



**HAL**  
open science

# Complete Variable-Length Codes: An Excursion into Word Edit Operations

Jean Néraud

► **To cite this version:**

Jean Néraud. Complete Variable-Length Codes: An Excursion into Word Edit Operations. LATA 2020, Mar 2020, Milan, Italy. hal-02389403v1

**HAL Id: hal-02389403**

**<https://hal.science/hal-02389403v1>**

Submitted on 2 Dec 2019 (v1), last revised 20 Jan 2020 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Complete Variable-Length Codes: An Excursion into Word Edit Operations

Jean NÉRAUD \*

## Abstract

Given an alphabet  $A$  and a binary relation  $\tau \subseteq A^* \times A^*$ , a language  $X \subseteq A^*$  is  $\tau$ -independent if  $\tau(X) \cap X = \emptyset$ ;  $X$  is  $\tau$ -closed if  $\tau(X) \subseteq X$ . The language  $X$  is complete if any word over  $A$  is a factor of some concatenation of words in  $X$ . Given a family of languages  $\mathcal{F}$  containing  $X$ ,  $X$  is maximal in  $\mathcal{F}$  if no other set of  $\mathcal{F}$  can strictly contain  $X$ . A language  $X \subseteq A^*$  is a variable-length code if any equation among the words of  $X$  is necessarily trivial. The study discusses the relationship between maximality and completeness in the case of  $\tau$ -independent or  $\tau$ -closed variable-length codes. We focus to the binary relations by which the images of words are computed by deleting, inserting, or substituting some characters.

**Keywords:** closed, code, complete, deletion, dependent, detection, distribution, edition, embedding, independent, insertion, Levenshtein, maximal, string, substitution, substring, subword, variable-length, word

## 1 Introduction

In formal language theory, given a property  $\mathcal{F}$ , the *embedding problem* with respect to  $\mathcal{F}$  consists in examining whether a language  $X$  satisfying  $\mathcal{F}$  can be included into some language  $\hat{X}$  that is *maximal* with respect to  $\mathcal{F}$ , in the sense that no language satisfying  $\mathcal{F}$  can strictly contain  $\hat{X}$ . In the literature, maximality is often connected to completeness: a language  $X$  over the alphabet  $A$  is *complete* if any string in the free monoid  $A^*$  (the set of the words over  $A$ ) is a factor of some word of  $X^*$  (the submonoid of all concatenations of words in  $X$ ). Such connection takes on special importance for codes: a language  $X$  over the alphabet  $A$  is a *variable-length code* (for short, a *code*) if every equation among the *words* (i.e. *strings*) of  $X$  is necessarily trivial.

A famous result due to M.P. Schützenberger states that, for the family of the so-called *thin* codes (which contains *regular* codes and therefore also finite ones),

---

\*Université de Rouen, Laboratoire d'Informatique, de Traitement de l'Information et des Systèmes, Avenue de l'Université, 76800 Saint-Étienne-du-Rouvray, France; jean.neraud@univ-rouen.fr, neraud.jean@gmail.com; <http://neraud.jean.free.fr>; orcid: 0000-0002-9630-461X

being maximal is equivalent to being complete. In connection with these two concepts lots of challenging theoretical questions have been stated. For instance, to this day the problem of the existence of a finite maximal code containing a given finite one is not known to be decidable. From this latter point of view, in [16] the author asked the question of the existence of a regular complete code containing a given finite one: a positive answer was brought in [4], where was provided a now classical formula for embedding a given regular code into some complete regular one. Famous families of codes have also been concerned by those studies: we mention *prefix* and *bifix* codes [2, Proposition 3.3.8, Proposition 6.2.1], codes with a *finite deciphering delay* [3], *infix* [10], *solid* [11], or *circular* [13].

Actually, with each of those families, a so-called *dependence system* can be associated. Formally, such a system is a family  $\mathcal{F}$  of languages constituted by those sets  $X$  that contain a non-empty finite subset in  $\mathcal{F}$ . Languages in  $\mathcal{F}$  are  $\mathcal{F}$ -*dependent*, the other ones being  $\mathcal{F}$ -*independent*. A special case corresponds to binary words relations  $\tau \subseteq A^* \times A^*$ , where a dependence system is constituted by those sets  $X$  satisfying  $\tau \cap (X \times X) \neq \emptyset$ :  $X$  is  $\tau$ -*independent* if we have  $\tau(X) \cap X = \emptyset$  (with  $\tau(X) = \{y : \exists x \in X, (x, y) \in \tau\}$ ). *Prefix codes* certainly constitute the best known example: they constitute those codes that are independent with respect to the relation obtained by removing each pair  $(x, x)$  from the famous *prefix* order. Bifix, infix or solid codes can be similarly characterized.

As regards to dependence, some extremal condition corresponds to the so-called *closed* sets: given a word relation  $\tau \subseteq A^* \times A^*$ , a language  $X$  is closed under  $\tau$  ( $\tau$ -*closed*, for short) if we have  $\tau(X) \subseteq X$ . Lots of topics are concerned by the notion. We mention the framework of prefix order where a one-to-one correspondence between independent and closed sets is provided in [2, Proposition 3.1.3] (cf. also [1, 18]). Congruences in the free monoid are also concerned [15], as well as their connections to DNA computing [7]. With respect to morphisms, involved topics are also provided by the famous *L*-systems [17] and, in the case of one-to-one (anti)-automorphisms, the so-called *invariant* sets [14].

As commented in [6], maximality and completeness concern the economy of a code. If  $X$  is a complete code then every word occurs as part of a message, hence no part of  $X^*$  is potentially useless. The present paper emphasizes the following questions: given a regular binary relation  $\tau \subseteq A^* \times A^*$ , in the family of regular  $\tau$ -independent (-closed) codes, are maximality and completeness equivalent notions? Given a non-complete regular  $\tau$ -independent (-closed) code, is it embeddable into some complete one?

