Representation and simulating of extended Markov chains over a finite field with a predefined precision

B F Eminov¹, V M Zakharov¹ and S V Shalagin¹

¹Kazan National Research Technical University name after A.N. Tupolev, Karl Marks str., 10, Kazan, Russia, 420111

e-mail: sshalagin@mail.ru, gilvv@mail.ru, bulfami@mail.ru

Abstract. A method is proposed for representing and simulating the extended Markov chains of a defined type by minimal polynomials over finite field GF(q). Simulation problem is being solved as a problem of constructing the minimal polynomial over field GF(q) using the Berlekamp-Massey algorithm. The polynomial produces a sequence of the length of N. A stochastic matrix relevant to that sequence approximates the given stochastic matrix of an extended Markov chain with a predefined precision proportional to the 1/N value. The polynomial constructed defines unambiguously the structure of a q-ary linear shift register for simulating extended Markov chains. The method allows constructing the implementations of this class of Markov sequences on linear q-ary shift registers with a pre-defined “linear complexity” defined by the value of N.

1. Introduction
This paper deals with the problem of modeling random discrete processes from the class of extended Markov chains (EMCs) [1] obtained based on simple Markov chains (MCs) [2]. Extending an MC allows gathering more information regarding the process under research. Random EMC-class sequences are characterized by the “tracked” (according to terms introduced by A.G. Postnikov [3]) daisy-chain links. In an EMC, the current state determines the future behavior of the system after r steps and can be used to forecast the system behavior.

Brumce-Markov chains [2] are a special case of EMC, which are defined by A.A. Markov as “non-true” chains in [4]. The law (stochastic matrix) of a Markov-Brumce chain is defined in [2]. In [1], the problem of defining the stochastic EMC matrix is solved for a special case. In [5, 6], algorithms are represented to build a stochastic EMC matrix sized m’×m’, m ≥ 2, r ≥ 2. [5] shows that, with some limitations on defining the EMC, the stochastic EMC matrix can be considered as the law of a certain r-complicated MC [2], r ≥ 2. [6, 7] represent a method of simulating EMCs on polynomial models described by polynomial functions over field GF(2^r). Within such approach, the problem of constructing polynomial models is solved as a problem of computing polynomial coefficients over field GF(2^r), based on the predefined stochastic matrices. A challenging task in presenting MCs over field GF(2^r) is to reduce the field order.

In [6, 8], an approach is proposed to simulate by the Berlekamp-Massey algorithm (BMA) [9] with minimal polynomials [10] over finite field GF(q), q ≥ 2, of a certain class [6] of nonuniform Markov chains defined by ergodic stochastic matrices [1] and lumped Markov chains [8] defined by regular
stochastic matrices. Within this approach, the main challenge is related to defining the precision of representing stochastic matrices by minimal polynomials.

The goal of this study is to solve the problems of representing extended Markov chains with a pre-defined accuracy and simulating them by minimal polynomials over field GF(q), q ≥ 2.

2. Problem statement

2.1. Defining the law and the algorithm of constructing an EMC

Suppose there is a given sequence of states s_1, s_2, ..., of a simple uniform MC with a finite set of states S = \{s_j\} by ergodic stochastic matrix \(P = (p_{ij})\) [1] sized \(m \times m\), \(i, j = 0, m - 1\).

Let us form from that initial chain, similarly to [1], a new, extended Markov chain as follows [5]. Compose all possible character strings \(s_j \in S\) with the length of \(r = v + \kappa, r ≥ 2, v ≥ 1, \kappa ≥ 1, j = 0, m - 1\), and represented as \((s_{j_1}, s_{j_2}, ..., s_{j_v}, s_{j_{v+1}}, ..., s_{j_{v+\kappa}})\). The adjacent strings contain \(\kappa = r - v\) common characters and \(v\) (“shift size”) different characters.

