

Sufficient Conditions for Existence of Binary Fix-Free Codes

Zsolt Kulkorelly and Kenneth Zeger, *Fellow, IEEE*

Abstract—Two sufficient conditions are given for the existence of binary fix-free codes (i.e., both prefix-free and suffix-free). Let L be a finite multiset of positive integers whose Kraft sum is at most $3/4$. It is shown that there exists a fix-free code whose codeword lengths are the elements of L if either of the following two conditions holds. i) The smallest integer in L is at least 2, and no integer in L , except possibly the largest one, occurs more than $2^{\min(L)-2}$ times. ii) No integer in L , except possibly the largest one, occurs more than twice. The results move closer to the Ahlswede–Balkenhol–Khachatrian conjecture that Kraft sums of at most $3/4$ suffice for the existence of fix-free codes.

Index Terms—Lossless source coding, Huffman codes, reversible codes.

I. INTRODUCTION

FOR each nonnegative integer n let $\{0, 1\}^n$ denote the set of all binary words of length n , and let $\{0, 1\}^*$ denote the set of all finite-length binary words, including the empty word ε . In particular, $\{0, 1\}^n = \{\varepsilon\}$ for $n = 0$. Let 0^n (resp., 1^n) denote a sequence of n zeros (resp., ones), with $0^0 = 1^0 = \varepsilon$. A binary code is any finite subset of $\{0, 1\}^*$. The elements of a code are called *codewords*. For any two words $u, v \in \{0, 1\}^*$, let uv denote the concatenation of u and v (where $u\varepsilon = \varepsilon u = u$, for all u). The word u is called a *prefix* of uv and v is called a *suffix* of uv . A *prefix-free code* is a code such that no codeword is a prefix of any other codeword. A *suffix-free code* is a code such that no codeword is a suffix of any other codeword. A *fix-free code* is a code that is both a prefix-free code and a suffix-free code. Fix-free codes are also known in the literature as *affix codes* (e.g., [6]), *biprefix codes* (e.g., [3]), *never-self-synchronizing codes* (e.g., [8]), and *reversible variable-length codes* (e.g., [18]).

The empty word ε is a prefix and a suffix of any word in $\{0, 1\}^*$, and the only prefix-free, suffix-free, or fix-free code that contains ε is $\{\varepsilon\}$. For any word $u \in \{0, 1\}^*$, let $l(u)$ denote the length of u in bits. For any two codes X and Y , let

$$XY = X \cdot Y = \{uv : u \in X, v \in Y\}.$$

In particular, $X \cdot \{\varepsilon\} = \{\varepsilon\} \cdot X = X$. For any word $u \in \{0, 1\}^*$ and any code X , let

$$u \cdot X = \{u\} \cdot X = \{uv : v \in X\}.$$

Manuscript received October 15, 2003; revised June 26, 2005. This work was supported in part by the National Science Foundation, the University of California, San Diego Center for Wireless Communications, and the Swiss National Science Foundation.

Z. Kulkorelly lives in Switzerland (e-mail: Zsolt.Kulkorelly@Consultant.ch).

K. Zeger is with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093-0407 USA (e-mail: zeger@code.ucsd.edu).

Communicated by S. A. Savari, Associate Editor for Source Coding.

Digital Object Identifier 10.1109/TIT.2005.855581

The set $X \cdot u$ is defined similarly. Accordingly

$$u \cdot \{\varepsilon\} = \{\varepsilon\} \cdot u = \{u\}.$$

In a fix-free code, any finite sequence of codewords can be decoded in both directions, thus reducing the decoding time and/or error propagation. Various properties of fix-free codes are known [3], [17] and algorithms were given in [6], [9] to construct a complete fix-free code for a given set of codeword lengths. Some other studies involving fix-free codes include [2], [10], [11], [16]. Fix-free codes have been used in the development of certain international video compression standards for robustness to channel errors [4], [7], [13], [15], [19], [22]–[24]. A special case of a fix-free code is a *palindromic code* (also called a *symmetrical reversible variable-length code*), which is defined as a prefix-free code, all of whose codewords are palindromes. Constructions of such codes were considered in [18], [20], [21].

If a probability distribution is put on a code C such that for each i , the i th codeword has probability p_i , then the *average length* of the code is defined as

$$\sum_{i \in C} l(i)p_i.$$

For data compression it is important to minimize the average length of a code with respect to a probability distribution. For a given distribution, Huffman codes are prefix codes for which no other prefix code has smaller average length [5]. However, Huffman codes cannot in general be decoded in both directions, making them sometimes less convenient than fix-free codes for use on a noisy channel. The average length properties of fix-free codes have been studied in many of the previously cited papers.

Let $m : \mathbb{Z} \rightarrow \mathbb{Z}$ be a nonnegative mapping defined on the positive integers. The *Kraft sum* of m is the quantity

$$S(m) = \sum_{i \geq 1} m(i)2^{-i}.$$

The *Kraft sum* of a code C is the quantity

$$\hat{S}(C) = \sum_{i \in C} 2^{-l(i)}.$$

If a code has exactly $m(i)$ codewords of length i for each i , then we say the code is an m -code. Every m -code C satisfies $\hat{S}(C) = S(m)$. The support of m is denoted

$$\text{supp}(m) = \{i : m(i) > 0\}$$

and the elements of $\text{supp}(m)$ are called *lengths* and each quantity $m(i)$ is called the *multiplicity* of the length i . The mapping

m is called a *multiplicity function*. For a multiplicity function m , let

$$l_{\min} = \min(\text{supp}(m))$$

$$l_{\max} = \max(\text{supp}(m)).$$

Kraft [14] showed in 1949 that every prefix-free code C must satisfy $\hat{S}(C) \leq 1$, and for every multiplicity function m with $S(m) \leq 1$ there exists a corresponding prefix-free m -code. The same result holds for suffix-free codes as well. Ahlswede, Balkenhol, and Khachatrian conjectured in 1996 that an analogous result holds for fix-free codes, but with the Kraft sum bound being $3/4$, instead of 1. The conjecture is stated as follows.¹

Conjecture I.1 (Ahlswede, Balkenhol, and Khachatrian [1]):

Let m be a multiplicity function.

If $S(m) \leq 3/4$, then there exists a fix-free m -code.

They proved the conjecture is true in the weaker case when the Kraft sum is at most $1/2$. They also proved a converse of the conjecture, namely, that any Kraft sum bound guaranteeing the existence of a fix-free code can be at most $3/4$. There are clearly fix-free codes whose Kraft sum is larger than $3/4$ (such as the set of all binary words of a given length, whose Kraft sum is 1), but these do not violate the conjecture. Instead, the conjecture gives the Kraft bound as a *sufficient* condition to guarantee the existence of a fix-free code.

Ahlswede, Balkenhol, and Khachatrian proved their conjecture in the special case where every two codewords either have the same length or have one codeword at least twice as long as the other codeword. Since their conjecture was made, several researchers have proven other special cases, although the general conjecture still remains an open problem. The following proposition is a summary of the known results relating to the conjecture.

Proposition I.2: Let m be a multiplicity function.

- a) (**Kraft** [14])
If $S(m) \leq 1$, then there exists a prefix-free m -code.
Conversely, every prefix-free m -code satisfies $S(m) \leq 1$.
- b) (**Ahlswede, Balkenhol, and Khachatrian** [1])
If $S(m) \leq 1/2$, then there exists a fix-free m -code.
Conversely, for every $x > 3/4$, there exists a multiplicity function m' , satisfying $S(m') \leq x$, such that no m' -code is fix-free.
- c) (**Ahlswede, Balkenhol, and Khachatrian** [1])
Suppose for every distinct $i, j \in \text{supp}(m)$, either $i \geq 2j$, or $j \geq 2i$.
If $S(m) \leq 3/4$, then there exists a fix-free m -code.
- d) (**Harada and Kobayashi** [12])
Suppose $|\text{supp}(m)| \leq 2$.
If $S(m) \leq 3/4$, then there exists a fix-free m -code.
- e) (**Ye and Yeung** [25])
Suppose $m(1) > 0$.
If $S(m) \leq 5/8$, then there exists a fix-free m -code.

- f) (**Ye and Yeung** [25])
Suppose $\max(\text{supp}(m)) \leq 7$.
If $S(m) \leq 3/4$, then there exists a fix-free m -code.
- g) (**Yekhanin** [26])
Suppose $\max(\text{supp}(m)) \leq 8$.
If $S(m) \leq 3/4$, then there exists a fix-free m -code.
- h) (**Yekhanin** [26])
Suppose $m(1) > 0$.
If $S(m) \leq 3/4$, then there exists a fix-free m -code.
- i) (**Yekhanin** [26])
Suppose there is $i \geq 2$ with $m(1) = \dots = m(i-1) = 0$ and $m(i) > 0$, such that

$$m(i)2^{-i} + m(i+1)2^{-(i+1)} \geq 1/2.$$

If $S(m) \leq 3/4$, then there exists a fix-free m -code.

- j) (**Yekhanin** [27])
If $S(m) \leq 5/8$, then there exists a fix-free m -code.

Parts f) and g) of Proposition I.2 were proven by exhaustive computer searches. In addition to these previous results, Ye and Yeung gave other conditions for the existence of fix-free codes, although not in the form of Kraft sum bounds. One such condition is stated as follows.

