On the Construction of Maximal Prefix-Synchronized Codes

Hiroyoshi Morita, Member, IEEE, Adriaan J. van Wijngaarden, Student Member, IEEE, and A. J. Han Vinck, Senior Member, IEEE

Abstract—We present a systematic procedure for mapping data sequences into codewords of a prefix-synchronized code (PS-code), as well as for performing the inverse mapping. A PS-code, proposed by Gilbert in 1960, belongs to a subclass of comma-free codes and is useful to recover word synchronization when errors have occurred in the stream of codewords. A PS-code is defined as a set of codewords with the property that each codeword has a known sequence as a prefix, followed by a coded data sequence in which this prefix is not allowed to occur. The largest PS-code proposed by Gilbert in 1960, belongs to a subclass of comma-free code, as well as for performing the inverse mapping. A PS-code, data sequences into codewords of a prefix-synchronized code (PS-code), was introduced by Gilbert [4]. These codes have the property that every codeword starts with a prefix \( P = p_1p_2 \ldots p_k \) of length \( k \), followed by a constrained sequence \( c_1c_2 \ldots c_m \) of length \( m \). Moreover, for any codeword \( p_1 \ldots p_kc_1 \ldots c_m \), prefix \( P \) does not appear as a block of \( k \) consecutive symbols anywhere in \( p_2 \ldots p_kc_1 \ldots c_{m-1}c_m \). Therefore, word synchronization can be easily established at the decoder side by scanning the incoming stream of symbols for the occurrence of prefix \( P \).

A major disadvantage of general comma-free codes is the need for an exhaustive search in the code set to decide whether or not a given string of \( n \) symbols is a codeword. To overcome this difficulty, a subclass of comma-free codes, called prefix-synchronized codes (PS-codes), was introduced by Gilbert [4]. These codes have the property that every codeword starts with a prefix \( P \), followed by a constrained sequence \( c_1c_2 \ldots c_m \) of length \( m \). Moreover, for any codeword \( p_1 \ldots p_kc_1 \ldots c_m \), prefix \( P \) does not appear as a block of \( k \) consecutive symbols anywhere in \( p_2 \ldots p_kc_1 \ldots c_{m-1}c_m \). Therefore, word synchronization can be easily established at the decoder side by scanning the incoming stream of symbols for the occurrence of prefix \( P \).

A BLOCK code \( C_n^\alpha \) of length \( n \) over an alphabet \( \mathcal{A}_\alpha \) of size \( \alpha \) is called a comma-free code, if and only if for any pair of codewords \( a_1a_2 \ldots a_n \) and \( b_1b_2 \ldots b_n \) in \( C_n^\alpha \), the \( n \) symbol overlaps

\[
a_2a_3 \ldots a_nb_1b_2b_3 \ldots b_n = b_2b_3 \ldots b_nb_1b_2b_3 \ldots a_n
\]

are not in \( C_n^\alpha \) [1]. In a communication system, a comma-free code can be used to enable the receiver to determine the location of the codewords in the incoming stream of symbols. Word synchronization can be recovered after having received at most \( 2n - 2 \) error-free symbols.

The cardinality of a comma-free code \( C_n^\alpha \), denoted by \( C_n^\alpha \), is bounded by

\[
C_n^\alpha \leq \frac{1}{n} \sum_{d|n} \mu(d) \alpha^{n/d}
\]

where \( \mu \) is the Möbius function [1]. A good approximation for this upper bound is given by \( C_n^\alpha \leq \alpha^n / n \), and therefore the redundancy, being equal to \( n - \log_\alpha C_n^\alpha \), is at least \( \log_\alpha n \). For any odd \( n \), comma-free codes having maximal size can be constructed [2], [3].

In the next section, we will give an overview of earlier work on PS-codes. Then, in Section III, we present the recursive structure of the constrained part of \( G_{P_G}^{(k+m)} \) from which a constructive mapping procedure of a data sequence to the constrained part of a codeword can be obtained. The encoding and decoding algorithms for \( G_{P_G}^{(k+m)} \) are presented in Section IV, as well as a proof of the correctness of both algorithms. The time complexity of the proposed coding scheme is proportional to the code length.
In Section V, we address the coding algorithm for the class of PS-codes with self-uncorrelated prefixes. It is known that if \( P \) is self-uncorrelated, then \( G_p^{(n)} \) has the same size as \( G_p^{(n)} \) [5]. However, neither any encoding nor decoding algorithm for such a \( G_p^{(n)} \) have been found in the literature. We give a two-step algorithm for encoding and decoding \( G_p^{(n)} \) for any self-uncorrelated \( P \). The total time complexity is also proportional to the code length.

II. PREVIOUS RESULTS

Gilbert [4] has shown that the redundancy of a binary PS-code \( G_p^{(n)} \) is upper-bounded by \( \log_2 n + 1.52 \), if \( P \) is of the form \( 11 \cdots 10 \) of length \( \lceil \log_2 n \log_2 e \rceil \). This specific prefix is referred to as Gilbert’s prefix, and the corresponding PS-code will be simply denoted by \( G^{(n)} \).

Gilbert conjectured that for a given code length \( n \), \( G^{(n)} \) is optimal in the sense that it is the largest PS-code among all binary PS-codes of length \( n \). For alphabet size \( \alpha \leq 4 \), the conjecture is proved by Guibas and Odlyzko [5] for sufficiently large \( n \). Surprisingly, it is also proved that for \( \alpha \geq 5 \) infinitely many values of \( n \) exist for which Gilbert’s prefix is not optimal [5]. Although it would be interesting to find out which prefix gives an optimal PS-code for a general finite alphabet, we will not consider this open problem in the current paper.

From an engineering point of view, the main practical difficulty of using PS-codes still remains. The encoding and decoding procedures generally become more complex as the length of the codewords increases. In fact, finding a constructive coding method for \( G^{(n)} \) without use of a lookup table has remained as an open problem.

A nearly-optimal construction method has been developed by Mandelbaum [6]. He presents an encoding and decoding procedure for a PS-code based on Fibonacci codes, as proposed by Kautz [7]. This method is constructive in the sense that no lookup table is required. Mandelbaum shows that a binary PS-code, denoted by \( M_{P_G}^{(k+m)} \) can be constructed by applying Kautz’s coding method. The redundancy of \( M_{P_G}^{(k+m)} \) is shown to be approximately equal to \( (\log_2 n) + 2 \), if \( n \approx 2^k \). His method was extended to runlength-limited codes [8]–[10]. However, being a PS-code, \( M_{P_G}^{(k+m)} \) is not optimal among all PS-codes of length \( k + m \) and prefix \( P_G \) of length \( k \). In fact, \( G_{P_G}^{(k+m)} \) is always larger than \( M_{P_G}^{(k+m)} \) for any \( n \) which is shown in Section III where the exact difference of size between \( M_{P_G}^{(k+m)} \) and \( G_{P_G}^{(k+m)} \) is determined.