Suppose the initial chain sequentially transfers within \(2v + \kappa - 1\) steps from a certain state \(s_j\) into \(s_{j_2}\), then from \(s_{j_2}\) into \(s_{j_3}\), ..., and from \(s_{j_{(2v+\kappa)-1}}\) into \(s_{j_{(2v+\kappa)}}\). We will consider adjacent strings having the length of \(r\) as one step of transferring the new process from state \((s_{j_1}, s_{j_2}, s_{j_{v+1}}, ..., s_{j_{v+\kappa}})\) into state \((s_{j_{v+1}}, s_{j_{2v+1}}, s_{j_{2v+2}}, ..., s_{j_{2v+\kappa}})\). This new, extended process (extended Markov chain) is MC having \(m^{\kappa v}\) states (character strings \(s_j\) with the length of \(r, j = 0, m - 1\)). Denote the states of the EMC with characters \(y_i\), from alphabet \(Y = \{y_i\}, i = 0, m^{\kappa v} - 1\).

We will further consider the EMC states ordered lexicographically:

\[
\begin{align*}
(s_0, ..., s_0, s_0), (s_0, ..., s_0, s_1), ..., (s_0, ..., s_0, s_{m-1}), \\
(s_0, ..., s_1, s_0), (s_0, ..., s_1, s_1), ..., (s_0, ..., s_1, s_{m-1}), \\
\vdots \\
(s_{m-1}, ..., s_{m-1}, s_0), (s_{m-1}, ..., s_{m-1}, s_1), ..., (s_{m-1}, ..., s_{m-1}, s_{m-1}).
\end{align*}
\]

Example 1. Forming \(y_i\) EMC states from symbol chains \(s_j \in S, i = 0, m - 1\).

Suppose \(r = v + \kappa = 3, v = 2, \kappa = 1\), then EMC states are represented as \(y_1 = (s_1, s_2, s_3), y_2 = (s_3, s_4, s_5), y_3 = (s_5, s_6, s_7), \ldots\). Suppose \(r = 3, v = 1, \kappa = 2\), then the EMC states are represented as \(y_1 = (s_1, s_2, s_3), y_2 = (s_2, s_3, s_4), y_3 = (s_3, s_4, s_5), \ldots\).

Denote by \(Q\) the stochastic matrix sized \(m^{\kappa v} \times m^{\kappa v} = t \times t\) of that EMC. Define EMC-matrix \(Q\) through matrix \(P\) of the initial MC for \(v ≥ 2, \kappa ≥ 2\) in accordance with [5]. For a special case, \(v = \kappa = 1\), this problem was solved in [1].

Let us obtain EMC-matrix \(Q\), based on the concept of “intermediary” stochastic matrix [5] sized \(m^{\kappa v} \times m^{\kappa v} = t \times t\), to be computed based on matrix \(P\) of the initial MC. We will denote the intermediary matrix by \(W = (w_{ij}), i, j = 0, m - 1\), is given. Then define the matrix \(W\) in accordance with [5] by formula

\[
W = \xi_m \otimes E \otimes C,
\]

where \(E\) and \(\xi_m\) are the unity matrix and the column vector sized \(m^{\kappa v} \times m^{\kappa v} = t\) and \(m\) respectively;

\(\otimes\) is the symbol of the operation of Kronecker (tensor) product [11] of matrices;

\(C = [B_0 B_1 \ldots B_{m-1}]\) is a matrix sized \(m \times m^r\) that is a sequential concatenation of matrices \(B_i = (b_{ij})\), \(i, j, a = 0, m - 1\), where \(b_{ij}^{(a)} = \begin{cases} p_{aj}, & \text{if } a = i \\ 0, & \text{if } a \neq i \end{cases}\).
Example 2. Suppose \( P = P_{(1)} = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix} \). Then, for \( r = 2 \):

\[
P_{(2)} = s_0 s_1 \begin{pmatrix} p_{00} & 0 & 0 \\ 0 & p_{10} & p_{11} \end{pmatrix} \quad (v = \kappa = 1).
\]

Matrix \( P_{(3)} \) for \( r = 3 \) \((v = 1, \kappa = 2)\) is shown in Figure 1.

