $b^n \in X$ with $n = 4$ or n a prime number: if the number of b's inside words of X is bounded by n, then X is factorizing (see also Theorem 11). The proof is simple and a separating box is constructed thanks to factorizations of cyclic groups.

6 Factorization of Cyclic Groups

In Sect. 4 and 5, several results have pointed out connections between finite maximal codes and factorizations of cyclic groups. The general structure of factorizations of cyclic groups is unknown, despite a lot of efforts. This open problem has consequences on the conjectures mentioned before for codes. We briefly explain them (for more details and further references, see the survey [18]).

G. Hajós was the first to introduce the notion of factorization of a cyclic group, by solving a conjecture proposed by H. Minkowski [23] (see Sect. 4.2). He has conjectured that any factorization (R, S) of a cyclic group $\mathbb{Z}_n$ was *periodic*, that is, $R + m = R \bmod n$ for some $m \in \mathbb{Z}_n \setminus \{0\}$. The conjecture was false, the smallest counter-example being the non periodic factorization [14] $(\{0,8,16,18,26,34\},\{0,1,5,6,12,25,29,36,42,48,49,53\})$ of $\mathbb{Z}_{72}$.

Cyclic groups have been classified into two classes,– the class of *good groups* for which *all* factorizations are periodic, and the class of *bad groups*. As a consequence of several papers, cyclic good groups are $\mathbb{Z}_{p^n q}$, $\mathbb{Z}_{p^2q^2}$, $\mathbb{Z}_{p^2qr}$, $\mathbb{Z}_{pqrs}$ and their subgroups, where p,q,r,s are distinct prime numbers [23].

G. Hajós [24] has found an interesting class of factorizations, whose construction is based on Krasner's factorizations (see Sect. 5.2). In [18], it is proved that they are exactly constructed as in (12): (R,S) is a *Hajós' factorization* of $\mathbb{Z}_n$ if and only if there exist a Krasner's factorization (I,J) of $\mathbb{Z}_n$ and two finite sets $M,N \subset \mathbb{N}$ such that

$$a^R = a^I(1+a^M(a-1)) \quad a^S = a^J(1+a^N(a-1)) \ .$$

It is unknown whether other factorizations than Hajós' ones appear in the structure of factorizing codes (see (11) (12)).

Hajós' factorizations is a subclass of the periodic factorizations, but it contains all the factorizations of the good groups. They have the following nice property [18].

Proposition 12. *If (R,S) is a Hajós' factorization of $\mathbb{Z}_n$, with R,S both containing 0, then there exists an integer $k \geq 2$, $k|n$, such that $R = 0 \bmod k$ or $S = 0 \bmod k$.*

It is an open question whether this property holds for every factorization [44]. A positive answer to this question would solve Conjecture 4.

Proposition 12 implies some consequences on Shor's code X_S and code $X_4 = \{a^5b, a^2b, ba, b\}$ (see (3) (4)), whose finite completion is an open problem (see Sect. 4.3). Sets $R = \{0,p,k\}$ and $S = \{0,1\}$ of Conjecture 4 cannot be embedded into a Hajós' factorization. Hence, codes X_S and X_4 are contained in no finite maximal code such that $a^n \in X$, with $\mathbb{Z}_n$ a good group. In [18], it is also proved that X_S and X_4 are contained in no finite maximal factorizing code.

As $\mathbb{Z}_{72}$ is the smallest bad group, if X_S or X_4 have a finite completion, one must add to them the word a^n with n at least equal to 72. This lower bound can be increased thanks to the following result [50].

Proposition 13. *Let $m,n \in \mathbb{N}$ be such that $\gcd(m,n) = 1$. If (R,S) is a factorization of $\mathbb{Z}_n$, then (R,mS) is also a factorization of $\mathbb{Z}_n$.*

Assume that the pair $(\{0,2,5\},\{0,1\})$ associated with code X_4 is contained in a factorization (R,S) of $\mathbb{Z}_n$. If $\gcd(2,n) = 1$, then $(R,2S)$ is still a factorization, which is impossible since $0+2 = 2+0$ in $R+2S$. Thus $2|n$. In the same way, one proves that 3, 5 and then 30 divides n. For the pair $(\{0,3,8,11\},\{0,1,7,13,14\})$ associated with Shor's code, similar computations lead to $330|n$.

