f-Words and Binary Solid Codes[1]

Stavros Konstantinidis[†] and Joshua Young[†]

[†]Department of Mathematics and Computing Science
Saint Mary's University
Halifax, Nova Scotia, B3H 3C3 Canada
s.konstantinidis@smu.ca, jyo04@hotmail.com

Abstract. Given any unbounded and non-decreasing sequence f of positive integers, we define an infinite set of binary words, called f-words, which constitute an overlap-free language. We investigate some properties of this language and then use these properties to define new classes of finite and infinite binary solid codes – solid codes have the strongest synchronization and error-delimiting capabilities in the hierarchies of codes. The finite class improves on an earlier construction of solid codes in terms of average word length (or information rate), without sacrificing their encoding complexity. The infinite class concerns maximal solid codes and builds on an earlier work on maximal binary solid codes. This work constitutes another step towards a systematic structural characterization of binary maximal solid codes.

Key words: construction, encoding complexity, maximal, overlap-free language, solid code, word.

1 Introduction

Solid codes constitute a proper subclass of comma-free codes, providing thus synchronization of messages without delay. In addition, they possess the following remarkable property: if a message over a solid code undergoes any type of errors, then the codewords containing no errors can be identified and decoded correctly without delay. We refer the reader to [3] for a relevant discussion and further references on these codes, as well as to [2], [7], [1], [11], [5], [9] for more recent results on these objects.

From a language theoretic point of view, a solid code is an infix code (no codeword is contained in another codeword) and an overlap-free language (no proper prefix of a codeword is also a proper suffix of some codeword, unless it is empty). The general problem of constructing "good" solid codes is of central importance in our context. Some of the criteria used in the literature for evaluating the quality of codes are the following: maximality, information ratio (or average word length), encoding/decoding complexity, error-detectability. In addition, there have been systematic efforts to characterize explicitly, or generate algorithmically, all possible solid codes of certain types [2], [7].

In [2], and then in [5], the authors presented constructions of binary solid codes (subsets of $a\{a,b\}^*b$) that are defined via pairs of functions h and g and have words of the form

$$a^{h(\dot{z})}b^{z_0}a^{z_1}\cdots b^{z_{2k-2}}a^{z_{2k-1}}b^{g(\dot{z})}, \tag{1}$$

such that $\dot{z}$ is the tuple $(z_0,\ldots,z_{2k-1})$ of the positive integers appearing in the *runs* $b^{z_0}, a^{z_1},\ldots$ of the above word – see the next section for definitions of technical terms. In [2] the authors showed a complete structural characterization of *all* maximal solid codes that are subsets of $a^+b^+a^+b^+$, by defining the exact requirements for the functions h and g. In [5], using again the approach of word runs, a class of finite solid codes is shown that has an asymptotically optimal information ratio and

[1]Research supported by a Discovery Research Grant of NSERC, Canada.

very simple linear encoding/decoding complexity. Moreover, some of these codes have certain good error-detecting capabilities. In [8], [9] the author presents a simple class of fixed-length solid codes that appear to be the best possible in terms of information ratio.

In this paper, we continue the systematic investigation of binary solid codes that are defined in terms of the runs of the words involved, in view of the fact, [2], that *every* binary solid code can be defined via a pair of functions h and g as shown in (1). It turns out that an explicit characterization of all maximal solid codes that are subsets of $(a^+b^+)^k$, for some integer $k \geq 2$, is quite difficult to describe. Our investigations have led to the discovery of f-words. These are words of the form shown in the expression (1), where h and g are defined explicitly via the choice of any non-decreasing and unbounded sequence f of positive integers. In particular, the sets of all f-words of the form $a^+b^+a^+b^+$ are exactly all the maximal solid codes defined in [2]. The f-words can also be used to improve the information rate of the finite solid codes defined in [5], without sacrificing their encoding/decoding complexity.

This paper is organized as follows. The next section contains the basic notions and notation used throughout the paper. Section 3 contains the definition of f-words and shows some basic properties of these words. In particular, the set of these words is an infinite overlap-free language. Section 4 uses f-words to improve the construction of the finite solid codes in [5], and Section 5 defines, again via f-words, new classes of infinite solid codes, including a class of infinite maximal solid codes that are subsets of

$$a^+b^+a^+b^+a^+b^+ \cup a^+b^+a^+b^+a^+b^+a^+b^+.$$

Finally, Section 6 contains a few concluding remarks.

2 Basic Notions and Notation

The symbol $\mathbb{N}$ denotes the set of positive integers. An *alphabet* is any finite and nonempty set of elements, which we call symbols or letters. We shall use the symbol Σ for the binary alphabet

$$\Sigma = \{a, b\}.$$

As usual, we use the notation Σ^* for the set of all words over the alphabet Σ, including the empty word λ. The length of a word u is denoted as $|u|$ and is defined to be the number of symbols occurring in u. Any set of words is called a *language*. The concatenation of two words u and v is written as uv. This notation is extended naturally to more than two words: $u_1u_2\cdots u_n$ is the concatenation of the words $u_1, u_2, \ldots, u_n$. When all the u_i's are the same, say equal to v, we write v^n for their concatenation. Obviously $u\lambda = \lambda u = u$, for all words u. If a word u is of the form xy then x is called a *prefix* of u and y is called a *suffix* of u. If x is not equal to u then it is called a proper prefix – proper suffixes are defined analogously. If u is of the form xyz then y is an *infix* of u – obviously prefixes and suffixes are special infixes of a word. A *run* of u is an infix of u of the form σ^n, for some $\sigma \in \Sigma$, such that u can be written as $x\sigma^n y$, and x does not end with σ and y does not start with σ.

Two nonempty words have an *overlap* z, if z is a nonempty word such that z is a proper prefix of one of the words and a proper suffix of the other. For example aa is an overlap of the words $aaabb$ and $ababaa$. A language L is an *overlap-free language* if no two words of L have an overlap. A language L is an *infix code*, if no L-word is a proper infix of another L-word. A language is called a *solid code* (or, code without overlaps) if it is both an infix code and an overlap-free language. If $\mathcal{C}$ is any class of languages/codes (infix, overlap-free, etc), then we say that a language/code L is a

maximal $\mathcal{C}$ language/code, if for any word w outside of L, namely $w \in \Sigma^* - L$, the set $L \cup \{w\}$ is not in the class $\mathcal{C}$.

