Prefix partitioned Gray codes for particular cross-bifix-free sets

A. Bernini* S. Bilotta* R. Pinzani* A. Sabri† V. Vajnovszki†

July 15, 2014

Abstract

A set of words with the property that no prefix of any word is the suffix of any other word is called cross-bifix-free set. We provide an efficient generating algorithm producing Gray codes for a remarkable family of cross-bifix-free sets.

Keywords: Gray codes, cross-bifix-free sets, CAT algorithms, q-ary words.

2010 Mathematics Subjects Classification: 68R15; 94B15; 69P30.

1 Introduction

Encoding and listing the objects of a particular class is a common problem to several research areas, such as computer science and hardware or software testing, chemistry, biology and biochemistry. A very special kind of list is the so called *Gray code*, where two successive objects differ 'in some pre-specified small way' [11]. Gray codes are involved in several combinatorial structures, as permutations [16], binary strings, Motzkin and Schröder words [19], derangements [3], involutions [20]. They are also used in other technological subjects as circuit testing, signal encoding [15], data compression and others.

The generation of a Gray code is often closely related with the nature of the objects we are dealing with. In this context we are going to generate Gray codes for particular words over the alphabet $\{0, 1, \dots q-1\}$, so that two successive words differ in a single position and by 1 or -1 in this position.

The set we are dealing with is called *cross-bifix-free* (this term appeared for the first time in [1], see also [2]) and consists of words having some constraints we are going to define in the following. We recall that a *bifix* (or *border*) of a word is a factor which is both a prefix and a suffix. A word is said to be *bifix-free* if it does not contain any bifix [17]. A set of bifix-free words is *cross-bifix-free* if, given any two words, any prefix of the first one is not a suffix of the second one.

Cross-bifix-free sets are involved in the study of distributed sequences for frame synchronization [14]. The problem of determining such sets is also related to several other scientific

*Dipartimento di Matematica e Informatica "Ulisse Dini", Università degli Studi di Firenze, Viale G.B. Morgagni 65, 50134 Firenze, Italy. `{bernini}{bilotta}{pinzani}@dsi.unifi.it`

†LE2I, Université de Bourgogne, BP 47 870, 21078 Dijon Cedex, France. `{ahmad.sabri}{vvajnov}@u-bourgogne.fr`

applications, for instance in pattern matching [7] and automata theory [4]. Moreover, in some applications, listing a cross-bifix-free set in Gray code manner can be of a particular interest.

Several methods for constructing cross-bifix-free sets have been recently proposed as in [2, 5, 6]. To our knowledge, for a fixed cardinality of the alphabet and of the length of the words, the construction giving the cross-bifix-free set having the greatest cardinality is the one proposed in [6]. In this paper we propose Gray codes for such cross-bifix-free sets which allow a deeper understanding and possible further applications of this combinatorial class.

Following the authors of [6], the cross-bifix-free set we consider here is denoted by $S_{n,q}^{(k)}$. It is formed by length n words over the q-ary alphabet $[q] = \{0, 1, \ldots, q-1\}$ containing a particular sub-word avoiding k consecutive 0s. Now we briefly summarize its definition and we refer the reader to [6] for more details about its features.

Let $n \geq 3$, $q \geq 2$ and $1 \leq k \leq n-2$. The cross-bifix-free set $S_{n,q}^{(k)}$ is the set of all length n words $s_1 s_2 \cdots s_n$ over $[q]$ satisfying:

- $s_1 = \cdots = s_k = 0$;
- $s_{k+1} \neq 0$;
- $s_n \neq 0$;
- the sub-word $s_{k+2} \ldots s_{n-1}$ does not contain k consecutive 0s.

The set $S_{5,4}^{(2)}$ is showed in Table 1 (c). Note that it is already listed in a Gray code manner and its generation will be presented in the next sections.