Proposition I.3 (Ye and Yeung [25]): For integers $1 \leq l_1 \leq \dots \leq l_n$, let h_i be the smallest index h with $l_h = l_{i+1}$, and define

$$L_i = 1 - 2 \sum_{j=1}^i 2^{-l_j} + (i+1-h_i) 2^{-l_{i+1}} + \sum_{\substack{1 \leq j, k \leq h_i - 1 \\ l_j + l_k \leq l_{i+1}}} 2^{-l_j - l_k}.$$

If L_1, \dots, L_{n-1} are all positive, then there is a fix-free code with codeword lengths l_1, \dots, l_n .

In this paper, we partly prove Conjecture I.1 by considering two special cases (Theorems II.4 and III.1). In both cases, we prove that the general conjecture holds if an additional constraint is put on the multiplicity function. Our proofs are constructive—specific codes are created in each case examined. We demonstrate that the classes of multiplicity functions for which our results hold contain many cases not covered by previous known results.

II. SUFFICIENT CONDITIONS FOR FIX-FREE CODES—PART I

First, some lemmas are given that are used to prove Theorem II.4.

Lemma II.1: If X and Y are fix-free codes then XY is a fix-free code.

Proof: The lemma holds if $X = \{\varepsilon\}$ or $Y = \{\varepsilon\}$. Suppose now that neither X nor Y contains ε . Let $u, v \in XY$ such that $u \neq v$. Then there exist $x_1, x_2 \in X$ and $y_1, y_2 \in Y$ such that $u = x_1y_1$ and $v = x_2y_2$. Suppose u is a prefix of v . Then x_1 is a prefix of x_2y_2 . If $l(x_1) < l(x_2)$ then x_1 is a prefix of x_2 , and if $l(x_1) > l(x_2)$ then x_2 is a prefix of x_1 , both violations of the fact that X is a prefix-free code. Thus, $l(x_1) = l(x_2)$, but this implies that $x_1 = x_2$ and hence, y_1 is a prefix of y_2 violating

¹The authors in [1] appropriately described this conjecture as “remarkable.”

the fact that Y is a prefix-free code. A similar argument holds if u is a suffix of v . \square

Lemma II.2: Let B and C be fix-free codes such that $l(b) < l(c)$ for all $b \in B$ and $c \in C$. Let $A \subseteq B$. Then $(B \setminus A) \cup C$ is fix-free if and only if the following two conditions hold:

- 1) any codeword in C has a prefix in A or no prefix in B ;
- 2) any codeword in C has a suffix in A or no suffix in B .

Proof: By symmetry, it is enough to prove that 1) implies that $(B \setminus A) \cup C$ is prefix-free. If $A = \emptyset$ or $A = B$, then the proof is easy. If $\varepsilon \in B$, then $B = \{\varepsilon\}$ and therefore, $A = \emptyset$ or $A = B$.

Now let $\emptyset \neq A \neq B$, and suppose that $(B \setminus A) \cup C$ is not prefix-free. Then there exists $c \in C$ and $b \in B \setminus A$ such that b is a prefix of c , that is, c is of the form bc_1 . By 1), c has a prefix in A , i.e., $c = ac_2$ for some $a \in A$, and thus $bc_1 = ac_2$. Both a and b are in B . If $l(a) < l(b)$, then a is a prefix of b . If $l(b) < l(a)$, then b is a prefix of a . Both possibilities contradict the fact that B is prefix-free. Therefore, $l(a) = l(b)$ and we conclude that $a = b$, which is also a contradiction as $a \in A$ and $b \notin A$. The converse of the lemma is easily seen to be true. \square

Lemma II.3: Let B, C , and D be binary codes such that $B \cup C$ and D are fix-free. Let $l(b) < l(c) = l(d)$ for all $b \in B, c \in C$, and $d \in D$. Let $A \subseteq B$. Then $(B \setminus A) \cup C \cup D$ is fix-free if the following two conditions hold:

- 1) any codeword in D has a prefix in A or no prefix in B ;
- 2) any codeword in D has a suffix in A or no suffix in B .

Proof: The sets B, C , and $(B \setminus A) \cup C$ are fix-free since $B \cup C$ was assumed to be fix-free. Using Lemma II.2, in the forward direction, then implies that any codeword in C has a prefix in A or no prefix in B , and has a suffix in A or no suffix in B . Therefore, using Conditions 1) and 2), the fact that $C \cup D$ is fix-free (since C and D have equal length codewords), and Lemma II.2 in the backward direction, we have that $(B \setminus A) \cup C \cup D$ is fix-free. \square

Theorem II.4: Let m be a multiplicity function such that $l_{\min} \geq 2$, and $m(i) \leq 2^{l_{\min}-2}$ for all $i \neq l_{\max}$. If $S(m) \leq 3/4$, then there exists a fix-free m -code.

Proof: If $l_{\max} \leq l_{\min} + 1$, then the theorem holds by Proposition I.2, part d). Assume now $l_{\max} \geq l_{\min} + 2$.

First, we construct a fix-free code E with Kraft sum equal to $3/4$, that has $2^{l_{\min}-2}$ codewords of length i , for all $i \neq l_{\max}$. Then we use the code E to construct a fix-free code H that has $m(i)$ codewords of length i for all $i \neq l_{\max}$, and whose Kraft sum satisfies $\hat{S}(H) \geq 3/4$. Such a code has

$$N = 2^{l_{\max}} \left(\hat{S}(H) - \sum_{i \neq l_{\max}} 2^{-i} m(i) \right)$$

codewords of length l_{\max} . Finally, we delete $N - m(l_{\max})$ codewords of length l_{\max} from H , to obtain a fix-free m -code.

Define the sets

$$\begin{aligned} C &= \{10^i 1 : 0 \leq i \leq l_{\max} - l_{\min} - 1\} \\ D_1 &= 0 \cdot \{0, 1\}^{l_{\max}-l_{\min}} \cdot 0 \cdot \{0, 1\}^{l_{\min}-2} \\ D_2 &= 10^{l_{\max}-l_{\min}+1} \cdot \{0, 1\}^{l_{\min}-2} \end{aligned}$$

$$\begin{aligned} D_3 &= 10^{l_{\max}-l_{\min}} 1 \cdot \{0, 1\}^{l_{\min}-2} \\ D &= D_1 \cup D_2 \cup D_3 \\ E &= C \cdot \{0, 1\}^{l_{\min}-2} \cup D. \end{aligned}$$

The set $\{0, 1\}^{l_{\min}-2}$ is fix-free (constant-length codewords) and C is fix-free as well. Therefore, the code $C \cdot \{0, 1\}^{l_{\min}-2}$ is fix-free by Lemma II.1 and consists of $2^{l_{\min}-2}$ words of length i for $l_{\min} \leq i < l_{\max}$. The words of D are all of length l_{\max} , and thus D is fix-free. We have

$$|D| = 2^{l_{\max}-2} + 2^{l_{\min}-1}$$

and therefore the Kraft sum of E is

$$\begin{aligned} \hat{S}(E) &= 2^{l_{\min}-2} \left(2^{-l_{\min}} + \dots + 2^{-(l_{\max}-1)} \right) \\ &\quad + 2^{-l_{\max}} (2^{l_{\max}-2} + 2^{l_{\min}-1}) = 3/4. \end{aligned}$$

It can be seen that no codeword of $C \cdot \{0, 1\}^{l_{\min}-2}$ is a prefix or a suffix of a word in D . Therefore, E is fix-free.

Let F be a subset of $C \cdot \{0, 1\}^{l_{\min}-2}$ containing $2^{l_{\min}-2} - m(i)$ words of length i , for each $i \neq l_{\max}$. Let G be a set of at least

$$\sum_{i=l_{\min}}^{l_{\max}-1} (2^{l_{\min}-2} - m(i)) \cdot 2^{l_{\max}-i}$$

words of length l_{\max} , none of which are in D . By removing the words of F from $C \cdot \{0, 1\}^{l_{\min}-2}$ and adding the words of G to D , we create the code

$$H = (C \cdot \{0, 1\}^{l_{\min}-2} \setminus F) \cup (D \cup G) = (E \setminus F) \cup G.$$

The Kraft sum of H is at least as large as the Kraft sum of E and thus is at least $3/4$, and H has $m(i)$ words of length i for $l_{\min} \leq i < l_{\max}$. It remains to choose F and G in such a way that H is fix-free. Lemma II.3 guarantees that H is fix-free if the following two conditions hold:

- 1) each word in G has a prefix in F or no prefix in $C \cdot \{0, 1\}^{l_{\min}-2}$;
 - 2) each word in G has a suffix in F or no suffix in $C \cdot \{0, 1\}^{l_{\min}-2}$.
- A. For each i satisfying $l_{\min} \leq i \leq l_{\max} - l_{\min} + 1$, proceed as follows. Include in F any $2^{l_{\min}-2} - m(i)$ words of the form $10^{i-l_{\min}} 1a$, where $a \in \{0, 1\}^{l_{\min}-2}$. For each of these words, include in G :

— the $2^{l_{\max}-i-1}$ words of the set

$$10^{i-l_{\min}} 1a \cdot \{0, 1\}^{l_{\max}-l_{\min}-i+1} \cdot 0 \cdot \{0, 1\}^{l_{\min}-2}.$$

These words have a prefix in F . They also have no suffix in $C \cdot \{0, 1\}^{l_{\min}-2}$ because they end with $0u$ for some $u \in \{0, 1\}^{l_{\min}-2}$ whereas the words in $C \cdot \{0, 1\}^{l_{\min}-2}$ end with $1v$ for some $v \in \{0, 1\}^{l_{\min}-2}$.