Example 3. Suppose \( P = P_{(1)} = \begin{pmatrix} p_{00} & p_{01} & p_{02} \\ p_{10} & p_{11} & p_{12} \\ p_{20} & p_{21} & p_{22} \end{pmatrix} \).

For \( r = 2 \) \((v = \kappa = 1)\), we will obtain matrix \( P_{(2)} \) shown in Figure 2.

Non-zero elements of the matrix \( W \) are computed by formula \([5, 6]\):

\[
W_{i,(i+m^\nu+d)} = P_{(i \mod m^\nu+d)}, \quad i = 0, m^\nu - 1, \quad d = 0, m - 1,
\]

where \( i \) is the current row number of matrix \( W \) and \( d \) is the current column number of matrix \( P \).

EMC-matrix \( Q \) is connected to the matrix \( W \) by relation \([5, 6]\)

\[
Q = W^v,
\]

where \((W)^v\) is the \( v \)-th power of matrix \( W \), \( v \geq 1 \), and \( \kappa \geq 1 \).

Let us introduce the EMC definition based on matrix \( Q \).

**Definition.** We will name a Markov chain with the stochastic matrix \( Q \) indicated (3) and sized \( m^{\nu x} \times m^{\nu x}, \nu \geq 1, \kappa \geq 1 \), obtained by ergodic stochastic matrix \( P = (p_{ij}) \) sized \( m \times m \), \( i, j = 0, m - 1 \), a general extended Markov chain.
Note the following properties of an EMC represented by matrix (3).

**Theorem 1** [5]. Suppose we are given matrix \( P = (p_{ij}) \), \( \forall p_{ij} > 0 \), \( i, j = 0, \ldots, m - 1 \), and EMC-matrix \( Q \) sized \( m' \times m' \), \( r = v + \kappa \geq 2 \), \( v \geq 1 \), \( \kappa \geq 1 \), is obtained based on it. Then this EMC is ergodic.

EMC ergodicity for \( r = v + \kappa = 2 \), \( v = 1 \), \( \kappa = 1 \), is shown in [1].

The limiting stochastic vector of EMC-matrix \( Q \) can be defined similarly to [1, p. 183] by relation

\[
\bar{\pi}_{pr} = \overline{(w)}_{pr} \cdot W,
\]

where \( \overline{(w)}_{pr} \) is the limiting vector of matrix \( W \).

**Statement.** Suppose we are given matrix \( P = (p_{ij}) \), \( \forall p_{ij} > 0 \), \( i, j = 0, \ldots, m - 1 \), and the EMC-matrix \( Q \) sized \( m' \times m' \), \( r = v + \kappa \geq 2 \), \( v = 1 \), \( \kappa \geq 1 \), is obtained based on it. Then this matrix \( Q \) defines the stochastic matrix of an \( r \)-complicated MC with transition probabilities defined by the elements of matrix \( P \).

The validity of the statement follows from the algorithm [5] of constructing matrix \( Q \).

**Example 4.** Based on the MC defined by the matrix \( P = \begin{bmatrix} 3/8 & 5/8 \\ 6/8 & 2/8 \end{bmatrix} \), we can use relation (3) as the basis of constructing the following EMCs defined by the stochastic matrices \( Q_1 \) and \( Q_2 \) (Figure 3, (a) and (b)), with the string lengths of \( r = 3 \) and different values of \( v \) and \( \kappa \) (\( Q_1 \) with \( v = 1 \), \( \kappa = 2 \); \( Q_2 \) with \( v = 2 \), \( \kappa = 1 \)). Matrix \( Q_1 \) defines the stochastic matrix of an \( r \)-complicated MC, \( r=3 \).