7 Completion of Codes

7.1 Embedding into a Complete Code

In Sect. 4, the problem of embedding a finite code into a complete one has been investigated. This question is part of a more general program.

Problem 7. Given a code $X \subseteq A^*$ with Property $\mathcal{P}$, is there a procedure to embed X into a complete code $Y \subseteq A^*$ sharing the same property?

The answer is simple for codes whose Property $\mathcal{P}$ is to be finite prefix, or regular prefix. In both cases, the answer is "yes". The finite case has already been discussed in Sect. 4.1: a complete finite prefix code Y is obtained by adding the lacking leaves to the tree associated with X. The regular case is solved similarly. Given a deterministic finite automaton $\mathcal{A} = (S, 1, T)$ associated with X (see Sect. 2.2), for any state s and letter a such that no transition (s, a, r) exists, add a new transition $(s, a, 1)$. The automaton remains deterministic, is now complete, and the related code Y is a regular complete prefix code containing X.

Embedding procedures are similarly defined for codes which are finite suffix, regular suffix, finite prefix-suffix composed, regular prefix-suffix composed.

When Property $\mathcal{P}$ is being a finite code, we recall that some codes has no finite completion (see Sect. 4.2). The smallest known example is the 4-word code $\{a^5, a^2b, ba, b\}$. However, for the class of regular codes, any regular code can be embedded into a complete one, and the embedding procedure is simple [21].

Theorem 14. *Any regular code can be effectively included in a regular code which is complete.*

Recently, the statement of this theorem has been reinforced: any regular code X can be realized as $X^* = Y^* \cap Z^*$ with Y, Z two regular maximal codes [9].

Problem 7 has made important progress for two families of codes, the bifix codes and the codes with bounded decoding delay. However, this problem is open for codes with bounded synchronization delay (see Sect. 2.1).

Problem 8. Is any regular code with bounded synchronization delay included in a maximal one?

7.2 Codes with Bounded Decoding Delay

Codes with bounded decoding delay naturally generalize prefix codes which have delay 0 (see Sect. 2.1). A central result due to M.-P. Schüzenberger [47] states that a finite complete code has either delay 0 or an infinite delay.

Theorem 15. *Any finite complete code with bounded decoding delay is prefix.*

In other words, there is no hope to embed a finite code with bounded decoding delay into a complete one, except if the code is prefix. However, if the code is regular instead of finite, such a procedure exists which keeps the same decoding delay [10]. The procedure of Theorem 14 is not suitable because the constructed complete code Y has always an infinite delay.

Theorem 16. *Any regular code with decoding delay d can be effectively included in a regular complete code with the same decoding delay d.*

Original proofs of both theorems use combinatorics on words; they are long and complex. Automata lead to simpler proofs, we want to sketch them.

In the same way deterministic automata correspond to prefix codes (Proposition 2), automata with bounded delay exactly describe codes with bounded decoding delay. Let $d \geq 0$, an automaton has a *bounded delay d* if [1, 8]

$$\begin{matrix} (r, a, s_1) \in T, \ (s_1, w, t_1) \in T^* \\ (r, a, s_2) \in T, \ (s_2, w, t_2) \in T^* \end{matrix} \ \text{with} \ |w| = d \quad \Rightarrow \quad s_1 = s_2 \ . \tag{15}$$

When $d = 0$, this is the definition of a deterministic automaton. The next proposition is proved in [8] for finite codes only.

