Block Codes for Asynchronous Data Transmission Designed from Binary Trees

Vladimir B. Balakirsky

Euler Institute for Discrete Mathematics
and its Applications (EIDMA)
c/o Eindhoven University of Technology
P.O.Box 513, 5600 MB Eindhoven, The Netherlands
v.b.balakirsky@ele.tue.nl

Abstract

We describe a class of codes that can be effectively used when one of q^n vectors of length n has to be delivered to the receiver over a noiseless q–ary channel in asynchronous mode assuming that the latter one receives the transmitted vector with the delay $\tau \in \{0, \ldots, n-1\}$, unknown in advance, while all other received symbols are arbitrarily chosen. The codes are specified for any n by a regular algorithm, which is based on properties of ordered, oriented, rooted, binary trees, and have length $N \approx n + 2\log_q n$. We show that these codes can be used in such a way that encoding and decoding complexities are measured by linear functions of n.

1 Statement of the synchronization problem and formulation of the result

Suppose that $q \geq 2$ is given and denote

$$\mathcal{Q} \triangleq \{0, \ldots, q-1\}.$$

We address the following problem.

- The sender has to deliver one of q^n vectors

$$\mathbf{X} = (X_0, \ldots, X_{n-1}) \in \mathcal{Q}^n$$

  to the receiver over a noiseless q–ary channel. Therefore the sender forms a q–ary vector (codeword) of length $N > n$, denoted by $f(\mathbf{X}) \in \mathcal{Q}^N$, and sends it to the channel.

- The receiver receives the transmitted vector with the delay $\tau \in \{0, \ldots, n-1\}$, which is unknown in advance, i.e.,

$$(Y_\tau, \ldots, Y_{\tau+N-1}) = f(\mathbf{X}) \tag{1}$$

  and all other symbols are arbitrary, where $Y_0, Y_1, \ldots \in \mathcal{Q}$ denote symbols of the received sequence.

- The receiver must uniquely decode the pair $(\tau, \mathbf{X})$ for any received sequence.

An obvious solution to the problem can be illustrated by the example for $q = 2$ and $n = 15$. Let $N = 2n = 30$. First, the sender sends "the syncro–prefix" $\mathbf{x}_0$ consisting of the packet of $(n+1)/2$ zeroes followed by $(n-1)/2$ ones, i.e.,

$$\mathbf{x}_0 = 000000001111111 = 00177\,(\text{octal})$$

and then sends $\mathbf{x}$. For example, if $\mathbf{X} = 00457$, then

$$f(\mathbf{X}) = 0017700457\,(\text{octal}).$$

It is easy to see that the receiver is capable to uniquely decode the pair $(\tau, \mathbf{X})$. However, the number of transmitted bits is increased twice as compared to n. We will present a different solution whose characteristics are summarized below.

Theorem. *For any $q \geq 2$ and n, there is an explicit construction of a q–ary code of length*

$$N = \begin{cases} n + \lceil \log_q n \rceil + \lceil \log_q (n+1) \rceil, & \\ & \text{if } q \text{ is even and } n \text{ is odd} \\ n + 1 + \lceil \log_q (n+1) \rceil + \lceil \log_q (n+2) \rceil, & \\ & \text{otherwise} \end{cases}$$

which brings a solution to the synchronization problem. Furthermore, complexities of encoding and decoding algorithms are measured by linear functions of n.

To prove the theorem we present the construction of a code that can be viewed as a result of adaptation of the coding algorithms for ordered, oriented, rooted, binary trees [1] (these trees will be referred to as regular/non–regular binary trees in foregoing considerations). In this paper, we do not present the analysis of the coding algorithm for trees in details and focus on the synchronization problem.

The following lower bound :

$$N \geq n + \log_q n \tag{2}$$

can be helpful to evaluate the statement of the theorem. Let us consider the set of vectors $\mathbf{Y} = (Y_0, \ldots, Y_{N+n-2}) \in \mathcal{Q}^{N+n-1}$. If $\mathbf{Y}$ is one of q^{n-1} vectors, for which (1) holds, then the decoder has to map this vector to the pair $(\tau, \mathbf{X})$ because of the requirement of unique decoding. There are nq^n pairs and we write

$$nq^n \cdot q^{n-1} \leq q^{N+n-1}$$

since q^{N+n-1} is the total number of vectors, and (2) follows.