Independence has some peculiar importance in the framework of coding theory. Informally, given some concatenation of words in  $X$ , each codeword  $x \in X$  is transmitted via a channel into a corresponding  $y \in A^*$ . According to the combinatorial structure of  $X$ , and the type of channel, one has to make use of codes with prescribed error-detecting constraints: some minimum-distance restraint is generally applied. In this paper, where we consider variable length codewords, we address to the Levenshtein metric [12]: given two different words  $x, y$ , their distance is the minimal total number of elementary edit operations that can transform  $x$  into  $y$ , such operation

consisting in a one character *deletion*, *insertion*, or *substitution*. Formally, it is the smallest integer  $p$  such that we have  $y \in \Lambda_p(x)$ , with  $\Lambda_p = \bigcup_{1 \leq k \leq p} (\delta_k \cup \iota_k \cup \sigma_k)^k$ , where  $\delta_k, \iota_k, \sigma_k$  are further defined below. From the point of view of error detection,  $X$  being  $\Lambda_p$ -independent guarantees that  $y \in \Lambda_p(x)$  implies  $y \neq x$ . In addition, a code satisfies the property of error correction if its elements are such that  $\Lambda_p(x) \cap \Lambda_p(y) = \emptyset$  unless  $x = y$ : according to [9, chap. 6], the existence of such codes is decidable. Denote by  $\text{Subw}(x)$  the set of the subsequences of  $x$ :

- $\delta_k$ , the *k-character deletion*, associates with every word  $x \in A^*$ , all the words  $y \in \text{Subw}(x)$  whose length is  $|x| - k$ . The *at most p-character deletion* is  $\Delta_p = \bigcup_{1 \leq k \leq p} \delta_k$ ;
- $\iota_k$ , the *k-character insertion*, is the converse relation of  $\delta_k$  and we set  $I_p = \bigcup_{1 \leq k \leq p} \iota_k$  (*at most p-character insertion*);
- $\sigma_k$ , the *k-character substitution*, associates with every  $x \in A^*$ , all  $y \in A^*$  with length  $|x|$  such that  $y_i$  (the letter of position  $i$  in  $y$ ), differs of  $x_i$  in exactly  $k$  positions  $i \in [1, |x|]$ ; we set  $\Sigma_p = \bigcup_{1 \leq k \leq p} \sigma_k$ ;
- We denote by  $\underline{\Lambda}_p$  the antireflexive relation obtained by removing all pairs  $(x, x)$  from  $\Lambda_p$  (we have  $\Lambda_1 = \underline{\Lambda}_1$ ).

For short, we will refer the preceding relations to *edit relations*. For reasons of consistency, in the whole paper we assume  $|A| \geq 2$  and  $k \geq 1$ . In what follows, we draw the main contributions of the study:

Firstly, we prove that, given a positive integer  $k$ , the two families of languages that are independent with respect to  $\delta_k$  or  $\iota_k$  are identical. In addition, for  $k \geq 2$ , no set can be  $\Lambda_k$ -independent. We establish the following result:

**Theorem A.** *Let  $A$  be a finite alphabet,  $k \geq 1$ , and  $\tau \in \{\delta_k, \iota_k, \sigma_k, \Delta_k, I_k, \Sigma_k, \underline{\Lambda}_k\}$ . Given a regular  $\tau$ -independent code  $X \subseteq A^*$ ,  $X$  is complete if, and only if, it is maximal in the family of  $\tau$ -independent codes.*

A code  $X$  is  $\underline{\Lambda}_k$ -independent if the Levenshtein distance between two distinct words of  $X$  is always larger than  $k$ : from this point of view, Theorem A states some noticeable characterization of maximal  $k$ -error detecting codes in the framework of the Levenshtein metric.

Secondly, we explore the domain of closed codes. A noticeable fact is that for any  $k$ , there are only finitely many  $\delta_k$ -closed codes and they have finite cardinality. Furthermore, one can decide whether a given non-complete  $\delta_k$ -closed code can be embedded into some complete one. We also prove that no closed code can exist with respect to the relations  $\iota_k, \Delta_k, I_k$ .

As regard to substitutions, beforehand, we focus to the structure of the set  $\sigma_k^*(w) = \bigcup_{i \in \mathbb{N}} \sigma_k^i$ . Actually, excepted for two special cases (that is,  $k = 1$  [5, 19], or  $k = 2$  with  $|A| = 2$  [8, ex. 8, p.77]), to our best knowledge, in the literature no general description is provided. In any event we provide such a description; furthermore we establish the following result:

**Theorem B.** *Let  $A$  be a finite alphabet and  $k \geq 1$ . Given a complete  $\sigma_k$ -closed code  $X \subseteq A^*$ , either every word in  $X$  has length not greater than  $k$ , or a unique integer*

$n \geq k + 1$  exists such that  $X = A^n$ . In addition for every  $\Sigma_k(\Lambda_k)$ -closed code  $X$ , some positive integer  $n$  exists such that  $X = A^n$ .

In other words, no  $\sigma_k$ -closed code can simultaneously possess words in  $A^{\leq k}$  and words in  $A^{\geq k+1}$ . As a consequence, one can decide whether a given non-complete  $\sigma_k$ -closed code  $X \subseteq A^*$  is embeddable into some complete one.

## 2 Preliminaries

We adopt the notation of the free monoid theory. Given a word  $w$ , we denote by  $|w|$  its length; for  $a \in A$ ,  $|w|_a$  denotes the number of occurrences of the letter  $a$  in  $w$ . The set of the words whose length is not greater (not smaller) than  $n$  is denoted by  $A^{\leq n}$  ( $A^{\geq n}$ ). Given  $x \in A^*$  and  $w \in A^+$ , we say that  $x$  is a *factor* of  $w$  if words  $u, v$  exist such that  $w = uxv$ ; a *subword* of  $w$  consists in any (perhaps empty) subsequence  $w_{i_1} \cdots w_{i_n}$  of  $w = w_1 \cdots w_{|w|}$ . We denote by  $F(X)$  ( $\text{Subw}(X)$ ) the set of the words that are factor (subword) of some word in  $X$  (we have  $X \subseteq F(X) \subseteq \text{Subw}(X)$ ). A pair of words  $w, w'$  is *overlapping-free* if no pair  $u, v$  exist such that  $uw' = wv$  or  $w'u = vw$ , with  $1 \leq |u| \leq |w| - 1$  and  $1 \leq |v| \leq |w'| - 1$ ; if  $w = w'$ , we say that  $w$  itself is overlapping-free.