Proposition 17. *A regular code X has a bounded decoding delay d if and only if $X = X_{\mathcal{A}}$ for some finite automaton $\mathcal{A}$ with bounded delay d.*

Proof. If $\mathcal{A}$ has bounded delay d, then it is not difficult to see that the code $X_{\mathcal{A}}$ has bounded decoding delay d. For the converse, we have to construct a particular automaton $\mathcal{A} = (S, 1, T)$ called *literal*. It is a "nearly deterministic" automaton in the sense that

$$(r, a, s_1), (r, a, s_2) \in T, \ s_1 \neq s_2 \quad \Rightarrow \quad s_1 = 1 \ \text{ or } \ s_2 = 1 \ .$$

Such an automaton always exists, with $S = \{w^{-1}X \mid w \in A^*\} \setminus \{\emptyset, \{1\}\}$ and state 1 equal to X. Transition (r, a, s) is in T if $a^{-1}r = s$ or if $a \in r$ and $s = 1$. Notice that this automaton is trim. Now, assume that X has decoding delay d. With the notations of (15), either s_1 or s_2 equals 1 as the automaton is literal. Suppose $s_1 = 1$. As $\mathcal{A}$ is trim, there exists some paths $(1, u, r)$ and $(t_2, v, 1)$. Reading the two paths of Fig. 8, we get (1) with $x = ua$ showing that $s_2 = 1 = s_1$. □

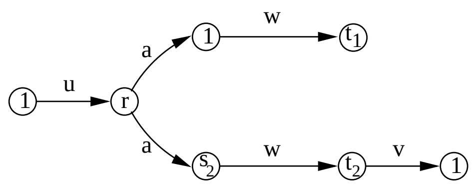

Fig. 8. Two paths with $|w| = d$.

Now, one can prove [8] that given an automaton $\mathcal{A}$ with delay d and a code $X = X_{\mathcal{A}}$ with bounded decoding delay d, X is complete if and only if $\mathcal{A}$ is d-complete. An automaton $\mathcal{A}$ is *d-complete* if

$$(r, w, s) \in T^* \ \text{ with } \ |w| = d \quad \Rightarrow \quad \forall a \in A, \exists t \in S, \ (r, wa, t) \in T^* \ .$$

When $d = 0$, a d-complete automaton with delay d is exactly a complete deterministic automaton.

A simple way to prove Theorem 15 is proposed in [8]. Let $\mathcal{A}$ be a finite d-complete automaton with delay d associated with the finite complete code X with bounded decoding delay d. As X is finite, one proves that all pairs of states s_1, s_2 such that $(r, a, s_1), (r, a, s_2) \in T$ can be merged without modifying X. The resulting automaton is deterministic showing that X is prefix.

In [1], it is shown how to make d-complete an automaton with delay d by adding it a *finite* number of new states and new transitions. This procedure leads to a simple proof of Theorem 16. It works as follows.

Example 10. Let $X = \{a, aab\}$ be a code with decoding delay 2. A literal automaton $\mathcal{A}$ for X is depicted on Fig. 9 (see the proof of Proposition 17). In a first step, add a "tree" of height $d = 2$ whose root is state 1 (see Fig. 10). In a second step, if $(r, w, s) \in T^*$ but $(r, wa, t) \notin T^*$ with $|w| = 2$, $a \in A$ (showing that $\mathcal{A}$ is not 2-complete), add a transition (r, b, q_u) such that $wa = bu$. This procedure keeps the delay 2 of the automaton, and makes it 2-complete. By Proposition 17, this new automaton leads to a regular code Y with delay 2 which is complete and contains X.

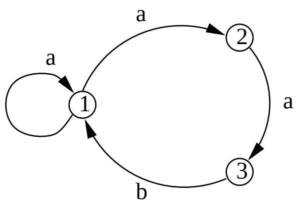

Fig. 9. A literal automaton for the code $\{a, aab\}$.

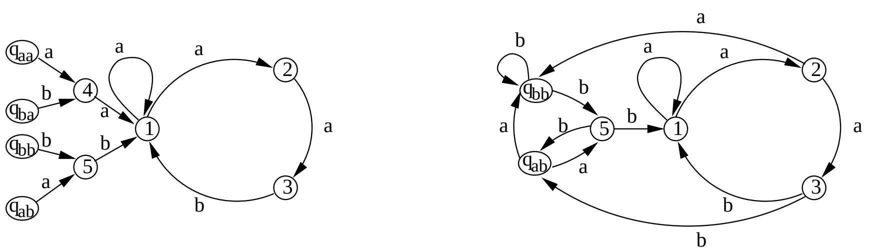

Fig. 10. The two steps for completing $\{a, aab\}$.

