ON EFFECTIVE ALGORITHMS SOLVING REGULARITY OF MARKOV CHAINS

Aleksander Dańko[1], Anna Sawicka[2], Wiktor Dańko[3]

[1] alxd@poczta.onet.pl

[2] Polish-Japanese Academy of Information Technology, Warsaw, Poland

[3] Faculty of Computer Science, Bialystok University of Technology, Bialystok, Poland

Abstract: We propose algorithms deciding whether a Markov chain with an $n \times n$ transition matrix M is regular. The lowest complexity of such an algorithm can be not greater than $O(n^3)$ and we argue that it cannot be essentially diminished.

Keywords: Markov chain, ergodic Markov chain, regular Markov chain

1. Introduction

We recall basic definitions (cf. [10]) concerning Markov chains. Let M be an $n \times n$ transition matrix of a Markov chain. Positions of the matrix M will be denoted by $M[i,j]$, $i,j \in \{1,2,...,n\}$. $M[i,j]$ will be called $[i,j]$ position of the matrix M. The chain is called *ergodic* if and only if for all $i,j \in \{1,2,...,n\}$ there exists a positive natural number p such that the $[i,j]$ position of p–th power of the matrix M^p is positive. Since the position $m_{ij}^{(p)}$ of the p–th power M^k of M gives the probability that the Markov chain, started in state i, will be in state j after p steps, we can say that a Markov chain is called an ergodic chain if and only if it is possible to go from every state to every other state (i.e., all transitions are ultimately possible). The chain is called *regular* if and only if there exists a positive natural number p, such that all position of the p–th power M^p of the matrix M are positive. Obviously, regular chains are ergodic.

It is well known that the Warshall-Floyd algorithm (cf. [1], [2]) can be used to solve whether a Markov chain is ergodic. Its complexity is $\Theta(n^3)$.

Advances in Computer Science Research, vol. 13, pp. 5-25, 2016.

The question of the existence of a polynomial algorithm for the problem whether an ergodic chain is regular remains open (according to the actual knowledge of the authors).

Moreover, we have not find any informal information about the complexity of a mathematical method (e.g., cf. [7]) related to deciding the regularity of Markov chains.

Some interesting and useful conditions characterizing regularity one can find in [5], [6], [7], [9], [10], [11], [12], [13].

The paper presents a $\Theta(n^3 \cdot \log n)$ algorithm solving the regularity of Markov chains. The data for the algorithm is an $n \times n$ matrix of a Markov chain. The algorithm consists of two modules. The first module realizes the Warshall-Floyd algorithm and decides whether M is a matrix of an ergodic Markov chain. Its (worst-case) complexity is $\Theta(n^3)$.

The second module verifies regularity of the Markov chain. The idea is to calculate powers of a Boolean matrix G – the adjacency matrix of the graph of the Markov chain (for details cf. Section 2). The main problem is to find, for a given n, an upper estimate $U(n)$, such that if there exists m satisfying $G^m = E$, then there exist $k \leq U(n)$ such that $G^k = E$, where E denotes the $n \times n$ Boolean matrix with all positions equal to 1. Another question is to minimize the number of matrix multiplication related to verification the condition $G^m = E$.

In the case, where we shall use an efficient algorithm for matrix multiplication, e.g., four Russian algorithm (cf. [1], [2]) or Strassen algorithm (cf. [1], [2]) we finally obtain the worst-case complexity of the second module at most $\Theta(n^3)$.

The position [10] can be treated as a report on programistic experiments illustrating realizations of the algorithm solving regularity of Markov chains.

2. Markov Chains

In this section we recall basic definition and facts related to Markov chains (cf. [5], [6], [7], [10]). We shall use the term "computation" to describe the process of changing states.