— the $2^{l_{\max}-i-1}$ words of the set

$$0 \cdot \{0, 1\}^{l_{\max}-i-1} \cdot 10^{i-l_{\min}} 1a.$$

These words have a suffix in F , and they have no prefix in $C \cdot \{0, 1\}^{l_{\min}-2}$ because they begin with 0 .

- B. For each i satisfying $l_{\max} - l_{\min} + 2 \leq i < l_{\max}$ and $m(i) \geq 2^{l_{\min}-3}$, proceed as follows. Include in F any $2^{l_{\min}-2} - m(i)$ words of the form $10^{i-l_{\min}}1a$, where $a \in \{0,1\}^{l_{\min}-2}$, that have a 0 in the $(l_{\max} - l_{\min} + 2)$ th bit from the left. (There are $2^{l_{\min}-3}$ such words, and $2^{l_{\min}-2} - m(i) \leq 2^{l_{\min}-3}$.) For each of these words, include in G

— the $2^{l_{\max}-i-1}$ words of the set

$$0 \cdot \{0,1\}^{l_{\max}-i-1} \cdot 10^{i-l_{\min}}1a.$$

These words have a suffix in F , and they have no prefix in $C \cdot \{0,1\}^{l_{\min}-2}$ because they begin with 0.

— the $2^{l_{\max}-i}$ words of the set $10^{i-l_{\min}}1a \cdot \{0,1\}^{l_{\max}-i}$.

These words have a prefix in F . They have no suffix in $C \cdot \{0,1\}^{l_{\min}-2}$ because they have a 0 in position $l_{\max} - l_{\min} + 2$, and thus end with $0u$ for some $u \in \{0,1\}^{l_{\min}-2}$ whereas the words in $C \cdot \{0,1\}^{l_{\min}-2}$ end with $1v$ for some $v \in \{0,1\}^{l_{\min}-2}$.

- C. For each i satisfying $l_{\max} - l_{\min} + 2 \leq i < l_{\max}$ and $m(i) < 2^{l_{\min}-3}$, proceed as follows. Include in F the $2^{l_{\min}-3}$ words of the form $10^{i-l_{\min}}1a$, where $a \in \{0,1\}^{l_{\min}-2}$, that have a 0 in position $l_{\max} - l_{\min} + 2$. For each of these words, include in G the same words as in Part B. Also, include in F any $2^{l_{\min}-3} - m(i)$ words of the form $10^{i-l_{\min}}1a$, $a \in \{0,1\}^{l_{\min}-2}$ that have a 1 in position $l_{\max} - l_{\min} + 2$. For each of these words, include in G the $2^{l_{\max}-i-1}$ words of the set $0 \cdot \{0,1\}^{l_{\max}-i-1} \cdot 10^{i-l_{\min}}1a$. As in Part B, these words have a suffix in F and no prefix in $C \cdot \{0,1\}^{l_{\min}-2}$.

By Lemma II.3, we conclude that H is fix-free. The size of G is (with sums whose lower limit is larger than their upper limit taken as 0)

$$\begin{aligned} |G| &= \sum_{i=l_{\min}}^{l_{\max}-l_{\min}+1} (2^{l_{\min}-2} - m(i))(2^{l_{\max}-i-1} + 2^{l_{\max}-i}) \\ &\quad + \sum_{\substack{i=l_{\max}-1 \\ i=l_{\max}-l_{\min}+2 \\ m(i) \geq 2^{l_{\min}-3}}}^{l_{\max}-1} (2^{l_{\min}-2} - m(i))(2^{l_{\max}-i-1} + 2^{l_{\max}-i}) \\ &\quad + \sum_{\substack{i=l_{\max}-l_{\min}+2 \\ m(i) < 2^{l_{\min}-3}}}^{l_{\max}-1} [2^{l_{\min}-3} \cdot (2^{l_{\max}-i-1} + 2^{l_{\max}-i}) \\ &\quad \quad + (2^{l_{\min}-3} - m(i)) \cdot 2^{l_{\max}-i-1}] \\ &= \sum_{i=l_{\min}}^{l_{\max}-l_{\min}+1} (2^{l_{\min}-2} - m(i)) \cdot 2^{l_{\max}-i} \\ &\quad + \sum_{\substack{i=l_{\max}-1 \\ i=l_{\max}-l_{\min}+2 \\ m(i) \geq 2^{l_{\min}-3}}}^{l_{\max}-1} \frac{3}{2} (2^{l_{\min}-2} - m(i)) \cdot 2^{l_{\max}-i} \\ &\quad + \sum_{\substack{i=l_{\max}-l_{\min}+2 \\ m(i) < 2^{l_{\min}-3}}}^{l_{\max}-1} \left(2^{l_{\min}-2} - \frac{m(i)}{2} \right) \cdot 2^{l_{\max}-i} \\ &\geq \sum_{i=l_{\min}}^{l_{\max}-1} (2^{l_{\min}-2} - m(i)) \cdot 2^{l_{\max}-i}. \quad \square \end{aligned}$$

TABLE I
FIX-FREE CODES FOR THE TWO CASES OF m SATISFYING
 $l_{\min} = 2, m(2) = 1, m(3) \leq 1$, AND $l_{\max} = 4$. THE TOP ROW
SPECIFIES THE MULTIPLICITY FUNCTION m

(0,1,1,6)	(0,1,0,8)
11	11
101	0000
0000	0010
0010	0100
0100	0110
0110	1001
1001	1000
1000	1010
	0101

III. SUFFICIENT CONDITIONS FOR FIX-FREE CODES—PART II

Theorem III.1: Let m be a multiplicity function with $m(i) \leq 2$ for all $i \neq l_{\max}$.

If $S(m) \leq 3/4$, then there exists a fix-free m -code.

Proof: We consider eight cases. The notation $m = (x_1, \dots, x_{l_{\max}})$ will be used to mean $m(i) = x_i$ for $1 \leq i \leq l_{\max}$.

Case 1 : $l_{\min} = 1$.

The existence of a fix-free code follows from Proposition I.2, part h).

Case 2 : $l_{\min} = 2$ and $m(2) = 2$.

The existence of a fix-free code follows from Proposition I.2, part i).

Case 3 : $l_{\min} \geq 3$.

The existence of a fix-free code follows from Theorem II.4.

Case 4 : $l_{\min} = 2, m(2) = 1$, and $m(3) \geq 2$.

The existence of a fix-free code follows from Proposition I.2, part i).

Case 5 : $l_{\min} = 2, m(2) = 1, m(3) \leq 1$, and $l_{\max} = 3$.

Then $|\text{supp}(m)| = 2$ and the theorem holds by Proposition I.2, part d).

Case 6 : $l_{\min} = 2, m(2) = 1, m(3) \leq 1$, and $l_{\max} = 4$.

There are only two possibilities, and for each of these two cases a fix-free code is given in Table I. Only the case where the Kraft sum exactly equals $3/4$ is shown in Table I. If the Kraft sum is less than $3/4$ then the desired m -code is a subcode of one of the two codes listed in Table I, formed by deleting some codewords of length l_{\max} .

Case 7 : $l_{\min} = 2, m(2) = 1, m(3) \leq 1$, and $l_{\max} = 5$.

There are six possibilities for m , and for each of these a fix-free code is given in Table II. As in Table I, all of the codes in Table II have Kraft sum exactly equal to $3/4$, and codewords of length l_{\max} can be removed to obtain codes with smaller Kraft sums.

Case 8 : $l_{\min} = 2, m(2) = 1, m(3) \leq 1$, and $l_{\max} \geq 6$.

We prove this lengthy case by constructing a fix-free code that has $m(i)$ codewords of length i for all $i \neq l_{\max}$ and a Kraft sum of at least $3/4$. Define the set

$$C = \{11, 101\} \cup \{10^{i-2}1, 01^{i-2}0 : 4 \leq i < l_{\max}\}$$

and notice that it is a fix-free code.

TABLE II
FIX-FREE CODES FOR THE SIX CASES OF m SATISFYING $l_{\min} = 2, m(2) = 1, m(3) \leq 1$, AND $l_{\max} = 5$. THE TOP ROW SPECIFIES THE MULTIPLICITY FUNCTION m

(0,1,1,2,8)	(0,1,1,1,10)	(0,1,1,0,12)	(0,1,0,2,12)	(0,1,0,1,14)	(0,1,0,0,16)
11	11	11	11	11	11
101	101	101	1001	1001	00000
1001	1001	00000	0110	00000	00010
0110	00000	00010	00000	00010	00100
00000	00010	00100	00010	00100	00110
00010	00100	00110	00100	00110	01000
00100	00110	01000	01000	01000	01010
01000	01000	01010	01010	01010	01100
01010	01010	01100	01110	01100	01110
01110	01100	01110	10001	01110	10001
10001	01110	10001	10000	10001	10000
10000	10001	10000	00001	10000	00001
	10000	10010	00101	00001	00101
		01001	10100	00101	10100
			10101	10100	10101
				10101	10010
					01001

We first construct a fix-free code E for the case where $m(3) = 1$ and $m(i) = 2$ for $4 \leq i < l_{\max}$. Then we use E to construct a fix-free code H satisfying the conditions of Case 8, and with Kraft sum at least $3/4$. Then, as in Theorem II.4, some codewords of length l_{\max} can be removed from H to ensure there are $m(l_{\max})$ such codewords. Define

$$\begin{aligned}
 D_1 &= 00 \cdot \{0, 1\}^{l_{\max}-4} \cdot 00 \\
 D_2 &= 00 \cdot \{0, 1\}^{l_{\max}-5} \cdot 010 \\
 D_3 &= 010 \cdot \{0, 1\}^{l_{\max}-5} \cdot 00 \\
 D_4 &= \{10^{l_{\max}-2}1, 01^{l_{\max}-2}0, 0^{l_{\max}-1}1, 10^{l_{\max}-1}\} \\
 D &= D_1 \cup D_2 \cup D_3 \cup D_4 \\
 E &= C \cup D.
 \end{aligned}$$

The code C has one codeword each of length 2 and 3 and two codewords of length i for $4 \leq i < l_{\max}$. All codewords of D have length l_{\max} and $|D| = 2^{l_{\max}-3} + 4$. Therefore, the Kraft sum of E is

$$\hat{S}(E) = \frac{1}{4} + \frac{1}{8} + \sum_{i=4}^{l_{\max}-1} 2 \cdot 2^{-i} + (2^{l_{\max}-3} + 4)2^{-l_{\max}} = 3/4.$$

Also, no codeword of C is a prefix or a suffix of a codeword in D . Therefore, E is fix-free.