It is assumed that the reader has a fundamental understanding with the main concepts of the theory of variable-length codes: we suggest, if necessary, that he (she) report to [2]. A set  $X$  is a *variable-length code* (a *code* for short) if for any pair of sequences of words in  $X$ , say  $(x_i)_{1 \leq i \leq n}, (y_j)_{1 \leq j \leq p}$ , the equation  $x_1 \cdots x_n = y_1 \cdots y_p$  implies  $n = p$ , and  $x_i = y_i$  for each integer  $i$  (equivalently the submonoid  $X^*$  is *free*). The two following results are famous ones from the variable length code theory:

**Theorem 2.1** Schützenberger [2, Theorem 2.5.16] *Let  $X \subseteq A^*$  be a regular code. Then the following properties are equivalent:*

- (i)  $X$  is complete;
- (ii)  $X$  is a maximal code;
- (iii) a positive Bernoulli distribution  $\pi$  exists such that  $\pi(X) = 1$ ;
- (iv) for every positive Bernoulli distribution  $\pi$  we have  $\pi(X) = 1$ .

**Theorem 2.2** [4] *Given a non-complete code  $X$ , let  $y \in A^* \setminus F(X^*)$  be an overlapping-free word and  $U = A^* \setminus (X^* \cup A^*yA^*)$ . Then  $Y = X \cup y(Uy)^*$  is a complete code.*

With regard to word relations, the following statement comes from the definitions:

**Lemma 2.3** *Let  $\tau \in A^* \times A^*$  and  $X \subseteq A^*$ . Each of the following properties holds:*

- (i)  $X$  is  $\tau$ -independent if, and only if, it is  $\tau^{-1}$ -independent ( $\tau^{-1}$  denotes the converse relation of  $\tau$ ).
- (ii)  $X$  is  $\delta_k(\Delta_k)$ -independent if, and only if, it is  $\iota_k(I_k)$ -independent.
- (iii)  $X$  is  $\tau$ -closed if, and only if, it is  $\tau^*$ -closed.

### 3 Complete independent codes

We start by providing a few examples:

**Example 3.1** For  $A = \{a, b\}$ ,  $k = 1$ , the prefix code  $X = a^*b$  is not  $\delta_k$ -independent (we have  $a^{n-1}b \in \delta_k(a^*b)$ ), whereas the following codes are  $\delta_1$ -independent:

- the regular (prefix) code:  $Y = \{a^2\}^+ \{b, aba, abb\}$ . Note that since it contains  $\{a^2\}^+$ ,  $\delta_1(Y)$  is not a code.

- the complete (non-regular) context-free Dyck bifix code  $D_1$ , which generates the Dyck free submonoid  $D_1^* = \{w \in A^* : |w|_a = |w|_b\}$  (for every word  $w \in D_1$  we have  $|\delta_1(w)|_a \neq |\delta_1(w)|_b$ ). Note that  $\delta_1(D_1)$  contains the empty word,  $\varepsilon$ , thus it cannot be a code; however  $\delta_1(D_1) \setminus A$  remains a (non-complete) bifix code

- the non-complete finite bifix code  $Z = \{ab^2, ba^2\}$ : actually,  $\delta_1(Z)$  is the complete uniform code  $A^2$ .

- for every pair of different integers  $n, p \geq 2$ , the prefix code  $T = aA^n \cup bA^p$ . We have  $\delta_1(T) = A^n \cup A^p$ , which is not a code, although it is complete.

In view of establishing the main result of Section 3, we will construct some peculiar word:

**Lemma 3.2** Let  $k \geq 1$ ,  $i \in [1, k]$ ,  $\tau \in \{\delta_i, \iota_i, \sigma_i\}$ . Given a non-complete code  $X \subseteq A^*$  some overlapping-free word  $y \in A^* \setminus F(X^*)$  exists such that  $\tau(y)$  does not intersect  $X$  and  $y \notin \tau(X)$ .

*Proof.* Let  $X$  be a non-complete code, and let  $w \in A^* \setminus F(X^*)$ . Trivially, we have  $w^{k+1} \notin F(X^*)$ . Moreover, in a classical way a word  $u \in A^*$  exists such that  $y = w^{k+1}u$  is overlapping-free (eg. [2, Proposition 1.3.6]). Since we assume  $i \in [1, k]$ , each word in  $\tau(y)$  is constructed by deleting (inserting, substituting) at most  $k$  letters from  $y$ , hence by construction it contains at least one occurrence of  $w$  as a factor. This implies  $\tau(y) \cap F(X^*) = \emptyset$ , thus  $\tau(y)$  does not intersect  $X$ .

By contradiction, assume that a word  $x \in X$  exists such that  $y \in \tau(x)$ . It follows from  $\delta_k^{-1} = \iota_k$  and  $\sigma_k^{-1} = \sigma_k$  that  $y = w^{k+1}u$  is obtained by deleting (inserting, substituting) at most  $k$  letters from  $x$ : consequently at least one occurrence of  $w$  appears as a factor of  $x \in X \subseteq F(X^*)$ : this contradicts  $w \notin F(X^*)$ , therefore we obtain  $y \notin \tau(X)$  (cf. Figure 1).  $\square$

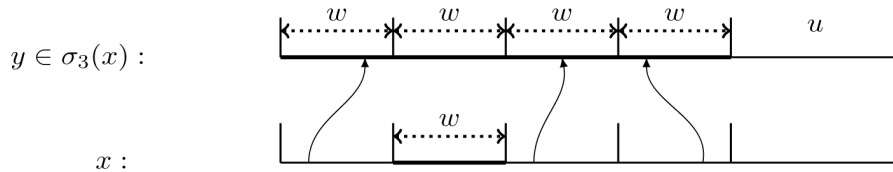


Fig. 1: Proof of Lemma 3.2:  $y \in \tau(X)$  implies  $w \in F(X)$ ; for  $i = k = 3$  and  $y = w^4u$ , the action of the substitution  $\tau = \sigma_3$  is represented by the arrows, in some extremal condition.