Investigations of the synchronization problem in coding theory were initiated by E. N. Gilbert [2]. In particular, it is known [3], [4] that if $q = 2$, then the bound (2) is asymptotically tight in a sense that there are codes for at least $\alpha_n M^2/n$ messages, where $\alpha_n \to \ln\sqrt{2}$, as $n \to \infty$. We design a different class of block codes whose cardinalities are "slightly" smaller in a sense that the ratio N/n still tends to 1, when n increases, as a function $\log n/n$, like for optimal codes. However these codes can be used with the encoding and decoding algorithms having complexities, which are linear in n, while implementation of the known encoding and decoding algorithms (see Section 2) seems to be a difficult computational problem. The construction of our codes for $q = 2$ is based on some properties of regular binary trees [6], which are indicated Section 3. In Sections 4, 5 we describe these codes and present encoding and decoding algorithms. In Section 6 we extend the approach to the case when $q > 2$.

2 Block codes without overlaps

Block codes without overlaps, which are oriented to the solution to the synchronization problem formulated above, were introduced by V. I. Levenshtein [3], [4]. The binary code $\mathcal{C}_n^*$ consisting of codewords of length n is called the "code without overlaps" if and only if $(c_0, \ldots, c_i) \neq (c'_{n-1-i}, \ldots, c'_{n-1})$ for all $(c_0, \ldots, c_{n-1}), (c'_0, \ldots, c'_{n-1}) \in \mathcal{C}_n^*$ and $i = 0, \ldots, n-1$. One can easily understand that any code brings a solution to the synchronization problem if and only if it has this property.

A subclass of binary codes considered in [3] can be presented in the following way. Let all codewords of a code begin with

$$r \stackrel{\triangle}{=} \lfloor \log n - \log \ln 4 \rfloor \tag{3}$$

zeroes and do not "contain" r zeroes, i.e., $(c_i, \ldots, c_{i+r-1}) \neq (0, \ldots, 0)$ for all $\mathbf{c} \in \mathcal{C}_n^*$ and $i = 1, \ldots, n-r$. The decoding algorithm is very simple : the decoder searches for r sequential zeroes in the received sequence $\mathbf{y}$ and decides that the delay is $\tau \in \{0, \ldots, n-1\}$ if $(y_\tau, \ldots, y_{\tau+r-1}) = (0, \ldots, 0)$ and $(y_i, \ldots, y_{i+r-1}) \neq (0, \ldots, 0)$ for all $i = \tau+1, \ldots, \tau+n-r$. The analysis of this scheme is based on the results by H. B. Mann [5], who showed that the total number of binary sequences of length t, do not containing r consecutive zeroes, is equal to the integer, which is the closest one to

$$\frac{\rho^{r+1} - \rho}{\rho^{r+1} - r} \cdot \rho^t$$

where ρ is the largest positive root of the equation

$$\rho^{r+1} - 2\rho^r - 1 = 0.$$

An assignment of the parameter r by (3) leads to the coincidence of the cardinality of a code and $2^n/n$ up to a constant, which tends to $\ln \sqrt{2}$ when n tends to infinity. The results are also rather good for finite values of n. In particular, any string of length 500 can be delivered to the receiver with 512 bits and any string of length 1011 can be delivered to the receiver with 1024 bits in asynchronous mode (the parameter r is 8 and 9 in these examples). Nevertheless, there is a problem with the implementation of the algorithm above meaning complexity of the mapping of the original string to a codeword and vise versa. We will describe another communication system, where we are able to deliver any of 494 and 1002 bits in the examples above.

3 Some properties of regular, ordered, oriented, rooted, binary trees

Let n be a given odd integer and let $\mathcal{T}_n$ denote the set regular binary trees having n nodes. The two trees with $n = 5$ are given in Figure 1. Any tree $T \in \mathcal{T}_n$ can be specified by a binary vector of length n called the preorder codeword [7], which is recurrently introduced as

$$\mathbf{c}(T) \triangleq \begin{cases} 0, & \text{if } T \text{ is an isolated vertex} \\ (1, \mathbf{c}(T_0), \mathbf{c}(T_1)), & \text{otherwise} \end{cases} \tag{4}$$

where T_0 and T_1 are the left and right subtrees of T, respectively.