In [1], one also finds a nice procedure which embeds a finite automaton with bounded bi-delay (d, d') into a (d, d')-complete one. By bi-delay (d, d'), we mean a (left) delay d as before and a right delay d'. See also [36] for the completion of deterministic local automata.

7.3 Bifix Codes

Bifix codes are particular prefix codes, since they are also suffix. Though the embedding of a prefix code into a complete one is easy, this is more complicated for bifix codes due to the two-sided condition defining them. For example, contrarily to prefix codes, there exist finite bifix codes not embeddable into a finite complete one.

Proposition 18. *The bifix code $X = \{a^n, b^m\}$, with $n \neq m$, is included in no finite complete bifix code.*

Proof. Assume that $n < m$ and that X is included in a finite complete bifix code Y. Thus, as a complete prefix code, Y satisfies (see Sect. 2.1)

$$\forall w \in A^*,\ wA^* \cap Y^* \neq \emptyset\ .$$

Hence for all i, $1 \leq i < m$, there exists $k_i \geq 1$ such that $b^i a^{k_i} \in Y$ (remember that Y is finite). The k_i's are pairwise distinct since Y is suffix. But $a^n \in Y$ with $n < m$, and a^n is a proper suffix of some $b^i a^{k_i}$. □

However, every regular bifix code is included in a complete one. The proof has been found recently [54]. It generalizes in an elegant way the proof proposed in [37] for finite bifix codes.

Theorem 19. *Any regular bifix code can be effectively included in a regular complete bifix code.*

Proof. We outline the construction in the finite case only. Given a bifix code X, its *indicator* L_X is defined on A^* by

$$L_X(w) = card\{(u, x, v) \mid w = uxv, u \in A^* \setminus A^*X, x \in X^*, v \in A^* \setminus XA^*\}\ .$$

The indicator is increasing

$$L_X(w) \leq L_X(wa) \leq L_X(w) + 1$$

and if X is not complete, L_X is bounded on no subset uA^*, $u \in A^*$.

Let X be a *finite* bifix code. Then its indicator is bounded on X; assume it is by n. The completion procedure runs as follows. Construct $Z = \{w \in A^* \mid L_X(w) = n + 1\}$. Then $Y = Pref(X \cup Z)$ is a regular complete bifix code, where $Pref(T) = T \setminus TA^+$. Indeed, Y is prefix by construction, Y is suffix and complete due to the increasing property of L_X mentioned above, and Y is regular because one can prove that Z is regular.

In the case X is a *regular* bifix code, L_X is no longer bounded on X and the previous method fails. However, it can be applied "locally" to get a regular bifix code Y which is not complete but strictly contains X. The construction is thus repeated on Y and after a finite number of steps, one gets a regular bifix code containing X, which is now complete. □

Example 11. The bifix code $X = \{a, b^2\}$ cannot be embedded into a finite complete one by Proposition 18. Its completion into a regular bifix code is depicted on Fig. 11 with $n = 2$ (value of $L_X(w)$ is indicated inside the node corresponding to w). This leads to the complete code $\{a\} \cup ba^*b$.

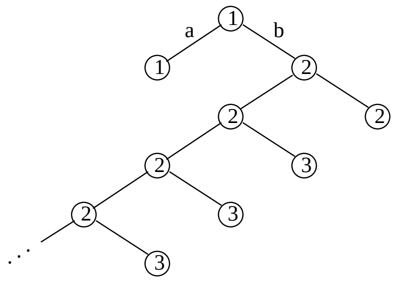

Fig. 11. Completion of the bifix code $\{a, b^2\}$.

Notice that the completion of automata with bounded bi-delay mentioned at the end of Sect. 7.2 does not lead to another proof of Theorem 19. Indeed, for some regular bifix codes X, there exists no finite automaton $\mathcal{A}$ with bi-delay $(0, 0)$ such that $X = X_{\mathcal{A}}$. It is then natural to try to generalize Theorem 19 to the class of regular codes with bounded decoding bi-delay.

Problem 9. Is any regular code with decoding bi-delay (d, d') included in a regular complete code with the same decoding bi-delay?

