Sufficient Conditions of Existence of Fix-Free Codes

Yekhanin Sergey
Moscow State University,
Faculty of Cybernetics & Computer Science,
Department of System Programming,
Email: gamov@cityline.ru

Abstract – Code is fix-free if no codeword is a prefix or a suffix of any other. This kind of codes has several applications [1]. For example, a file compressed by fix-free code can be decoded in forward direction and in reverse direction simultaneously, thus reducing the decoding time to half compared with decoding in one direction only. The problem of existence for fix-free codes was studied by several authors [2, 3]. In this paper we improve the best-known sufficient conditions on existence of fix-free codes by a new explicit construction. We also discuss the well-known Kraft-type conjecture on the existence of fix-free codes basing on the results obtained by computer checking.

1 Problem Statement

Let $C(k_1, k_2, \ldots, k_n)$ denote a binary variable-length code C with k_1 codewords of length 1, k_2 codewords of length 2, ... and k_n codewords of length n. Assume without loss of generality that $k_n \neq 0$.

Let $k_1, \ldots, k_n$ be arbitrary positive integers. To simlify further presentation we denote $\sum_{i=1}^{n} \frac{k_i}{2^i}$ by $\chi(k_1, \ldots, k_n)$.

Recall that a code is called prefix-free {resp. suffix-free}, if no codeword is beginning {resp. ending} of another one. Code C, which is simultaneously prefix-free and suffix-free is called fix-free.

Our goal is to develop sufficient conditions on the existence of fix-free codes. The well-known Kraft-type conjecture [2] is that for any values of parameters $k_1, k_2, \ldots, k_n$, such that

$$\chi(k_1, k_2, \ldots, k_n) \leq \frac{3}{4}, \tag{1}$$

there exists a fix-free code $C(k_1, k_2, \ldots, k_n)$

The next section gives a short overview of particular cases in which the conjecture is proved.

The following lemma [2] shows that, if true, bound (1) is the best possible.

Lemma 1: For any $\varepsilon > 0$ there exist parameters $k_1, k_2, \ldots, k_n$ such that $\chi(k_1, k_2, \ldots, k_n) \leq \frac{3}{4} + \varepsilon$ and there exists no fix-free code $C(k_1, k_2, \ldots, k_n)$.

2 Statement of Results

In this section we formulate the known sufficient conditions on existence of fix-free codes. Proofs of the next two theorems can be found in [2].

Theorem 1: If $\chi(k_1,\ldots,k_n) \leq \frac{1}{2}$ then there exists a fix-free $C(k_1,\ldots,k_n)$.

Theorem 2: Suppose that if $i < j$ and $k_i > 0$ and $k_j > 0$, then $i < 2j$. Then $\chi(k_1,\ldots,k_n) \leq \frac{3}{4}$ implies the existence of a fix-free code $C(k_1,\ldots,k_n)$.

The main result presented in current paper is formulated by the next theorem. It proves conjecture (1) for a wide class of parameters $k_1,\ldots,k_n$. Proof of theorem 3 is given in the next section.

Theorem 3: Let $k_1,k_2,\ldots,k_n$ be arbitrary nonnegative integers. Suppose that the following statements are true:

$$\chi(k_1,\ldots,k_n) \leq \frac{3}{4},$$

$$\exists p : k_1 = \ldots = k_{p-1} = 0, \text{ and } \frac{k_p}{2^p} + \frac{k_{p+1}}{2^{p+1}} \geq \frac{1}{2}.$$

This implies the existence of fix-free code $C(k_1,k_2,\ldots,k_n)$.

The following particular case of theorem 3 improves corollary 3 from [3].

Corollary 1: If $\chi(k_1,\ldots,k_n) \leq \frac{3}{4}$ and $k_1 = 1$ then there exists a fix-free code $C(k_1,k_2,\ldots,k_n)$.

Note, that all the theorems formulated above are particular cases of conjecture (1). We have applied computer programming to check the conjecture for the small values of n. Thus the following theorem was obtained.