Capocelli [11] proposes another coding scheme for \( G_{P_G}^{(k+m)} \) as a part of unbounded integer coding by showing an example of the scheme for \( k = 3 \). In fact, for a given \( k \), the infinite union

\[
\bigcup_{m=0}^{\infty} G_{P_G}^{(k+m)}
\]

is a code set which can be used to encode arbitrary positive integers. In his method, \( G_{P_G}^{(3(k+m))} \) is partitioned into two subsets: one set of codewords starting with 0 and the other set of codewords starting with 1. The integer to be encoded is compared with the size of the first subset to obtain the first bit of the corresponding codeword. Continuing these steps recursively, the codeword will be determined bit by bit. The size of the latter set is easily shown to be equal to the second-order Fibonacci number. Therefore, the size of the former set can be also represented using these Fibonacci numbers, although the obtained formula will be complicated. A more formal description of Capocelli’s algorithm for any \( k \) is found in [12].

Unlike Capocelli’s algorithm, we partition \( G_{P_G}^{(k+m)} \) into \( k \) subsets, which gives us a much more convenient formula for enumerating the number of codewords. Moreover, derivation of the encoding and decoding algorithms has become straightforward.

III. MAXIMAL PREFIX-SYNCHRONIZED CODES

In this section we investigate a recursive structure of \( G_{P_G}^{(k+m)} \) from which a coding scheme for \( G_{P_G}^{(k+m)} \) is directly deduced. The exact analysis on the difference between \( M_{P_G}^{(k+m)} \) and \( G_{P_G}^{(k+m)} \) is also deduced using the recursive structure. Before developing the theory, we introduce a useful definition of the correlation between two sequences [5], [13] using a slightly different notation.

**Definition 1:** For two sequences \( X \) and \( Y \) of length \( |X| \) and \( |Y| \), respectively, the correlation of \( X \) over \( Y \), denoted by \( X \circ Y \), is a binary sequence \( [b_1b_2 \cdots b_{|X|}] \) of the same length as \( X \). Let \( s = \max (|X| - |Y|, 0) \). Each element \( b_i \) with \( 1 \leq i \leq |X| \) is defined by

\[
b_i = \begin{cases} 
\gamma(x_1 \cdots x_{|Y|} y_{i-1} \cdots y_{|Y|}), & 1 \leq i \leq s \\
\gamma(x_1 \cdots x_{|Y|} y_1 \cdots y_{|X|-i+1}), & s < i \leq |X|
\end{cases}
\]

where \( \gamma(Z_1, Z_2) \) is 1 if two sequences \( Z_1 \) and \( Z_2 \) are identical, and 0 otherwise.

For example, if \( X = 1021 \) and \( Y = 10102 \), then \( X \circ Y = 00011 \) and \( Y \circ X = 1100 \). Note that in general \( X \circ Y \neq Y \circ X \). The correlation \( X \circ X \) is called the autocorrelation of \( X \). We will denote a sequence of \( s \) consecutive symbols \( b \in A_\alpha \) by \( b^s \). Then, for a self-uncorrelated sequence \( X \), \( X \circ X = 100 \cdots 0 \) holds.

Let the concatenation of two sequences \( X \) and \( Y \) be denoted by \( XY \). In terms of correlation, we can represent the necessary and sufficient condition that a sequence \( PY \) of length \( n = k + m \) is a codeword of \( G_{P_G}^{(k+m)} \) by

\[
PY \circ P = 10^{k+m-1}(s)^{k-1}
\]

where the character \( * \) is used to denote an arbitrary symbol of \( A_\alpha \), and \( (s)^{k} \) represents a sequence of \( A_\alpha^s \).

For a prefix \( P \) of length \( k \geq 1 \), let \( F_{P}^{(m)} \) denote the set of sequences of length \( m \) such that no \( P \) appears in any position as a string of \( k \) consecutive symbols. Therefore, \( F_{P}^{(m)} \) is defined by

\[
F_{P}^{(m)} = \begin{cases} 
A_\alpha^m, & m < k \\
A_\alpha^m \setminus \{P\}, & m = k \\
\{X \in A_\alpha^m | X \circ P = 0^{m-k+1}(s)^{k-1}\}, & m > k
\end{cases}
\]

The following lemma can be easily derived from the definition of \( F_{P}^{(m)} \), and is useful to obtain the structure of \( G_{P_G}^{(k+m)} \).
Lemma 1: Let $p \in \mathcal{A}_n^k$. For every $Q \in \mathcal{A}_n^s$ with $1 \leq s \leq \min(k-1,m)$, $\mathcal{F}_Q^{(m)}$ contains at least one sequence with prefix $Q$, and at least one sequence with suffix $Q$.

Proof: If $m < k$, $\mathcal{F}_Q^{(m)} = \mathcal{A}_n^k$, and the correctness of the lemma immediately follows. If $m \geq k$, let $P = p_1p_2 \cdots p_k$, and let $p_i \in A_n \setminus \{p_1\}$. It is obvious that for every $Q \in \mathcal{A}_n^s$, the sequences $\bar{Q}(p_k)^{m-s}$ and $(\bar{Q})^{m-s}Q$ are elements of $\mathcal{F}_Q^{(m)}$ if $1 \leq s \leq \min(k-1,m)$, and therefore the lemma holds. □

For a sequence $P$ and a set of sequences $S$, let us denote the set of concatenations of $P$ and all the sequences in $S$ by $PS$, that is, $PS = \{PW | W \in S\}$. The null string, denoted by $\emptyset$, is introduced to represent a string of length 0, for which $\emptyset x = x \emptyset = x$. In this context, $\mathcal{F}_P^{(0)} = \{P\}$ and $\mathcal{F}_P^{(0)} = \{\emptyset\}$ with cardinality 1. For any $m < 0$, $\mathcal{F}_P^{(m)}$ is empty.

Theorem 1: For any PS-code $G_P^{(k+m)}$ with prefix $P \in \mathcal{A}_n^k$,

$$G_P^{(k+m)} \subseteq P \mathcal{F}_P^{(m)}.$$  
Equality always holds if $P$ is self-uncorrelated. Moreover, if (3) holds with equality for any $m \geq k-1$, then $P \circ P = 10^{k-1}$.

Proof: For $m \leq k$, (3) holds according to (2). Since any sequence $PY \in G_P^{(k+m)}$ satisfies

$$PYP \circ P = 10^{k-1+m} 1(s)^{k-1}, \text{ for } m > k$$

$Y \circ P = 0^{m+k-1}1(s)^{k-1}$ holds. Thus $Y \in \mathcal{F}_P^{(m)}$.

If $P$ is self-uncorrelated,

$$PYP \circ P = 10^{m+k-1}10^{k-1}$$
holds. Since $Y$ is any sequence in $\mathcal{F}_P^{(m)}$, $P \mathcal{F}_P^{(m)}$ is a PS-code and it is the largest one among all PS-codes of length $k + m$ and with prefix $P$ of length $k$. That is, $P \mathcal{F}_P^{(m)} = G_P^{(k+m)}$.

Now we assume that

$$P \mathcal{F}_P^{(m)} = G_P^{(k+m)}, \text{ for } m \geq k-1.$$ 

Then

$$PW \circ P = 10^{m+k-1}1(s)^{k-1}$$
for any $W \in \mathcal{F}_P^{(m)}$. Hence, for $1 \leq i \leq k-1$, no subblocks $L_i = p_{k+i-1} \cdots p_{k+i-1}$ of $PW \circ P$ equal $P$. Thus the first $i$ symbols of $P$ must be different from the last $i$ symbols of $P$, which shows $P \circ P = 10^{k-1}$. □

The key equation to construct the encoding/decoding algorithms for $G_P^{(k+m)}$ is presented in the following theorem.