A *tuple* is any finite sequence of positive integers written as

$$\dot{z} = (z_0, \ldots, z_{k-1}). \tag{2}$$

We use the dot notation $\dot{z}$ for tuples. The empty tuple is ( ). The length of $\dot{z}$ is the number of components in $\dot{z}$ – this is equal to k in the tuple shown in (2). The set of all tuples of length k is denoted as $\mathbb{N}^k$ and the set of all tuples as $\mathbb{N}^*$. The *size* of the tuple $\dot{z}$ is $\sum \dot{z} = \sum_{i=0}^{k-1} z_i$. We also define $\max \dot{z} = \max\{z_i \mid i = 0, \ldots, k-1\}$.

Consider any unbounded and non-decreasing function $f : \mathbb{N} \to \mathbb{N}$ (as usual, non-decreasing means $f(i) \leq f(i+1)$ for all $i \in \mathbb{N}$). The *near inverse* of f is the function $\hat{f} : \mathbb{N} \to \mathbb{N}$ such that

$$\hat{f}(j) = \min\{i \mid f(i) > j\}.$$

The term near-inverse arises from the fact that, if φ is any real non-decreasing function with $\varphi(i) = f(i)$ for all $i \in \mathbb{N}$, then

$$\hat{f}(j) = 1 + \lfloor \varphi^{-1}(i) \rfloor.$$

In [4], it is shown that, like f, the near-inverse $\hat{f}$ is unbounded and non-decreasing. A basic property of near-inverses is

$$\text{For all } i, j \in \mathbb{N}: \quad f(i) > j \text{ if and only if } i \geq \hat{f}(j). \tag{3}$$

When $f(i) = 2i$ it is easy to see that $\hat{f}(j) = 1 + \lfloor j/2 \rfloor$. We shall also use the identity function $\delta(i) = i$ and its near-inverse $\hat{\delta}(j) = 1 + j$.

3 f-Words

We define a certain type of words, called f-words, with the aim that any two such words are overlap free. The definition of these words is inspired from the constructions of solid codes in [2] and [5]. In particular, in [2] it is noted that every binary solid code that is a subset of $a\Sigma^*b$ is defined using a set of tuples $\mathbb{T}$ of even length, and two functions $h, g : \mathbb{T} \to \mathbb{N}$ such that each word of the code is of the form shown in (1), where $\dot{z}$ is a tuple in $\mathbb{T}$ of some length $2k$. In [5], the set $\mathbb{T}$ consists of all tuples $\dot{z}$ such that $\sum \dot{z} = n$, for some fixed word length n, and

$$h(\dot{z}) = \max\{f(z_{2i}) : i = 0, \ldots, k-1\} \text{ and } g(\dot{z}) = \max\{\hat{f}(z_{2i-1}) : i = 1, \ldots, k\},$$

where $f : \mathbb{N} \to \mathbb{N}$ is any non-decreasing and unbounded function. Thus $h(\dot{z})$ is the f-value on the length of the longest a-run, and $g(\dot{z})$ is the f-value on the length of the longest b-run in the word $b^{z_0}a^{z_1} \cdots b^{z_{2k-2}}a^{z_{2k-1}}$.

For reasons of notational convenience we view a tuple $\dot{z}$ of even length as consisting of two interleaved tuples $\dot{x}$ and $\dot{y}$ of the same length k, where we use the components of the second one in reverse order. More specifically, for any two tuples $\dot{x}, \dot{y}$ of some length k, we define the word

$$\text{wd}(\dot{x}, \dot{y}) = b^{x_0}a^{y_{k-1}}b^{x_1} \cdots a^{y_{k-i}}b^{x_i} \cdots a^{y_1}b^{x_{k-1}}a^{y_0},$$

where we assume that $\text{wd}(\dot{x}, \dot{y})$ is empty if $k = 0$.

In the sequel we assume that $f : \mathbb{N} \to \mathbb{N}$ is any non-decreasing and unbounded function. We use this function to define, for any tuples $\dot{x}$ and $\dot{y}$, the quantities $f_{\dot{x},\dot{y}} = h(\dot{x}, \dot{y})$ and $\hat{f}_{\dot{y},\dot{x}} = g(\dot{x}, \dot{y})$ that will be used to make overlap-free words as shown in the expression (1). One possibility is to use

$f_{\dot{x},\dot{y}} = f(\max \dot{x})$ and $\hat{f}_{\dot{y},\dot{x}} = \hat{f}(\max \dot{y})$, as in [5]. However, we can do better than that. Instead of giving immediately the definitions for $f_{\dot{x},\dot{y}}$ and $\hat{f}_{\dot{y},\dot{x}}$, we provide a couple of examples that motivate the choice of our definitions. Any two (possibly equal) f-words will be of the form

$$a^{f_{\dot{x},\dot{y}}}\mathrm{wd}(\dot{x},\dot{y})b^{\hat{f}_{\dot{y},\dot{x}}}, \quad a^{f_{\dot{s},\dot{t}}}\mathrm{wd}(\dot{s},\dot{t})b^{\hat{f}_{\dot{t},\dot{s}}}. \tag{4}$$