\[
(a) \begin{pmatrix}
\frac{3}{8} & \frac{5}{8} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{6}{8} & \frac{2}{8} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{3}{8} & \frac{5}{8} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{6}{8} & \frac{2}{8} & \frac{3}{8} \\
\frac{3}{8} & \frac{5}{8} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{6}{8} & \frac{2}{8} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{3}{8} & \frac{5}{8} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{6}{8} & \frac{2}{8} & \frac{3}{8}
\end{pmatrix}, \quad (b) \begin{pmatrix}
\frac{9}{32} & \frac{15}{32} & \frac{15}{32} & \frac{5}{32} & 0 & 0 & 0 & 0 \\
0 & \frac{9}{32} & \frac{15}{32} & \frac{15}{32} & \frac{7}{32} & \frac{1}{16} & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{9}{32} & \frac{15}{32} & \frac{15}{32} & \frac{7}{32} \\
0 & 0 & 0 & 0 & 0 & \frac{9}{32} & \frac{15}{32} & \frac{15}{32} \\
\frac{9}{32} & \frac{15}{32} & \frac{15}{32} & \frac{5}{32} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{15}{32} & \frac{15}{32} & \frac{7}{32} & \frac{1}{16} \\
0 & 0 & 0 & 0 & \frac{9}{32} & \frac{15}{32} & \frac{15}{32} & \frac{7}{32} \\
0 & 0 & 0 & 0 & 0 & \frac{9}{32} & \frac{15}{32} & \frac{15}{32} \\
\end{pmatrix}
\]

**Figure 3.** (a) matrix \( Q_1 \ (v = 1, \kappa = 2) \) and (b) matrix \( Q_2 \ (v = 2, \kappa = 1) \).

2.2. Problem statement for simulating a given EMC indicated (3) over field \( GF(q) \) by minimal polynomials

We will use the term of “sequence over field \( GF(q) \)” for any function \( u: Z \to GF(q) \) defined within set \( Z \) of nonnegative integers and taking its values within field \( GF(q) \) [10]. The sequence \( u = (u_i), i \in Z \), is called an \( L \)-order linear recurrence sequence (LRS) over field \( GF(q) \), if there are constants \( b_0, b_1, \ldots, b_{L-1} \in GF(q) \), such that

\[
u(i + L) = \sum_{j=0}^{L-2} b_j \cdot u(i + j), i \geq 0 \ [10].
\]

The polynomial

\[
f(x) = x^L - \sum_{j=0}^{L-1} b_j \cdot x^j
\]

is called the characteristic polynomial of LRS [10].

The vector \( \bar{u} = (u(0), \ldots, u(L-1)) \) is the initial vector of LRS. Characteristic polynomial of LRS \( u \), having the minimal power, is its minimal polynomial [10]. Let us denote LRS \( u \) of random length \( N \) by
where the length of LRS is the number of characters in the LRS. The LRS is implemented by the linear feedback shift register (feedback LSR) [10], where the power of polynomial $f(x)$ determines the number of $q$-ary bits of the register, while the coefficients are a form of feedback. We will consider the minimal polynomial indicated (4) and constructed over field $\text{GF}(q)$ by the Berlekamp-Massey algorithm [9] as the characteristic polynomial of the LRS that can be obtained based on LSR.

The matrix $Q$ indicated (3) can, in accordance with [12], be associated via algorithm [12] of approximating elements $w_{ij}$, $i, j = 0, m^{-1}$, by rational elements with ergodic stochastic matrix $P_\phi = (p_{ij}^{(\phi)})$, $i, j = 0, t - 1$, sized $t \times t = m' \times m'$, where elements $p_{ij}^{(\phi)} = a_{ij}^{(\phi)} / a_{ij}$ satisfy relation

$$P_\phi = (P_{ij}^{(\phi)}) = (a_{ij}^{(\phi)} / a_{ij}), a_{ij}^{(\phi)} = \sum_{j=0}^{t-1} a_{ij}^{(\phi)} = \sum_{j=0}^{t-1} a_{ji}^{(\phi)} \text{ and } \sum_{j=0}^{t-1} a_{ii}^{(\phi)} = N$$

and the limiting vector of matrix $P_\phi$ is

$$\pi_\phi = (\pi_i^{(\phi)} = a_i / N), \ i = 0, t - 1. \ (6)$$

Assume that:

1) The error of approximating matrix $Q$ by matrix $P_\phi = (p_{ij}^{(\phi)})$, $i, j = 0, t - 1$, satisfies conditions

$$|p_{ij}^{(\phi)} - w_{ij}| \leq \epsilon, 0 < \epsilon < 1; \ (7)$$

$$P_{ij}^{(\phi)} = \begin{cases} 0, \text{if } w_{ij} = 0 \\ > 0, \text{if } w_{ij} > 0. \end{cases} \ (8)$$

2) The value $\epsilon$ is associated with $N$ linear relation [12]

$$N \geq N', \ N' = \max \{ \max_{i, j=0, t-1} \{|1/(w_{ij} \pi_i)|, \max_{i, j=0, t-1} \{(1 + w_{ij} + \epsilon) / (\pi_i \epsilon)\} \} \}. \ (9)$$

For assumption (7)-(9), the reachable precision of approximating the elements of matrix $Q$ by elements $p_{ij}^{(\phi)}$ depends linearly on $N$.

The problem being solved is stated 1) as a problem of constructing by the Berlekamp-Massey algorithm the minimal polynomial over field $\text{GF}(q)$, describing sequence $u_N$ with the length of $N$, such that stochastic matrix $P_\phi = (p_{ij}^{(\phi)})$, $i, j = 0, t - 1$, sized $t \times t$ and relevant to that sequence, must approximate the given stochastic matrix $Q$ with a given precision proportional to the $1/N$ value; and 2) as constructing based on the obtained minimal polynomial of LSR, which allows reproducing sequence $u_N$ with the length of $N$.

3. Method representing and simulating extended Markov chains, based on the minimal polynomial

Let us introduce value $N'$ satisfying condition

$$|N' - N| \leq t - 1, \ (10)$$

**Theorem 2.** Suppose we are given ergodic stochastic matrix $Q$ sized $t \times t$ and numbers $0 < \epsilon < 1$, $N \geq N'$. Then there exists minimal polynomial $f(x)$ over field $\text{GF}(q)$, which produces sequence $u_{N'+1}$ with the length of $N' + 1$ and law $P_\phi = (p_{ij}^{(\phi)})$, both satisfying conditions (5)-(10),

$$|\pi_i^{(\phi)} - \pi_i| \leq \frac{1}{N} + \frac{\pi_i |N' - N|}{N}, \ (11)$$

while power $L$ of polynomial $f(x)$ satisfies condition

$$2L \leq N' + 1. \ (12)$$

The proof of Theorem 2 can be constructed in accordance with the scheme of proving theorem 2 presented in [8].
From Theorem 2, the method below follows for simulating an EMC by minimal polynomials \( f(x) \) over field \( GF(q) \) with the given precision \( \varepsilon \) of representing matrix \( Q \).

Stage 1. Suppose a given MC is defined by ergodic stochastic matrix \( P = (p_{ij}) \) sized \( m \times m \), with a finite set of states, \( S = \{ s_j \}, \ i, j = 0, t - 1 \).

Compute EMC-matrix \( Q \) sized \( m' \times m' \) through matrix \( P \) for the given \( r = v + \kappa \geq 2, v \geq 1, \) and \( \kappa \geq 1 \), by formulas (2) and (3).

It should be noted that matrix \( Q \) with the given \( r = v + \kappa \geq 2, v \geq 1, \) and \( \kappa \geq 1 \), defines the stochastic matrix of the \( r \)-complicated MC with transition probabilities defined by the elements of matrix \( P \).

Stage 2. By pre-defined \( Q, \varepsilon \) using the following algorithm [12] (denoted as \( A(N^r) \)) of approximating elements \( w_{ij}, i, j = 0, m' - 1 \) by rational elements, we construct matrix \( P_\rho = (p_\rho^{(\rho)}) \), \( i, j = 0, t - 1 \), satisfying conditions (4)-(10).