Theorem 2: For $\alpha \geq 2$ and $m \geq 1$,

$$\mathcal{F}_P^{(m)} = \{1^m\} \cup \bigcup_{i=2}^{k-1} \{1^{i-1}0\mathcal{F}_{P_0}^{(m-i)}\} \cup \bigcup_{a \in \mathcal{A}_n, a \neq 1} \{a \mathcal{F}_{P_0}^{(m)}\}$$
(4)

where $P_G$ is of the form $1^{k-1}0$.

Proof: According to the definition, $1^m \in \mathcal{F}_{P_0}^{(m)}$. Any other sequence $W \in \mathcal{F}_{P_0}^{(m)}$ starts at one of

$$\{1^{i-1}0 | 2 \leq i < k\} \cup \{0^m \setminus \{1^m\}\}.$$ 

If $W = 1^{i-1}0V$, then $V \in \mathcal{F}_{P_0}^{(m-i)}$ since

$$1^{i-1}0 \circ P_G = 0^{m-1}(s)^{k-1}$$
implies that

$$V \circ P_G = 0^{m-i-1}(s)^{k-1}.$$ 
Similarly, it is shown that if $W \in \mathcal{F}_{P_0}^{(m)}$ is represented as $aV$ with $a \in \mathcal{A}_n \setminus \{1\}$, then $V \in \mathcal{F}_{P_0}^{(m-1)}$.

Conversely, the first $i$ components of $1^{i-1}0V \circ P_G$ are all 0 for any $V \in \mathcal{A}_n^{m-i}$ since $1^{i-1}0 \circ 0^{m-i} = 0$ where $i < k$. Moreover, the last $m - i$ components of $1^{i-1}0V \circ P$ equal $0^{m-i-1}(s)^{k-1}$ for $V \in \mathcal{F}_{P_0}^{(m-i)}$ from (2). Therefore, we obtain that

$$1^{i-1}0V \in \mathcal{F}_{P_0}^{(m)}, \text{ for } V \in \mathcal{F}_{P_0}^{(m-i)}.$$ 
In the same way, we can show that

$$aV \in \mathcal{F}_{P_0}^{(m)}, \text{ for } V \in \mathcal{F}_{P_0}^{(m-1)}.$$ □

Remark 1: Let $\overline{P_G}$ be the negation of $P_G$, that is, $\overline{P_G} = 0^{k-1}$. Then we obtain

$$\mathcal{F}_{\overline{P_G}}^{(m)} = \{0^m\} \cup \bigcup_{i=2}^{k-1} \\{0^{i-1}1\mathcal{F}_{P_0}^{(m-i)}\} \cup \bigcup_{a \in \mathcal{A}_n, a \neq 0} \{a \mathcal{F}_{P_0}^{(m)}\}.$$ 

Let $\overline{P_G}$ be the reverse order of $P_G$, that is, $\overline{P_G} = 01^{k-1}$. Then we obtain

$$\mathcal{F}_{\overline{P_G}}^{(m)} = \{1^m\} \cup \bigcup_{i=2}^{k-1} \\{\mathcal{F}_{P_0}^{(m-i)}01^{i-1}\} \cup \bigcup_{a \in \mathcal{A}_n, a \neq 1} \{\mathcal{F}_{P_0}^{(m-1)}a\}.$$ 

For a prefix $P_G = 1^{k-1}0$, we denote the cardinality of $G_{P_G}^{(k+m)}$ by $G_{k,m}$. Note that $G_{k,m}$ is written as

$$G_{k,m} = |P_G \mathcal{F}_{P_G}^{(m)}| = |\mathcal{F}_{P_G}^{(m)}|.$$ 

Theorem 3: For a given $k \geq 1$, a sequence $G_{k,0}, G_{k,1}, G_{k,2}, \ldots$ satisfies the following recursion:

$$G_{k,m} = \begin{cases} \alpha^m, & m < k \\ (\alpha - 1)G_{k,m-1} + \sum_{i=2}^{k-1} G_{k,m-i} + 1, & m \geq k. \end{cases}$$
(5)

Proof: If $m < k$, $G_{k,m}$ equals $\alpha^m$ according to (2). Since the sets on the right-hand side of the formula in Theorem 2 are distinct, we obtain

$$|\mathcal{F}_{P_0}^{(m)}| = 1 + \sum_{i=2}^{k-1} |1^{i-1}0\mathcal{F}_{P_0}^{(m-i)}| + \sum_{a \in \mathcal{A}_n, a \neq 1} |a \mathcal{F}_{P_0}^{(m)}|$$

$$= (\alpha - 1)|\mathcal{F}_{P_0}^{(m-1)}| + \sum_{i=2}^{k-1} |\mathcal{F}_{P_0}^{(m-i)}| + 1.$$ 
Equation (5) follows by replacing $|\mathcal{F}_{P_0}^{(m-i)}|$ with $G_{k,m-i}$. □
Remark 2: Mandelbaum’s code $\mathcal{M}_{P_G}^{(k+m)}$ is a binary PS-code defined by $\mathcal{M}_{P_G}^{(k+m)} = 1-k-10^2_{1-k-1}$. It is also subdivided into $k-1$ subsets as follows:

$$\mathcal{M}_{P_G}^{(k+m)} = \bigcup_{i=1}^{k-1} \left\{ \frac{1}{10^i}02^{(m-i)} \right\}. \quad (6)$$

Since the derivation of (6) is similar to Theorem 2, we omit it. Moreover, let $M_{k,m}$ be the cardinality of $\mathcal{M}_{P_G}^{(k+m)}$. Then, we obtain that

$$M_{k,m} = \begin{cases} 2^m, & m \leq k-2 \\ M_{k,m-1} + \cdots + M_{k,m-k+1}, & m \geq k-1. \end{cases} \quad (7)$$

By comparing (5) in case of $\alpha = 2$ with (7), we immediately know that $G_{k,m} > M_{k,m}$.

The exact difference in size between $\mathcal{M}_{P_G}^{(k+m)}$ and $\mathcal{G}_{P_G}^{(k+m)}$ is analyzed using the generating functions for $G_{k,m}$ and $M_{k,m}$, which are defined as follows:

$$G_k(z) = \sum_{m=0}^{\infty} G_{k,m} z^m \quad (8)$$

$$M_k(z) = \sum_{m=0}^{\infty} M_{k,m} z^m. \quad (9)$$

Then, using the recursions of (5) and (7), $G_k(z)$ and $M_k(z)$ can be written as

$$G_k(z) = \frac{1}{1-2z+z^2} \quad (10)$$

$$M_k(z) = \frac{1}{1-2z} \quad (11)$$

Since $G_k(z) = (1-z^{-k})G_k(z)$, we obtain

$$M_{k,m} = G_{k,m} - G_{k,m-k}, \quad m \geq 0 \quad (12)$$

where $G_{k,i} = 0, i < 0$. A variation of (12) is given by

$$G_{k,m} = G_{k,m-1} + M_{k,m-1}, \quad m \geq 1. \quad (13)$$

To obtain (13), we modify (5) as follows:

$$G_{k,m} = G_{k,m-1} + \cdots + G_{k,m-k+1} + 1$$

$$= G_{k,m-1} + (G_{k,m-1} - G_{k,m-k})$$

$$= G_{k,m-1} + M_{k,m-1}$$

where we use (12) to obtain the last equality.