Below we recall the cardinality of $S_{n,q}^{(k)}$: let

$$f_{n,q}^{(k)} = \begin{cases} q^n & \text{if } 0 \leq n < k, \\ (q-1)\left(f_{n-1,q}^{(k)} + f_{n-2,q}^{(k)} + \ldots + f_{n-k,q}^{(k)}\right) & \text{if } n \geq k, \end{cases} \tag{1}$$

be the sequence enumerating the words of n length words over $[q]$ avoiding k consecutive zeros [15, 18] (observe that in the particular case of $q = 2$, the well known *k-generalized Fibonacci sequences* [12] are obtained). It is not difficult to realize that, from the above description of $S_{n,q}^{(k)}$ we have:

$$|S_{n,q}^{(k)}| = (q-1)^2 f_{n-k-2,q}^{(k)}. \tag{2}$$

In this work we propose a Gray code which is *prefix partitioned* in the sense that all the words with the same prefix are consecutive. For constructive reasons we give the Gray code $\mathcal{S}_{n+k,q}^{(k)}$ for the set $S_{n+k,q}^{(k)}$ starting from particular list of length n words, then prepending the 0^k prefix. Our strategy in order to obtain $\mathcal{S}_{n+k,q}^{(k)}$ is the following:

- first, we adapt a q-ary generalization of the Binary Reflected Gray Code [9] and obtain the list $\mathcal{H}_{n,q}^{(k)}(u)$, a Gray code for the set of q-ary words with no 0^k factors and beginning by at most u 0s, $0 \leq u \leq k-1$, then
- we restrict this Gray code to words which end by a non-zero symbol and we denote by $\mathcal{J}_{n,q}^{(k)}(u)$ the obtained list; in particular $\mathcal{J}_{n,q}^{(k)}(0)$ is a Gray code for the set of q-ary words with no 0^k factors which begin and end by a non-zero symbol;

- finally, prepending a 0^k prefix to each word in $\mathcal{J}^{(k)}_{n,q}(0)$, the desired Gray code $\mathcal{S}^{(k)}_{n+k,q}$ is obtained.

In the following we will use the notations below:

- For a list of words $\mathcal{L}$, $\overline{\mathcal{L}}$ denotes the list obtained by covering $\mathcal{L}$ in reverse order; and for $i \geq 0$, $(\mathcal{L})^{\underline{i}}$ denotes the list $\mathcal{L}$ if i is even, and the list $\overline{\mathcal{L}}$ if i is odd;
- $\mathsf{first}(\mathcal{L})$ and $\mathsf{last}(\mathcal{L})$ denotes, respectively, the first and the last word of $\mathcal{L}$, and clearly, $\mathsf{first}(\overline{\mathcal{L}}) = \mathsf{last}(\mathcal{L})$ and $\mathsf{last}(\overline{\mathcal{L}}) = \mathsf{first}(\mathcal{L})$;
- If α is a word, then α^n is the word which consists of n copies of α; and $\alpha \cdot \mathcal{L}$ is the list obtained by concatening α to each word of $\mathcal{L}$;
- For two lists $\mathcal{L}$ and $\mathcal{M}$, $\mathcal{L} \circ \mathcal{M}$ denotes their concatenation, and for two integers, $p \leq r$, and the lists $\mathcal{L}_p, \mathcal{L}_{p+1}, \ldots, \mathcal{L}_r$, we denote by $\bigcirc_{i=p}^{r} \mathcal{L}_i$ the list $\mathcal{L}_p \circ \mathcal{L}_{p+1} \circ \ldots \circ \mathcal{L}_r$;
- $\mathcal{L} \subset \mathcal{M}$ means that $\mathcal{L}$ is a (possibly scattered) sublist of $\mathcal{M}$; in this case the corresponding underlying sets L and M satisfy $L \subset M$.

2 The Gray codes construction

For our purpose we need a Gray code list for the set of words of a certain length over the q-ary alphabet $[q]$, $q \geq 2$. An obvious generalization of the Binary Reflected Gray Code [9] to the alphabet $[q]$ is the list $\mathcal{G}_{n,q}$ for the set of words $[q]^n$ (i.e. the length n words over $[q]$) defined in [8, 22] where it is also shown that $\mathcal{G}_{n,q}$ is a Gray code with Hamming distance 1. We recall that the Hamming distance between two successive words in a Gray code list is the number of positions where the two words differ. Moreover, if the Hamming distance is d, then the associated Gray code is said d-Gray code. The list $\mathcal{G}_{n,q}$ is defined as:

$$\mathcal{G}_{n,q} = \begin{cases} \lambda & \text{if } n = 0, \\ \bigcirc_{i=0}^{q-1} i \cdot (\mathcal{G}_{n-1,q})^{\underline{i}} & \text{if } n > 0, \end{cases} \tag{3}$$

where λ is the empty word. The reader can easily verify the following proposition.