We now construct a fix-free code H from E . Let F be a subset of C containing $1 - m(3)$ codewords of length 3 and $2 - m(i)$ codewords of length i for $4 \leq i < l_{\max}$. Let G be a set containing at least

$$(1 - m(3))2^{l_{\max}-3} + \sum_{i=4}^{l_{\max}-1} (2 - m(i)) \cdot 2^{l_{\max}-i}$$

words of length l_{\max} , none of which are in D . By removing the words of F from C and adding the words of G to D , we create the code

$$H = (C \setminus F) \cup (D \cup G) = (E \setminus F) \cup G.$$

Such a set H has a Kraft sum at least as large as the Kraft sum of E and thus at least $3/4$, and H has $m(i)$ words of length i for $2 \leq i < l_{\max}$. It remains to choose F and G in such a way

that H is fix-free. Lemma II.3 guarantees that H is fix-free if the following two conditions hold:

- 1) each word in G has a prefix in F or no prefix in C ;
- 2) each word in G has a suffix in F or no suffix in C .

We first choose F .

If $m(3) = 0$, then include in F the word 101. For $4 \leq i < l_{\max}$

- if $m(i) = 1$ then include in F the word $01^{i-2}0$;
- if $m(i) = 0$ then include in F the words $01^{i-2}0$ and $10^{i-2}1$.

The set G is then constructed as follows.

- Include in G the $2^{l_{\max}-6}$ words of the set

$$010 \cdot \{0, 1\}^{l_{\max}-6} \cdot 010.$$

These words have neither a prefix nor a suffix in C .

- For each word $10^{i-2}1$ in F , for $3 \leq i \leq l_{\max} - 2$, include in G
 - the $2^{l_{\max}-i-2}$ words of the set

$$00 \cdot \{0, 1\}^{l_{\max}-i-2} \cdot 10^{i-2}1.$$

These words have a suffix in F and no prefix in C ;

- the $2^{l_{\max}-i-2}$ words of the set

$$10^{i-2}1 \cdot \{0, 1\}^{l_{\max}-i-2} \cdot 00.$$

These words have a prefix in F and no suffix in C .

- For each word $01^{i-2}0$ in F , for $4 \leq i \leq l_{\max} - 2$, include in G
 - the $2^{l_{\max}-i-2}$ words of the set

$$00 \cdot \{0, 1\}^{l_{\max}-i-2} \cdot 01^{i-2}0.$$

These words have a suffix in F and no prefix in C ;

- the $2^{l_{\max}-i-2}$ words of the set

$$01^{i-2}0 \cdot \{0, 1\}^{l_{\max}-i-2} \cdot 00.$$

These words have a prefix in F and no suffix in C .

- For each word $10^{i-2}1$ in F , for $3 \leq i \leq l_{\max} - 3$, include in G
 - the $2^{l_{\max}-i-3}$ words of the set

$$010 \cdot \{0, 1\}^{l_{\max}-i-3} \cdot 10^{i-2}1.$$

These words have a suffix in F and no prefix in C ;
— the $2^{l_{\max}-i-3}$ words of the set

$$10^{i-2}1 \cdot \{0, 1\}^{l_{\max}-i-3} \cdot 010.$$

These words have a prefix in F and no suffix in C .

- For each word $01^{i-2}0$ in F , for $4 \leq i \leq l_{\max} - 3$, include in G

— the $2^{l_{\max}-i-3}$ words of the set

$$010 \cdot \{0, 1\}^{l_{\max}-i-3} \cdot 01^{i-2}0.$$

These words have a suffix in F and no prefix in C ;
— the $2^{l_{\max}-i-3}$ words of the set

$$01^{i-2}0 \cdot \{0, 1\}^{l_{\max}-i-3} \cdot 010.$$

These words have a prefix in F and no suffix in C .

- If F contains the word $10^{l_{\max}-3}1$, include in G
 - the word $010^{l_{\max}-3}1$. This word has a suffix in F and no prefix in C ;
 - the word $10^{l_{\max}-3}10$. This word has a prefix in F and no suffix in C .
- If F contains the word $01^{l_{\max}-3}0$, include in G
 - the word $001^{l_{\max}-3}0$. This word has a suffix in F and no prefix in C ;
 - the word $01^{l_{\max}-3}00$. This word has a prefix in F and no suffix in C .
- If F contains the word $01^{l_{\max}-4}0$, include in G
 - the word $0101^{l_{\max}-4}0$. This word has a suffix in F and no prefix in C ;
 - the word $01^{l_{\max}-4}010$. This word has a prefix in F and no suffix in C .
- For all $i, j \geq 3$, if $i + j \leq l_{\max}$ and F contains $10^{i-2}1$ and $10^{j-2}1$, then include in G
 - the $2^{l_{\max}-i-j}$ words of the set

$$10^{i-2}1 \cdot \{0, 1\}^{l_{\max}-i-j} \cdot 10^{j-2}1.$$

These words have both a prefix and a suffix in F .

- For all $i, j \geq 4$ if $i + j \leq l_{\max}$ and F contains $01^{i-2}0$ and $01^{j-2}0$, then include in G
 - the $2^{l_{\max}-i-j}$ words of the set

$$01^{i-2}0 \cdot \{0, 1\}^{l_{\max}-i-j} \cdot 01^{j-2}0.$$

These words have both a prefix and a suffix in F .

- For all $i \geq 3, j \geq 4$, if $i + j \leq l_{\max}$ and F contains $10^{i-2}1$ and $01^{j-2}0$, then include in G
 - the $2^{l_{\max}-i-j}$ words of the set

$$10^{i-2}1 \cdot \{0, 1\}^{l_{\max}-i-j} \cdot 01^{j-2}0.$$

These words have both a prefix and a suffix in F ;
— the $2^{l_{\max}-i-j}$ words of the set

$$01^{j-2}0 \cdot \{0, 1\}^{l_{\max}-i-j} \cdot 10^{i-2}1.$$

These words have both a prefix and a suffix in F .

- For all $i, j \geq 3$, if $i + j = l_{\max} + 1$ and F contains $10^{i-2}1$ and $10^{j-2}1$, then include in G
 - the word $10^{i-2}10^{j-2}1$. This word has both a prefix and a suffix in F .
- For all $i, j \geq 4$, if $i + j = l_{\max} + 1$ and F contains $01^{i-2}0$ and $01^{j-2}0$, then include in G
 - the word $01^{i-2}01^{j-2}0$. This word has both a prefix and a suffix in F .
- For all $i \geq 3, j \geq 4$, if $i + j = l_{\max} + 2$ and F contains $10^{i-2}1$ and $01^{j-2}0$, then include in G
 - the word $10^{i-2}10^{j-2}0$. This word has both a prefix and a suffix in F ;
 - the word $01^{j-2}01^{i-2}1$. This word has both a prefix and a suffix in F .

By Lemma II.3, H is fix-free. It remains to prove that

$$|G| \geq (1 - m(3))2^{l_{\max}-3} + \sum_{i=4}^{l_{\max}-1} (2 - m(i)) \cdot 2^{l_{\max}-i}.$$

Order the elements of

$$\{j : 3 \leq j < l_{\max} \text{ and } m(j) = 0\}$$

and of

$$\{j : 4 \leq j < l_{\max} \text{ and } m(j) \leq 1\}$$

and then denote them by r_1, r_2, \dots and s_1, s_2, \dots , respectively. Note that

$$\{r_1, r_2, \dots\} \setminus \{3\} \subseteq \{s_1, s_2, \dots\}.$$

Then

$$\begin{aligned} (1 - m(3))2^{l_{\max}-3} + \sum_{i=4}^{l_{\max}-1} (2 - m(i)) \cdot 2^{l_{\max}-i} \\ = \sum_i 2^{l_{\max}-r_i} + \sum_i 2^{l_{\max}-s_i}. \end{aligned}$$

Recall that

$$\begin{aligned} 10^{i-2}1 \in F &\iff m(i) = 0 \iff i \in (r_j) \\ 01^{i-2}0 \in F &\iff m(i) \leq 1 \iff i \in (s_j). \end{aligned}$$