As a consequence, we obtain the following result:

**Theorem 3.3** *Let  $k \geq 1$  and  $\tau \in \{\delta_k, \iota_k, \sigma_k\}$ . Given a regular  $\tau$ -independent code  $X \subseteq A^*$ ,  $X$  is complete if, and only if, it is maximal as an  $\tau$ -independent codes.*

*Proof.* According to Theorem 2.1, every complete  $\tau$ -independent code is a maximal code, hence it is maximal in the family of  $\tau$ -independent codes. For proving the converse, we make use of the contrapositive. Let  $X$  be a non-complete  $\tau$ -independent code, and let  $y \in A^* \setminus F(X^*)$  satisfying the conditions of Lemma 3.2. With the notation of Theorem 2.2, necessarily  $X \cup \{y\}$ , which is a subset of  $Y = X \cup y(Uy)^*$ , is a code. According to Lemma 3.2, we have  $\tau(y) \cap X = \tau(X) \cap \{y\} = \emptyset$ . Since  $X$  is  $\tau$ -independent and  $\tau$  antireflexive, this implies  $\tau(X \cup \{y\}) \cap (X \cup \{y\}) = \emptyset$ , thus  $X$  non-maximal as a  $\tau$ -independent code.  $\square$

We notice that for  $k \geq 2$  no  $\Lambda_k$ -independent set can exist (indeed, we have  $x \in \sigma_1^2(x) \subseteq \Lambda_k(x)$ ). However, the following result holds:

**Corollary 3.4** *Let  $\tau \in \{\Delta_k, I_k, \Sigma_k, \underline{\Delta}_k\}$ . Given a regular  $\tau$ -independent code  $X \subseteq A^*$ ,  $X$  is complete if, and only if, it is maximal as a  $\tau$ -independent code.*

*Proof.* As indicated above, if  $X$  is complete, it is maximal as a  $\tau$ -independent code. For the converse, once more we argue by contrapositive that is, with the notation of Lemma 3.2, we prove that  $X \cup \{y\}$  remains independent. By definition, for each  $\tau \in \{\Delta_k, I_k, \Sigma_k, \underline{\Delta}_k\}$ , we have  $\tau \subseteq \bigcup_{1 \leq i \leq k} \tau_i$ , with  $\tau_i \in \{\delta_i, \iota_i, \sigma_i\}$ . According to Lemma 3.2, since  $\tau_i$  is antireflexive, for each  $i \in [1, k]$  we have  $\tau_i(X \cup \{y\}) \cap (X \cup \{y\}) = \emptyset$ : this implies  $(X \cup \{y\}) \cap \bigcup_{1 \leq i \leq k} \tau_i(X \cup \{y\}) = \emptyset$ , thus  $X \cup \{y\}$   $\tau$ -independent.  $\square$

With regard to the relation  $\underline{\Delta}_k$ , Corollary 3.4 expresses some interesting property in term of error detection. Indeed, as indicated in Section 1, every code is  $\underline{\Delta}_k$ -independent if the Levenshtein distance between its (distinct) elements is always larger than  $k$ . From this point, Corollary 3.4 states some characterization of the maximality in the family of such codes.

It should remain to develop some method in view of embedding a given non-complete  $\underline{\Delta}_k$ -code into a complete one. Since the construction from the proof Theorem 2.2 does not preserve independence, this question remains open.

## 4 Complete closed codes with respect to deletion or insertion

We start with relation the  $\delta_k$ . A noticeable fact is that corresponding closed codes are necessarily finite, as attested by the following result:

**Proposition 4.1** *Given a  $\delta_k$ -closed code  $X$ , and  $x \in X$ , we have  $|x| \in [1, k^2 - k - 1] \setminus \{k\}$ .*

*Proof.* It follows from  $\varepsilon \notin X$  and  $X$  being  $\delta_k$ -closed that  $|x| \neq k$ . By contradiction, assume  $|x| \geq (k-1)k$  and let  $q, r$  be the unique pair of integers such that  $|x| = qk + r$ , with  $0 \leq r \leq k-1$ . Since we have  $0 \leq rk \leq (k-1)k \leq |x|$ , an integer  $s \geq 0$  exists such that  $|x| = rk + s$ , thus words  $x_1, \dots, x_k, y$  exist such that  $x = x_1 \cdots x_k y$ , with  $|x_1| = \cdots = |x_k| = r$  and  $|y| = s$ . By construction, every word  $t \in \text{Sub}(x)$  with  $|t| \in \{r, s\}$  belongs to  $\delta_k^*(x) \subseteq X$  (indeed, we have  $r = |x| - qk$  and  $s = |x| - rk$ ). This implies  $x_1, \dots, x_k, y \in X$ , thus  $x \in X^{k+1} \cap X$ : a contradiction with  $X$  being a code.  $\square$

**Example 4.2** (1) According to Proposition 4.1, no code can be  $\delta_1$ -closed. This can be also drawn from the fact that, for every set  $X \subseteq A^+$  we have  $\varepsilon \in \delta_1^*(X)$ .

(2) Let  $A = \{a, b\}$  and  $k = 3$ . According to Proposition 4.1, every word in any  $\delta_k$ -closed code has length not greater than 5. It is straightforward to verify that  $X = \{a^2, ab, b^2, a^4b, ab^4\}$  is a  $\delta_k$ -closed code. In addition, a finite number of examinations leads to verify that  $X$  is maximal as a  $\delta_k$ -closed code. Taking for  $\pi$  the uniform distribution we have  $\pi(X) = 3/4 + 1/16 < 1$ : thus  $X$  is non-complete.

According to Example 4.2 (2), no result similar to Theorem 3.3 can be stated in the framework of  $\delta_k$ -closed codes. We also notice that, in Proposition 4.1 the bound does not depend of the size of the alphabet, but only depends of  $k$ .

**Corollary 4.3** Given a finite alphabet  $A$  and a positive integer  $k$ , one can decide whether a non-complete  $\delta_k$ -closed code  $X \subseteq A^*$  is included into some complete one. In addition there are a finite number of such complete codes, all of them being computable, if any.