An overlap of these words has the form $a^{f_{\dot{x},\dot{y}}}b^{\hat{f}_{\dot{t},\dot{s}}}$ (case "$i = -1$"), or

$$a^{f_{\dot{x},\dot{y}}}b^{x_0}(a^{y_{k-1}}b^{x_1})\cdots(a^{y_{k-i}}b^{x_i})a^{y_{k-i-1}}b^{\hat{f}_{\dot{t},\dot{s}}} = a^{f_{\dot{x},\dot{y}}}b^{s_{l-i-1}}(a^{t_i}b^{s_{l-i}})\cdots(a^{t_1}b^{s_{l-1}})a^{t_0}b^{\hat{f}_{\dot{t},\dot{s}}}, \tag{5}$$

where $\hat{f}_{\dot{t},\dot{s}} \leq x_{i+1}$ and $f_{\dot{x},\dot{y}} \leq t_{i+1}$, and for $i \geq 0$ the tuples match as follows $(x_0,\ldots,x_i) = (s_{l-i-1},\ldots,s_{l-1})$ and $(y_{k-1},\ldots,y_{k-i-1}) = (t_i,\ldots,t_0)$. In the case of $i = -1$, the overlap is prevented when we *require* that always $f(x_0) \leq f_{\dot{x},\dot{y}}$ and $\hat{f}(t_0) \leq \hat{f}_{\dot{t},\dot{s}}$. Indeed, in this case, we get $f(x_0) \leq t_0$ and $\hat{f}(t_0) \leq x_0$, and by the property of near-inverses, $x_0 < \hat{f}(t_0)$; a contradiction.

When the overlap in (5) is equal to $a^{f_{\dot{x},\dot{y}}}b^{x_0}a^{y_{k-1}}b^{\hat{f}_{\dot{t},\dot{s}}}$, we have $f_{\dot{x},\dot{y}} \leq t_1$ and $\hat{f}_{\dot{t},\dot{s}} \leq x_1$, and $x_0 = s_{l-1}$, $t_0 = y_{k-1}$. Then, if $y_{k-1} \geq f(x_1)$, we have $\hat{f}(t_0) = \hat{f}(y_{k-1}) > x_1 \geq \hat{f}_{\dot{t},\dot{s}}$, which already contradicts our first requirement that $\hat{f}_{\dot{t},\dot{s}} \geq \hat{f}(t_0)$. Analogously, $\hat{f}(t_1) \leq s_{l-1}$ leads to a contradiction. Hence, $y_{k-1} < f(x_1)$ and $\hat{f}(t_1) > s_{l-1}$. To get a contradiction and prevent the overlap we *require* that $f_{\dot{x},\dot{y}} \geq f(x_1)$ if $y_{k-1} < f(x_1)$ and, analogously, $\hat{f}_{\dot{t},\dot{s}} \geq \hat{f}(t_1)$ if $\hat{f}(t_1) > s_{l-1}$. Indeed, $f_{\dot{x},\dot{y}} \geq f(x_1)$ implies $f_{\dot{x},\dot{y}} \geq f(\hat{f}_{\dot{t},\dot{s}})$, and $\hat{f}_{\dot{t},\dot{s}} \geq \hat{f}(t_1)$ implies $\hat{f}_{\dot{t},\dot{s}} \geq \hat{f}(f_{\dot{x},\dot{y}})$, which in turn gives $f(\hat{f}_{\dot{t},\dot{s}}) > f_{\dot{x},\dot{y}}$.

For longer overlaps of the form shown in (5) the requirements for $f_{\dot{x},\dot{y}}$ and $\hat{f}_{\dot{t},\dot{s}}$ become more complex. Next we give the definition for these quantities. It is based on the following sets of indices

$$I_f(\dot{x},\dot{y}) = \{0\} \cup \{i \mid 0 < i < k,\ \max(y_{k-1},\ldots,y_{k-i}) < f(x_i)\} \tag{6}$$

$$I_{\hat{f}}(\dot{y},\dot{x}) = \{0\} \cup \{j \mid 0 < j < k,\ \hat{f}(y_j) > \max(x_{k-j},\ldots,x_{k-1})\} \tag{7}$$

For example, using the function $f(i) = 2i$, with $\hat{f}(j) = 1 + \lfloor j/2 \rfloor$, and the word

$$\mathrm{wd}(\dot{x},\dot{y}) = ba^3bab^2a^4b^2a^2$$

we see that $I_f(\dot{x},\dot{y}) = \{0,2\}$ and $I_{\hat{f}}(\dot{y},\dot{x}) = \{0,1\}$.

We note that if $i \in I_f(\dot{x},\dot{y}) - \{0\}$ then none of $k-1,\ldots,k-i$ belongs to $I_{\hat{f}}(\dot{y},\dot{x})$. This is because $y_{k-1},\ldots,y_{k-i} < f(x_i)$ and, by Property 3 of near-inverses, $\hat{f}(y_{k-1}),\ldots,\hat{f}(y_{k-i}) \leq x_i$. So we cannot have $\hat{f}(y_{k-j}) > x_j,\ldots,x_{k-1}$ for any $j \in \{1,\ldots,i\}$.

Definition 1 *For any non-decreasing and unbounded function $f : \mathbb{N} \to \mathbb{N}$, and tuples $\dot{x}$ and $\dot{y}$ of the same length k,*

$$f_{\dot{x},\dot{y}} = \max\{f(x_i) \mid i \in I_f(\dot{x},\dot{y})\} \tag{8}$$

$$\hat{f}_{\dot{y},\dot{x}} = \max\{\hat{f}(y_j) \mid j \in I_{\hat{f}}(\dot{y},\dot{x})\} \tag{9}$$

An f-word is any word of the form

$$\mathrm{wd}_f(\dot{x},\dot{y}) = a^{f_{\dot{x},\dot{y}}}\,\mathrm{wd}(\dot{x},\dot{y})\,b^{\hat{f}_{\dot{y},\dot{x}}} = a^{f_{\dot{x},\dot{y}}}\,b^{x_0}a^{y_{k-1}}b^{x_1}\cdots a^{y_{k-i}}b^{x_i}\cdots a^{y_1}b^{x_{k-1}}a^{y_0}\,b^{\hat{f}_{\dot{y},\dot{x}}}.$$

The language of all f-words is

$$L_f = \{\,\mathrm{wd}_f(\dot{x},\dot{y}) \mid \dot{x},\dot{y} \in \mathbb{N}^k,\ \textit{for some } k > 0\,\}.$$

Using again the above word $\mathrm{wd}(\dot{x},\dot{y}) = ba^3bab^2a^4b^2a^2$, we see that $\mathrm{wd}_f(\dot{x},\dot{y}) = a^4ba^3bab^2a^4b^2a^2b^3$.