We have

$$\begin{aligned} |G| &= 2^{l_{\max}-6} \\ &+ \sum_{\substack{3 \leq j \leq l_{\max}-2 \\ m(j)=0}} 2 \cdot 2^{l_{\max}-j-2} \\ &+ \sum_{\substack{4 \leq j \leq l_{\max}-2 \\ m(j) \leq 1}} 2 \cdot 2^{l_{\max}-j-2} \\ &+ \sum_{\substack{3 \leq j \leq l_{\max}-3 \\ m(j)=0}} 2 \cdot 2^{l_{\max}-j-3} \\ &+ \sum_{\substack{4 \leq j \leq l_{\max}-3 \\ m(j) \leq 1}} 2 \cdot 2^{l_{\max}-j-3} \\ &+ 2 \cdot \max(0, 1 - m(l_{\max} - 1)) \end{aligned}$$

$$\begin{aligned}
 &+ 2 \cdot \min(1, 2 - m(l_{\max} - 1)) \\
 &+ 2 \cdot \min(1, 2 - m(l_{\max} - 2)) \\
 &+ \sum_{\substack{i,j \geq 3 \\ i+j \leq l_{\max} \\ m(i)=m(j)=0}} 2^{l_{\max}-i-j} \\
 &+ \sum_{\substack{i,j \geq 4 \\ i+j \leq l_{\max} \\ m(i) \leq 1, m(j) \leq 1}} 2^{l_{\max}-i-j} \\
 &+ 2 \cdot \sum_{\substack{i > 3, j \geq 4 \\ i+j \leq l_{\max} \\ m(i)=0, m(j) \leq 1}} 2^{l_{\max}-i-j} \\
 &+ \sum_{\substack{i \geq 3, j \geq 3 \\ i+j = l_{\max} + 1 \\ m(i)=m(j)=0}} 1 \\
 &+ \sum_{\substack{i \geq 4, j \geq 4 \\ i+j = l_{\max} + 1 \\ m(i) \leq 1, m(j) \leq 1}} 1 \\
 &+ \sum_{\substack{i > 3, j \geq 4 \\ i+j = l_{\max} + 2 \\ m(i)=0, m(j) \leq 1}} 2 \\
 &= 2^{l_{\max}-6} \\
 &+ 2 \cdot \sum_{j:r_j \leq l_{\max}-2} 2^{l_{\max}-r_j-2} \\
 &+ 2 \cdot \sum_{j:s_j \leq l_{\max}-2} 2^{l_{\max}-s_j-2} \\
 &+ 2 \cdot \sum_{j:r_j \leq l_{\max}-3} 2^{l_{\max}-r_j-3} \\
 &+ 2 \cdot \sum_{j:s_j \leq l_{\max}-3} 2^{l_{\max}-s_j-3} \\
 &+ 2 \cdot |\{j : r_j = l_{\max} - 1\}| \\
 &+ 2 \cdot |\{j : s_j = l_{\max} - 1\}| \\
 &+ 2 \cdot |\{j : s_j = l_{\max} - 2\}| \\
 &+ \sum_{(i,j):r_i+r_j \leq l_{\max}} 2^{l_{\max}-r_i-r_j} \\
 &+ \sum_{(i,j):s_i+s_j \leq l_{\max}} 2^{l_{\max}-s_i-s_j} \\
 &+ 2 \cdot \sum_{(i,j):r_i+s_j \leq l_{\max}} 2^{l_{\max}-r_i-s_j} \\
 &+ |\{(i,j) : r_i + r_j = l_{\max} + 1\}| \\
 &+ |\{(i,j) : s_i + s_j = l_{\max} + 1\}| \\
 &+ 2|\{(i,j) : r_i + s_j = l_{\max} + 2\}|.
 \end{aligned}$$

After some manipulations, one sees that

$$|G| \geq \sum_i 2^{l_{\max}-r_i} + \sum_i 2^{l_{\max}-s_i}$$

if and only if $U \geq 0$, where

$$\begin{aligned}
 U &= 2^{l_{\max}-6} \\
 &- 2 \sum_{j:r_j \leq l_{\max}-3} 2^{l_{\max}-r_j-3} \\
 &- 2 \sum_{j:s_j \leq l_{\max}-3} 2^{l_{\max}-s_j-3}
 \end{aligned}$$

$$\begin{aligned}
 &- 2 \cdot |\{j : r_j = l_{\max} - 2\}| \\
 &+ \sum_{(i,j):r_i+r_j \leq l_{\max}} 2^{l_{\max}-r_i-r_j} \\
 &+ \sum_{(i,j):s_i+s_j \leq l_{\max}} 2^{l_{\max}-s_i-s_j} \\
 &+ 2 \cdot \sum_{(i,j):r_i+s_j \leq l_{\max}} 2^{l_{\max}-r_i-s_j} \\
 &+ |\{(i,j) : r_i + r_j = l_{\max} + 1\}| \\
 &+ |\{(i,j) : s_i + s_j = l_{\max} + 1\}| \\
 &+ 2|\{(i,j) : r_i + s_j = l_{\max} + 2\}|. \tag{3.1}
 \end{aligned}$$

We prove in the Appendix that $U \geq 0$. This completes the proof of Case 8 and hence of Theorem III.1. \square

IV. NEW CODES FROM THEOREM II.4

Theorem II.4 constructs codes that establish sufficient conditions for the existence of fix-free codes. Such codes include new fix-free codes that have not been accounted for in prior literature. As an example, we next demonstrate that a subfamily of fix-free codes given by Theorem II.4 was not implied by Propositions I.2 and I.3.

By Theorem II.4, for each $n \geq 9$, there exists a fix-free m -code satisfying

$$m(j) = \begin{cases} 0, & \text{if } 1 \leq j \leq 3 \\ 4, & \text{if } 4 \leq j \leq n - 1 \\ 2^{n-2} + 8, & \text{if } j = n. \end{cases}$$

It can be verified that these codes are not implied by either Proposition I.2 or Theorem III.1. We now prove that they are also not implied by Proposition I.3.

Let i and a be integers such that $i = 4(n - 4) + a$, and $1 \leq a \leq 2^{n-2} + 7$. Then $h_i = 4(n - 4) + 1$, $l_{i+1} = n$, and

$$\begin{aligned}
 L_i &= 1 - 2 \left(4 \cdot \sum_{m=4}^{n-1} 2^{-m} + a2^{-n} \right) \\
 &\quad + (4(n - 4) + a + 1 - (4(n - 4) + 1)) \cdot 2^{-n} \\
 &\quad + \sum_{\substack{1 \leq j, k \leq 4(n-4) \\ l_j + l_k \leq n}} 2^{-l_j - l_k} \\
 &= 1 - 8(2^{-3} - 2^{1-n}) - a2^{-n} + \sum_{\substack{1 \leq j, k \leq 4(n-4) \\ l_j + l_k \leq n}} 2^{-l_j - l_k} \\
 &= 2^{4-n} - a2^{-n} + \sum_{\substack{1 \leq j, k \leq 4(n-4) \\ l_j + l_k \leq n}} 2^{-l_j - l_k}.
 \end{aligned}$$

Note that in the multiset

$$\{(l_j, l_k) : 1 \leq j, k \leq 4(n - 4)\}$$

each element of

$$\{(c, d) : 4 \leq c, d \leq n - 1\}$$

appears exactly 16 times. Therefore, for each m satisfying $8 \leq m \leq n$, the multiset

$$\{l_j + l_k : 1 \leq j, k \leq 4(n - 4), l_j + l_k \leq n\}$$

contains $16(m-7)$ occurrences of the number m . Thus,

$$\begin{aligned} \sum_{\substack{1 \leq j, k \leq 4(n-4) \\ l_j + l_k \leq n}} 2^{-l_j - l_k} &= \sum_{m=8}^n 16(m-7)2^{-m} \\ &= 2^{-2} - (n-5)2^{4-n} \end{aligned}$$

and therefore,

$$L_i = 2^{4-n} - a2^{-n} + 2^{-2} - (n-5)2^{4-n}.$$

From this it follows that $L_i \leq 0$ if and only if

$$a \geq 2^{n-2} - 16n + 96.$$

Since a was chosen to satisfy $a \leq 2^{n-2} + 7$, there exists an a that makes $L_i \leq 0$, provided that

$$2^{n-2} + 7 \geq 2^{n-2} - 16n + 96$$

or, equivalently, if $n \geq 6$.

Therefore, for all $n \geq 6$, we have $L_i \leq 0$ and hence, the supposition of Proposition I.3 is not satisfied. This demonstrates that Proposition I.3 does not anticipate the fix-free codes constructed in Theorem II.4.

V. NEW CODES FROM THEOREM III.1

Analogous to Section IV, we demonstrate here that Theorem III.1 constructs new fix-free codes that have not been accounted for in prior literature. As an example, we next demonstrate that a subfamily of fix-free codes given by Theorem III.1 was not implied by Propositions I.2 and I.3.

By Theorem III.1, for each $n \geq 9$, there exists a fix-free m -code satisfying

$$m(j) = \begin{cases} 0, & \text{if } j = 1, 3 \\ 1, & \text{if } j = 2 \\ 2, & \text{if } 4 \leq j \leq n-1 \\ 2^{n-2} + 4, & \text{if } j = n. \end{cases}$$

It can be verified that these codes are not implied by either Proposition I.2 or Theorem II.4. We now prove that they are also not implied by Proposition I.3.