Next, we will deduce approximated expressions for $G_{k,m}$ and $M_{k,m}$, which indicate the asymptotic behavior of the code size. Since their derivations are similar to those in [4], [7], we only give the results and the intermediate steps are omitted. Let $r_k$ be the real root but 1 of the equation $x = 1/(2-z^{-k})$. Then, for large $m$, we obtain

$$G_{k,m} \approx \frac{1}{k-1} \left( 1 - \frac{r_k}{2kr_k - k - 1} \right) r_k^{-(m+1)} \quad (14)$$

$$M_{k,m} \approx \frac{1}{k-1} \left( 1 - \frac{r_k}{2kr_k - k - 1} \right) r_k^{-(m+1)} \quad (15)$$

As an example, approximations of $G_{4,m}$ and $M_{4,m}$ are shown in Table I. Since the first term of the right-hand side of (14) is a fractional number for any integer $k > 2$ and the second term increases as $m$ goes to infinity, we state that

$$M_{k,m} \approx \frac{1}{r_k} G_{k,m}, \quad \text{for } m \gg 1. \quad (16)$$

Thus the difference in redundancy between $\mathcal{M}_{P_G}^{(k+m)}$ and $\mathcal{G}_{P_G}^{(k+m)}$ is approximately given by $\log_2 r_k/(1-r_k)$. Note that this difference is determined only by the prefix length $k$ and does not depend on the constrained sequence length $m$. For example, for $k$ equal to 4, 6, and 8, the difference is equal to 0.253, 0.050, and 0.011, respectively.

### IV. CODING ALGORITHMS FOR MPS-CODES OF PREFIX $P_G$

In this section, we present the encoding and decoding algorithm for a class of PS-codes $G_{P_G}^{(k+m)}$ of prefix $P_G = 1-k-10$ for arbitrary $k \geq 1$ and $m \geq 1$. Note that this class contains the class of binary Gilbert's PS-codes $G_n$. The algorithms will be extended for any self-uncorrelated prefix in Section IV. For the sake of simplicity, we will only discuss the binary alphabet case in this paper. The extension to nonbinary alphabet, however, can be easily obtained using the same arguments that have been developed here.

#### A. Encoding Algorithm

Theorem 2 shows that $F_{P_G}^{(m)}$ can be subdivided into $k+\alpha-2$ distinct subsets. By recursively applying this theorem to each subset except the singleton set (consisting of only one element), we know that $F_{P_G}^{(m)}$ can be represented as a collection of $G_{k,m}$ singleton sets. We assume that input data is represented as a stream of binary block sequences, each of which corresponds to a number $x$ with $0 \leq x < G_{k,m}$. For a given $m$ and $y$, with $0 \leq y < 2^m$, let $\beta_m(y)$ be an $m$-bit binary sequence $\beta_m(y) = b_1 b_2 \cdots b_m$ such that

$$y = \sum_{i=1}^{m} b_{m+1-i} 2^{i-1}$$

Conversely, for each binary sequence $W$ of length $m$, let $\beta_m^{-1}(W)$ be a number $y$ such that $\beta_m(y) = W$.

The main task of the encoding algorithm is to find a singleton set corresponding to an input $x$ with $0 \leq x < G_{k,m}$. The encoding algorithm consists of two parts: EncodePSC($k,m,x$) and CodePSC($k,m,x$). EncodePSC($k,m,x$) calls
CodePSC\((k, m, x)\) to get sequence \(\Omega_{k,m}(x)\) corresponding to a number \(x, 0 \leq x < G_{k,m}\) and then returns the concatenation of \(P_G\) and \(\Omega_{k,m}(x)\). The task of CodePSC\((k, m, x)\) is to construct \(\Omega_{k,m}(x)\) with recursive calls
\[
X = \text{EncodePSC}(k, m, x)
\]
Return \(X = P_G\text{CodePSC}(k, m, x)\).

(End of EncodePSC)

\[
Y = \text{CodePSC}(k, m, x)
\]
begin
1. if \((m \geq k)\) then begin
2. \(t := 1; y := x;\)
3. while \((y \geq G_{k,m-t})\) do begin
4. \(y := y - G_{k,m-t};\)
5. \(t := t + 1;\)
end;
6. if \((t = k)\) then return \((Y = 1^m)\)
7. else return \((Y = 1^{t-1}0\text{CodePSC}(k, m - t, y))\)
8. end else return \((Y = \beta_m(x))\)
end

(End of CodePSC)

**Example:** Let us consider the encoding procedure of \(G_{10}^{(4)}\) with prefix \(P_G = 1110\) where \(m = 6\) and \(k = 4\).

EncodePSC\((4, 6, x)\) converts a number \(x\) from 0 to 51 into a codeword in \(G_{10}^{(4)}\). For instance, tracing the encoding procedure for \(x = 17\), we obtain the codeword 1110\(\Omega_{4,6}(17) = 1110010010\). Similarly, we obtain \(\Omega_{4,6}(3) = 000011\) and \(\Omega_{4,6}(42) = 101111\).

As shown in this example, the value of \(\Omega_{k,m}(x)\) is recursively determined during the encoding process. Let
\[
L_{k,m}[i] = \begin{cases} 0, & i = 1 \\ \sum_{j=1}^{i-1} G_{k,m-j}, & 1 < i < k. \end{cases}
\]

Let us denote the set of integers \(\{0, 1, \cdots, G_{k,m} - 1\}\) by \(I_{k,m}\) for \(m \geq 1\). Then, we divide \(I_{k,m}\) into \(k\) distinct sets \(I_{k,m}[i]\) with \(1 \leq i \leq k\), which are defined by
\[
I_{k,m}[i] = \begin{cases} \{ j | L_{k,m}[i] \leq j < L_{k,m}[i+1] \}, & 1 \leq i < k \\ \{ G_{k,m} - 1 \}, & i = k. \end{cases}
\]

(17)

**Theorem 4:** \(\Omega_{k,m}\) is a one-to-one mapping from \(I_{k,m}\) onto \(F_{P_G}^{(m)}\).

**Proof:** If \(m < k\), then \(\Omega_{k,m}(x)\) is the \(m\)-bit binary representation of \(x\) with \(0 \leq x < G_{k,m} = 2^m\), and \(F_{P_G}^{(m)} = A_2^m\). Therefore, the theorem holds. For \(m \geq k\), we use induction. We assume that \(\Omega_{k,m-1}\) is a one-to-one mapping from \(I_{k,m-1}\) onto \(F_{P_G}^{(m-1)}\). Then, we prove that \(\Omega_{k,m}\) is a one-to-one mapping from \(I_{k,m}\) onto \(F_{P_G}^{(m)}\).