Lemma 1 *Let $\mathrm{wd}_f(\dot{x},\dot{y})$ be any f-word, and let w be any nonempty word having a^t as a longest run of a's and b^s as a longest run of b's, with $s,t>0$.*

1. *If w is a proper prefix of $\mathrm{wd}_f(\dot{x},\dot{y})$ then $\hat{f}(t) > s$.*
2. *If w is a proper suffix of $\mathrm{wd}_f(\dot{x},\dot{y})$ then $f(s) > t$.*

Proof. We only prove the first statement, as the second one follows by symmetry. We argue by contradiction, assuming that $\hat{f}(t) \leq s$ and, therefore, $t < f(s)$. As $a^{f_{\dot{x},\dot{y}}}$ is a run in w, we have $f_{\dot{x},\dot{y}} \leq t$. Hence,

$$f_{\dot{x},\dot{y}} < f(s).$$

We continue by distinguishing two cases. First, $s \leq x_i$ for some index i. Then, as f is non-decreasing, $f(x_i) > t$, so $f(x_i)$ must be greater than all the y_j's appearing in w. This implies that $i \in I_f(\dot{x},\dot{y})$ and, therefore, $f_{\dot{x},\dot{y}} \geq f(x_i) \geq f(s)$, which is a contradiction. In the second case, we assume that $s > \max \dot{x}$, which implies that w ends in the run $b^{\hat{f}_{\dot{y},\dot{x}}}$ of $\mathrm{wd}_f(\dot{x},\dot{y})$, and b^s is part (in fact a proper prefix) of $b^{\hat{f}_{\dot{y},\dot{x}}}$. Hence,

$$\hat{f}_{\dot{y},\dot{x}} > s \geq \hat{f}(t).$$

As $\hat{f}_{\dot{y},\dot{x}} = \hat{f}(y_j)$ for some index j, and a^{y_j} is a run of w, we have that $\hat{f}_{\dot{y},\dot{x}} \leq \hat{f}(t)$, which again is a contradiction. ■

Theorem 1 *For every non-decreasing and unbounded function f, the language L_f of all f-words is an overlap-free language.*

Proof. If w is a proper prefix and proper suffix of two f-words, then a contradiction arises using the above lemma and the properties of near-inverses. Hence, L_f is overlap-free. ■

A natural question that arises here is whether L_f is a maximal overlap-free language. As shown next, this is not the case. However, every word w outside of L_f that has no overlap with any L_f-word must be a proper prefix, or a proper suffix of some L_f-word. For example, for $f(i) = 2i$, with $\hat{f}(j) = 1 + \lfloor j/2 \rfloor$, we have the following

$$w = a^6ba^2b^3 \notin L_f,\ \ a^2ba^2b^2 \in L_f,\ \ wa^4b^2a^3b^3 \in L_f.$$

Obviously, the word w itself is overlap-free. If a proper nonempty prefix, say v, of w is a proper suffix of some L_f-word then the word v is also a proper prefix of the L_f-word $wa^4b^2a^3b^3$, which is impossible. Similarly, if a proper nonempty suffix of w is a proper prefix of some f-word $a^{f_{\dot{x},\dot{y}}}\mathrm{wd}(\dot{x},\dot{y})b^{\hat{f}_{\dot{y},\dot{x}}}$ then this suffix must be in

$$\{a^iba^2b^3, a^jb^3 \mid i = 1,\ldots,5;\ j = 1,2\},$$

which implies that $a^{f_{\dot{x},\dot{y}}} = a^i$, or $a^{f_{\dot{x},\dot{y}}} = a^j$, as shown above and, therefore, $f_{\dot{x},\dot{y}} < 6$. On the other hand, it must be that $f_{\dot{x},\dot{y}} \geq f(3) = 6$, which is again impossible. Hence, $L_f \cup \{w\}$ is an overlap-free language.

Proposition 1 *Let w be any word in $\Sigma^* - L_f$. Then w has an overlap with some word in L_f, or w is a proper prefix, or proper suffix, of some word in L_f.*

Proof. Consider any word w in $\Sigma^* - L_f$, and assume for the sake of contradiction that w has no overlap with any word in L_f, and that w is neither a prefix nor a suffix of any word in L_f.

If w starts with b or ends with a then w has an overlap with every word in L_f; a contradiction. If w is of the form $a^t b^s$ then we consider the word $u = a^{f(s)} b^s a^t b^{\hat{f}(t)} \in L_f$, and we observe that, as $a^t b^{\hat{f}(t)}$ is not a proper prefix of w and w is not a proper prefix of u, we must have $\hat{f}(t) > s$. Similarly, we must have $f(s) > t$, which leads to a contradiction by the properties of near-inverses.

Now suppose that w is of the form

$$w = a^{y_m} b^{x_0} a^{y_{m-1}} b^{x_1} \cdots b^{x_{m-1}} a^{y_0} b^{x_m},$$

for some $m \geq 1$. We show that, for all $i = 0, \ldots, m$,

$$x_i < \max\{\hat{f}(y_m), \ldots, \hat{f}(y_{m-i})\}. \tag{10}$$

Indeed, assume that there is an index i such that $x_i > \hat{f}(y_m), \ldots, \hat{f}(y_{m-i})$, and consider the word

$$u = a^{f_{\dot{s},i}} b^{x_0} a^{y_m} b^{x_0} \cdots a^{y_{m-i+1}} b^{x_{i-1}} a^{y_{m-i}} b^{\hat{f}_{\dot{t},\dot{s}}} \in L_f,$$

where $\dot{s} = (x_0, x_0, \ldots, x_{i-1})$ and $\dot{t} = (y_m, \ldots, y_{m-i})$. Then the word

$$z = a^{y_m} b^{x_0} \cdots a^{y_{m-i+1}} b^{x_{i-1}} a^{y_{m-i}} b^{\hat{f}_{\dot{t},\dot{s}}}$$

is a proper suffix of u. If $i < m$, or $i = m$ and $\hat{f}_{\dot{t},\dot{s}} < x_m$ then z is an overlap of w and u, which is a contradiction. If $i = m$ and $\hat{f}_{\dot{t},\dot{s}} = x_m$ then w is a proper suffix of u, which is again a contradiction. Hence, (10) holds. By symmetry, we also get that, for all $j = 0, \ldots, m$,

$$y_{m-j} < \max\{f(x_j), \ldots, f(x_m)\}. \tag{11}$$

Now it follows that $\max \dot{x} < \hat{f}(y_{m-j})$, for some index j, which implies $f(\max \dot{x}) \leq y_{m-j}$. Also, by (11), we have $y_{m-j} < f(\max \dot{x})$, which is a contradiction, as required. ■

We close this section by proving an interesting property of an f-word.