Let $\mathbf{c} = (c_0, \ldots, c_{n-1})$ be the preorder codeword for some tree having n nodes. Then

$$\begin{cases} c_0 + \ldots + c_{t-1} \geq t/2, & \text{for all } \ t = 1, \ldots, n-1 \\ c_0 + \ldots + c_{n-1} = (n-1)/2 \end{cases} \tag{5}$$

and

$$\begin{cases} \sigma_0 = \sigma_{n-1} = 0, \ \sigma_n = -1 \\ \sigma_1, \ldots, \sigma_{n-2} \geq 0 \end{cases} \tag{6}$$

where

$$\sigma_0 \triangleq 0; \quad \sigma_t \triangleq 2(c_0 + \ldots + c_{t-1}) - t, \quad \text{for all } \ t = 1, \ldots, n. \tag{7}$$

The vector $\boldsymbol{\sigma} = (\sigma_0, \ldots, \sigma_n)$, constructed in this way, can be represented as a polygonal line whose vertices have abscissas $0, \ldots, n$, ordinates $\sigma_0, \ldots, \sigma_n$, and interpreted as a path from the origin to the point $(n, -1)$ (see Figure 1). The relationships (5) and (6) specifying a one–to–one mapping of preorder codewords to the paths on the plane are usually understood as exposition of the *dominating property* in a sense that the number of ones is not less than the number of zeroes in the first $t-1$ positions of the vector $\mathbf{c}$ for all $t = 1, \ldots, n-1$, or, equivalently, in a sense that the path specified by the vector $\boldsymbol{\sigma}$ crosses the X–axes only in the point $(n-1, 0)$.

Let us also introduce the following transformations of the preorder codeword corresponding to sequential cuts of the leaves of the tree (see Figure 2) : the preorder codeword is scanned from right to the left and, as soon as the triple (1,0,0) is found, it is replaced with the triple $(0, *, *)$, where $*$ is a special symbol. As a result, the vector consisting of 0 followed by $n-1$ symbols $*$ is constructed. This procedure will be used in the decoding algorithm.

4 Transformations $\mathbf{x} \to \mathbf{x}' \to \mathbf{z}$ and the encoding algorithm for the case $q = 2$ and n is odd

We denote

$$x_i \triangleq X_i, \quad i = 0, \ldots, n-1. \tag{8}$$

4.1 Transformations $\mathbf{x} \to \mathbf{x}' \to \mathbf{z}$

- Find $j \in \{0, \ldots, n\}$ such that the inversion of the first j components of the vector $\mathbf{x}$ transforms it to the vector $\mathbf{x}' = (x'_0, \ldots, x'_{n-1})$ having $(n+1)/2$ zeroes and $(n-1)/2$ ones, i.e.,

  $$x'_0 + \ldots + x'_{n-1} = (n-1)/2$$

  where

  $$x'_i \triangleq \begin{cases} x_i \oplus 1, & \text{if } i \in \{0, \ldots, j-1\} \\ x_i, & \text{if } i \in \{j, \ldots, n-1\}. \end{cases}$$

- Find $k \in \{0, \ldots, n-1\}$ such that the cyclic shift of the vector $\mathbf{x}'$ by k positions to the right transforms it to the vector $\mathbf{z} = (z_0, \ldots, z_{n-1})$ having the dominating property, i.e.,

  $$z_0 + \ldots + z_{t-1} \geq t/2, \quad \text{for all } \ t = 1, \ldots, n-1$$

  where

  $$z_i \triangleq \begin{cases} x'_{n+i-k}, & \text{if } i \in \{0, \ldots, k-1\} \\ x'_{i-k}, & \text{if } i \in \{k, \ldots, n-1\}. \end{cases}$$

Example of transformations $\mathbf{x} \to \mathbf{x}' \to \mathbf{z}$ is given in Table 1.