Proposition 1. *For $q \geq 2$,*

- $\mathsf{first}(\mathcal{G}_{n,q}) = 0^n$*;*
- $\mathsf{last}(\mathcal{G}_{n,q}) = (q-1)0^{n-1}$ *if q is even, and $(q-1)^n$ if q is odd.*

Now, let fix a $k \geq 2$ and for $0 \leq u \leq k-1$ let us define the list $\mathcal{H}^{(k)}_{n,q}(u) = \mathcal{H}_{n,q}(u)$ as:

$$\mathcal{H}_{n,q}(u) = \begin{cases} \lambda & \text{if } n = 0, \\ \bigcirc_{i=1}^{q-1} i \cdot (\mathcal{H}_{n-1,q}(k-1))^{\underline{i}} & \text{if } n > 0,\ u = 0, \\ 0 \cdot \mathcal{H}_{n-1,q}(u-1) \circ \bigcirc_{i=1}^{q-1} i \cdot (\mathcal{H}_{n-1,q}(k-1))^{\underline{i}} & \text{if } n, u > 0. \end{cases} \tag{4}$$

Notice that, when $q = 2$, the case $n > 0$ and $u = 0$ of this definition becomes $\mathcal{H}_{n,2}(0) = 1 \cdot \overline{\mathcal{H}_{n-1,2}(k-1)}$.

Let $\alpha = \alpha_1\alpha_2\alpha_3 \dots$ be the infinite word $(1(q-1)0^{k-1})^\infty$. The following propositions hold.

Proposition 2. *The last and first words in the list $\mathcal{H}_{n,q}(u)$ are given by:*

- *If q is even, then*
  - $\mathsf{first}(\mathcal{H}_{n,q}(u)) = 0^u\alpha_1\alpha_2\dots\alpha_{n-u}$,
  - $\mathsf{last}(\mathcal{H}_{n,q}(u)) = (q-1)0^{k-1}\alpha_1\alpha_2\dots\alpha_{n-k}$.
- *If q is odd, then*
  - $\mathsf{first}(\mathcal{H}_{n,q}(u)) = 0^u1(q-1)^{n-u-1}$,
  - $\mathsf{last}(\mathcal{H}_{n,q}(u)) = (q-1)^n$.

Proof. Let $q \geq 2$ be even and $0 \leq u < n$. By relation (4) it follows that

$$\begin{aligned}
\mathsf{first}(\mathcal{H}_{n,q}(u)) &= 0^u \cdot \mathsf{first}(\mathcal{H}_{n-u,q}(0)) \\
&= 0^u1 \cdot \mathsf{first}(\overline{\mathcal{H}_{n-u-1,q}(k-1)}) \\
&= 0^u1 \cdot \mathsf{last}(\mathcal{H}_{n-u-1,q}(k-1))
\end{aligned}$$

and when $u < n-1$, since $q-1$ is odd, we have

$$\begin{aligned}
\mathsf{first}(\mathcal{H}_{n,q}(u)) &= 0^u1(q-1) \cdot \mathsf{last}(\overline{\mathcal{H}_{n-u-2,q}(k-1)}) \\
&= 0^u1(q-1) \cdot \mathsf{first}(\mathcal{H}_{n-u-2,q}(k-1)),
\end{aligned}$$

and induction on n completes the proof. Similarly,

$$\begin{aligned}
\mathsf{last}(\mathcal{H}_{n,q}(u)) &= (q-1) \cdot \mathsf{last}(\overline{\mathcal{H}_{n-1,q}(k-1)}) \\
&= (q-1) \cdot \mathsf{first}(\mathcal{H}_{n-1,q}(k-1)),
\end{aligned}$$

and by the previous result, the statement holds.
Now, if $q \geq 3$ is odd and $n > 1$, for any u we have

$$\mathsf{last}(\mathcal{H}_{n,q}(u)) = (q-1) \cdot \mathsf{last}(\mathcal{H}_{n-1,q}(k-1)),$$

and thus, $\mathsf{last}(\mathcal{H}_{n,q}(u)) = (q-1)^n$. And when $u < n$, we have

$$\begin{aligned}
\mathsf{first}(\mathcal{H}_{n,q}(u)) &= 0^u \cdot \mathsf{first}(\mathcal{H}_{n-u,q}(0)) \\
&= 0^u1 \cdot \mathsf{first}(\overline{\mathcal{H}_{n-u-1,q}(k-1)}) \\
&= 0^u1 \cdot \mathsf{last}(\mathcal{H}_{n-u-1,q}(k-1)) \\
&= 0^u1(q-1)^{n-u-1},
\end{aligned}$$

and the result holds. □

Proposition 3. *The list $\mathcal{H}_{n,q}(u)$ is a Gray code for the set of words in $[q]^n$ with no 0^k factors and beginning by at most u 0s.*