First, we will show \(\Omega_{k,m}\) maps \(I_{k,m}\) into \(F_{P_G}^{(m)}\). Suppose that \(x \in I_{k,m}[k]\), that is, \(x = G_{k,m} - 1\). Then, the while-loop at step 3 in CodePSC\((k, m, x)\) is repeated \(k - 1\) times since
\[
G_{k,m} - 1 = G_{k,m-1} + G_{k,m-2} + \cdots + G_{k,m-k+1}.
\]

At the \(k\)th repetition of step 3, \(y = 0, t = k\), and \(G_{k,m-k} > 0\). Therefore, the sequence \(1^m\) is returned (step 6). It implies that \(\Omega_{k,m}\) maps \(G_{k,m} - 1\) into \(1^m\). Next, suppose that \(x \in I_{k,m}[i] i < k\). Then
\[
\sum_{j=1}^{i-1} G_{k,m-j} \leq x < \sum_{j=1}^{i} G_{k,m-j}.
\]

(18)

The while-loop at step 3 is repeated \(i - 1\) times until the \(i\)th repetition, when
\[
y = x - \sum_{j=1}^{i-1} G_{k,m-j} < G_{k,m-i}
\]
and \(t = i\). In step 6, since \(i < k\) holds, CodePSC\((k, m - i, y)\) is called in step 7. Hence, if \(x \in I_{k,m}[i] (i < k)\), we can write
\[
\Omega_{k,m}(x) = 1^{i-1}0\Omega_{k,m-i}(y)
\]
where
\[
y = x - \sum_{j=1}^{i-1} G_{k,m-j}.
\]

From the assumption of induction, \(\Omega_{k,m-i}(z) \in F_{P_G}^{(m-i)}\) holds for \(z \in I_{k,m}[m-i] \) and \(1 \leq i \leq m\). Since \(0 \leq y < G_{k,m-i}, y\) must be in \(I_{k,m-i}\). Therefore, \(\Omega_{k,m-i}(y) \in F_{P_G}^{(m-i)}\). Equation (19) shows that \(P_G\) does not appear in \(1^{i-1}0\Omega_{k,m-i}(y)\) if \(1 \leq i < k - 1\). Thus
\[
\Omega_{k,m}(x) \in F_{P_G}^{(m)}.
\]

(20)

Since (20) holds for any \(x \in I_{k,m}[i]\) and \(1 \leq i \leq k\), we have \(\Omega_{k,m}(I_{k,m}) \subset F_{P_G}^{(m)}\).

Now, we will show that \(\Omega_{k,m}(x)\) is one-to-one. If \(x \in I_{k,m}[i]\) and \(y \in I_{k,m}[j] (i \neq j)\), the sequences corresponding to those numbers have distinct prefixes. Therefore, \(\Omega_{k,m}(x) \neq \Omega_{k,m}(y)\). In case that \(x\) and \(y\) belong to the same \(I_{k,m}[i]\), the corresponding sequences can be represented as
\[
\Omega_{k,m}(x) = 1^{i-1}0\Omega_{k,m-i}(x')
\]
\[
\Omega_{k,m}(y) = 1^{i-1}0\Omega_{k,m-i}(y')
\]
respectively, where
\[
x' = x - \sum_{j=1}^{i-1} G_{k,m-j}
\]
and
\[
y' = y - \sum_{j=1}^{i-1} G_{k,m-j}.
\]

From the assumption of induction, \(\Omega_{k,m-i}(x') \neq \Omega_{k,m-i}(y')\). Hence, \(\Omega_{k,m}(x) \neq \Omega_{k,m}(y)\) holds.

The time complexity of the encoding algorithm is evaluated as the number of comparisons of possibly large numbers \(y\) and \(G_{k,m-i}\) at step 3 and the number of recursive calls at step 7. The sum of the numbers of comparisons and recursive calls is upper-bounded by \(m\). Hence, the time complexity is \(O(m)\). At most \(m\) values of \(G_{k,i}\) \((1 \leq i \leq m)\) must be stored in memory to invoke CodePSC\((k, m, x)\). Since \(G_{k,i}\) can be represented by at most \(m\) bits, the total amount of memory is \(O(m^2)\).
B. Decoding Algorithm

Suppose that the decoder receives a series of codewords in $G^{(k+m)}_{P_G} = P_{G'}^{(m)}$. After finding prefix $P_G$ followed by an $m$-bit block $W = (w_1, w_2, \ldots, w_m)$ from the received sequence, the decoder converts $W$ into a number $x$ where $0 \leq x < G_{k,m}$. The following decoding algorithm returns a unique number for any $W \in P_{P_G}^{(m)}$:

$x = \text{DecodePSC}(k, m, W)$
begin
  1. if $(m \geq k)$ then begin
  2. if there exists $1 \leq i < k$ such that $W = 1^{i-1}0V$ then
  3. return $(x = \sum_{j=1}^{i-1} G_{k,m-j} + \text{DecodePSC}(k, m-i, V))$
  4. else return $(x = G_{k,m} - 1)$
  5. end else return $(x = \beta^{-1}_m(W))$
end

(End of DecodePSC)

Let $\Xi_{k,m}(W)$ denote the returned value of DecodePSC$(k, m, W)$. Then, $\Xi_{k,m}$ maps $P_{P_G}^{(m)}$ to the set of integers.

**Theorem 5:** $\Xi_{k,m}$ is the inverse mapping of $\Omega_{k,m}$.

**Proof:** If $m < k$, then

$$\Xi_{k,m}(W) = \text{DecodePSC}(k, m, W)$$

is a number $x$ such that $W$ equals the $m$-bit binary representation of $x$. Given $x$, EncodePSC$(k, m, x)$ returns the $m$-bit binary representation of $x$ ($0 \leq x < 2^m$). Hence, $\Xi_{k,m}(\Omega_{k,m}(x)) = x$. Now, suppose that

$$\Xi_{k,m-1}(\Omega_{k,m-1}(x)) = x$$

holds for $m \geq k$ and $x \in I_{k,m-1}$. We will show that the assumption also holds when $m - 1$ is replaced by $m$. First we consider the case $x = G_{k,m} - 1$. As shown in the proof of Theorem 4, $\Omega_{k,m}(G_{k,m} - 1) = 1^m$ holds. Moreover, DecodePSC$(k, m, 1^m)$ returns $G_{k,m} - 1$. Hence,

$$\Xi_{k,m}(\Omega_{k,m}(G_{k,m} - 1)) = G_{k,m} - 1$$

holds. Assume that $x \in I_{k,m}[i]$ for $1 \leq i < k$. Then, there exists a value $r$ such that

$$x = \sum_{j=1}^{i-1} G_{k,m-j} + r$$

for which $0 \leq r < G_{k,m-i})$. Moreover, according to (19), $\Omega_{k,m}(x)$ can be written as

$$\Omega_{k,m}(x) = 1^{i-1}0\Omega_{k,m-i}(r).$$

With $W = \Omega_{k,m}(x)$ and $V = \Omega_{k,m-i}(r)$, $\Xi_{k,m}(W)$ can be written as

$$\Xi_{k,m}(W) = \sum_{j=1}^{i-1} G_{k,m-j} + \Xi_{k,m-i}(V).$$

From the assumption of induction, $\Xi_{k,m-i}(V) = r$. Therefore

$$\Xi_{k,m}(\Omega_{k,m}(x)) = \sum_{j=1}^{i-1} G_{k,m-j} + \Xi_{k,m-i}(\Omega_{k,m-i}(r))$$

$$= \sum_{j=1}^{i-1} G_{k,m-j} + r$$

$$= x.$$

The proof is complete. $\square$

Although DecodePSC needs the same amount of memory for storing the values of $G_{k,j}$ as EncodePSC does, the decoder is much faster than the encoder since no comparisons of two large integers are required in the decoding process.