4.2 Encoding algorithm

- Form the vector $\mathbf{z}$ and transmit it in the reverse order, i.e., $z_{n-1}, \ldots, z_0$ are sent to the channel.

- Form the standard binary representation of the integer k with $\lceil \log n \rceil$ bits and send it to the channel.

– Form the standard binary representation of the integer j with $\lceil \log(n+1) \rceil$ bits and send it to the channel.

For the example, given in Table 1, the vector $\mathbf{z}$ in the reverse order is

$$001000001111101$$

and the standard binary representations of k and j are

$$\begin{aligned} k = 6 &\quad \rightarrow \quad 01110 \\ j = 1 &\quad \rightarrow \quad 0001. \end{aligned}$$

Thus, the vector

$$f(\mathbf{x}) = 00100000111110101100001$$

of length $15 + 8 = 23$ is the transmitted codeword.

5 The $(0,0,1) \rightarrow (*,*,0)$ procedure and the decoding algorithm for the case $q = 2$ and n is odd

We denote

$$y_i \triangleq Y_i, \quad i = 0, 1, \ldots \tag{9}$$

5.1 Unique decoding of the vectors having the dominating property in asynchronous mode

Definition 1. *For any odd integer n, let the code $\mathcal{C}_n$ consist of preorder codewords for all regular binary trees having n nodes, which are taken in the reverse order, i.e., if $\mathbf{c} = (c_0, \ldots, c_{n-1})$ is a preorder codeword, then $\overline{\mathbf{c}} \triangleq (c_{n-1}, \ldots, c_0) \in \mathcal{C}_n$.*

For example, $\mathcal{C}_5 = \left\{ 00011, 00101 \right\}$.

Proposition 1. *For any $\tau \in \{0,\ldots,n-1\}$, any of transmitted codewords $\overline{\mathbf{c}} \in \mathcal{C}_n$ can be uniquely decoded if*

$$(y_\tau, \ldots, y_{\tau+N-1}) = \overline{\mathbf{c}} \tag{10}$$

and $y_0, \ldots, y_{\tau-1} \in \{0,1\}$ are arbitrary.

Proof : The decoder can represent the received vector as the path on the plane going through the points (t, σ_t'), where $\sigma_t' \triangleq 2(y_0 + \ldots + y_{t-1}) - t$ and $t = 0, 1, \ldots$ (see Figure 3). His goal is to find the position of the axes (τ, σ_τ'), where $\tau \in \{0, \ldots, n-1\}$, in such a way that the first n steps of the path have the dominating property in the reverse order, i.e., $\sigma_i' < \sigma_\tau'$ for all $i = \tau+1, \ldots, \tau+n-1$ and $\sigma_{\tau+n-1}' = \sigma_\tau' - 1$. If a codeword of the code $\mathcal{C}_n$ was transmitted, then a solution to this problem exists, and we only have to check that this solution is unique.

Suppose that $\sigma_i' < \sigma_\xi'$ for all $i = \xi+1, \ldots, \xi+n-1$ and $\sigma_{\xi+n-1}' = \sigma_\xi' - 1$ for some $\xi \in \{0, \ldots, n-1\} \backslash \{\tau\}$. If $\xi > \tau$, then $\sigma_\xi' < \sigma_\xi'$ and $\sigma_{\tau+n-1}' = \sigma_\xi' - 1$ contradicts $\sigma_{\tau+n-1}' < \sigma_\xi'$. If $\xi < \tau$ and $\sigma_\xi' \geq \sigma_\xi$, then $\sigma_{\xi+n-1}' = \sigma_\xi' - 1$ contradicts $\sigma_{\xi+n-1}' < \sigma_\xi'$. It is easy to see that the case $\xi < \tau$ and $\sigma_\xi' < \sigma_\xi'$ is also not possible.

5.2 The $(0,0,1) \rightarrow (*,*,0)$ procedure

Let us introduce the following algorithm for searching for the vector, having the dominating property, in the received sequence $\mathbf{y}$ (see Table 2). The decoder scans the received sequence and maps

$$(0,0,1) \;\rightarrow\; (*,*,0).$$

As soon as the vector consisting of $n-1$ symbols $*$ followed by 0 has been constructed, the value of τ is determined since the position of 0 of this vector is the position of the last bit of the transmitted vector, which has the dominating property.