Proof. Clearly, by the definition given by relation (4), the list $\mathcal{H}_{n,q}(u)$ contains exactly once each length n q-ary word with no 0^k factors and beginning by at most u 0s, and only them. Now we show that two consecutive words in this list differ in a single position and by 1 or -1 in this position. To do this, considering the recursive definition (4) it is enough to show that the last and first words in two consecutive lists in (4) differ in this way. Let $\mathcal{L}$ and $\mathcal{M}$ be such consecutive lists.

When q is even, then either

- $\mathcal{L} = 0 \cdot \mathcal{H}_{n-1,q}(u-1)$ and $\mathcal{M} = 1 \cdot (\overline{\mathcal{H}_{n-1,q}(k-1)})$, or
- $\mathcal{L} = i \cdot \mathcal{H}_{n-1,q}(k-1)$ and $\mathcal{M} = (i+1) \cdot (\overline{\mathcal{H}_{n-1,q}(k-1)})$ with even $i \geq 2$, or
- $\mathcal{L} = i \cdot \overline{\mathcal{H}_{n-1,q}(k-1)}$ and $\mathcal{M} = (i+1) \cdot (\mathcal{H}_{n-1,q}(k-1))$ with odd i.

In the first two cases we have:

- $\mathsf{last}(\mathcal{L}) = 0 \cdot \mathsf{last}(\mathcal{H}_{n-1,q}(u-1)) = 0(q-1)0^{k-1}\alpha_1\alpha_2 \dots \alpha_{n-k-1}$, and
- $\mathsf{first}(\mathcal{M}) = 1 \cdot \mathsf{first}(\overline{\mathcal{H}_{n-1,q}(k-1)}) = 1 \cdot \mathsf{last}(\mathcal{H}_{n-1,q}(k-1)) =$
  $= 1(q-1)0^{k-1}\alpha_1\alpha_2 \dots \alpha_{n-k-1}$,

and in the last case

- $\mathsf{last}(\mathcal{L}) = i \cdot \mathsf{last}(\overline{\mathcal{H}_{n-1,q}(k-1)}) = i \cdot \mathsf{first}(\mathcal{H}_{n-1,q}(k-1)) = i0^{k-1}\alpha_1\alpha_2 \dots \alpha_{n-k}$, and
- $\mathsf{first}(\mathcal{M}) = (i+1) \cdot \mathsf{first}(\mathcal{H}_{n-1,q}(k-1)) = (i+1)0^{k-1}\alpha_1\alpha_2 \dots \alpha_{n-k}$,

and in any case $\mathsf{last}(\mathcal{L})$ and $\mathsf{first}(\mathcal{M})$ differ in the desired way.

When q is odd, the proof is similar and if i is even we have

- $\mathsf{last}(\mathcal{L}) = i(q-1)^{n-1}$, and
- $\mathsf{first}(\mathcal{M}) = (i+1)(q-1)^{n-1}$,

and if i is odd

- $\mathsf{last}(\mathcal{L}) = i0^{k-1}1(q-1)^{n-k-1}$, and
- $\mathsf{first}(\mathcal{M}) = (i+1)0^{k-1}1(q-1)^{n-k-1}$,

and again, in both cases $\mathsf{last}(\mathcal{L})$ and $\mathsf{first}(\mathcal{M})$ differ in the desired way. □

Corollary 1. The list $\mathcal{H}_{n,q}(k-1)$ is a Gray code for the set of words in $[q]^n$ with no 0^k factors.

See Table 1 (a) for the list $\mathcal{H}_{6,2}(1)$ with $k=2$.