Proposition 2 *For every f-word $\mathrm{wd}_f(\dot{x}, \dot{y})$ we have that*

$$(f_{\dot{x},\dot{y}} = f(\max \dot{x}) > \max \dot{y} \;\; \textit{and} \;\; \hat{f}_{\dot{y},\dot{x}} \leq \max \dot{x}) \;\; \textit{OR} \;\; (\hat{f}_{\dot{y},\dot{x}} = \hat{f}(\max \dot{y}) > \max \dot{x} \;\; \textit{and} \;\; f_{\dot{x},\dot{y}} \leq \max \dot{y})$$

Proof. First we show that one of $f_{\dot{x},\dot{y}} > \max \dot{y}$ and $\hat{f}_{\dot{y},\dot{x}} > \max \dot{x}$ must hold, using contradiction. So assume $f_{\dot{x},\dot{y}} \leq \max \dot{y}$ and $\hat{f}_{\dot{y},\dot{x}} \leq \max \dot{x}$. We can write

$$\mathrm{wd}_f(\dot{x}, \dot{y}) = a^{f_{\dot{x},\dot{y}}} w_1 \sigma_1^{r_1} w_2 \sigma_2^{r_2} w_3 b^{\hat{f}_{\dot{y},\dot{x}}},$$

with $\{\sigma_1^{r_1}, \sigma_2^{r_2}\} = \{a^{\max \dot{y}}, b^{\max \dot{x}}\}$. Then, as $a^{f_{\dot{x},\dot{y}}} w_1 \sigma_1^{r_1} w_2 \sigma_2^{r_2}$ is a proper prefix of $\mathrm{wd}_f(\dot{x}, \dot{y})$, we have $\hat{f}(\max \dot{y}) > \max \dot{x}$, which implies $\max \dot{y} \geq f(\max \dot{x})$. Now, as $\sigma_1^{r_1} w_2 \sigma_2^{r_2} w_3 b^{\hat{f}_{\dot{y},\dot{x}}}$ is a proper suffix of $\mathrm{wd}_f(\dot{x}, \dot{y})$, we have $f(\max \dot{x}) > \max \dot{y}$, which is a contradiction.

Now we show that $f_{\dot{x},\dot{y}} \geq \max \dot{y}$ implies $f_{\dot{x},\dot{y}} = f(\max \dot{x})$ – then, by symmetry, we also have that $\hat{f}_{\dot{y},\dot{x}} \geq \max \dot{x}$ implies $\hat{f}_{\dot{y},\dot{x}} = \hat{f}(\max \dot{y})$. Assume $f_{\dot{x},\dot{y}} \geq \max \dot{y}$. Then, $a^{f_{\dot{x},\dot{y}}}$ is a longest run of a's in the word $\mathrm{wd}_f(\dot{x}, \dot{y})$, and this word has a proper prefix of the form $a^{f_{\dot{x},\dot{y}}} w b^{\max \dot{x}}$. By Lemma 1, $\hat{f}(f_{\dot{x},\dot{y}}) > \max \dot{x}$ and, therefore, $f_{\dot{x},\dot{y}} \geq f(\max \dot{x})$. Also, as always $f_{\dot{x},\dot{y}} \leq f(\max \dot{x})$, we have that $f_{\dot{x},\dot{y}} = f(\max \dot{x})$.

The statement of the proposition will follow easily when we show that $f_{\dot{x},\dot{y}} > \max \dot{y}$ implies $\hat{f}_{\dot{y},\dot{x}} \leq \max \dot{x}$. Indeed this can be easily verified using the previous facts. ■

4 Finite solid codes based on f-words

In [5], the authors showed that the language

$$K_n = \{a^{\max \dot{x}} \mathrm{wd}(\dot{x}, \dot{y}) b^{1+\max \dot{y}} \mid \mathrm{wd}(\dot{x}, \dot{y}) = bua, \text{ for some word } u \text{ of length } n\}$$

is a solid code. Moreover, the information rate $\mathrm{rt}(K_n)$ of K_n tends to 1, as $n \to \infty$ – recall the *information rate* $\mathrm{rt}(C)$ of a finite code C is the quantity $\log |C|/\bar{\ell}(C)$, where $\bar{\ell}(C)$ is the average word length of C. Obviously, this code contains exactly 2^n codewords. Moreover, it is evident that every binary word u of length n can be encoded to a unique codeword in linear time – the time to identify the largest runs of b's and a's in bua. Here we use the same idea and the identity function $\delta(i) = i$, whose near-inverse is $\hat{\delta}(j) = 1 + j$, to define a new sequence of solid codes that has a better information rate without sacrificing the order of magnitude of the encoding complexity.

Theorem 2 *The language*

$$F_n = \{a^{\delta_{\dot{x},\dot{y}}} \mathrm{wd}(\dot{x}, \dot{y}) b^{\hat{\delta}_{\dot{y},\dot{x}}} \mid \mathrm{wd}(\dot{x}, \dot{y}) = bua, \textit{ for some word } u \textit{ of length } n\}$$

is a solid code of cardinality 2^n such that $\mathrm{rt}(F_n) \geq \mathrm{rt}(K_n)$ and, therefore, $\mathrm{rt}(F_n) \to 1$, as $n \to \infty$. Moreover, there is an algorithm that encodes (maps) every binary word of length n to a unique codeword of F_n in linear time.