*Proof.* According to Proposition 4.1 only a finite number of  $\delta_k$ -closed codes over  $A$  can exist, each of them being a subset of  $A^{\leq k^2 - k - 1} \setminus A^k$ .  $\square$

We close the section by considering the relations  $\Delta_k$ ,  $\iota_k$  and  $I_k$ :

**Proposition 4.4** No code can be  $\iota_k$ -closed,  $\Delta_k$ -closed, nor  $I_k$ -closed.

*Proof.* By contradiction assume that some  $\iota_k$ -closed code  $X \subseteq A^*$  exists. Let  $x \in X$ ,  $n = |x|$  and  $u, v \in A^*$  such that  $x = uv$ . It follows from  $|(vu)^k| = kn$ , that  $u(vu)^k v \in \iota_k^*(x)$ . According to Lemma 2.3(iii), we have  $\iota_k^*(X) \subseteq X$ , thus  $u(vu)^k v \in X$ . Since  $u(vu)^k v = (uv)^{k+1} = x^{k+1} \in X^{k+1}$ , we have  $X^{k+1} \cap X \neq \emptyset$ : a contradiction with  $X$  being a code. Consequently no  $I_k$ -closed codes can exist. According to Example 4.2(1), given a code  $X \subseteq A^*$ , we have  $\delta_1(X) \not\subseteq X$ : this implies  $\Delta_k(X) \not\subseteq X$ , thus  $X$  not  $\Delta_k$ -closed.  $\square$

## 5 Complete codes closed under substitutions

Beforehand, given a word  $w \in A^+$ , we need a thorough description of the set  $\sigma_k^*(w)$ . Actually, it is well known that, over a binary alphabet, all  $n$ -bit words can be computed by making use of some Gray sequence [5]. With our notation, we



have  $A^n = \sigma_1^*(w)$ . Furthermore, for every finite alphabet  $A$ , the so-called  $|A|$ -arity Gray sequences allow to generate  $A^n$  [8, 19]: once more we have  $\sigma_1^*(w) = A^n$ . In addition, in the special case where  $k = 2$  and  $|A| = 2$ , it can be proved that we have  $|\sigma_2(w)| = 2^{n-1}$  [8, Exercise 8, p. 28]. However, except in these special cases, to the best of our knowledge no general description of the structure of  $\sigma_k^*(w)$  appears in the literature. In any event, in the next paragraph we provide an exhaustive description of  $\sigma_k(w)$ . Strictly speaking, the proofs, that we have reported in Section 5.2, are not involved in  $\sigma_k$ -closed codes: we suggest the reader that, in a first reading, after para. 5.1 he (she) directly jumps to para. 5.3.

### 5.1 Basic results concerning $\sigma_k^*(w)$

**Proposition 5.1** *Assume  $|A| \geq 3$ . For each  $w \in A^{\geq k}$ , we have  $\sigma_k^*(w) = A^{|w|}$ .*

In the case where  $A$  is a binary alphabet, we set  $A = \{0, 1\}$ : this allows a well-known algebraic interpretation of  $\sigma_k$ . Indeed, denote by  $\oplus$  the addition in the group  $\mathbb{Z}/2\mathbb{Z}$  with identity 0, and fix a positive integer  $n$ ; given  $w, w' \in A^n$ , define  $w \oplus w'$  as the unique word of  $A^n$  such that, for each  $i \in [1, n]$ , the letter of position  $i$  in  $w \oplus w'$  is  $w_i \oplus w'_i$ . With this notation the sets  $A^n$  and  $(\mathbb{Z}/2\mathbb{Z})^n$  are in one-to-one correspondence. Classically, we have  $w' \in \sigma_1(w)$  if, and only if, some  $u \in A^n$  exists such that  $w' = w \oplus u$  with  $|u|_1 = 1$  (thus  $|u|_0 = n - 1$ ). From the fact that  $\sigma_k(w) \subseteq \sigma_1^k(w)$ , the following property holds:

$$w' \in \sigma_k(w) \iff \exists u \in A^n : w = w' \oplus u, \quad |u|_1 = k. \quad (1)$$

In addition  $w \oplus u = w'$  is equivalent to  $u = w \oplus w'$ . Let  $d = |\{i \in [1, n] : w_i = w'_i = 1\}|$ . The following property follows from  $|u|_1 = (|w|_1 - d) + (|w'|_1 - d)$  and  $|w|_1 + |w'|_1 = |w|_1 + |w'|_1 - 2|w'|_1 \pmod{2}$ :

$$|w|_1 + |w'|_1 = |w|_1 - |w'|_1 \pmod{2} = |u|_1 \pmod{2}. \quad (2)$$

Finally, for  $a \in A$  we denote by  $\bar{a}$  its complementary letter that is,  $\bar{a} = a \oplus 1$ ; for  $w \in A^n$  we set  $\bar{w} = \bar{w}_1 \cdots \bar{w}_n$ .

**Lemma 5.2** *Let  $A = \{0, 1\}$ ,  $n \geq k + 1$ . Given  $w, w' \in A^n$  the two following properties hold:*

- (i) *If  $k$  is even and  $w' \in \sigma_k^*(w)$  then  $|w'|_1 - |w|_1$  is an even integer;*
- (ii) *If  $|w'|_1 - |w|_1$  is even then we have  $w' \in \sigma_k^*(w)$ , for every  $k \geq 1$ .*

Given a positive integer  $n$ , we denote  $A_0^n$  ( $A_1^n$ ) the set of the words  $w \in A^n$  such that  $|w|_1$  is even (odd).

**Proposition 5.3** *Assume  $|A| = 2$ . Given  $w \in A^{\geq k}$  exactly one of the following conditions holds:*

- (i)  $|w| \geq k + 1$ ,  $k$  is even, and  $\sigma_k^*(w) \in \{A_0^{|w|}, A_1^{|w|}\}$ ;
- (ii)  $|w| \geq k + 1$ ,  $k$  is odd, and  $\sigma_k^*(w) = A^{|w|}$ ;
- (iii)  $|w| = k$  and  $\sigma_k^*(w) = \{w, \bar{w}\}$ .