As previously, let fix a $k \geq 2$ and for $0 \leq u \leq k-1$ let us define the list $\mathcal{J}^{(k)}_{n,q}(u) = \mathcal{J}_{n,q}(u)$ as:

$$\mathcal{J}_{n,q}(u) = \begin{cases} \lambda & \text{if } n = 0, \\ \bigcirc_{i=1}^{q-1} i \cdot (\mathcal{J}_{n-1,q}(k-1))^{\underline{i}} & \text{if } n = 1 \text{ or} \\ & n > 1 \text{ and } u = 0, \\ 0 \cdot \mathcal{J}_{n-1,q}(u-1) \circ \bigcirc_{i=1}^{q-1} i \cdot (\mathcal{J}_{n-1,q}(k-1))^{\underline{i}} & \text{if } n > 1 \text{ and } u > 0. \end{cases} \tag{5}$$

1	0 1 1 0 1 1
2	0 1 1 0 1 0
3	0 1 1 1 1 0
4	0 1 1 1 1 1
5	0 1 1 1 0 1
6	0 1 0 1 0 1
7	0 1 0 1 1 1
8	0 1 0 1 1 0
9	1 1 0 1 1 0
10	1 1 0 1 1 1
11	1 1 0 1 0 1
12	1 1 1 1 0 1
13	1 1 1 1 1 1
14	1 1 1 1 1 0
15	1 1 1 0 1 0
16	1 1 1 0 1 1
17	1 0 1 0 1 1
18	1 0 1 0 1 0
19	1 0 1 1 1 0
20	1 0 1 1 1 1
21	1 0 1 1 0 1

(a)

1	1 2 2 2	22	2 1 2 1
2	1 2 2 1	23	2 1 1 1
3	1 2 1 1	24	2 1 1 2
4	1 2 1 2	25	2 1 0 2
5	1 2 0 2	26	2 1 0 1
6	1 2 0 1	27	2 2 0 1
7	1 1 0 1	28	2 2 0 2
8	1 1 0 2	29	2 2 1 2
9	1 1 1 2	30	2 2 1 1
10	1 1 1 1	31	2 2 2 1
11	1 1 2 1	32	2 2 2 2
12	1 1 2 2		
13	1 0 2 2		
14	1 0 2 1		
15	1 0 1 1		
16	1 0 1 2		
17	2 0 1 2		
18	2 0 1 1		
19	2 0 2 1		
20	2 0 2 2		
21	2 1 2 2		

(b)

1	0 0 1 3 1	22	0 0 2 3 3
2	0 0 1 3 2	23	0 0 2 3 2
3	0 0 1 3 3	24	0 0 2 3 1
4	0 0 1 2 3	25	0 0 3 3 1
5	0 0 1 2 2	26	0 0 3 3 2
6	0 0 1 2 1	27	0 0 3 3 3
7	0 0 1 1 1	28	0 0 3 2 3
8	0 0 1 1 2	29	0 0 3 2 2
9	0 0 1 1 3	30	0 0 3 2 1
10	0 0 1 0 3	31	0 0 3 1 1
11	0 0 1 0 2	32	0 0 3 1 2
12	0 0 1 0 1	33	0 0 3 1 3
13	0 0 2 0 1	34	0 0 3 0 3
14	0 0 2 0 2	35	0 0 3 0 2
15	0 0 2 0 3	36	0 0 3 0 1
16	0 0 2 1 3		
17	0 0 2 1 2		
18	0 0 2 1 1		
19	0 0 2 2 1		
20	0 0 2 2 2		
21	0 0 2 2 3		

(c)

Table 1: The Gray code list: (a) $\mathcal{H}^{(2)}_{6,2}(1)$ for the set of length 6 binary words with no two consecutive 0s; (b) $\mathcal{J}^{(2)}_{4,3}(0)$ for the set of length 4 ternary words with no two consecutive 0s and beginning and ending by a non-zero symbol; and (c) $\mathcal{S}^{(2)}_{5,4}$ for the set of cross-bifix-free words $S^{(2)}_{5,4}$ obtained by prepending 00 to each word in $\mathcal{J}^{(2)}_{3,4}(0)$.

Proposition 4. *The last and first words in the list $\mathcal{J}_{n,q}(u)$ are given by:*

- *If q is even, let $f = f_1 f_2 \dots f_n = \mathsf{first}(\mathcal{H}_{n,q}(u))$ and $\ell = \ell_1 \ell_2 \dots \ell_n = \mathsf{last}(\mathcal{H}_{n,q}(u))$. Then*
  - $\mathsf{first}(\mathcal{J}_{n,q}(u)) = \begin{cases} f_1 f_2 \dots f_n & \text{if} \quad f_n \neq 0, \\ f_1 f_2 \dots f_{n-1} 1 & \text{if} \quad f_n = 0. \end{cases}$
  - $\mathsf{last}(\mathcal{J}_{n,q}(u)) = \begin{cases} \ell_1 \ell_2 \dots \ell_n & \text{if} \quad \ell_n \neq 0, \\ \ell_1 \ell_2 \dots \ell_{n-1} 1 & \text{if} \quad \ell_n = 0. \end{cases}$
- *If q is odd, then*
  - $\mathsf{first}(\mathcal{J}_{n,q}(u)) = 0^u 1 (q-1)^{n-u-1}$,
  - $\mathsf{last}(\mathcal{J}_{n,q}(u)) = (q-1)^n$.