Proof. As $F_n \subseteq L_\delta$, the language F_n is overlap-free. Moreover, it is not difficult to see that no word of F_n is a proper infix of another word in F_n. Hence, the language is a solid code. The statement about the information rate of F_n follows easily from the fact that $\delta_{\dot{x},\dot{y}} \leq \max \dot{x}$ and $\hat{\delta}_{\dot{y},\dot{x}} \leq 1 + \max \dot{y}$. Regarding the encoding complexity, it is sufficient to show that $\delta_{\dot{x},\dot{y}}$ can be computed from $\mathrm{wd}(\dot{x}, \dot{y})$ in linear time. Recall, $\delta_{\dot{x},\dot{y}}$ is the largest value $\delta(x_r) = x_r$ over all r with $r = 0$ or $\max(y_{k-1}, \ldots, y_{k-r}) < \delta(x_r)$. The algorithm reads the runs of $\mathrm{wd}(\dot{x}, \dot{y})$ left to right and uses the variables X and Y. The variable Y keeps track of the current largest length of the a-runs seen so far, and is initially zero. The variable X keeps track of the largest length of a b-run that is also greater than Y. Once the first run b^{x_0} has been read, X is set to x_0 and, then, the following loop is performed

while (there are another two runs a^i, b^j) {
  if ($Y < i$) then $Y = i$;
  if ($X < j$ AND $Y < j$) then $X = j$;
}

The final value of the variable X is equal to $\delta_{\dot{x},\dot{y}}$. ■

The improvement $\mathrm{rt}(F_n) \geq \mathrm{rt}(K_n)$ in the above result could be nontrivial. In particular there are many pairs of tuples $(\dot{x}, \dot{y})$ for which the corresponding words in F_n are strictly shorter than those in K_n. To see this consider all tuples $\dot{x}, \dot{y}$ of some fixed length k such that

$$\sum \dot{x} + \sum \dot{y} = n + 2, \;\; y_0 = 1, \;\; \max \dot{x} = x_{k-1} > \max \dot{y}.$$

Then $\delta_{\dot{x},\dot{y}} = x_{k-1}$ and $\hat{\delta}_{\dot{y},\dot{x}} = 2$. Moreover, for all of these tuples we have

$$a^{x_{k-1}} \mathrm{wd}(\dot{x}, \dot{y}) b^{1+\max \dot{y}} \in K_n \;\text{ and }\; a^{x_{k-1}} \mathrm{wd}(\dot{x}, \dot{y}) b^2 \in F_n.$$

Clearly, each of these F_n-words is $(\max \dot{y} - 1)$ bits shorter than the corresponding K_n-word. For example, for $n = 16$ and $k = 3$, if we consider all tuples $\dot{x}, \dot{y}$ as above such that $\max \dot{x} = x_{k-1} = 5$, $y_0 = 1$, and $\max \dot{y} = 4$ we get the equation

$$x_0 + x_1 + y_1 + y_2 = 12, \;\; \text{with} \;\; \max(x_0, x_1) \leq 5, \; \max(y_1, y_2) = 4.$$

This equation has 29 solutions, which implies that there are at least 29 words in F_{16} with each one being 3 bits shorter than the corresponding word in K_{16}.

For a finite nonempty language L, we use $\ell(L)$ to denote the length of the longest word in L. We recall from [5] that when the code K_n is restricted as follows

$$K_C = \{a^{\max \dot{x}} \mathrm{wd}(\dot{x}, \dot{y}) b^{1+\max \dot{y}} \mid \mathrm{wd}(\dot{x}, \dot{y}) = bua, \text{ for some word } u \in C\},$$

where C is any code of length n that is error-detecting for the combinatorial channel[2] $\sigma(1, n)$, then K_C^* is error-detecting for the channel that permits at most one substitution, insertion, or deletion error in any segment of length $\ell(K_C)$ of the transmitted message. This result is shown in Theorem 7.2 of [5]. Using essentially the same proof as in that theorem, it turns out that when

$$F_C = \{a^{\delta_{\dot{x},\dot{y}}} \mathrm{wd}(\dot{x}, \dot{y}) b^{\hat{\delta}_{\dot{y},\dot{x}}} \mid \mathrm{wd}(\dot{x}, \dot{y}) = bua, \text{ for some word } u \in C\}$$

the language F_C^* is also error-detecting for the channel that permits at most one substitution, insertion, or deletion error in any segment of length $\ell(F_C)$ of the transmitted message.

5 Infinite maximal solid codes based on f-words

In [2] the authors obtained a complete structural characterization of *all* infinite maximal solid codes that are subsets of $a^+b^+a^+b^+$. In particular, for any non-decreasing and unbounded function $f : \mathbb{N} \to \mathbb{N}$, the language

$$C_{f,1} = \{a^{f(x)} b^x a^y b^{\hat{f}(y)} \mid x, y \in \mathbb{N}\}$$

is a maximal solid code and, conversely, every maximal solid code that is a subset of $a^+b^+a^+b^+$ must be equal to $C_{f,1}$, for some function f. It should be clear that $C_{f,1} \subseteq L_f$. A natural question then is to consider, for each $k \geq 1$, the language

$$C_{f,k} = \{a^{f_{\dot{x},\dot{y}}} b^{x_0} (a^{y_{k-1}} b^{x_1}) \cdots (a^{y_1} b^{x_{k-1}}) a^{y_0} b^{\hat{f}_{\dot{y},\dot{x}}} \mid \dot{x}, \dot{y} \in \mathbb{N}^k\}.$$