Let i and a be integers such that

$$i = 2(n-4) + 1 + a$$

and

$$1 \leq a \leq 2^{n-2} + 3.$$

Then

$$\begin{aligned} h_i &= 2(n-4) + 2 \\ l_{i+1} &= n \end{aligned}$$

and

$$\begin{aligned} L_i &= 1 - 2 \left(2^{-2} + 2 \cdot \sum_{m=4}^{n-1} 2^{-m} + a2^{-n} \right) \\ &\quad + (2(n-4) + a + 2 - (2(n-4) + 2)) \cdot 2^{-n} \\ &\quad + \sum_{\substack{1 \leq j, k \leq 2(n-4)+1 \\ l_j + l_k \leq n}} 2^{-l_j - l_k} \end{aligned}$$

$$\begin{aligned} &= 1 - 2^{-1} - 4(2^{-3} - 2^{1-n}) - a2^{-n} \\ &\quad + \sum_{\substack{1 \leq j, k \leq 2(n-4)+1 \\ l_j + l_k \leq n}} 2^{-l_j - l_k} \\ &= 2^{3-n} - a2^{-n} + \sum_{\substack{1 \leq j, k \leq 2(n-4)+1 \\ l_j + l_k \leq n}} 2^{-l_j - l_k}. \end{aligned}$$

Note that in the multiset

$$\{(l_j, l_k) : 1 \leq j, k \leq 2(n-4) + 1\},$$

the pair $(2, 2)$ appears once, each element of

$$\{(2, k), (k, 2) : 4 \leq k \leq n-2\}$$

appears twice, and each element of

$$\{(j, k) : 4 \leq j, k \leq n-4\}$$

appears four times. Therefore, the multiset

$$\{l_j + l_k : 1 \leq j, k \leq 2(n-4) + 1 \text{ and } l_j + l_k \leq n\}$$

contains one occurrence of the number 4, four occurrences of the number 6, and $4(m-6)$ occurrences of the number m , for $7 \leq m \leq n$. Thus,

$$\begin{aligned} \sum_{\substack{1 \leq j, k \leq 2(n-4)+1 \\ l_j + l_k \leq n}} 2^{-l_j - l_k} &= 2^{-4} + 4 \cdot 2^{-6} + \sum_{m=7}^n 4(m-6)2^{-m} \\ &= 2^{-3} + 2^{-3} - (n-4)2^{2-n} \end{aligned}$$

and therefore,

$$L_i = 2^{3-n} - a2^{-n} + 2^{-2} - (n-4)2^{2-n}.$$

From this it follows that $L_i \leq 0$ if and only if

$$a \geq 2^{n-2} - 4n + 24$$

or, equivalently, if $n \geq 6$.

Therefore, for all $n \geq 6$, we have $L_i \leq 0$ and hence, the supposition of Proposition I.3 is not satisfied. This demonstrates that Proposition I.3 does not anticipate the fix-free codes constructed in Theorem III.1.

APPENDIX

PROOF OF $U \geq 0$, WITH U DEFINED IN (3.1)

We first prove two more general inequalities. The inequality in (3.1) is then a special case of one or the other, depending on the parity of l_{\max} .

Lemma 1: Let n and k be integers with $0 \leq n \leq k-3$. Let (r_i) and (s_i) be sets of nonnegative integers, such that $k-n+1 \leq r_i, s_i \leq 2k-1$ for all i , and $(r_i) \subseteq (s_i)$. Then $V \geq 0$, where

$$V = 2^{2n} - 2 \sum_{i: r_i \leq k+n} 2^{k+n-r_i} - 2 \sum_{i: s_i \leq k+n} 2^{k+n-s_i}$$

$$\begin{aligned}
 & -2|\{\ell : r_\ell = k + n + 1\}| + \sum_{(i,j):r_i+r_j \leq 2k} 2^{2k-r_i-r_j} \\
 & + \sum_{(i,j):s_i+s_j \leq 2k} 2^{2k-s_i-s_j} + 2 \sum_{(i,j):r_i+s_j \leq 2k} 2^{2k-r_i-s_j} \\
 & + |\{(i,j) : r_i + r_j = 2k + 1\}| \\
 & + |\{(i,j) : s_i + s_j = 2k + 1\}| \\
 & + 2|\{(i,j) : r_i + s_j = 2k + 2\}|. \tag{A1}
 \end{aligned}$$

Proof: First, note that

$$\begin{aligned}
 & \sum_{j:r_j \leq k+n} 2^{k+n-r_j} \\
 & = \sum_{j=k-n+1}^{k+n} 2^{k+n-j} |\{\ell : r_\ell = j\}| \\
 & = \sum_{j=-n+1}^n 2^{n-j} |\{\ell : r_\ell = k+j\}|, \\
 & \sum_{(i,j):r_i+r_j \leq 2k} 2^{2k-r_i-r_j} \\
 & = \sum_{i=k-n+1}^{k+n-1} \sum_{j=k-n+1}^{2k-i} 2^{2k-i-j} |\{\ell : r_\ell = i\}| \cdot |\{\ell : r_\ell = j\}| \\
 & = \sum_{i=-n+1}^{n-1} \sum_{j=-n+1}^{-i} 2^{-i-j} [|\{\ell : r_\ell = k+i\}| \cdot |\{\ell : r_\ell = k+j\}|], \\
 & |\{(i,j) : r_i + r_j = 2k + 1\}| \\
 & = \sum_{i=k-n+1}^{k+n} |\{\ell : r_\ell = i\}| \cdot |\{\ell : r_\ell = 2k + 1 - i\}| \\
 & = \sum_{i=-n+1}^n |\{\ell : r_\ell = k + i\}| \cdot |\{\ell : r_\ell = k + 1 - i\}|.
 \end{aligned}$$

The expressions for (s_i) and for the mixed elements are similar. Let a_i (resp., b_i) be the number of occurrences of the number $k + i$ in (r_j) (resp., in (s_j)). Note that because $(r_i) \subseteq (s_i)$, we have $0 \leq a_i \leq b_i \leq 1$ for all i . Then we can write

$$\begin{aligned}
 V & = 2^{2n} - 2 \sum_{i=-n+1}^n 2^{n-i} a_i - 2 \sum_{i=-n+1}^n 2^{n-i} b_i - 2a_{n+1} \\
 & + \sum_{i=-n+1}^{n-1} \sum_{j=-n+1}^{-i} 2^{-i-j} a_i a_j + \sum_{i=-n+1}^{n-1} \sum_{j=-n+1}^{-i} 2^{-i-j} b_i b_j \\
 & + 2 \sum_{i=-n+1}^{n-1} \sum_{j=-n+1}^{-i} 2^{-i-j} a_i b_j \\
 & + \sum_{i=-n+1}^n a_i a_{1-i} + \sum_{i=-n+1}^n b_i b_{1-i} + 2 \sum_{i=-n+1}^{n+1} a_i b_{2-i}
 \end{aligned}$$

which we write as $V_n(a_{-n+1}, b_{-n+1}, \dots, a_{n+1}, b_{n+1})$, or simply as V_n .

We show that $V_n \geq 0$ by induction. If $n = 0$, then $a_1 = 0$ since $k + 1 \leq r_j$, and therefore,

$$V_0 = 1 - 2a_1 = 1 \geq 0.$$

Now, let $n = 1$. Then

$$\begin{aligned}
 V_1 & = 4 - 4a_0 - 2a_1 - 4b_0 - 2b_1 - 2a_2 + a_0^2 + b_0^2 + 2a_0b_0 \\
 & + 2a_0a_1 + 2b_0b_1 + 2a_0b_2 + 2a_1b_1 + 2a_2b_0 \\
 & = \begin{cases} 2(b_2 + a_1b_1), & \text{if } a_0 = b_0 = 1 \\ 1 - 2a_1(1 - b_1), & \text{if } a_0 = 0 \\ & \text{and } b_0 = 1 \\ 2[(1 - a_2) + (1 - a_1)(1 - b_1)], & \text{if } a_0 = b_0 = 0 \end{cases} \\
 & \geq 0
 \end{aligned}$$

since $0 \leq a_i \leq b_i \leq 1$.

Let $n \geq 2$ and assume that $V_{n-1} \geq 0$. We now consider the three cases $(a_{-n+1}, b_{-n+1}) = (1, 1), (0, 1), (0, 0)$ separately.

Case I: $a_{-n+1} = b_{-n+1} = 1$

$$\begin{aligned}
 & V_n(1, 1, a_{-n+2}, b_{-n+2}, \dots, a_{n+1}, b_{n+1}) \\
 & = 2^{2n} - 2^{2n} - 2 \sum_{i=-n+2}^n 2^{n-i} a_i - 2^{2n} \\
 & - 2 \sum_{i=-n+2}^n 2^{n-i} b_i - 2a_{n+1} + 2^{2n-2} \\
 & + \sum_{j=-n+2}^{n-1} 2^{n-1-j} a_j + \sum_{i=-n+2}^{n-1} 2^{n-1-i} a_i \\
 & + \sum_{i=-n+2}^{n-2} \sum_{j=-n+2}^{-i} 2^{-i-j} a_i a_j + 2^{2n-2} \\
 & + \sum_{j=-n+2}^{n-1} 2^{n-1-j} b_j + \sum_{i=-n+2}^{n-1} 2^{n-1-i} b_i \\
 & + \sum_{i=-n+2}^{n-2} \sum_{j=-n+2}^{-i} 2^{-i-j} b_i b_j + 2 \cdot 2^{2n-2} \\
 & + 2 \sum_{j=-n+2}^{n-1} 2^{n-1-j} b_j + 2 \sum_{i=-n+2}^{n-1} 2^{n-1-i} a_i \\
 & + 2 \sum_{i=-n+2}^{n-2} \sum_{j=-n+2}^{-i} 2^{-i-j} a_i b_j + 2a_n \\
 & + \sum_{i=-n+2}^{n-1} a_i a_{1-i} + 2b_n + \sum_{i=-n+2}^{n-1} b_i b_{1-i} \\
 & + 2a_{n+1} + 2b_{n+1} + 2 \sum_{i=-n+2}^n a_i b_{2-i} \\
 & = -2 \sum_{i=n}^n 2^{n-i} a_i - 2 \sum_{i=n}^n 2^{n-i} b_i \\
 & + \sum_{i=-n+2}^{n-2} \sum_{j=-n+2}^{-i} 2^{-i-j} a_i a_j \\
 & + \sum_{i=-n+2}^{n-2} \sum_{j=-n+2}^{-i} 2^{-i-j} b_i b_j \\
 & + 2 \sum_{i=-n+2}^{n-2} \sum_{j=-n+2}^{-i} 2^{-i-j} a_i b_j + 2a_n
 \end{aligned}$$