Theorem 4: Let $k_1,\ldots,k_n$ be arbitrary nonnegative integers such that $\chi(k_1,\ldots,k_n) \leq \frac{3}{4}$ and $n \leq 10$ then there exists a fix-free code $C(k_1,\ldots,k_n)$.

One more sufficient condition of existence of fix-free codes is given in [3].

3 Proof of Theorem 3

3.1 Notations and definitions

Define p-suffix {p-prefix} of codeword w as binary vector of length p consisting of p last {p first} symbols of w.

Let $C(k_1,\ldots,k_n)$ be a fix-free code. We say that a vector $w \in \{0,1\}^n$ is prefix-free {suffix-free} over C if no codeword $c \in C$ is a prefix {suffix} of w. If vector w is not prefix {suffix}-free over C we say that it is prefix {suffix}-forbidden. Futher by $M(C)$ $\{\hat{M}(C)\}$ we denote the set all binary vectors of length n that are prefix-free {suffix-free} over C.

We say that set $D \subseteq \{0,1\}^n$ is *left regular* if all $(n-1)$-prefixes of words from D are pairwisely distinct. Similarly, set $D \subseteq \{0,1\}^n$ is *right regular* if all $(n-1)$-suffixes of words from D are pairwisely distinct.

If D is simultaneously left regular and right regular we say that D is *regular*.

Definition 1: We say that fix-free code $C(k_1, \ldots, k_n)$ is a *π-system* if $M(C)$ is right regular, $\hat{M}(C)$ is left regular and $\chi(k_1, \ldots, k_n) = \frac{1}{2}$.

For example, one can easily check that a set Q_n of all binary vectors of length n with odd Hamming weight is a π-system.

Definition 2: We say that fix-free code C is a *$\hat{\pi}$-system* if $M(C)$ is right regular and $\hat{M}(C)$ is left regular.

Definition 3: A *monolith* with parameters k and n - (denotation $V(k,n)$), is a regular k-subset of $\{0,1\}^n$ such that every $(n-1)$-prefix of codeword from $V(k,n)$ is an $(n-1)$-suffix of a codeword from $V(k,n)$.

3.2 Sketch of the proof

We split the proof into two sections. In the next section we study the properties and develop explicit constructions of π-systems. The main result of this section is formulated by lemma 5. In section "fix-free codes" we show the relationship between π-systems and fix-free codes. The main result of that section is summarized by lemma 8. Using the constructions from lemma 5 and lemma 8 one can easily get the proof of theorem 3.

3.3 π-Systems

It is obvious, that if the union of non-intersecting monoliths is regular then it's a monolith. Similarly, if $V(k_2,n) \subseteq V(k_1,n)$, then set difference of $V(k_1,n)$ and $V(k_2,n)$ is a monolith.

Definition 4: We say that monolith $V(k,n)$ is *undecomposable* if it can not be presented as a union of two non-empty monoliths.

Lemma 2: Monolith $V(k,n)$ defines a sequence of monoliths:

$$V(k,n) \to V(k,n+1) \to V(k,n+2) \to \ldots.$$

Construction: Any monolith $V(k,n)$ can be represented as a union of undecomposable monoliths. $V(k,n) = \bigcup_i V(k_i,n)$, where $\sum_i k_i = k$. Fix an arbitrary undecomposable monolith $V(k_i,n)$. We order its codewords $v_1, \ldots, v_{k_i}$. v_s is followed by v_{s+1} if $(n-1)$-suffix of v_s is an $(n-1)$-prefix of v_{s+1}. As $V(k_i,n)$ is undecomposable, this ordering is cyclic. Basing on this ordering one can notice that $V(k_i,n)$ is defined by a binary "key" sequence of k_i symbols. Every codeword from $V(k_i,n)$ is just n symbols, read from "key" sequence in a cyclic manner. As "key" sequence does not depend on n, $V(k_i,n) \to V(k_i,n+1) \to \ldots$.