Proof. The proof is similar to the one in Proposition 2 and imposing that the last symbol of $\mathsf{first}(\mathcal{J}_{n,q}(u))$ and $\mathsf{last}(\mathcal{J}_{n,q}(u))$ is not 0, then the thesis holds. □

Proposition 5. *The list $\mathcal{J}_{n,q}(u)$ is a Gray code for the set of words in $[q]^n$ which begin by at most u 0s, have no 0^k factors and end by a non-zero symbol.*

Proof. The proof is similar to the one in Proposition 3. $\square$

Corollary 2. The list $\mathcal{J}_{n,q}(0)$ is a prefix partitioned 1-Gray code for the set of words in $[q]^n$ which have no 0^k factors and begin and end by a non-zero symbol.

Note that, from the recursive relation (5), the words in $\mathcal{J}_{n,q}(0)$ with the same prefix are consecutively generated, as the reader can easily check using inductive arguments. See Table 1 (b) for the list $\mathcal{J}_{4,3}(0)$ with $k = 2$.

Remark 1.

- For $u = 0, 1, \ldots k-1$, $\mathcal{J}_{n,q}(u) \subset \mathcal{H}_{n,q}(u)$; and
- for $u = 0, 1, \ldots k-2$, $\mathcal{J}_{n,q}(u) \subset \mathcal{J}_{n,q}(u+1)$ and $\mathcal{H}_{n,q}(u) \subset \mathcal{H}_{n,q}(u+1)$.

Finally, the Gray code list $\mathcal{S}^{(k)}_{n+k,q}$ for the set of cross-bifix-free words $S^{(k)}_{n+k,q}$ is obtained from the list $\mathcal{J}_{n,q}(0)$ by prepending to each word in $\mathcal{J}_{n,q}(0)$ the prefix 0^k; see Table 1 (c) for the list $\mathcal{S}^{(2)}_{5,4}$.

3 Generating algorithms

Algorithm `gen_J` in Figure 1 is a direct implementation of the recursive definition of the list $\mathcal{J}_{n,q}(u)$ given by relation (5). Integer variables n, q, k and array b are global and the call of `gen_J(1,0,0)` produces the Gray code $\mathcal{J}^{(k)}_{n,q}(0)$ for the set of length n q-ary words with no 0^k factors and beginning and ending by a non-zero symbol; and prepending 0^k to each word in $\mathcal{J}^{(k)}_{n,q}(0)$ the Gray code list $\mathcal{S}^{(k)}_{n+k,q}$ is obtained. For example, when $n = 4$, $q = 3$ and $k = 2$, the call of `gen_J(1,0,0)` produces the list in Table 1 (b).

Before proving Proposition 7 we need the following result.

Proposition 6. *For $q, k \geq 2$, the integer sequence $(f^{(k)}_{n,q})_{n\geq 0}$ defined in relation (1) satisfies*

$$\sum_{i=0}^{n-1} f^{(k)}_{i,q} \leq 2 \cdot f^{(k)}_{n,q}. \tag{6}$$

Proof. Indeed, for $n < k$, $f^{(k)}_{n,q} = q^n$, and it is routine to check the result. Now let us suppose the result true for any $n < m$, and let us prove it for m.

$$\begin{aligned}
\sum_{i=0}^{m} f^{(k)}_{i,q} &= \sum_{i=0}^{m-2} f^{(k)}_{i,q} + f^{(k)}_{m-1,q} + f^{(k)}_{m,q} \\
&\leq \sum_{i=0}^{m-2} f^{(k)}_{i,q} + f^{(k)}_{m+1,q} \\
&\leq 2f^{(k)}_{m-1,q} + f^{(k)}_{m+1,q} \\
&\leq f^{(k)}_{m-1,q} + f^{(k)}_{m,q} + f^{(k)}_{m+1,q} \\
&\leq 2f^{(k)}_{m+1,q},
\end{aligned}$$