5.3 Decoding algorithm

- Decode τ by the $(0,0,1) \rightarrow (*,*,0)$ procedure. Set $z_i = y_{\tau+n-1-i}$ for all $i = 0, \ldots, n-1$.

– Decode k from the standard binary representation $(y_{\tau+n}, \ldots, y_{\tau+n+l-1})$ and j from the standard binary representation $(y_{\tau+n+l}, \ldots, y_{\tau+n+2l_1-1})$, where $l = \lceil \log n \rceil$ and $l_1 = \lceil \log(n+1) \rceil$.

– Cyclically shift the vector $\mathbf{z}$ by k positions to the left and invert first j components of the resulting vector.

It is easy to see that complexities of all transformations of the decoding algorithm can be measured by linear functions of n.

Note that the reverse order of the preorder codewords in the code $\mathcal{C}_n$ allows us to organize the decoding procedure in a real time mode. Another important point is that synchronization between the sender and the receiver has been established as soon as the transmitted codeword is decoded. Therefore, the sender can attach any bits to the transmitted codeword, and they will be noiselessly recognized by the receiver at time instants $\tau+n, \tau+n+1, \ldots$ This possibility is used in our transmission scheme to deliver integers k and j to the receiver.

6 Extension of the approach to noiseless q–ary channels

6.1 The case $q = 2$ and n is even

We attach arbitrary "the parity bit" to the vector $\mathbf{x}$ (for example, we set $x_n = 1$ for any input vector $\mathbf{x}$), increase n by 1 and proceed as before.

6.2 The case q is even

We cancel notations (8) and (9) and project the set $\mathcal{Q}$ to $\{0, 1\}$ by

$$x_i \triangleq \begin{cases} 0, & \text{if } X_i \in \{0, \ldots, q/2-1\} \\ 1, & \text{if } X_i \in \{q/2, \ldots, q-1\} \end{cases} \tag{11}$$

for all $i = 0, \ldots, n-1$ and

$$y_i \triangleq \begin{cases} 0, & \text{if } Y_i \in \{0, \ldots, q/2-1\} \\ 1, & \text{if } Y_i \in \{q/2, \ldots, q-1\} \end{cases} \tag{12}$$

for all $i = 0, 1, \ldots$

Having received the vector $\mathbf{X}$, let the encoder construct the vector $\mathbf{x}$ by (11) and compute the integers j and k, like for the binary case, where we assume that n is odd. Then he forms two vectors having components

$$X_i' \triangleq \begin{cases} q-1-X_i, & \text{if } i \in \{0, \ldots, j-1\} \\ X_i, & \text{if } i \in \{j, \ldots, n-1\} \end{cases} \tag{13}$$

and

$$Z_i \triangleq \begin{cases} X_{n+i-k}', & \text{if } i \in \{0, \ldots, k-1\} \\ X_{i-k}', & \text{if } i \in \{k, \ldots, n-1\} \end{cases} \tag{14}$$

for all $i = 0, \ldots, n-1$. The vector $(Z_{n-1}, \ldots, Z_0)$ is transmitted over a channel. Then the sender attaches the q–ary representations of the integers k and j with $\lceil \log_q n \rceil$ and $\lceil \log_q (n+1) \rceil$ symbols, respectively.

Having received the symbols of the sequence $\mathbf{Y}$, let the decoder construct the sequence $\mathbf{y}$ by (12) and determine the value of $\tau \in \{0, \ldots, n-1\}$ based on the dominating property of the codeword included in the this vector. If τ is known, then the vector $\mathbf{Z}$ and the pair of integers (k, j) are also known. The transformation of $\mathbf{Z}$ to $\mathbf{X}$ is straightforward.

If n is even, then we translate the q–ary case to the binary one in a similar way.

6.3 The case $q = 3$

We return to the notations (8) and (9).