V. CONSTRUCTION OF MPS CODES WITH ARBITRARY SELF-UNCORRELATED PREFIXES

In practical situations, one might want to use another prefix than $P_G = 1^{k-1}0$. Although we can easily obtain the encoding and decoding algorithms for the negation of $P_G$ or the reversed $P_G$, it seems to be hard to obtain a recursive relation on the partitions even for any self-uncorrelated prefix other than $P_G$. We will present the encoding and decoding algorithms for $G^{(k+m)}$ with a self-uncorrelated prefix $Q$. As in the previous section, we will only consider the binary alphabet case for the sake of simplicity. The algorithm can be easily extended to nonbinary alphabets.

We will describe a mapping $\Phi_Q$ to transform each sequence in $P_{P_G}^{(m)}$ into another $P_{Q}^{(m)}$ where $Q$ is any prefix but $P_G$. If $Q$ is self-uncorrelated, that is, $Q \circ Q = 10^{k-1}$, then it is shown that $\Phi_Q(P_{P_G}^{(m)}) = P_{Q}^{(m)}$. As a byproduct of this result, we obtain another proof of the statement that $G^{(k+m)}$ with a self-uncorrelated prefix $Q$ has the same size as $G^{(k+m)}$ [5].

The main idea of the mapping is to uniquely transform a sequence $X = \Omega_{k,m}(x)$ obtained from EncodePSC$(k, m, x)$, to another sequence in $P_{Q}^{(m)}$ where $Q$ has the same length as $P_G$. Scanning $X$ from the left to the right, check pattern $Q$ for the occurrence of $Q$. If we find $Q$ as a subsequence of $X$, this subsequence is replaced by $P_G$. Let us denote the sequence obtained after the transformation by $\tau_{Q-P_G}(X)$. After the conversion, no $Q$ is supposed to appear in anywhere in $\tau_{Q-P_G}(X)$, which means that the obtained sequence would be in $P_{Q}^{(m)}$. Unfortunately, $\tau_{Q-P_G}(X) \in P_{Q}^{(m)}$ does not always hold, since the replacement of $Q$ with $P_G$ may cause $Q$ to occur at a position which has been scanned before. For example, for $P_G = 1110$ and $Q = 1011$, let us convert the sequence $W = 0101011001 \in P_{P_G}^{(10)}$. When we scan this sequence, $Q$ is found at the fourth position, and $W$ is converted to $0101110001$. However, another $Q$ now appears at the second position. Hence, replacing $Q$ by $P_G$ might result in scanning the sequence again and again. Fortunately, if we replace $P_G$ by $\overline{P_G}$, then no $Q$ appears at the position of $W$ scanned before. In fact, $W$ is converted to $0100001001$ which belongs to $P_{\overline{P_G}}^{(10)}$. In general, there exists a one-path scanning from left to right to uniquely transform $P_{P_G}^{(m)}$ to $P_{Q}^{(m)}$ or $P_{\overline{P_G}}^{(m)}$ to $P_{Q}^{(m)}$. In the rest of this section, we will show this method.
Lemma 2: Let $S$ and $T$ be distinct sequences of length $k$. Let $X$ be a sequence in $\mathcal{F}_S^{(m)}$. If $T \circ S = 0^k$, then $S \tau \circ S(X) \in \mathcal{F}_T^{(m)}$.

Proof: Let us assume that $T$ is found for the first time at the $i$th position of $X$. It means that $X$ can be written as $X = VTW$, where $V$ is a string of length $i = 1$ such that $VT \circ T = 0^{i-1}(s)k^{-1}$ and $W$ is the remaining part of $X$. By replacing $T$ with $S$ at the $i$th position, $X$ is converted to $VSW$.

To prove the Lemma, it is sufficient to show that $VS \circ T = 0^i(s)k^{-1}$. If another $T$ is found at the $j$th position, the prefix $V'$ of length $j - 1$ followed by $T$ satisfies $V'T \circ T = 0^{j-1}(s)k^{-1}$ and the situation is equivalent.

Suppose that the $h$th symbol of $VS \circ T$ is one, where $1 \leq h < i - 1$. Then, $T$ can be written as $T = T_1T_2$, where $T_1$ is equal to a suffix of $V$ and $T_2$ is a prefix of $S$. However, this contradicts $T \circ S = 0^k$. Therefore, we have $VS \circ T = 0^i(s)k^{-1}$.

Lemma 3: For $Q \in \mathcal{A}_2^k$, if the last symbol of $Q$ is zero and $Q \neq P_G$, then $Q \circ P_G = 0^k$. If $Q = P_G$, then $Q \circ P_G = 0^k$.

Proof: If $q_k = 0$, $q_k$ is different from any symbol of $P_G$ but the last one. Hence,

$$Q \circ P_G = 0^k, \quad \text{if } Q \neq P_G.$$  

If $q_k = 1$, it is different from any symbol of $\overline{P_G}$ but the last one. Hence,

$$Q \circ \overline{P_G} = 0^k, \quad \text{if } Q \neq \overline{P_G}. \quad \square$$

Lemma 3 guarantees that if $P_G$ or $\overline{P_G}$ is chosen appropriately depending on the “target” prefix $Q$, then the sufficient condition of Lemma 2 holds.

Using Lemmas 2 and 3, let us construct a mapping

$$\Phi^{(m)}_Q : \mathcal{F}^{(m)}_{P_G} \rightarrow \mathcal{F}^{(m)}_Q$$

as follows:

$$\Phi^{(m)}_Q(X) = \begin{cases} \tau_{Q \rightarrow P_G}(X), & \text{if } q_k = 0 \\ \tau_{Q \rightarrow \overline{P_G}}(X), & \text{otherwise} \end{cases}. \quad (21)$$

We note that $\Phi^{(m)}_Q(X) \in \mathcal{F}^{(m)}_Q$ holds for $X \in \mathcal{F}^{(m)}_{P_G}$.

Theorem 6: For any $Q \in \mathcal{A}_2^k$, $\Phi^{(m)}_Q$ is a one-to-one mapping from $\mathcal{F}^{(m)}_{P_G}$ to $\mathcal{F}^{(m)}_Q$ for $m \geq 1$.