$$\begin{aligned}
& + \sum_{i=-n+2}^{n-1} a_i a_{1-i} + 2b_n + \sum_{i=-n+2}^{n-1} b_i b_{1-i} \\
& + 2b_{n+1} + 2 \sum_{i=-n+2}^n a_i b_{2-i} \\
= & \sum_{i=-n+2}^{n-2} \sum_{j=-n+2}^{-i} 2^{-i-j} a_i a_j \\
& + \sum_{i=-n+2}^{n-2} \sum_{j=-n+2}^{-i} 2^{-i-j} b_i b_j \\
& + 2 \sum_{i=-n+2}^{n-2} \sum_{j=-n+2}^{-i} 2^{-i-j} a_i b_j + \sum_{i=-n+2}^{n-1} a_i a_{1-i} \\
& + \sum_{i=-n+2}^{n-1} b_i b_{1-i} + 2b_{n+1} + 2 \sum_{i=-n+2}^n a_i b_{2-i} \\
\geq & 0.
\end{aligned}$$

Case II: $a_{-n+1} = 0$ and $b_{-n+1} = 1$

$$\begin{aligned}
V_n(0, 1, a_{-n+2}, b_{-n+2}, \dots, a_{n+1}, b_{n+1}) \\
= & 2^{2n} - 2 \sum_{i=-n+2}^n 2^{n-i} a_i - 2^{2n} - 2 \sum_{i=-n+2}^n 2^{n-i} b_i \\
& - 2a_{n+1} + \sum_{i=-n+2}^{n-2} \sum_{j=-n+2}^{-i} 2^{-i-j} a_i a_j \\
& + \sum_{j=-n+2}^{n-1} 2^{n-1-j} b_j + \sum_{i=-n+2}^{n-1} 2^{n-1-i} b_i + 2^{2n-2} \\
& + \sum_{i=-n+2}^{n-2} \sum_{j=-n+2}^{-i} 2^{-i-j} b_i b_j + 2 \sum_{i=-n+2}^{n-1} 2^{n-1-i} a_i \\
& + 2 \sum_{i=-n+2}^{n-2} \sum_{j=-n+2}^{-i} 2^{-i-j} a_i b_j + \sum_{i=-n+2}^{n-1} a_i a_{1-i} \\
& + 2b_n + \sum_{i=-n+2}^{n-1} b_i b_{1-i} + 2a_{n+1} + \sum_{i=-n+2}^n a_i b_{2-i} \\
= & 2^{2n-2} - 2 \sum_{i=-n+2}^{n-1} 2^{n-1-i} a_i - 2 \sum_{i=-n+2}^{n-1} 2^{n-1-i} b_i \\
& - 2a_n - 2b_n + \sum_{i=-n+2}^{n-2} \sum_{j=-n+2}^{-i} 2^{-i-j} a_i a_j \\
& + \sum_{i=-n+2}^{n-2} \sum_{j=-n+2}^{-i} 2^{-i-j} b_i b_j \\
& + 2 \sum_{i=-n+2}^{n-2} \sum_{j=-n+2}^{-i} 2^{-i-j} a_i b_j + 2b_n \\
& + \sum_{i=-n+2}^{n-1} a_i a_{1-i} + \sum_{i=-n+2}^{n-1} b_i b_{1-i} + 2 \sum_{i=-n+2}^n a_i b_{2-i} \\
= & V_{n-1}(a_{-n+2}, b_{-n+2}, \dots, a_n, b_n) \\
\geq & 0 \text{ (by the induction hypothesis).}
\end{aligned}$$

Case III: $a_{-n+1} = b_{-n+1} = 0$

For this case we consider the a_i 's and b_i 's as real variables, and by differentiating V_n , we will show that V_n is monotonic nonincreasing in each variable, and thus achieves a minimum on the boundary (i.e., when all the variables equal 1). The minimum value of V_n is shown to be zero, which will establish $V_n \geq 0$. Various useful derivatives are

$$\begin{aligned}
\frac{d}{da_k} 2 \sum_{i=-n+1}^n 2^{n-i} a_i & = \begin{cases} 2^{n+1-k}, & \text{if } k \leq n \\ 0, & \text{if } k = n+1 \end{cases} \\
\frac{d}{da_k} \sum_{i=-n+1}^{n-1} \sum_{j=-n+1}^{-i} 2^{-i-j} a_i a_j & = \begin{cases} \sum_{j=-n+1}^{-k} 2^{1-k-j} a_j, & \text{if } k \leq n-1 \\ 0, & \text{if } k \geq n \end{cases} \\
\frac{d}{da_k} \sum_{i=-n+1}^n a_i a_{1-i} & = \begin{cases} 2a_{1-k}, & \text{if } k \leq n \\ 0, & \text{if } k = n+1 \end{cases}
\end{aligned}$$

and similar expressions exist for the derivative with respect to b_k of the quantities containing only b_i 's. For the quantities containing both a_i 's and b_i 's

$$\begin{aligned}
\frac{d}{da_k} 2 \sum_{i=-n+1}^{n+1} a_i b_{2-i} & = 2b_{2-k} \\
\frac{d}{db_k} 2 \sum_{i=-n+1}^{n+1} a_i b_{2-i} & = 2a_{2-k} \\
\frac{d}{da_k} 2 \sum_{i=-n+1}^{n-1} \sum_{j=-n+1}^{-i} 2^{-i-j} a_i b_j & = \begin{cases} \sum_{j=-n+1}^{-k} 2^{1-k-j} b_j, & \text{if } k \leq n-1 \\ 0, & \text{if } k \geq n \end{cases} \\
\frac{d}{db_k} 2 \sum_{i=-n+1}^{n-1} \sum_{j=-n+1}^{-i} 2^{-i-j} a_i b_j & = \begin{cases} \sum_{j=-n+1}^{-k} 2^{1-k-j} a_j, & \text{if } k \leq n-1 \\ 0, & \text{if } k \geq n. \end{cases}
\end{aligned}$$

Thus, for $k \leq n-1$

$$\begin{aligned}
\frac{d}{da_k} V_n & = -2^{n+1-k} + \sum_{j=-n+1}^{-k} 2^{1-k-j} a_j \\
& + \sum_{j=-n+1}^{-k} 2^{1-k-j} b_j + 2a_{1-k} + 2b_{2-k} \\
\frac{d}{db_k} V_n & = -2^{n+1-k} + \sum_{j=-n+1}^{-k} 2^{1-k-j} a_j \\
& + \sum_{j=-n+1}^{-k} 2^{1-k-j} b_j + 2b_{1-k} + 2a_{2-k}.
\end{aligned}$$

Since $a_{-n+1} = b_{-n+1} = 0$, the first term in every sum disappears, giving

$$\begin{aligned} \frac{d}{da_k} V_n &\leq -2^{n+1-k} + 2^{2-k} \sum_{j=-n+2}^{-k} 2^{-j} + 4 \\ &= -2^{n+1-k} + 2^{2-k}(2^{n-1} - 2^k) + 4 \\ &= 0 \end{aligned}$$

and similarly $\frac{d}{db_k} V_n \leq 0$. Furthermore

$$\begin{aligned} \frac{d}{da_n} V_n &= -2 + 2a_{-n+1} + 2b_{-n+2} \\ &= -2 + 2b_{-n+2} \\ &\leq 0 \\ \frac{d}{db_n} V_n &= -2 + 2b_{-n+1} + 2a_{-n+2} \\ &= -2 + 2a_{-n+2} \\ &\leq 0 \\ \frac{d}{da_{n+1}} V_n &= -2 + 2b_{-n+1} \\ &\leq 0 \\ \frac{d}{db_{n+1}} V_n &= 2a_{-n+1} \\ &= 0. \end{aligned}$$

Thus, for $a_{-n+1} = b_{-n+1} = 0$, V_n is minimized in the cube

$$\{(a_{-n+2}, b_{-n+2}, \dots, a_{n+1}, b_{n+1}) : 0 \leq a_i, b_i \leq 1\}$$

when $a_{-n+2} = b_{-n+2} = \dots = a_{n+1} = b_{n+1} = 1$. At this minimum

$$\begin{aligned} V_n(0, 0, 1, \dots, 1) &= 2^{2n} - 2 \sum_{i=-n+2}^n 2^{n-i} - 2 \sum_{i=-n+2}^n 2^{n-i} - 2 \\ &\quad + 4 \sum_{i=-n+2}^{n-2} \sum_{j=-n+2}^{-i} 2^{-i-j} + 2 \sum_{i=-n+2}^{n-1} 1 + 2 \sum_{i=-n+2}^n 1 \\ &= 0. \quad \square \end{aligned}$$