## 5.2 Proofs of the statements 5.1, 5.2 and 5.3

Actually, Proposition 5.1 is a consequence of the following property:

**Lemma 5.4** *Assume  $|A| \geq 3$ . For every word  $w \in A^{\geq k}$  we have  $\sigma_1(w) \subseteq \sigma_k^2(w)$ .*

*Proof.* Let  $w' \in \sigma_1(w)$  and  $n = |w| = |w'| \geq k$ . We prove that  $w'' \in A^*$  exists with  $w'' \in \sigma_k(w)$  and  $w' \in \sigma_k(w'')$ . By construction,  $i_0 \in [1, n]$  exists such that:

(a)  $w'_i = w_i$  if, and only if,  $i \neq i_0$ .

It follows from  $k \leq n$  that some  $(k-1)$ -element subset  $I \subseteq [1, n] \setminus \{i_0\}$  exists. Since we have  $|A| \geq 3$ , some letter  $c \in A \setminus \{w_{i_0}, w'_{i_0}\}$  exists. Let  $w'' \in A^n$  such that:

(b)  $w''_{i_0} = c$  and, for each  $i \neq i_0$ :  $w''_i \neq w_i$  if, and only if,  $i \in I$ .

By construction we have  $w'' \in \sigma_k(w)$ , moreover  $c \neq w'_{i_0}$  implies  $w'_{i_0} \neq w''_{i_0}$ . According to (a) and (b), we obtain:

(c)  $w'_{i_0} \neq c = w''_{i_0}$ ,

(d)  $w'_i = w_i \neq w''_i$  if  $i \in I$ , and:

(e)  $w'_i = w_i = w''_i$  if  $i \notin I \cup \{i_0\}$ .

Since we have  $|I \cup \{i_0\}| = k$ , this implies  $w' \in \sigma_k(w'')$ .  $\square$

*Proof of Proposition 5.1.* Let  $w' \in A^n \setminus \{w\}$ : we prove that  $w' \in \sigma_k^*(w)$ . Let  $I = \{i_0, \dots, i_p\} = \{i \in [1, n] : w'_i \neq w_i\}$  and let  $(w^{(i_j)})_{0 \leq j \leq p}$  be a sequence of words such that  $w = w^{(i_0)}$ ,  $w^{(i_p)} = w'$  and, for each  $j \in [0, p-1]$ :  $w_\ell^{(i_{j+1})} \neq w_\ell^{(i_j)}$  if, and only if,  $\ell = i_{j+1}$ . Since we have  $w^{(i_{j+1})} \in \sigma_1(w^{(i_j)})$  ( $1 \leq j < p$ ), by induction over  $j$  we obtain  $w' \in \sigma_1^*(w)$  thus, according to Lemma 5.4,  $w' \in \sigma_k^*(w)$ .  $\square$

In view of proving Lemma 5.2 and Proposition 5.3, we need some new lemma:

**Lemma 5.5** *Assume  $|A| = 2$ . For every  $w \in A^{\geq k+1}$ , we have  $\sigma_2(w) \subseteq \sigma_k^2(w)$ .*

*Proof.* Set  $A = \{0, 1\}$ . It follows from  $\sigma_2 \subseteq \sigma_1^2$  that the result holds for  $k = 1$ . Assume  $k \geq 2$  and let  $n = |w|$ ,  $w' \in \sigma_2(w)$ . By construction, there are distinct integers  $i_0, j_0 \in [1, n]$  such that the following holds:

(a)  $w'_i = \overline{w}_i$  if, and only if,  $i \in \{i_0, j_0\}$ .

Since some  $(k-1)$ -element set  $I \subseteq [1, n] \setminus \{i_0, j_0\}$  exists, words  $w'', w''' \in A^n$  such that:

(b)  $w''_i = \overline{w}_i$  if, and only if,  $i \in \{i_0\} \cup I$ , and:

(c)  $w'''_i = \overline{w}''_i$  if, and only if,  $i \in \{j_0\} \cup I$ .

By construction, we have  $w'' \in \sigma_k(w)$  and  $w''' \in \sigma_k(w'')$ , thus  $w''' \in \sigma_k^2(w)$ . Moreover, the fact that we have  $w''' = w'$  is attested by the following equations:

(d)  $w'''_{j_0} = \overline{w}''_{j_0} = \overline{w}_{j_0} = w'_{j_0}$ ,

(e)  $w'''_{i_0} = w''_{i_0} = \overline{w}_{i_0} = w'_{i_0}$ , and:

(f) for  $i \notin \{i_0, j_0\}$ :  $w'''_i = \overline{w}''_i = w_i = w'_i$  if, and only if,  $i \in I$ .  $\square$

*Proof of Lemma 5.2.* Assume  $k$  even. According to Property (1) we have  $w' = w \oplus u$  with  $|u|_1 = k$ . According to (2),  $|w'|_1 - |w|_1$  is even: hence (i) follows.

Conversely, assume  $|w'|_1 - |w|_1$  even and let  $u = w \oplus w'$ . According to (2),  $|u|_1$  is also even, moreover according to (1) we obtain  $w' = \sigma_{|u|_1}(w)$ : this implies  $w' \in \sigma_2^*(w)$ . According to Lemma 5.5, we have  $w' \in \sigma_k^*(w)$ : this establishes (ii).  $\square$

*Proof of Proposition 5.3.* Let  $w \in A^{\geq k}$  and  $n = |w|$ . (iii) is trivial and (i) follows from Lemma 5.2(i): indeed, since  $k$  is even,  $\sigma_k^*(w)$  is the set of the words  $w' \in A^n$  such that  $|w'|_1 - |w|_1$  is even. Assume  $k$  odd and let  $w' \in A^n \setminus \{w\}$ ; we will prove that  $w' \in \sigma_k^*(w)$ . If  $|w'|_1 - |w|_1$  is even, the result comes from Lemma 5.2(ii). Assume  $|w'|_1 - |w|_1$  odd and let  $t \in \sigma_1(w')$ , thus  $w' \in \sigma_1(t) \subseteq \sigma_k \circ \sigma_{k-1}(t)$  that is,  $w' \in \sigma_k(t')$  for some  $t' \in \sigma_{k-1}(t)$ . It follows from  $w' \in \sigma_1(t)$  that  $|t|_1 - |w'|_1$  is odd, whence  $|t|_1 - |w|_1 = (|t|_1 - |w'|_1) + (|w'|_1 - |w|_1)$  is even: according to Lemma 5.2(ii), this implies  $t \in \sigma_k^*(w)$ . But since  $k-1$  is even, we have  $t' \in \sigma_{k-1}(t) \subseteq \sigma_2^*(t)$ : according to Lemma 5.5, this implies  $t' \in \sigma_k^*(t)$  (we have  $|t| = |w'| = n$ ). We obtain  $w' \in \sigma_k(t') \subseteq \sigma_k^*(t) \subseteq \sigma_k^*(\sigma_k^*(w)) = \sigma_k^*(w)$ : this completes the proof.  $\square$