As this language also is a subset of L_f, it is an overlap-free language. Moreover, it is not difficult to see that $C_{f,k}$ is an infix code and, therefore, a solid code. It turns out, however, that $C_{f,k}$ is not a maximal solid code, and it appears that a complete characterization of maximal solid codes containing $C_{f,k}$ is rather difficult to describe. In the sequel we show that, for any choice of the function f, the language

$$K_{f,2} = C_{f,2} \cup \{\mathrm{wd}_f(\dot{x}, \dot{y}) \mid \dot{x}, \dot{y} \in \mathbb{N}^3, f(x_1) > y_2, \hat{f}(y_1) > x_2\}$$

is a maximal solid code. Again, as this language is a subset of L_f, it is an overlap-free language. Moreover, $K_{f,2}$ is an infix code. Indeed, if there were a word in $K_{f,2}$ that is a proper infix of another word in $K_{f,2}$ then clearly these words must be of the form

$$a^{f_{\dot{s},\dot{t}}} b^{s_0} a^{t_1} b^{s_1} a^{t_0} b^{\hat{f}_{\dot{t},\dot{s}}}, \quad a^{f_{\dot{x},\dot{y}}} b^{x_0} a^{y_2} b^{x_1} a^{y_1} b^{x_2} a^{y_0} b^{\hat{f}_{\dot{y},\dot{x}}}. \tag{12}$$

One possibility is that the shorter word is an infix of $a^{f_{\dot{x},\dot{y}}} b^{x_0} a^{y_2} b^{x_1} a^{y_1} b^{x_2}$ such that $y_1 = t_0$ and $x_2 \geq \hat{f}_{\dot{t},\dot{s}}$.Then, $\hat{f}(t_0) > \hat{f}_{\dot{t},\dot{s}}$, which is impossible by the definition of $\hat{f}_{\dot{t},\dot{s}}$. Similarly one shows that

[2]This channel permits at most one substitution error in any segment of length n of the transmitted message. A language L is error-detecting for a channel γ, if no word of $L \cup \{\lambda\}$ (the transmitted message) results in *another* word of $L \cup \{\lambda\}$ using the errors permitted by the channel γ – see [3], [6] for details on these concepts.

the shorter word cannot be an infix of the last six runs of the longer one. Hence, we have that $K_{f,2}$ is a solid code.

Our aim now is to show that $K_{f,2}$ is a maximal solid code. We need to establish a special notation. For any word $w = \mathrm{wd}(\dot{x}, \dot{y})$, with $\dot{x}, \dot{y}$ tuples of length m, we define the *pattern* $\tilde{w} = P_0 P_1 \cdots P_{m-1}$ of w as follows. Each P_i is either the symbol F or G, depending on whether $\hat{f}(y_{m-i}) > x_i$ or $y_{m-i} < f(x_i)$, respectively. Equivalently, P_i is either the symbol F or G, depending on whether $y_{m-i} \geq f(x_i)$ or $\hat{f}(y_{m-i}) \leq x_i$, respectively. For example, for $f(i) = i$ and $\hat{f}(j) = 1 + j$, we have that the pattern of $a^3b^2a^2b^3ab^2$ is FGG. It is easy to see that the pattern of any f-word starts with F and ends with G. Moreover, the pattern of any word in $(K_{f,2} - a^+b^+a^+b^+a^+b^+)$ is equal to FGFG, and the pattern of any word in $K_{f,2} \cap a^+b^+a^+b^+a^+b^+$ is FGG or FFG. Note that, with this notation, it is immediate to see that $K_{f,2}$ is an infix code.

We write $P_i > P_{i+1}$ when $\hat{f}(y_{m-i}) > x_{i+1}$, and $P_i < P_{i+1}$ when $y_{m-i} < f(x_{i+1})$. Obviously, by the properties of near-inverses, exactly one of $P_i > P_{i+1}$ and $P_i < P_{i+1}$ is true, for each index i. In the former case, we write the predicate

$$P_0 \cdots (P_i > P_{i+1}) \cdots P_{m-1} \in \vec{w},$$

and we say that $P_0 \cdots (P_i > P_{i+1}) \cdots P_{m-1}$ is an enhanced pattern of w. We use a similar notation for the latter case. For example, the word $a^3b^2a^2b^3ab^2$ considered above has (F>G)G as an enhanced pattern.

Lemma 2 *Let u be any word in $K_{f,2}$ and w be any word in $a\Sigma^*b$.*

1. *If $\tilde{u} = \mathrm{FFG}$ then $\mathrm{F(F < G)} \in \vec{u}$.*
2. *If $\tilde{u} = \mathrm{FGG}$ then $\mathrm{(F > G)G} \in \vec{u}$.*
3. *If $\tilde{u} = \mathrm{FGFG}$ then $\mathrm{(F > G)(F < G)} \in \vec{u}$.*
4. *If $\tilde{w} = \mathrm{G}$ then w begins with a nonempty suffix of $K_{f,2}$.*
5. *If $\tilde{w} = \mathrm{F}$ then w ends with a nonempty prefix of $K_{f,2}$.*
6. *If $\mathrm{F(F < G)} \in \vec{w}$, or $\mathrm{(F > G)G} \in \vec{w}$, or $\mathrm{(F > G)(F < G)} \in \vec{w}$ then w must contain a word of $K_{f,2}$.*
7. *If $\mathrm{(F > G)(F > G)} \in \vec{w}$ then w must end with a nonempty proper prefix of $K_{f,2}$.*
8. *If $\mathrm{(F < G)(F < G)} \in \vec{w}$ then w must begin with a nonempty proper suffix of $K_{f,2}$.*
9. *If $\mathrm{(F < G)G} \in \vec{w}$ then w must begin with a nonempty proper suffix of $K_{f,2}$.*
10. *If $\mathrm{F(F > G)} \in \vec{w}$ then w must end with a nonempty proper prefix of $K_{f,2}$.*

Proof. The proof of each statement is based on the definitions of pattern and enhanced pattern, using the form of the words in $K_{f,2}$ as shown in (12). For the first statement, if $\mathrm{F(F > G)} \in \vec{u}$ then $\hat{f}(t_1) > \hat{f}_{\dot{t},\dot{s}}$, which implies that $\hat{f}_{\dot{t},\dot{s}} = \hat{f}(t_0)$ and $\hat{f}(t_1) \leq s_1$; hence, the pattern of u must be FGG, which is a contradiction. Thus, $\mathrm{F(F < G)} \in \vec{u}$. The second statement is symmetric. For the third statement, one can use the same arguments as before. For the fourth statement, w is of the form $a^q b^p$ with $q < f(p)$; hence, $\hat{f}(q) \leq p$. The statement follows when we note that $a^q b^{\hat{f}(q)}$ is a suffix of $a^{f(1)}baba^q b^{\hat{f}(q)} \in K_{f,2}$. The fifth statement is symmetric to the fourth one.