Definition 2. *For an integer n, let the code $\tilde{\mathcal{C}}_n$ consist of preorder codewords for all non–regular binary trees having n nodes, which are taken in the reverse order.*

For example,

$$\tilde{\mathcal{C}}_5 = \Big\{ 00112, 01102, 01012, 00121, 01021, 00211, 00022, 00202, 01111 \Big\}.$$

The generalized version of the mapping $\mathbf{x} \to \mathbf{x}' \to \mathbf{z}$ is presented below.

– Set

$$x_n \triangleq \begin{cases} 1, & \text{if } n - n_1 \text{ is odd} \\ 0, & \text{if } n - n_1 \text{ is even} \end{cases}$$

where n_1 is the number of 1's in the vector $(x_0, \ldots, x_{n-1})$.

– Find $j \in \{0, \ldots, n+1\}$ such that the substitutions $0 \to 2$ and $2 \to 0$ at the first $j-1$ components of the vector $\mathbf{x} = (x_0, \ldots, x_n)$ transform it to the vector $\mathbf{x}' = (x'_0, \ldots, x'_n)$ such that

$$(x'_0 - 1) + \ldots + (x'_n - 1) = (n-1)/2$$

where

$$x'_i \triangleq \begin{cases} 2, & \text{if } x_i = 0 \text{ and } i \in \{0, \ldots, j-1\} \\ 0, & \text{if } x_i = 2 \text{ and } i \in \{0, \ldots, j-1\} \\ x_i, & \text{if } x_i = 1 \text{ or } i \in \{j, \ldots, n\}. \end{cases}$$

– Find $k \in \{0, \ldots, n\}$ such that the cyclic shift of the vector $\mathbf{x}'$ by k positions to the right transforms it to the vector $\mathbf{z} = (z_0, \ldots, z_{n-1})$ having the dominating property, i.e.,

$$(z_0 - 1) + \ldots + (z_{t-1} - 1) \geq t/2, \quad \text{for all } \ t = 1, \ldots, n$$

where

$$z_i \triangleq \begin{cases} x'_{n+i-k}, & \text{if } \ i \in \{0, \ldots, k-1\} \\ x'_{i-k}, & \text{if } \ i \in \{k, \ldots, n\}. \end{cases}$$

After the integers k and j have been found, the encoder proceeds as follows :

– send the vector $(z_n, \ldots, z_0)$ over a channel;

– form the standard q–ary representation of the integer k with $\lceil \log_q n \rceil$ symbols and send it to the channel;

– form the standard q–ary representation of the integer j with $\lceil \log_q (n+1) \rceil$ symbols and send it to the channel.

The generalized version of the $(0,0,1) \to (*,*,0)$ procedure included in the decoding algorithm is presented below (example is given in Table 3).

1. Set $t = 1$.

2. If $y_t = d \neq 0$ and there are d indices $t_1, \ldots, t_d$ such that $t > t_1 > \ldots > t_d \geq 0$ with $y_{t_1} = \ldots = y_{t_d} = 0$ and $y_j = *$ for all $j \in \{t_d, \ldots, t-1\} \backslash \{t_1, \ldots, t_d\}$, then set $y_t = 0$ and $y_{t_1} = \ldots = y_{t_d} = *$.

3. If $y_t = 0$ and $y_{t-j} = *$ for $j = 1, \dots, n$, then set $\tau = t - n$ and go to 5.

4. Increase t by 1 and go to 2.

5. End : the transmitted codeword is written at positions $\tau, \dots, \tau + n$ of the received sequence.

After the receiver discovers τ, he extracts $\mathbf{z}$ and (k, j) from the received sequence. The transformation $\mathbf{z} \to \mathbf{x}$ is obvious.