Proof: We will show that

$$\Phi^{(m)}_Q(X) \neq \Phi^{(m)}_Q(Y) \quad \text{if } X \neq Y$$

for any pair $X, Y \in \mathcal{F}^{(m)}_{P_G}$. This statement will be proved using induction on $m$. First, let us consider the case that $q_k = 0$. If $m < k$, then the statement holds since $\Phi^{(m)}_Q(X) = X$ for $X \in \mathcal{A}_2^m$ and $X$ belongs to both $\mathcal{F}^{(m)}_{P_G}$ and $\mathcal{F}^{(m)}_Q$ since $\mathcal{F}^{(m)}_{P_G} = \mathcal{F}^{(m)}_Q = \mathcal{A}_2^m$ from (2). Next, assume $m = k$. Then, if $X = Q$, then $\Phi^{(k)}_Q(Q) = P_G$. If $X \in \mathcal{F}^{(k)}_{P_G} \setminus \{Q\}$, then $\Phi^{(k)}_Q(X) = X$. Since $Q \in \mathcal{F}^{(k)}_{P_G}$ and $P_G \in \mathcal{F}^{(k)}_Q$, $\Phi^{(k)}_Q$ is one-to-one. Assuming that the statement holds for $m = t$, we will now prove that the statement holds for $m = t + 1$. Let $X = x_1x_2 \cdots x_{t+1}$ and $Y = y_1y_2 \cdots y_{t+1}$ be in $\mathcal{F}^{(t+1)}_{P_G}$. We assume $X \neq Y$. Let $W'$ denote the subsequence of $W$ obtained by removing the first symbol. That is, $X' = (x_2, \cdots , x_{t+1})$ and $Y' = (y_2, \cdots , y_{t+1})$.

According to the values of the first $k$ symbols of $X$ and $Y$, we have four cases to consider:

i) $X_k \neq Q$ and $Y_k \neq Q$,

ii) $X_k = Q$ and $Y_k = Q$,

iii) $X_k = Q$ and $Y_k \neq Q$,

iv) $X_k \neq Q$ and $Y_k = Q$,

where $X_k = x_1x_2 \cdots x_k$ and $Y_k = y_1y_2 \cdots y_k$. For case i), we obtain

$$\begin{align*} \Phi^{(t+1)}_Q(X) &= x_1 \Phi^{(t)}_Q(X') \\ \Phi^{(t+1)}_Q(Y) &= y_1 \Phi^{(t)}_Q(Y'). \end{align*} \quad (22)$$

By the hypothesis of induction, we obtain

$$\begin{align*} \Phi^{(t+1)}_Q(X) &= 1 \Phi^{(k-2)}_Q(1^{k-2}0R) \\ \Phi^{(t+1)}_Q(Y) &= 1 \Phi^{(k-2)}_Q(1^{k-2}0S). \end{align*} \quad (23)$$

By the hypothesis of induction, we obtain $\Phi^{(t+1)}_Q(X) \neq \Phi^{(t+1)}_Q(Y)$ since $R \neq S$. Because of the symmetry of both cases iii) and iv), it is sufficient to consider case iii). In case iii), $X$ and $Y$ can be written as $X = QR$ and $Y = QS$, respectively. Then, we have

$$\begin{align*} \Phi^{(t+1)}_Q(X) &= 1 \Phi^{(k-2)}_Q(1^{k-2}0) \\ \Phi^{(t+1)}_Q(Y) &= 1 \Phi^{(k-2)}_Q(1^{k-2}0). \end{align*} \quad (24)$$

If $y_1 \neq 1$, then the statement is obviously true. Otherwise, $X'$ or $Y'$ must hold from the assumption. Then, by the inductive hypothesis, we have $\Phi^{(t)}_Q(X') \neq \Phi^{(t)}_Q(Y')$. Next, we consider case ii). By choosing appropriate sequences $R$ and $S$, $(R \neq S)$ of length $t + 1 - k$, $X$ and $Y$ can be written as $X = QR$ and $Y = QS$, respectively. Then, we have

$$\begin{align*} \Phi^{(t+1)}_Q(X) &= 1 \Phi^{(t)}_Q(1^{k-2}0) \\ \Phi^{(t+1)}_Q(Y) &= 1 \Phi^{(t)}_Q(1^{k-2}0). \end{align*} \quad (25)$$

By the hypothesis of induction, we obtain $\Phi^{(t+1)}_Q(X) \neq \Phi^{(t+1)}_Q(Y)$ since $R \neq S$. Because of the symmetry of both cases iii) and iv), it is sufficient to consider case iii). In case i), $X$ and $Y$ can be written as $X = QR$ and $Y = QS$, respectively. Then, we obtain

$$\begin{align*} \Phi^{(t+1)}_Q(X) &= 1 \Phi^{(t)}_Q(1^{k-2}0) \\ \Phi^{(t+1)}_Q(Y) &= 1 \Phi^{(t)}_Q(1^{k-2}0). \end{align*} \quad (26)$$

If $y_1 \neq 1$, then the statement is true. Otherwise, $Y' \neq 1^{k-2}0$, since $Y$ is assumed to be in $\mathcal{F}^{(m)}_{P_G}$. Hence, $Y' \neq 1^{k-2}0T$. By the inductive hypothesis, we have

$$\begin{align*} \Phi^{(t+1)}_Q(1^{k-2}0T) &= \Phi^{(t+1)}_Q(Y'U) \quad (27) \end{align*}$$

Now we consider the case that $q_k = 1$. Using the same argument as for $q_k = 0$, we take the negation of $Q$, and show $\tau_{Q \rightarrow P_G}$ is a one-to-one mapping from $\mathcal{F}^{(m)}_{P_G}$ to $\mathcal{F}^{(m)}_Q$. Note that

$$\mathcal{F}^{(m)}_Q = \overline{\mathcal{F}^{(m)}_Q}$$

and the correspondence between $X$ and $\overline{X}$ is one-to-one. Thus

$$\Phi^{(m)}_Q = \tau_{Q \rightarrow P_G}$$

is one-to-one if $q_k = 1$. This completes the proof. \qed
Using Theorem 6, we can construct the coding algorithms of a PS-code $\psi_{Q}^{(m)}$ for a self-uncorrelated prefix $Q$ by combining the algorithms for $P_G$ and one of the transformations $\tau_{Q-P_G}$ and $\tau_{Q-P_{G'}}$. The PS-code $\psi_{Q}^{(m)}$ constructed in this way is a subset of $\psi_{Q}^{(m)}$ because of the existence of a one-to-one mapping from $\mathcal{T}(m)$ to $\mathcal{T}(m)$. In what follows, we will show that $\psi_{Q}^{(m)}$ equals $\psi_{Q}^{(m)}$.

Now let us assume that $qk = 0$, and consider a conversion $\tau_{Q-P_G}$ from $\mathcal{T}(m)$ to $\mathcal{T}(m)$ as follows: by scanning $W \in \mathcal{T}(m)$ from right to left, look for pattern $P_G$ in $W$. If $P_G$ is found as a subsequence of $W$, then it will be replaced by $Q$. We continue the above operation until we reach the left-end of $W$. The obtained sequence is denoted by $\mathcal{T}(m)$. That is, $\tau_{Q-P_G}$ is a mapping from $\mathcal{T}(m)$ into $\mathcal{T}(m)$. If $qk = 1$, replace $P_G$ by $P_G$. Then we similarly obtain a conversion $\tau_{Q-P_G}$ which transforms $\mathcal{T}(m)$ into $\mathcal{T}(m)$. Now we define a mapping $\psi_{Q}^{(m)}(X)$ as follows:

$$
\psi_{Q}^{(m)}(X) = \begin{cases} 
\tau_{Q-P_G}(X), & \text{if } qk = 0 \\
\tau_{Q-P_{G'}}(X), & \text{if } qk = 1.
\end{cases}
$$

From the definition, $\psi_{Q}^{(m)}(X)$ is a mapping from $\mathcal{T}(m)$ to $\mathcal{T}(m)$. Let us note that

$$
\tau_{Q-P_G}(X) = \tau_{Q-P_{G'}}(X). \quad (28)
$$

**Theorem 7:** If $Q$ is self-uncorrelated, then $\psi_{Q}^{(m)}$ is a one-to-one mapping from $\mathcal{T}(m)$ to $\mathcal{T}(m)$.