**Algorithm \( A(N^r) \)**

At input: Matrix \( Q \) sized \( r \times r \); number \( \varepsilon, 0 < \varepsilon < 1 \).

At output: Matrix \( P_\rho \) sized \( r \times r \), satisfying conditions (4)-(10).

1. Number \( N^r \) is calculated by formula (8).

2. Square matrix \( B = (b_{ij}), \ i, j = 0, t - 1 \), is calculated, the elements of which are \( b_{ij} = p_{ij} \times \pi \times N^r \). Elements \( b_{ij} \) are formed resulting from transferring the significant part of decimal fraction (considering accuracy) to the sequence of natural numbers.

   Elements \( b_i \) of vector \( (b_i), i, j = 0, t - 1 \), are formed by summing up the elements of the \( i \)th line (or the \( i \)th column) \( b_i = \sum_{j=0}^{t-1} b_{ij} = \sum_{j=0}^{t-1} b_{ji} \) matrix \( B \), where \( \sum_{i=0}^{t-1} b_i = N^r \).

3. Let us compute matrix \( (a_{ij}), \ i, j = 0, t - 1 \), containing the fractional parts of elements \( b_{ij} \) that have not been considered in constructing matrix \( B \). Its elements are computed as \( a_{ij} = b_{ij} - \lfloor b_{ij} \rfloor \).

4. Similarly, vector \( (a_i) \) is computed, \( i, j = 0, t - 1 \) - its elements \( a_i \) are equal to the sum of the remainders of the elements of the \( i \)th line of matrix \( B \):

   \[
   a_i = \sum_{j=0}^{t-1} b_{ij} - \lfloor \sum_{j=0}^{t-1} b_{ij} \rfloor = \sum_{j=0}^{t-1} (b_{ij} - \lfloor b_{ij} \rfloor) = b_i - \lfloor b_i \rfloor.
   \]

5. Binary features \( C = (c_{ij}) \) are computed, \( i, j = 0, t - 1 \) : \( c_{ij} = \begin{cases} 1, \text{ if } a_{ij} > 0 \\ 0, \text{ if } a_{ij} = 0 \end{cases} \).

Based on matrix \( (a_{ij}) \), two vectors, \( \vec{k} = (k_i) \) and \( \vec{r} = (r_i) \), \( i, j = 0, t - 1 \), are constructed:

- By lines, \( k_j = \sum_{i=0}^{t-1} a_{ij} \) defines the sum of the elements of matrix \( (a_{ij}) \) by the \( j \)th line (the sum of the fractional parts of the elements of the \( j \)th line of matrix \( B \));

- By columns, \( r_j = \sum_{j=0}^{t-1} a_{ij} \) defines the sum of the elements of matrix \( (a_{ij}) \) by the \( j \)th column (the sum of the fractional parts of the elements of the \( j \)th column of matrix \( B \)).

6. Binary matrix \( D = (d_{ij}), i, j = 0, t - 1 \), is computed by algorithm [12]. Matrix \( D \) must meet the following conditions:

   - \( \sum_{j=0}^{t-1} d_{ij} = k_j \); - \( \sum_{i=0}^{t-1} d_{ij} = r_j \);

   - \( 0 \leq d_{ij} \leq c_{ij} \) (if \( c_{ij} = 0 \), then \( d_{ij} = 0 \), too);
- \( \forall d_{ij} \in \{0, 1\} \).

Existence of matrix \( D \) is proven in [12]. Algorithm [12] used to compute matrix \( D \) is similar to that of constructing the maximum matching in two-partite graph [13] (unities are re-distributed among lines and columns in accordance with the conditions (13)).

Elements of vectors \( \tilde{K} \) and \( \tilde{r} \) are changed in computing matrix \( D \).

Assume matrix \( D \) to be a zero matrix in the beginning. It starts being filled with unities as follows.