6.4 The case q is odd

Let us cancel notations (8) and (9) again and project the set $\mathcal{Q}$ to $\{0, 1, 2\}$ by

$$x_i \triangleq \begin{cases} 0, & \text{if } X_i \in \{0, \dots, (q-1)/2 - 1\} \\ 1, & \text{if } X_i = (q-1)/2 \\ 2, & \text{if } X_i \in \{(q+1)/2, \dots, q-1\} \end{cases} \tag{15}$$

for all $i = 0, \dots, n-1$ and

$$y_i \triangleq \begin{cases} 0, & \text{if } Y_i \in \{0, \dots, (q-1)/2 - 1\} \\ 1, & \text{if } Y_i = (q-1)/2 \\ 2, & \text{if } Y_i \in \{(q+1)/2, \dots, q-1\} \end{cases} \tag{16}$$

for all $i = 0, 1, \dots$

Having received the vector $\mathbf{X}$, let the encoder construct the vector $\mathbf{x}$ by (15), attach the n–th bit x_n, and compute the integers j and k, like for the ternary case. To be definite, let us assume that $X_n \triangleq 0$ if $x_n = 0$ and $X_n \triangleq q-1$ if $x_n = 1$. Then the encoder forms two vectors whose components are defined by (13) and (14). The vector $(Z_n, \dots, Z_0)$ is transmitted over a channel. Then the sender attaches the q–ary representations of the integers k and j with $\lceil \log_q(n+1) \rceil$ and $\lceil \log_q(n+2) \rceil$ symbols, respectively.

Having received the symbols of the sequence $\mathbf{Y}$, let the decoder construct the sequence $\mathbf{y}$ by (16) and determine the value of $\tau \in \{0, \dots, n-1\}$ based on the dominating property of the codeword included in this vector. All further transformations are also clear.

7 Conclusions

Our mathematical problem seems to be close to practical requests. We present two situations.

1). The participants are moving. The receiver wants to know the current distance and simultaneously receive messages from the sender. Therefore, the latter one splits data in pieces of length n and sends them with some fixed delay known to the receiver. The value of the variable delay τ gives information about the distance.

2). Two people communicate with each other at dialog by sending packets consisting of n bits. The time for the reaction depends on many factors and varies from packet to packet. We allow the variation by at most $n-1$ bits.

Table 1: Example of transformations $\mathbf{x} \to \mathbf{x}' \to \mathbf{z}$ for $q = 2$ and $n = 15$.

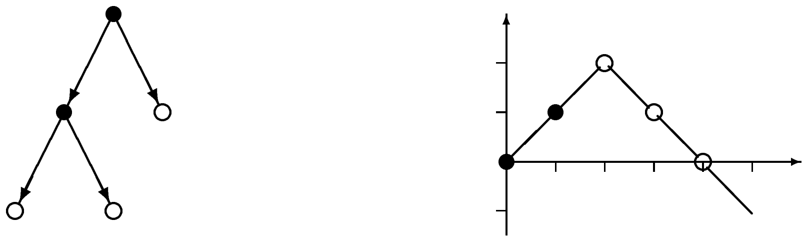
	$\mathbf{x} =$	000000100101111
$j = 1,$	$\mathbf{x}' =$	100000100101111
$k = 6,$	$\mathbf{z} =$	101111100000100

Table 2: The $(0,0,1) \rightarrow (*,*,0)$ procedure for $q=2$, $n=7, \tau=6$, and $\mathbf{y} = 1100110001011\ldots$

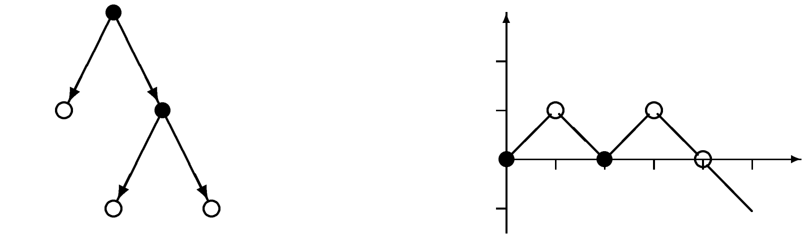
$\mathbf{y}$												
1	1	0	0	1	1	0	0	0	1	0	1	1
1	1	$*$	$*$	0	1	0	0	0	1	0	1	1
1	1	$*$	$*$	0	1	0	$*$	$*$	0	0	1	1
1	1	$*$	$*$	0	1	0	$*$	$*$	$*$	$*$	0	1
1	1	$*$	$*$	0	1	$*$	$*$	$*$	$*$	$*$	$*$	0

Table 3: Example of the generalized version of the $(0,0,1) \to (*,*,0)$ procedure for $q = 3$, $n = 4, \tau = 3$, and $\mathbf{y} = 00201012\ldots$