Example 1:

$$V(k=3,n=3) \to\ V(k=3,n=4) \to\ V(k=3,n=5)$$

"Key sequence" 001.

$$\begin{array}{ccc} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{array} \Rightarrow \begin{array}{cccc} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{array} \Rightarrow \begin{array}{ccccc} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{array}$$

Let w and v be arbitrary binary vectors of length n. We say that vector v is dual of w if they coincide in all coordinates except the first and the last. Let V be an arbitrary set of codewords. By $\bar{V}$ we denote the set of their dual.

It's simple to check the following lemma:

Lemma 3: If $C(k_1, \ldots, k_n)$ is a π-system and monolith $V(k,n) \subseteq C$, then replacing $V(k,n)$ with $V(k,n+1) \bigcup \bar{V}(k,n+1)$, (here $V(k,n+1)$ is generated by the previous lemma), we obtain a new π-system $C'(k_1, \ldots, k_n - k, 2k)$.

Lemma 4: For every n there exists a π-system $C_n(k_1, k_2, \ldots, k_n)$, such that: $k_1 = k_2 = \ldots = k_{n-1} = 0$, $k_n = 2^{n-1}$, $C_n(k_1, \ldots, k_n) = \bigcup\limits_{i=0}^{n-1} V(p_i, n)$, where $p_0 = p_1 = 1$, $p_2 = 2, \ldots, p_{n-1} = 2^{n-2}$.

Proof: We apply the method of mathematical induction on n.

$$\underline{n=1}: \; C_1(k_1 = 1) = \{0\}, V(p_0 = 1, n = 1) = \{0\},$$

$$\underline{n=2}: \; C_2(k_1 = 0, k_2 = 2) = \{00, 11\},$$

$$V(p_0 = 1, n = 2) = \{00\}, V(p_1 = 1, n = 2) = \{11\}.$$

Suppose that lemma is proved for $n = N$. Hence, $C_N(k_1 = 0, \ldots, k_N = 2^{N-1}) = \bigcup\limits_{i=0}^{N-1} V(p_i, N)$. We prove lemma for $n = N+1$. Let us apply the construction from lemma 3 to every monolith $V(p_i, N)$ from C_N. Hence, $C_N \Rightarrow C_{N+1} = \left(\bigcup\limits_{i=0}^{N-1} V(p_i, N+1) \right) \bigcup D$, where $D = \bigcup\limits_{i=0}^{N-1} \bar{V}(p_i, N+1)$. Obviously, C_{N+1} is a π-system and a monolith. Note that D is also a monolith as D is set difference of monolith C_{N+1} and monolith $\bigcup\limits_{i=0}^{N-1} V(p_i, N+1)$. Hence, $D = V(p_N = 2^{N-1}, n = N+1)$ and lemma 4 is proved.

Lemma 5: If $k_1 = \ldots = k_{n-2} = 0$, $\frac{k_{n-1}}{2^{n-1}} + \frac{k_n}{2^n} = \frac{1}{2}$, then there exists a π-system $C(k_1, k_2, \ldots, k_n)$.

Proof: The existence of π-system $C_{n-1}(k_1 = \ldots = k_{n-2} = 0, k_{n-1} = 2^{n-2}) = \bigcup\limits_{i=0}^{n-2} V(p_i, n-1)$, $p_0 = 1$, $p_i = 2^{i-1}$, is proved by the previous lemma. There are all the powers of 2 up to $n-3$ among the sizes of the monoliths contained in C_{n-1}. Hence, uniting some monoliths $V(p_i, n-1)$, we obtain $V(k_{n-1}, n-1)$. Applying the construction from lemma 3 to the rest of the monoliths we prove lemma 5.