and induction on n completes the proof. $\square$

Proposition 7. *Procedure* `gen_J` *has a constant amortized time complexity.*

Proof. Let $J_{n,q}(u)$ be the underlying set of the list $\mathcal{J}_{n,q}(u)$, that is the set of words in $[q]^n$ which begin by at most u 0s, have no 0^k factors and end by a non-zero symbol. By the second point of Remark 1 we have

$$J_{n,q}(0) \subset J_{n,q}(1) \subset \ldots \subset J_{n,q}(k-1),$$

and since

$$|J_{n,q}(0)| = (q-1)^2 f^{(k)}_{n-2,q}$$

it follows that

$$(q-1)^2 f^{(k)}_{n-2,q} \leq |J_{n,q}(u)| \tag{7}$$

for $n \geq 2$ and $0 \leq u \leq k-1$, and with $f^{(k)}_{n,q}$ defined in relation (1), see also relation (2).

Similarly, let $H_{n,q}(u)$ be the underlying set of the list $\mathcal{H}_{n,q}(u)$, that is the set of words in $[q]^n$ which begin by at most u 0s, and have no 0^k factors. Again by the second point of Remark 1 we have

$$H_{n,q}(0) \subset H_{n,q}(1) \subset \ldots \subset H_{n,q}(k-1),$$

and since

$$|H_{n,q}(k-1)| = f^{(k)}_{n,q}$$

it follows that

$$|H_{n,q}(u)| \leq f^{(k)}_{n,q} \tag{8}$$

for $n \geq 0$ and $0 \leq u \leq k-1$.

The total amount of computation of the procedure `gen_J` is proportional to the number of recursive calls of this procedure. Each call of `gen_J` produces either a proper prefix of a word in $J_{n,q}(u)$, that is a word in $H_{i,q}(u)$ for some $i < n$, or a word in $J_{n,q}(u)$ which is output, and so, the total amount of computation of this procedure is proportional to

$$\sum_{i=0}^{n-1} |H_{i,q}(u)| + |J_{n,q}(u)|,$$

and the average complexity is proportional to

$$\begin{aligned}
\frac{\sum_{i=0}^{n-1} |H_{i,q}(u)| + |J_{n,q}(u)|}{|J_{n,q}(u)|} &= \frac{\sum_{i=0}^{n-1} |H_{i,q}(u)|}{|J_{n,q}(u)|} + 1 \\
&\leq \frac{\sum_{i=0}^{n-1} f^{(k)}_{i,q}}{(q-1)^2 \cdot f^{(k)}_{n-2,q}} + 1 \text{ (by relations (7) and (8))} \\
&\leq \frac{2 \cdot f^{(k)}_{n,q}}{(q-1)^2 \cdot f^{(k)}_{n-2,q}} + 1 \text{ (by Proposition 6)} \\
&\leq \frac{2 \cdot q^2 \cdot f^{(k)}_{n-2,q}}{(q-1)^2 \cdot f^{(k)}_{n-2,q}} + 1 \text{ (considering } f^{(k)}_{i,q} \leq q \cdot f^{(k)}_{i-1,q}) \\
&\leq 9.
\end{aligned}$$

Thus, the average complexity of the procedure `gen_J` is bounded by a constant.

□

```
procedure gen_J(pos,dir,u)
if pos = n+1
then output b;
else if dir = 0
     then if u > 0 and pos ≠ n
          then b[pos] := 0; gen_J(pos+1,0,u−1);
          end if
          for i := 1 to q−1 do
               b[pos] := i; gen_J(pos+1,i mod 2,k−1);
     else for i := q−1 downto 1 do
               b[pos] := i; gen_J(pos+1,(i+1) mod 2,k−1);
          if u > 0 and pos ≠ n
          then b[pos] := 0; gen_J(pos+1,1,u−1);
          end if
     end if
end if
end procedure.
```

Figure 1: Algorithm producing the list $\mathcal{J}_{n,q}(u)$, defined in relation (5).

4 Conclusions and further developments

We presented a prefix partitioned Gray code for the cross-bifix-free set of q-ary words of length n containing exactly once the factor 0^k studied in [6], which for fixed q, n and k has, to our knowledge, the greatest cardinality.