			$\mathbf{y}$				
0	0	2					
$*$	$*$	0	0	1			
$*$	$*$	0	$*$	0	0	1	
$*$	$*$	0	$*$	0	$*$	0	2
$*$	$*$	0	$*$	$*$	$*$	$*$	0



$\mathbf{c} = 11000$ $\boldsymbol{\sigma} = (0, 1, 2, 1, 0, -1)$



$\mathbf{c} = 10100$ $\boldsymbol{\sigma} = (0, 1, 0, 1, 0, -1)$

Figure 1: The regular binary trees having $n = 5$ nodes and corresponding paths $(0, 0), \ldots, (5, -1)$.

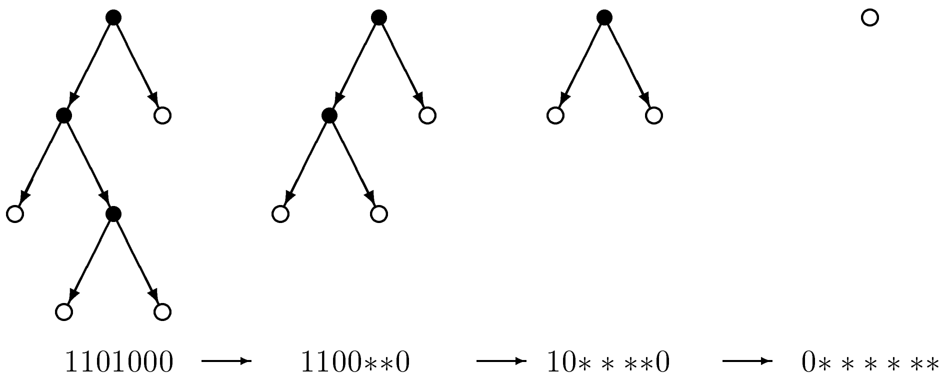


Figure 2: Transformations of the preorder codeword 1101000 of length $n = 7$ and corresponding regular binary trees.

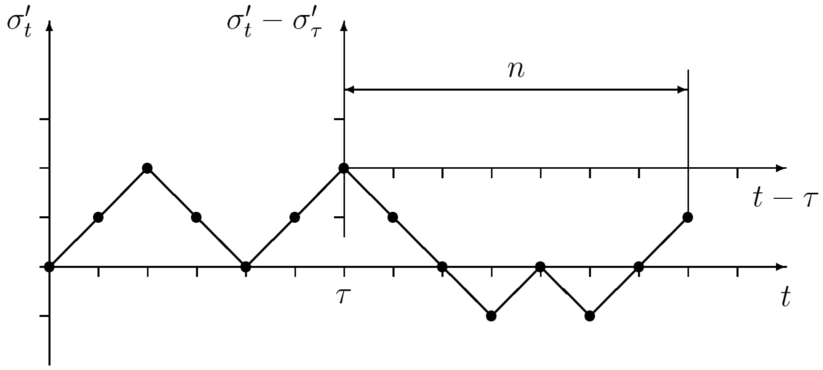


Figure 3: The path corresponding to the received sequence $\mathbf{y} = 1100110001011\ldots$ for $q = 2$ and $n = 7$.

References

[1] Balakirsky, V. B. (2002) A new coding algorithm for trees. *Comp. J.*, **45**.

[2] Gilbert, E. N. (1960) Synchronization of binary messages. *IRE Trans.,* **6**, 470–477.

[3] Levenshtein, V. I. (1964) Decoding automata that are invariant to the initial state. *Problems of Cybernetics,* **12** (In Russian), 125–136.

[4] Levenshtein, V. I. (1970) On the maximum number of words in codes without overlaps. *Probl. Inform. Transmission,* **4**, 88–90.

[5] Mann, H. B. (1962) On the number of information symbols in Bose–Chaudhuri codes. *Inform. Control,* **5**, 153–162.

[6] Even, S. (1973) *Algorithmic Combinatorics.* Macmillan, London.

[7] Mäkinen, E. (1996) A survey on binary tree codings. *Comp. J.,* **34**, 438–443.