As long as \( \forall r_j > 0 \) and \( \forall k_t > 0 \), for each \( i \)-th line being searched through (starting from 0 to \( t-1 \)), columns \( j, i, j = 0, t - 1 \), are searched through:

- If conditions \( c_{ij} = 1 \) and \( d_{ij} = 0 \) are met (the unity has not been placed into a cell yet) and \( r_j > 0 \) ( \( r_j \) unities can still be placed in the \( j \)-th column), then we set \( d_{ij} = 1 \), and the counters of the number of unities by columns (\( r_j = r_j - 1 \)) and lines (\( k_t = k_t - 1 \)) are decreased;
- Otherwise, in the \( j \)-th column, element \( d_{ij} = 1 \), \( k = 0, t - 1 \) (to which the unity has been assigned earlier) is searched for; if it exists, the following is replaced: \( d_{ij} = 0 \) and \( d_{ij} = 1 \).

7. Elements \( p^{(s)}_{ij} \), \( i, j = 0, t - 1 \), of the required matrix \( P_\varphi \) are computed as

\[
p^{(s)}_{ij} = a_j \sum_{j=0}^{t-1} a_{ij}
\]

where \( a_{ij} = \lceil b_{ij} + 1 \rceil \).

Correctness of the algorithm used in constructing matrix \( P_\varphi \) is confirmed in [12], while its computational complexity is \( O(t^3) \) [12].

Stage 3. Based on matrix \( P_\varphi = (p^{(s)}_{ij}) \), using the probabilistic algorithm [14], we construct in alphabet \( Y = \{y_i\}, \; i = 0, m^{\infty x} - 1 \), sequence \( u_{N'} + 1 \) with the length of \( N' + 1 \) and law \( P_\varphi = (p^{(s)}_{ij}) \), \( i, j = 0, t - 1 \), both satisfying conditions (5)-(11). Solving this problem is comprehensively described in [14], where sequence \( u_{N'} + 1 \) is constructed by the algorithm of isolating Euler’s chains [13], including the probabilistic procedure [14] of choosing by matrix \( P_\varphi \) an arc in each vertex.

Stage 4. Let us code the characters of alphabet \( Y \) with the elements of field \( \text{GF}(q) \), where \( q \geq t \). Based on sequence \( u_{N'} + 1 \), we will construct minimal \( L \)-power polynomial \( f(x) \) using the software implementation [15] of BMA, where \( L \) satisfies condition (12) of theorem 2. We keep initial vector \( \tilde{u} = (u(0), \ldots, u(L-1)) \) of sequence \( u_{N'} + 1 \) in the memory.

It should be noted that Berlekamp-Massey algorithm construct, based on sequence \( u_{N'} + 1 \), the only minimal polynomial \( f(x) \) with the order of \( L \) satisfying condition (12), which follows from the theorem below.

**Theorem 3** [9]. Suppose there is an \( N \)-long sequence \( u_N \) consisting of the elements of field \( \text{GF}(q) \). Then, based on sequence \( u_N \), Berlekamp-Massey algorithm constructs the only minimal polynomial with order \( L \) satisfying condition \( 2L \leq N \).

The polynomial constructed unambiguously identifies matrix \( P_\varphi \).

Stage 5. Based on the \( L \)-order polynomial \( f(x) \) obtained, we construct the software implementation of LSR [15] with the length of \( L \) and \( q \)-ary bits, where \( L \) is defined by expression

\[
L = \begin{cases} 
(N' + 1)/2, & \text{if } N' \text{ is odd}; \\
((N' + 1) + 1)/2, & \text{if } N' \text{ is even}.
\end{cases}
\]

It should be noted that Berlekamp-Massey algorithm construct by sequences \( u_{N'} + 1 \) the only \( L \)-order minimal polynomial \( f(x) \) satisfying condition (12). Having defined \( \tilde{u} \) as the initial state of LSR, we obtain sequence \( u_{N'} + 1 \) with the length of \( N' + 1 \) and law \( P_\varphi \) at the \( i \)-th output, \( i = 1, L \), of the \( q \)-ary bit of the LSR program model.
4. Conclusion
The problem of simulating extended Markov chains and a certain type of complicated Markov chains by minimal polynomials over a finite field is solved in our study as a problem of constructing by Berlekamp-Massey algorithm a minimal polynomial over field GF(q), characteristics $q \geq 2$. Polynomial is developing a sequence of a length, so that stochastic matrix $P_\phi$, relevant to that sequence and having a predefined precision proportional to the value of $1/N$, approximates the initial ergodic stochastic matrix $Q$. The precision of representing stochastic matrices $Q$ by minimal polynomials depends linearly on the polynomial order.