Problem 2 is solved for bifix codes, since it is decidable whether a finite bifix code has a finite completion into a bifix code [3]. However the decision procedure does not give a real characterization of the subsets of finite complete bifix codes.

8 Conclusion

In this survey, we have focused on finite and regular maximal codes. They play a central role in the theory of variable-length codes. We have emphasized the richness and the variety of tools introduced to study the difficult conjectures on maximal codes. The important developments obtained during the last ten years seem promising to solve the still open problems.

Acknowledgments

We thank J. Berstel, D. Perrin and A. Restivo for stimulating discussions, and we are grateful to C. De Felice for numerous comments on a preliminary version of this paper. We also wish to thank H. Jürgensen for providing helpful references.

References

1. J. Ashley, B. Marcus, D. Perrin, S. Tuncel, Surjective extensions of sliding block codes, *SIAM J. Discrete Math.* **6** (1993) 582–611.
2. J. Berstel, D. Perrin, *Theory of Codes*, Academic Press (1985).
3. J. Berstel, D. Perrin, Trends in the theory of codes, *Bull. EATCS* **29** (1986) 84–95.
4. J. Berstel, C. Reutenauer, *Rational series and their languages*, EATCS Monogr. Theoret. Comput. Sci. **12**, Springer-Verlag (1988).
5. J.-M. Boë, Une famille remarquable de codes indécomposables, *Lecture Notes in Comput. Sci.* **62** (1978) 105–112.
6. J.-M. Boë, Sur les codes synchronisants coupants, in: *Non-commutative Structures in Algebra and Geometric Combinatorics*, A. De Luca, Ed., *Quaderni de la Ricerca Scientifica* **109** (1981) 7–10.
7. J.-M. Boë, Factorisation par excès du monoïde libre, *Technical Report* **94-005**, University of Montpellier (1994) 6 pages.
8. V. Bruyère, Automata and codes with bounded deciphering delay, *Lecture Notes in Comput. Sci.* **583** (1992) 99–107.
9. V. Bruyère, D. Derencourt, M. Latteux, The meet operation in the lattice of codes *submitted* (1996) 16 pages.
10. V. Bruyère, L. Wang, L. Zhang, On completion of codes with finite deciphering delay, *European J. Combin.* **11** (1990) 513–521.
11. J. Brzozowski, Open problems about regular languages, in: *Formal Language Theory: Perspectives and Open Problems*, R.V. Book, Ed., Academic Press (1980) 23–45.
12. A. Carpi, On synchronizing unambiguous automata, *Theoret. Comput. Sci.* **60** (1988) 285–296.
13. Y. Césari, Sur l'application du théorème de Suschkevitch à l'étude des codes rationnels complets, in: *Automata, Languages and Programming*, *Lecture Notes in Comput. Sci.* (1974) 342–350.
14. N.G. de Bruijn, On the factorization of cyclic groups, *Indag. Math.* **15** (1953) 258–264.
15. D. Derencourt, A three-word code which is not prefix-suffix composed, to appear in *Theoret. Comput. Sci* (1996) 15 pages.
16. D. Derencourt, personal communication (1996).
17. C. De Felice, A partial result about the factorization conjecture for finite variable-length codes, *Discrete Math.* **122** (1993) 137–152.
18. C. De Felice, An application of Hajós factorizations to variable-length codes, to appear in *Theoret. Comput. Sci.* (1996) 31 pages.
19. C. De Felice, A. Restivo, Some results on finite maximal codes, *RAIRO Inform. Théor. Appl.* **19** (1985) 383–403.
20. C. De Felice, C. Reutenauer, Solution partielle de la conjecture de factorisation des codes *C. R. Acad. Sci. Paris* **302** (1986) 169–170.
21. A. Ehrenfeucht, G. Rozenberg, Each regular code is included in a regular maximal code, *RAIRO Inform. Théor. Appl.* **20** (1985) 89–96.
22. S. Eilenberg, *Automata, Languages and Machines*, Vol. A, Academic Press (1974).
23. L. Fuchs, *Abelian Groups*, Pergamon Press (1960).
24. G. Hajós, Sur la factorisation des groupes abéliens, *Casopis Pest. Mat. Fys.* **74** (1950) 157–162.
25. R.W. Hamming, *Coding and Information Theory*, Prentice-Hall (1986).