For the sixth statement we only show the case $(\mathrm{F} > \mathrm{G})(\mathrm{F} < \mathrm{G}) \in \vec{w}$. In this case,

$$w = \mathrm{wd}(\dot{x}, \dot{y}) = a^{y_3} b^{x_0} a^{y_2} b^{x_1} a^{y_1} b^{x_2} a^{y_0} b^{x_3},$$

with $\dot{x}$ and $\dot{y}$ of length 4. By the pattern of w we must have $y_3 \geq f(x_0)$, $y_3 \geq f(x_1)$, $\hat{f}(y_1) \leq x_3$, $\hat{f}(y_0) \leq x_3$. Hence, the word $\mathrm{wd}_f(\dot{x}, \dot{y}) \in K_{f,2}$ is contained in w. For the seventh statement, we note that $y_1 \geq f(x_2)$, $y_1 \geq f(x_3)$, and $y_0 < f(x_3)$. Hence, the word $a^{\max(f(x_2), f(x_3))} b^{x_2} a^{y_0} b^{x_3} a^1 b^{\hat{f}(1)}$ is in $K_{f,2}$, and w ends with a proper and nonempty prefix of that word, as required. The eighth statement is symmetric to the seventh one. The last two statements can be dealt with using similar arguments. ■

Theorem 3 *For every non-decreasing and unbounded function $f : \mathbb{N} \to \mathbb{N}$, the language $K_{f,2}$ is a maximal solid code.*

Proof. We use contradiction. So, assume there is a word $w \in a\Sigma^* b - K_{f,2}$ such that $K_{f,2} \cup \{w\}$ is a solid code. First we note that, by Lemma 2(4-5), the pattern $\tilde{w}$ must start with F and end with G. Now we claim that $\tilde{w}$ can contain neither FF nor GG. With this claim $\tilde{w}$ must be of the form $\mathrm{FG} \cdots \mathrm{FG}$, and then by Lemma 2(6), we must have that one of $(\mathrm{F} > \mathrm{G}) \cdots (\mathrm{F} > \mathrm{G})$ and $(\mathrm{F} < \mathrm{G}) \cdots (\mathrm{F} < \mathrm{G})$ must be an enhanced pattern of w. The desired contradiction obtains when we use Lemma 2(7-8). Now we prove the FF case of our claim in the following paragraph (the other case is symmetric).

Assume that $\tilde{w}$ contains FF. As it ends with G, it must also contain FFG. By Lemma 2(6), w has an enhanced pattern of the form

$$\tilde{w}_1 \mathrm{F}(\mathrm{F} > \mathrm{G})\tilde{w}_2,$$

with no occurrence of FFG in $\tilde{w}_2$. Now let r be the largest value such that $(\mathrm{F} > \mathrm{G})^r \tilde{w}_3$ is an enhanced pattern of w_2; hence, no enhanced pattern of w_3 can start with $(\mathrm{F} > \mathrm{G})$. Moreover, by Lemma 2(6), no enhanced pattern of w_3 can start with $(\mathrm{F} < \mathrm{G})$. Hence, $\tilde{w}_3$ cannot start with FG. By Lemma 2(10), w_2 is nonempty. If w_3 is empty then $r \geq 1$ and an enhanced pattern of w ends with $(\mathrm{F} > \mathrm{G})(\mathrm{F} > \mathrm{G})$, which is a contradiction by the above lemma. So w_3 must be nonempty and $\tilde{w}_3$ must start with F. Moreover, as $\tilde{w}_3$ contains no FFG, it must start with FG; a contradiction. ■

6 Discussion

We have presented new classes of both, finite and infinite, binary solid codes based on word runs. These utilize the new concept of f-word. Our work constitutes another step towards the explicit structural characterization of large classes of binary solid codes satisfying various good criteria. Can we find a reasonable characterization of all maximal binary solid codes that are subsets of $(a^+b^+)^k$, for any $k \geq 2$? Then, of such codes that are subsets of $(a^+b^+)^+$? Would such constructions also lead to improved finite solid codes?

References

[1] V.B. Balakirsky. Block codes for asynchronous data transmission designed from binary trees. *The Computer Journal* **45**.2 (2002), 243–248.

[2] H. Jürgensen, M. Katsura and S. Konstantinidis. Maximal solid codes. *J. Automata, Languages and Combinatorics* **6** (2001), 25–50.

[3] H. Jürgensen and S. Konstantinidis. Codes. In [10], Vol. 1, pp 511–607.

[4] H. Jürgensen and S. Konstantinidis. (Near-)inverses of sequences. *International J. of Computer Mathematics* **83**.2 (2006), 203–222.

[5] H. Jürgensen, S. Konstantinidis and N.H. Lâm. Asymptotically optimal low-cost solid codes. *Journal of Automata, Languages and Combinatorics* **9**.1 (2004), 81–102.

[6] S. Konstantinidis and A. O'Hearn. Error-detecting properties of languages. *Theoretical Computer Science* **276**.1-2 (2002), 355–375.

[7] N.H. Lâm. Finite maximal solid codes. *Theoretical Computer Science* **262**.1-2 (2001), 333–347.

[8] V.I. Levenshtein. Maximum number of words in codes without overlaps. *Probl. Inform. Transm.* **6** (1970), 355–357.

[9] V.I. Levenshtein. Combinatorial problems motivated by comma-free codes. *J. Combinatorial Designs* **12** (2004), 184–196.

[10] G. Rozenberg and A. Salomaa (eds). *Handbook of Formal Languages, Vol. 1.* Springer-Verlag, Berlin, 1997.

[11] S.S. Yu. *Languages and Codes.* Tsang Hai Publishing Co., Taichung, Taiwan, 2005.