We demonstrate the method for constructing π-systems by the following

Example 2: Construction of π-system $C(k_3 = 3, k_4 = 1, k_5 = 2)$.

$$0 \to \begin{matrix} 0 & 0 \\ 1 & 1 \end{matrix} \Rightarrow \begin{matrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{matrix} \Rightarrow \begin{matrix} 0 & 0 & 0 & \\ 1 & 0 & 1 & \\ 0 & 1 & 0 & \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{matrix} \Rightarrow \begin{matrix} 0 & 0 & 0 & & \\ 1 & 0 & 1 & & \\ 0 & 1 & 0 & & \\ 0 & 1 & 1 & 0 & \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{matrix}$$

3.4 Fix-free codes

Let $M_1, M_2 \subseteq \{0,1\}^n$. Then *concatenation* of M_1 and M_2 (denotation $M_1 \otimes M_2$) is a set of all vectors $w \in \{0,1\}^{n+1}$, such that n-prefix of w belongs to M_1, and n-suffix of w belongs to M_2. Note that if C is a fix-free code, then we can add $M(C) \otimes \hat{M}(C)$ to C not breaking the fix-free property. It's simple to check that if M_1 is left regular, M_2 is right regular and $|M_1| = |M_2| \geq 2^{n-2} + \delta$, then $|M_1 \otimes M_2| \geq 2\delta$.

Let $w \in \{0,1\}^n$. We define left {right} dual of w as vector v of length n that coincides with w in all the coordinates except the first {last}.

Lemma 6: Let $C_1(k_1, \ldots, k_n)$ be an arbitrary π-system. Then for any $p \leq 2^{n-2}$, we can extend C_1 with p codewords of length n. Thus

$$C_1(k_1, \ldots, k_n) \Rightarrow C_2(k_1, \ldots, k_n + p),$$

where C_2 is a $\hat{\pi}$-system, left dual of added p codewords are prefix-forbidden over C_1 and right dual are suffix-forbidden.

Lemma 7: Let $C_1(k_1, \ldots, k_n = b + p)$ be an arbitrary $\hat{\pi}$-system such that $\chi(k_1, \ldots, k_n = b + p) = \frac{3}{4}$. For fixed p codewords of length n left dual are prefix-forbidden and right dual are suffix-forbidden. Then we can convert:

$$C_1(k_1, \ldots, k_n = b + p) \Rightarrow C_2(k_1, \ldots, k_n = b, k_{n+1} = 2p).$$

in such a way that C_2 is a $\hat{\pi}$-system, left dual of all the codewords of length $n+1$ are prefix-forbidden and right dual are suffix-forbidden.

Proof: As $\chi(k_1, \ldots, b+p) = \frac{3}{4}$, $|M(C_1)| = |\hat{M}(C_1)| = 2^{n-2}$. Deleting p fixed codewords from C_1 we convert it to $C'(k_1, \ldots, k_n = b)$. $|M(C')| = |\hat{M}(C')| = 2^{n-2} + p$. As left duals of the deleted codewords were prefix-forbidden $M(C')$ is right regular. Similarly, $\hat{M}(C')$ is left regular. Hence, $|M(C') \otimes \hat{M}(C')| \geq 2p$. Adding $2p$ codewords from $M(S') \otimes \hat{M}(S')$ to C' we obtain $C_2(k_1, k_2, \ldots, k_n = b, k_{n+1} = 2p)$.

Note that construction from lemma 6 can be reapplied recursively.

The obvious corollary from lemmas 6 and 7 is:

Lemma 8: Let $C_1(k_1, k_2, \ldots, k_p)$ be an arbitrary π-system. Then for any nonnegative values of parameters $b, k_{p+1}, \ldots, k_n$ such that

$$\chi(k_1, \ldots, k_p + b, \ldots, k_n) \leq \frac{3}{4},$$

we can extend $C_1(k_1, \ldots, k_n)$ to $C_2(k_1, \ldots, k_p + b, \ldots, k_n)$, where C_2 is a fix-free code.

References

[1] J.L. Peterson, Computer programs for detecting and correcting spelling errors. In *Comm. ACM, 23*, pp. 676-687, 1980.

[2] R. Ahlswede, B. Balkenhol, L. Kharchatrian, Some Properties of Fix-Free Codes. In *Proc. First INTAS International Seminar on Coding Theory and Combinatorics*, pp. 20-33, Thazkhadzor, Armenia, 1996.

[3] Chunxuan Ye, Raymond W. Yeung, On Fix-Free Codes. In *Proc. ISIT 2000*, p. 426, Sorrento, Italy, June 25-30, 2000.