**Proof:** Using the same arguments as in Theorem 6, we show that $\psi_{Q}^{(m)}$ is one-to-one for $m \leq k$. Next, assuming that the statement holds for $m = t$, we will now prove that it also holds for $m = t + 1$ with the condition $qk = 0$. Let two distinct sequences $X = (x_1, x_2, \ldots, x_{t+1})$ and $Y = (y_1, y_2, \ldots, y_{t+1})$ be in $\mathcal{T}(m)$. Using the same notation as in Theorem 6, $X$ and $Y$ can be written as $x_1 X'$ and $y_1 Y'$, respectively. Let us write

$$
\psi_{Q}^{(t)}(X') = M_X R_X \quad (29)
$$
$$
\psi_{Q}^{(t)}(Y') = M_Y R_Y \quad (30)
$$

where $M_X$ and $M_Y$ are sequences of length $k - 1$, and $R_X$ and $R_Y$ are sequences of length $t + 1 - k$. Before the last step of the conversion $\tau_{Q-P_G}$, sequences $X$ and $Y$ must have been converted into $x_1 M_X R_X$ and $y_1 M_Y R_Y$, respectively. Then, $R_X \neq R_Y$ or $R_Y \neq R_Y$ may occur. In the former case, it always holds that $\psi_{Q}^{(t)}(X') \neq \psi_{Q}^{(t)}(Y')$ since neither $R_X$ nor $R_Y$ changes when converting the last part of $X$ and $Y$. Hence, the only case where $\psi_{Q}^{(t)}(X') = \psi_{Q}^{(t)}(Y')$ holds is the latter case.

Assume that $R_X = R_Y$. If $x_1 M_X = y_1 M_Y$, then $\psi_{Q}^{(t)}(X') = \psi_{Q}^{(t)}(Y')$ follows. From the inductive hypothesis, $X' = Y'$ holds. Hence, $X = Y$ holds since $x_1 = y_1$. This contradicts $X \neq Y$. Thus $x_1 M_X \neq y_1 M_Y$ always holds when $R_X = R_Y$.

Now, we assume that $x_1 M_X \neq y_1 M_Y$. Without loss of generality, it is sufficient to consider the following four cases:

i) $x_1 M_X = P_G$ and $y_1 M_Y = Q$.

ii) $x_1 M_X = P_G$ and $y_1 M_Y \neq Q$.

iii) $x_1 M_X \neq P_G$ and $y_1 M_Y = Q$.

iv) $x_1 M_X \neq P_G$, $x_1 M_X \neq Q$, $y_1 M_Y \neq P_G$, and $y_1 M_Y \neq Q$.

In cases iii) and iv), neither $x_1 M_X$ nor $y_1 M_Y$ change at the last step of conversion $\tau_{Q-P_G}$. From the assumption, $x_1 M_X \neq y_1 M_Y$ holds. Hence, we have

$$
\psi_{Q}^{(t+1)}(X) \neq \psi_{Q}^{(t+1)}(Y).
$$

In case ii), $x_1 M_X$ is converted into $Q$, while $y_1 M_Y$ keeps the same values. Hence,

$$
\psi_{Q}^{(t+1)}(X) \neq \psi_{Q}^{(t+1)}(Y)
$$

also follows.

The only remaining case to be considered is case i). We aim to show that case i) never holds if $Y \in \mathcal{T}(m)$. Let us break $Y$ into two parts, $Y_M = (y_2, \ldots, y_k)$ and $Y_R = (y_{k+1}, \ldots, y_{t+1})$. Since $Y \in \mathcal{T}(m)$, $Y_M$ is not equal to $Q$ while $y_1 M_Y = Q$, which implies that a $P_G$ exists with overlapping $Y_M$ and $y_1 M_Y$. That is, a suffix of $M_Y$ equals a prefix of $P_G$ and a suffix of $P_G$ equals a prefix of $Y_R$. Then $Y_M$ is the sequence obtained after converting the suffix of $Y_M$ into the corresponding prefix of $Q$. And let $Y_R$ be the sequence obtained after converting the prefix of $Y_R$ into the corresponding suffix of $Q$. If $Y_M Y_R$ contains no $P_G$, then it equals $M_Y R_Y$. Hence, a suffix of $M_Y$ equals a prefix of $Q$. Even if $P_G$ is found again before the scan reaches the left end of $Y_M Y_R$, a suffix of $M_Y$ equals a prefix of $Q$ after all. This contradicts the assumption that $Q \circ Q = 10^{k-1}$.

If $qk = 0$ and $Q$ is replaced by $Q$, we can use the same arguments as for $qk = 0$ to show that $\tau_{Q-P_G}$ is a one-to-one mapping from $\mathcal{T}(m)$ to $\mathcal{T}(m)$.

Since $qk = 0$, from the arguments developed above, we have that $\tau_{Q-P_G}$ is a one-to-one mapping from $\mathcal{T}(m)$ to $\mathcal{T}(m)$.

$$
\psi_{Q}^{(m)} = \mathcal{T}(m) \quad (28)
$$

$\psi_{Q}^{(m)}$ is proved to be a one-to-one mapping from $\mathcal{T}(m)$ to $\mathcal{T}(m)$.

**Corollary 1:** Let $P_G$ be $10^{k-1}0$ and let $Q$ be a sequence of length $k$. If $Q \circ Q = 10^{k-1}$, then

$$
|\mathcal{T}(m)| = |\mathcal{T}(m)|, \quad m \geq 1.
$$

**Proof:** According to Theorem 6, there exists a one-to-one mapping from $\mathcal{T}(m)$ to $\mathcal{T}(m)$. Thus, the inequality $|\mathcal{T}(m)| \leq |\mathcal{T}(m)|$ always holds. If $Q \circ Q = 10^{k-1}$, there exists a one-to-one mapping from $\mathcal{T}(m)$ to $\mathcal{T}(m)$ according to Theorem 7. Hence, $\mathcal{T}(m)$ has the same size as $\mathcal{T}(m)$.
VI. CONCLUSION

Encoding and decoding algorithms for a class of MPS-codes have been presented. The key idea used in the algorithms is to partition recursively the set $\mathcal{F}_m$ of the constrained sequences of length $m$ in which pattern $P_G = 1^8-10$ does not appear, from which it is straightforward to obtain the algorithms.

Moreover, a method to transform $\mathcal{F}_m$ into $\mathcal{F}_m$ with a self-uncorrelated prefix $P$ has been obtained. Based on this method, the algorithms have been extended to construct MPS codes with arbitrary self-uncorrelated prefixes. The obtained algorithms provide us a variety of options of selecting a prefix since the majority of prefixes used in practical applications of frame synchronization [14] are known to be self-uncorrelated.

The time complexity of the algorithms is proportional to the code length $n$ since we can adopt one of the linear-time string matching algorithms [15] for the transformation.

REFERENCES