Lemma 2: Let n and k be integers such that $0 \leq n \leq k - 4$. Let (r_i) and (s_i) be finite sets of nonnegative integers, such that $k - n \leq r_i, s_i \leq 2k - 2$ for all i , and $(r_i) \subseteq (s_i)$. Then $W \geq 0$, where

$$\begin{aligned} W &= 2^{2n+1} - 2 \sum_{i:r_i \leq k+n} 2^{k+n-r_i} - 2 \sum_{i:s_i \leq k+n} 2^{k+n-s_i} \\ &\quad - 2|\{\ell : r_\ell = k + n + 1\}| \\ &\quad + \sum_{(i,j):r_i+r_j \leq 2k-1} 2^{2k-1-r_i-r_j} \\ &\quad + \sum_{(i,j):s_i+s_j \leq 2k-1} 2^{2k-1-s_i-s_j} \\ &\quad + 2 \sum_{(i,j):r_i+s_j \leq 2k-1} 2^{2k-1-r_i-s_j} \\ &\quad + |\{(i,j) : r_i + r_j = 2k\}| \\ &\quad + |\{(i,j) : s_i + s_j = 2k\}| \\ &\quad + 2|\{(i,j) : r_i + s_j = 2k + 1\}|. \end{aligned} \quad (\text{A2})$$

Proof: The proof is essentially the same as the one of Lemma 1. \square

We are now ready to prove (3.1).

Corollary 3: Let $l_{\max} \geq 6$, and let (r_i) and (s_i) be finite lists of integers, with $3 \leq r_i \leq l_{\max} - 1$, and $4 \leq s_i \leq l_{\max} - 1$, in which each integer occurs at most once, and $(r_i) \setminus \{3\} \subseteq (s_i)$. Then $U \geq 0$.

Proof: First let $3 \in (r_i)$. We extract from the sums each occurrence of $r_i = 3$ and obtain

$$\begin{aligned} U &= 2^{l_{\max}-6} \\ &\quad - 2 \cdot 2^{l_{\max}-6} - 2 \cdot \sum_{4 \leq r_i \leq l_{\max}-3} 2^{l_{\max}-3-r_i} \\ &\quad - 2 \cdot \sum_{4 \leq s_i \leq l_{\max}-3} 2^{l_{\max}-3-s_i} \\ &\quad - 2 \cdot |\{j : r_j = l_{\max} - 2\}| \\ &\quad + 2 \cdot \sum_{4 \leq r_i \leq l_{\max}-3} 2^{l_{\max}-3-r_i} + 2^{l_{\max}-6} \\ &\quad + \sum_{(i,j):r_i+r_j \leq l_{\max}, r_i, r_j \geq 4} 2^{l_{\max}-r_i-r_j} \\ &\quad + \sum_{(i,j):s_i+s_j \leq l_{\max}} 2^{l_{\max}-s_i-s_j} \\ &\quad + 2 \cdot \sum_{4 \leq s_i \leq l_{\max}-3} 2^{l_{\max}-3-s_i} \\ &\quad + 2 \cdot \sum_{(i,j):r_i+s_j \leq l_{\max}, r_i \geq 4} 2^{l_{\max}-r_i-s_j} \\ &\quad + 2 \cdot |\{j : r_j = l_{\max} - 2\}| \\ &\quad + |\{(i,j) : r_i + r_j = l_{\max} + 1, r_i \geq 4, r_j \geq 4\}| \\ &\quad + |\{(i,j) : s_i + s_j = l_{\max} + 1\}| \\ &\quad + 2 \cdot |\{(i,j) : r_i + s_j = l_{\max} + 2\}| \\ &= \sum_{(i,j):r_i+r_j \leq l_{\max}, r_i, r_j \geq 4} 2^{l_{\max}-r_i-r_j} \\ &\quad + \sum_{(i,j):s_i+s_j \leq l_{\max}} 2^{l_{\max}-s_i-s_j} \\ &\quad + \sum_{(i,j):r_i+s_j \leq l_{\max}, r_i \geq 4} 2^{l_{\max}-r_i-s_j+1} \\ &\quad + |\{(i,j) : r_i + r_j = l_{\max} + 1, r_i \geq 4, r_j \geq 4\}| \\ &\quad + |\{(i,j) : s_i + s_j = l_{\max} + 1\}| \\ &\quad + 2 \cdot |\{(i,j) : r_i + s_j = l_{\max} + 2\}| \\ &\geq 0. \end{aligned}$$

Now suppose $3 \notin (r_i)$. If l_{\max} is even, set $k = l_{\max}/2$ and use Lemma 1 with $n = k - 3$ in (A1). If l_{\max} is odd, set $k = (l_{\max} + 1)/2$ and use Lemma 2 with $n = k - 4$ in (A2). \square

ACKNOWLEDGMENT

The authors wish to thank both anonymous reviewers for catching several errors in the original manuscript.

REFERENCES

- [1] R. Ahlswede, B. Balkenhol, and L. Khachatryan, "Some properties of fix-free codes," in *Proc. 1st Intas Seminar on Coding Theory and Combinatorics*, Thakadzor, Armenia, 1996, pp. 20–33.

- [2] R. Bauer and J. Hagenauer, "Iterative source/channel-decoding using reversible variable length codes," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 2000, pp. 93–102.
- [3] J. Berstel and D. Perrin, *Theory of Codes*. New York: Academic, 1985.
- [4] M. Bystrom, S. Kaiser, and A. Kopansky, "Soft source decoding with applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 10, pp. 1108–1120, Oct. 2001.
- [5] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
- [6] A. S. Fraenkel and S. T. Klein, "Bidirectional Huffman coding," *Comput. J.*, vol. 33, no. 4, pp. 296–307, 1990.
- [7] S.-S. Gao and G.-F. Tu, "Robust H.263+ video transmission using partial backward decodable bit stream (PBDBS)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 2, pp. 182–187, Feb. 2003.
- [8] E. N. Gilbert and E. F. Moore, "Variable-length binary encodings," *Bell Syst. Tech. J.*, vol. 38, pp. 933–967, July 1959.
- [9] D. Gillman and R. L. Rivest, "Complete variable-length fix-free codes," *Des., Codes, Cryptogr.*, vol. 5, no. 2, pp. 109–114, Mar. 1995.
- [10] B. Girod, "Bidirectionally decodable streams of prefix code-words," *IEEE Commun. Lett.*, vol. 3, no. 8, pp. 245–247, Aug. 1999.
- [11] S. Kaiser and M. Bystrom, "Soft decoding of variable-length codes," in *Proc. IEEE Int. Conf. Communications*, vol. 3, New Orleans, LA, Jun. 2000, pp. 1203–1207.
- [12] K. Harada and K. Kobayashi, "A note on the fix-free property," *IEICE Trans. Fundam.*, vol. E82-A, no. 10, pp. 2121–2128, Oct. 1999.
- [13] *Coding of Audio-Visual Objects: Visual*, Final Draft, ISO/IEC Int. Std. 14496-2, Oct. 1998.
- [14] L. G. Kraft, "A device for quantizing, grouping, and coding amplitude modulated pulses," Master's thesis, Dep. Elec. Eng., MIT, Cambridge, MA, 1949.
- [15] K. Laković and J. Villasenor, "On design of error-correcting reversible variable length codes," *IEEE Commun. Lett.*, vol. 6, no. 8, pp. 337–339, Aug. 2002.
- [16] —, "An algorithm for construction of efficient fix-free codes," *IEEE Commun. Lett.*, to be published.
- [17] M. P. Schützenberger, "On a special class of recurrent events," *Ann. Math. Statist.*, vol. 32, pp. 1201–1213, 1961.
- [18] Y. Takishima, M. Wada, and H. Murakami, "Reversible variable length codes," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 158–162, Feb./Mar./Apr. 1995.
- [19] C.-W. Tsai, T.-J. Huang, K.-L. Fang, and J.-L. Wu, "A hybrid and flexible H.263-based error resilient and testing system," in *IEEE Region 10 International Conference on Electrical and Electronic Technology*, vol. 1, Aug. 2001, pp. 122–128.
- [20] C.-W. Tsai and J.-L. Wu, "On constructing the Huffman-code-based reversible variable-length codes," *IEEE Trans. Commun.*, vol. 49, no. 9, Sep. 2001.
- [21] —, "Modified symmetrical reversible variable-length code and its theoretical bounds," *IEEE Trans. Inf. Theory*, vol. 47, no. 6, pp. 2543–2548, Sep. 2001.
- [22] J. L. H. Webb, "Efficient table access for reversible variable-length decoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 8, pp. 981–985, Aug. 2001.
- [23] J. Wen and J. D. Villasenor, "A class of reversible variable length codes for robust image and video coding," in *Proc. IEEE Int. Conf. Image Processing*, vol. 2, Washington, DC, Oct. 1997, pp. 65–68.
- [24] —, "Reversible variable length codes for efficient and robust image and video coding," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 1998, pp. 471–480.
- [25] C. Ye and R. W. Yeung, "Some basic properties of fix-free codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 1, pp. 72–87, Jan. 2001.
- [26] S. Yekhanin, "Sufficient conditions of existence of fix-free codes," in *Proc. IEEE Int. Symp. Information Theory*, Washington, DC, Jun. 2001, p. 284.
- [27] —, "Improved upper bound for the redundancy of fix-free codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 11, pp. 2815–2818, Nov. 2004.