### 5.3 The consequences for $\sigma$ -closed codes

Given a  $\sigma_k$ -closed code  $X \subseteq A^*$ , we say that the tuple  $(k, A, X)$  satisfies Condition (3) if each of the three following properties holds:

$$(a) \ k \text{ is even, } (b) \ |A| = 2, \quad (c) \ X \not\subseteq A^{\leq k}. \quad (3)$$

We start by proving the following technical result:

**Lemma 5.6** *Assume  $|A| = 2$  and  $k$  even. Given a pair of words  $v, w \in A^+$ , if  $|w| \geq \max\{|v| + 1, k + 1\}$  then the set  $\sigma_k^*(w) \cup \{v\}$  cannot be a code.*

*Proof.* Let  $v, w \in A^+$ , and  $n = |w| \geq \max\{|v| + 1, k + 1\}$  (hence we have  $v \notin \sigma_k^*(w) \subseteq A^{|w|}$ ). By contradiction, we assume that  $\sigma_k^*(w) \cup \{v\}$  is a code. We are in Condition (i) of Proposition 5.3 that is, we have  $\sigma_k^*(w) \in \{A_0^n, A_1^n\}$ . On a first hand, since  $A^{n-1}$  is a right-complete prefix code [2, Theorem 3.3.8], it follows from  $|v| \leq n - 1$  that a (perhaps empty) word  $s$  exists such that  $vs \in A^{n-1}$ . On another hand, it follows from  $A^{n-1}A = A^n = A_0^n \cup A_1^n$  that, for each  $u \in A^{n-1}$ , a unique pair of letters  $a_0, a_1$ , exists such that  $ua_0 \in A_0^n$ ,  $ua_1 \in A_1^n$  with  $a_1 = \bar{a}_0$  that is,  $a \in A$  exists with  $vs a \in \sigma_k^*(w)$ . According to Lemma 5.2(i),  $|sav|_1 - |w|_1 = |vsa|_1 - |w|_1$  is even; according to Lemma 5.2(ii), this implies  $sav \in \sigma_k^*(w)$ . Since we have  $(vsa)v = v(sav)$ , the set  $\sigma_k^*(w) \cup \{v\}$  cannot be a code.  $\square$

As a consequence of Lemma 5.6, we obtain the following result:

**Lemma 5.7** *Given a  $\sigma_k$ -closed code  $X \subseteq A^*$ , if  $(k, A, X)$  satisfies Condition (3) then either we have  $X \subseteq A^{\leq k}$ , or we have  $X \in \{A_0^n, A_1^n, A^n\}$  for some  $n \geq k + 1$ .*

*Proof.* Assume that we have  $X \not\subseteq A^{\leq k}$ . Firstly, consider two words  $v, w \in X \cap A^{\geq k+1}$  and by contradiction, assume  $|v| \neq |w|$  that is, without loss of generality  $|v| + 1 \leq |w|$ .

Since  $X$  is  $\sigma_k$ -closed, we have  $\sigma_k^*(w) \subseteq X$ , whence the set  $\sigma_k^*(w) \cup \{v\}$ , which a subset of  $X$  is a code: this contradicts the result of Lemma 5.6. Consequently, we have  $X \subseteq A^{\leq k} \cup A^n$ , with  $n = |v| = |w| \geq k + 1$ . Secondly, once more by contradiction assume that words  $v \in X \cap A^{\leq k}$ ,  $w \in X \cap A^{\geq k+1}$  exist. As indicated above, since  $X$  is  $\sigma_k$ -closed,  $\sigma_k^*(w) \cup \{v\}$  is a code: since we have  $|w| \geq k + 1$  and  $|w| \geq |v| + 1$ , once more this contradicts the result of Lemma 5.6. As a consequence, if  $X \not\subseteq A^{\leq k}$  then necessarily we have  $X \subseteq A^n$ , for some  $n \geq k + 1$ . With such a condition, according to Proposition 5.3 for each pair of words  $v, w \in X$ , we have  $\sigma_k^*(v), \sigma_k^*(w) \in \{A_0^n, A_1^n\}$ : this implies  $X \in \{A_0^n, A_1^n, A^n\}$ .  $\square$

According to Lemma 5.7, with Condition (3) no  $\sigma_k$ -closed code can simultaneously possess words in  $A^{\leq k}$  and words in  $A^{\geq k+1}$ .

**Lemma 5.8** *Given a  $\sigma_k$ -closed code  $X \subseteq A^*$ , if  $(k, A, X)$  does not satisfy Condition 3 then either we have  $X \subseteq A^{\leq k}$ , or we have  $X = A^n$ , with  $n \geq k + 1$ .*

*Proof.* If Condition (3) doesn't hold then exactly one of the three following conditions holds:

- (a)  $X \subseteq A^{\leq k}$ ;
- (b)  $X \not\subseteq A^k$  and  $|A| \geq 3$ ;
- (c)  $X \not\subseteq A^{\leq k}$  with  $|A| = 2$  and  $k$  odd.

With each of the two last conditions, let  $w \in X \cap A^{\geq k+1}$ . Since  $X$  is  $\sigma_k$ -closed, according to the propositions 5.1 and 5.3(ii), we have  $A^n = \sigma_k^*(w) \subseteq \sigma_k^*(X)$ . Since  $A^n$  is a maximal code, it follows from Lemma 2.3(iii) that  $X = A^n$ .  $\square$

As a consequence, every  $\sigma_k$ -closed code is finite. In addition, we state:

**Theorem 5.9** *Given a complete  $\sigma_k$   $(\Sigma_k, \Lambda_k)$ -closed code  $X$ , exactly one of the following conditions holds:*

- (i)  $X$  is a subset of  $A^{\leq k}$ ;
- (ii) a unique integer  $n \geq k + 1$  exists such that  $X = A^n$ .