26. J. Hopcroft, J. Ullman, *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley (1979).
27. H. Jürgensen, S. Konstantinidis, Codes, in: *Handbook of Formal Languages*, G. Rozenberg, A. Salomaa, Eds, Springer-Verlag, to appear (1996).
28. R. König, Lectures on codes, *Technical Report* **93-3**, University of Erlangen (1993) 66 pages.
29. M. Krasner, B. Ranulac, Sur une propriété des polynômes de la division du cercle, *C. R. Acad. Sci. Paris* **240** (1937) 397–399.
30. N.H. Lam, On codes having no finite completion, *Lecture Notes in Comput. Sci.* **775** (1994) 691–698.
31. N.H. Lam, A property of finite maximal codes, *preprint* (1996) 8 pages.
32. D. Lind, B. Marcus, *Symbolic Dynamics and Coding*, Cambridge University Press (1996).
33. A.A. Markov, An example of an independent system of words which cannot be included in a finite complete system (in Russian), *Mat. Zametki* **1** (1967) 87–90.
34. R.J. McEliece, *The Theory of Information and Coding*, Enc. of Math. **3**, Addison-Wesley (1977).
35. F.J. McWilliams, N.J.A. Sloane, *The Theory of Error-Correcting Codes*, North-Holland (1977).
36. R. Montalbano, Local automata and completion, *Lecture Notes in Comput. Sci.* **665** (1993) 333–342.
37. D. Perrin, Completing biprefix codes, *Theoret. Comput. Sci.* **28** (1984) 329–336.
38. D. Perrin, Finite automata, in: *Handbook of Theoretical Computer Science*, vol. B, J. Van Leeuwen, Ed., Elsevier (1990) 2–57.
39. D. Perrin, M.-P. Schützenberger, Un problème élémentaire de la théorie de l'information, *Théorie de l'Information, Colloques Internat. CNRS* **276**, Cachan (1977) 249–260.
40. A. Restivo, On codes having no finite completions, *Discrete Math.* **17** (1977) 309–316.
41. A. Restivo, S. Salemi, T. Sportelli, Completing codes, *RAIRO Inform. Théor. Appl.* **23** (1989) 135–147.
42. C. Reutenauer, Non commutative factorization of variable length codes, *J. Pure Appl. Algebra* **36** (1985) 157–186.
43. A. Salomaa, *Jewels of Formal Language Theory*, Washington, D.C.: Computer Science Press (1981).
44. A.D. Sands, On a conjecture og G. Hajós, *Glasgow Math. J.* **15** (1974) 88–89.
45. M.-P. Schützenberger, Une théorie algébrique du codage, *Séminaire Dubreil-Pisot* 1955-56, exposé no 15 (1955).
46. M.-P. Schützenberger, Sur certains sous-monoïdes libres, *Bull. Soc. Math. France* **93** (1965) 209–223.
47. M.-P. Schützenberger, On a question concerning certain free submonoids, *J. Combin. Theory* **1** (1966) 437–442.
48. P. Shor, A counterexample to the triangle conjecture, *J. Combin. Theor. Ser. A* **38** (1983) 110–112.
49. H.J. Shyr, *Free Monoids and Languages*, Hon Min Book Company, Taichung, second ed. (1991).
50. S. Szabo, personal communication to A. Restivo (1992).
51. J.H. Van Lindt, *Introduction to Coding Theory*, Graduate Texts in Math. **86**, Springer-Verlag (1982).

52. L. Zhang, Every finite maximal code is a factorizing code, *manuscript* (1993) 19 pages.
53. L. Zhang, C.K. Gu, Two classes of factorizing codes – (p, p) codes and $(4, 4)$ codes, In: *Words, languages and combinatorics II*, M. Ito, H. Jürgensen, Eds., World Scientific (1994) 477–483.
54. L. Zhang, Z. Shen, Completion of recognizable bifix codes, *Theoret. Comput. Sci.* **145** (1995) 345–355.