Our strategy, outlined in Introduction, uses two strictly related lists (namely, the Gray codes $\mathcal{H}_{n,q}(u)$ and $\mathcal{J}_{n,q}(u)$ defined in Section 2) in order to define the desired Gray code, denoted by $\mathcal{S}^{(k)}_{n+k,q}$. Moreover, the two lists are crucial in the proof of Proposition 7 stating the efficiency of procedure gen_J, generating our Gray code $\mathcal{S}^{(k)}_{n+k,q}$. In particular, this Gray code is obtained by prepending the prefix 0^k to each word in the list $\mathcal{J}_{n,q}(0)$ of q-ary words of length n with no 0^k factors and which begin and end with a non-zero symbol.

From Proposition 4 it is directly seen that when q is odd, the first and the last word in the list $\mathcal{J}_{n,q}(0)$ differ in the same way as any two consecutive words in this list (in a single position and by 1 or -1 in this position), and thus $\mathcal{J}_{n,q}(0)$ is a *cyclic* Gray code; consequently, for q odd so is $\mathcal{S}^{(k)}_{n+k,q}$. This is not longer true for q even: the first and the last word in $\mathcal{S}^{(k)}_{n+k,q}$ can differ in arbitrary many positions; and it should be interesting to find a cyclic Gray code for the set of cross-bifix-free words under consideration, independently on the parity of q.

References

[1] Bajic, D. On Construction of Cross-Bifix-Free Kernel Sets. 2nd MCM COST 2100, TD(07)237, Lisbon, Portugal (2007).

[2] Bajic, D. A simple suboptimal construction of cross-bifix-free codes. *Cryptography and Communications* **6**(1) 27–37 (2014).

[3] Baril, J., Vajnovszki, V. Gray code for derangements. *Discrete Applied Mathematics* **140** 207–221 (2004).

[4] Berstel, J., Perrin, D. and Reutenauer, C. Codes and Automata (Encyclopedia of Mathematics and its Applications). Cambridge University Press (2009).

[5] Bilotta, S., Pergola, E. and Pinzani, R. A new approach to cross-bifix-free sets. *IEEE Transactions on Information Theory* **58** 4058–4063 (2012).

[6] Chee, Y.M., Kiah, H.M., Purkayastha, P. and Wang, C. Cross-bifix-free codes within a constant factor of optimality. *IEEE Transactions on Information Theory* **59** 4668–4674 (2013).

[7] Crochemore, M., Hancart, C. and Lecroq, T. (2007) Algorithms on strings. Cambridge University Press, Cambridge (2007).

[8] Er, M.C. On generating the N-ary reflected Gray code. *IEEE Transactions on Computer* **33** 739–741 (1984).

[9] Gray, F. Pulse Code Communication. U.S. Patent 2 632 058 (1953).

[10] Hamming, R.W. Error detecting and error correcting codes. *Bell System Technical Journal* **29** 147–160 (1950).

[11] Joichi, J., White, D.E. and Williamson, S.G., Combinatorial Gray codes. *Siam J. on Computing* **9** 130–141 (1980).

[12] Knuth, D.E. The Art of Computer Programming. Vol. 3, "Sorting and Searching", Addison-Wesley, Reading, MA (1973).

[13] Levesque, C. On m-th order linear recurrences. *Fibonacci Quarterly* **23** 290–293 (1985).

[14] de Lind van Wijngaarden, A.J. and Willink, T. J. Frame synchronization using distributed sequences. *IEEE Transactions on Commununications* **48** 2127–2138 (2000).

[15] Ludman, J.E. Gray code generation for MPSK signals. *IEEE Transactions on Communications* **29** 1519–1522 (1981).

[16] Johnson, S.M. Generation of permutations by adjacent transpositions. *Math. Comp.* **17** 282–285 (1963).

[17] Nielsen, P.T. A Note on Bifix-Free Sequences. *IEEE Transaction Information Theory* **29** 704–706 (1973).

[18] Shork, M. The r-generalized Fibonacci numbers and Polynomial coefficients. *International Journal of Contemporary Mathematical Sciences* **3** 1157–1163 (2008).

[19] Vajnovszki, V. Gray visiting Motzkin *Acta Informatica* **38** 793–811 (2002).

[20] Walsh, T. Gray codes for involutions. *Journal of Combinatorial Mathematics and Combinatorial Computing* **36** 95–118 (2001).

[21] Walsh, T. Generating Gray Codes in O(1) worst-case time per word. *Lecture Notes in Computer Science* **2731** 73–88 (2003).

[22] Williamson, S.G. Combinatorics for computer science. Computer Science Press, Rockville, Maryland (1985).