*In addition, every  $\Sigma_k(\Lambda_k)$ -closed code is equal to  $A^n$ , for some  $n \geq 1$ .*

*Proof.* Let  $X$  be a complete  $\sigma_k$ -closed code. If Condition (3) does not hold, the result is expressed by Lemma 5.8. Assume that Condition (3) holds with  $X \not\subseteq A^{\leq k}$ . According to Lemma 5.7, in any case some integer  $n \geq k + 1$  exists such that  $X \in \{A_0^n, A_1^n, A^n\}$ . Taking for  $\pi$  the uniform distribution, we have  $\pi(A_0^n) = \pi(A_1^n) = 1/2$  and  $\pi(A^n) = 1$  thus, according to Theorem 2.1:  $X = A^n$ . Recall that we have  $\sigma_1^*(w) = A^{|w|}$  (eg. [8]). Let  $w \in X$  and  $n = |w|$ ; if  $X$  is  $\Sigma_k$ -closed, we have  $A^n = \sigma_1^*(X) \subseteq \Sigma_k^*(X) \subseteq X$  thus  $X = A^n$  (indeed,  $A^n$  is a maximal code). Since  $\Sigma_k \subseteq \Lambda_k$ , if  $X$  is  $\Lambda_k$ -closed then it is  $\Sigma_k$ -closed, thus we have  $X = A^n$ .  $\square$

As a corollary, in the family of  $\Sigma_k(\Lambda_k)$ -closed codes, maximality and completeness are equivalent notions. With regard to  $\sigma_k$ -closed codes, things are otherwise: indeed, as shown in [16], there are finite codes that have no finite

completion. Let  $X$  be one of them, and  $k = \max\{|x| : x \in X\}$ . By definition  $X$  is  $\sigma_k$ -closed. Since every  $\sigma_k$ -closed code is finite, no complete  $\sigma_k$ -closed code can contain  $X$ .

**Proposition 5.10** *Let  $X$  be a (finite) non-complete  $\sigma_k$ -closed code. Then one can decide whether some complete  $\sigma_k$ -closed code containing  $X$  exists. More precisely, there is only a finite number of such codes, each of them being computable, if any.*

*Proof sketch.* We draw the scheme of an algorithm that allows to compute every complete  $\sigma_k$ -closed code  $\hat{X}$  containing  $X$ . In a first step, we compute  $Y = X \cap A^{\leq k}$ . If  $Y = X$ , according to Theorem 5.9, we have  $\hat{X} \subseteq A^{\leq k}$ :  $\hat{X}$ , if any, can be computed in a finite number of steps. Otherwise,  $\hat{X}$  exists if, and only if, for some  $n \geq k + 1$  we have  $X \subseteq A^n$ : this can be straightforwardly checked.  $\square$

## References

- [1] Berstel, J., Felice, C.D., Perrin, D., Reutenauer, C., Rindonne, G.: Bifix codes and sturmian words. *J. of Algebra* **369**, 146–202 (2012)
- [2] Berstel, J., Perrin, D., Reutenauer, C.: *Codes and Automata*. Cambridge University Press (2010)
- [3] Bruyère, V., Wang, L., Zhang, L.: On completion of codes with finite deciphering delay. *European J. Comb.* **11**, 513–521 (1990)
- [4] Ehrenfeucht, A., Rozenberg, S.: Each regular code is included in a regular maximal one. *RAIRO - Theor. Inform. Appl.* **20**, 89–96 (1986)
- [5] Ehrlich, G.: Loopless algorithms for generating permutations, combinations, and other combinatorial configurations. *J. ACM* **20**, 500–513 (1973)
- [6] Jürgensen, H., Konstantinidis, S.: Codes. In: *Handbook of Formal Languages*, chap. 8, pp. 511–607. Springer Verlag, Berlin (1997)
- [7] Kari, L., Păun, G., Thierrin, G., Yu, S.: At the crossroads of linguistic, DNA computing and formal languages: characterizing RE using insertion–deletion systems. In: *Proc. of the Third DIMACS Workshop on DNA Based Computing*, pp. 318–333 (1997)
- [8] Knuth, D.: *The art of computer programming, vol.4, Fascicule 2 : generating all tuples and permutations*. Addison Wesley (2005)
- [9] Konstantinidis, S.: *Error Correction and Decodability*. Ph.D. thesis, The University of Western Ontario, London, Canada (1996)
- [10] Lam, N.: Finite maximal infix codes. *Semigr. Forum* **61**, 346–356 (2000)
- [11] Lam, N.: Finite maximal solid codes. *Theor. Comput. Sci.* **262**, 333–347 (2001)

- [12] Levenshtein, V.: Binary codes capable of correcting deletions, insertion and reversals. Soviet Physics Dokl. Engl. trans. in: Dokl. Acad. Nauk. SSSR **163**, 845–848 (1965)
- [13] Néraud, J.: Completing circular codes in regular submonoids. Theoret. Comp. Sci. **391**, 90–98 (2008)
- [14] Néraud, J., Selmi, C.: Embedding a  $\theta$ -invariant code into a complete one. Theoret. Comput. Sci. (2019, Available online 29 August 2018, in press, DOI:101016/j.tcs201808022)
- [15] Nivat, M.: Congruences parfaites et quasi-parfaites. Séminaire Dubreil. Algèbre et théorie des nombres **25**, 1–9 (1971)
- [16] Restivo, A.: On codes having no finite completion. Discr. Math. **17**, 309–316 (1977)
- [17] Rozenberg, G., Salomaa, A.: The Mathematical Theory of L-Systems. Academic Press (1980)
- [18] Rudi, K., Wonham, W.M.: The infimal prefix-closed and observable superlanguage of a given language. Systems and Control Letters **15**, 361–371 (1990)
- [19] Savage, C.: A survey of combinatorial gray codes. SIAM Rev. **39(4)**, 605–629 (1997)