The solution proposed includes the following stages: 1) By the pre-defined ergodic stochastic matrix $P$ sized $m \times m$, we will compute EMC-matrix $Q$ sized $m' \times m'$. 2) By the pre-defined $Q$, $\epsilon$, and $N \geq N'$, we construct matrix $P_\phi = (p_\phi^{(\epsilon)})$. 3) By matrix $P_\phi = (p_\phi^{(\epsilon)})$, we construct sequence $u_{N'+1}$. 4) By sequence $u_{N'+1}$, we construct minimal polynomial $f(x)$ of order $L$, using Berlekamp-Massey algorithm. 5) By the obtained polynomial $f(x)$ of order $L$, we construct the software implementation of the $L$-long LSR.

The polynomial constructed unambiguously identifies matrix $P_\phi$ and unambiguously defines the structure of the $q$-ary linear shift register for simulating extended Markov chains. The method allows constructing the implementations of this class of Markov sequences on linear $q$-ary shift registers with a pre-defined “linear complexity” defined by the value of $N$.

5. References
[1] Kemeny J G and Snell J L 1960 Finite Markov chains (Princeton: Van Nostrand) p 210
[2] Romanovskiy V I 1949 Discrete Markov chains (Moscow: Gostekhizdat) p 436
[3] Postnikov A G 1960 Arithmetic modeling of random processes Proceedings of the Mathematical Institute Steklov Academy of Sciences of the USSR 57 1-84
[4] Markov A A 1911 About related values that do not form a real chain Imperial Academy of Sciences 5 113-126
[5] Zakharov V M and Eiminov B F 2007 Representation of expanded Markov chains over the field GF(2n) Process modeling Proceedings of the Kazan scientific seminar Modeling Methods 3 270-286
[6] Eiminov B F 2013 Algorithm for constructing expanded Markov chains over a field GF(2n) Management systems and information technologies 3.1(53) 186-191
[7] Eiminov B F and Zakharov V M 2014 Polinomial models of expanded Markov chains over the field GF(2n) Management systems and information technologies 2(56) 26-30
[8] Zakharov V M, Eiminov B F and Shalagin S V 2018 Representing lumped Markov chains by minimal polynomials over field F(q) Journal of Physics: Conference series 2015 1-6
[9] Massey J L 1969 Shift-register synthesis and BCH decoding IEEE Trans. Inform. Theory IT-15 122-127
[10] Alferov A P, Zubov A Yu, Kuzmin A S and Cheremushkin A V 2002 Basics of cryptography (Moscow: GeliosARV) p 480
[11] Sigorsky V P 1975 The mathematical apparatus of the engineer (Kiev: Technique) p 768
[12] Zakharov V M and Kuznetsov S E 1987 Complexity of the problem of approximation stochastic matrix by rational elements Fundamental of Computation Theory International conf. 483-487
[13] Kharari F 1973 Theory of graphs (Moscow: Mir) p 300
[14] Eiminov B F and Zakharov V M 2011 Simulation of Markov sequences with inaccuracy less than the standard error Herald of KSTU named after Tupolev 1 115-122
[15] Khusainov R N, Eiminov B F, Galimov M D and Kryukov A I 2015 Development of a software implementation of the Berlekamp-Massey algorithm for the analysis and synthesis of binary recurrent sequences Herald of Technological University 18(24) 89-91

Acknowledgments
This work was supported by RFBR Grant 18-01-00120, “Specialized devices for generating and processing data sets in the architecture of programmable logic devices class FPGA.”