A Markov chain is determined by a finite set of states S. Without loss of generality we shall assume that S is a subset of natural numbers of the form $S = \{1, 2, ..., n\}$. A *computation* of a Markov chain will be understand as a finite or infinite sequence of states $\{s_m\}_{m \in L}$, where L is the set of natural numbers N or an initial subset of natural numbers of the form $N_k = \{0, 1, 2, ..., k\}$. We shall say that computation starts at time 0 and moves successively from one state to another at unit time intervals. s_0 will be called the initial state. For a given state s_i, the next state s_{i+1} is chosen in a random way. The probabilities of passing from a state i to another next state j of a computation, denoted by m_{ij} or $M[i,j]$, for $i,j \in S$, are fixed and form a square, $n \times n$ transition matrix M of the Markov chain. The matrix M satisfies the following conditions:

$$M[i,j] \geq 0, \text{for all } i,j \in \{1,2,...,n\}, \tag{1}$$

$$\sum_{j=1}^{n} M[i,j] = 1, \text{for all } i \in \{1,2,...,n\}. \tag{2}$$

Each matrix M, with positions being real numbers, satisfying the conditions (1), (2), will be called *probabilistic matrix.*

2.1 Theorem (cf., e.g., [7], [10])

The product of probabilistic matrices is a probabilistic matrix.

□

Let $c =< s_0, s_1, s_2, ..., s_p >$ be a computation of the Markov chain. The number p will be called the number of steps (transitions) of the computation c or the length of the computation c and denoted by $|c|$. By the probability of realization of the computation c we shall mean the number

$$pr(c) = \prod_{i=1}^{p} M[s_{i-1}, s_i] \tag{3}$$

It is easy to note that $pr(c) > 0$ if and only if $M[s_{i-1}, s_i] > 0$, for $i = 1, ..., p$.

Let $C_p(i,j)$ denote the set of all computations of the length p that starts in the state i and ending, after p steps, in the state j. The following theorem gives a method of determining the sum of probabilities of computations of $C_p(i,j)$:

2.2 Theorem (cf., e.g., [10])

Let M be transition matrix of a Markov chain. The position $m_{ij}^{(p)}$ of the p–th power M^p of M gives the probability that the Markov chain, started in state s_i will be in state s_j after a computation containing p steps.

□

We shall now formulate the above theorem in a more technical manner. Let $i,j \in S$ be fixed states.

2.3 Proposition (compare [5], [6], [7], [10])

$$M^p[i,j] = \sum_{c \in C_p(i,j)} pr(c) \tag{4}$$

□

2.4 Corollary (compare [5], [6], [7], [10])

$M^p[i,j] > 0$ *if and only if there exists a computation c of the length p that starts in the state i and ending, after p steps, in the state j and such that* $pr(c) > 0$.

□

Now, we shall formulate the question of regularity of a Markov chain, represented by a transition $n \times n$ matrix M, by means of a Boolean, we shall formulate the question of regularity of a Markov chain, represented by a transition $n \times n$ matrix G_M, being the adjacency matrix of the transition graph $\mathcal{G}_M$ of the chain.

Let M be transition matrix of a Markov chain. We shall define the transition graph $\mathcal{G}_M =< V, E >$ of the Markov chain. The set V_M of vertices of $\mathcal{G}_M$ is the set S of states, $V = S$, and the set $E_M \in S \times S$ of edges is defined in the following way:

$$< i, j > \in E_M \text{ if and only if } M[i,j] > 0,$$

for $i, j \in \{1, 2, ..., n\}$.

We recall that the adjacency matrix of the graph $\mathcal{G}_M$ is the Boolean matrix G_M or defined as follows:

$$G_M[i,j] = \begin{cases} 1 & \text{if } < i, j > \in E_M, \\ 0 & \text{if } < i, j > \notin E_M. \end{cases}$$

The Boolean matrix G_M will be denoted by G, if it does not lead to any misunderstanding.

In the sequel we shall consider paths of graphs. We recall that by a path of a graph we shall understand a finite sequence of vertices $d =< s_0, s_1, s_2, ..., s_p >$, such that $< s_{i-1}, s_i > \in E_M$, for $i = 1, 2, ..., p$. s_0 and s_p are called, respectively, the initial and the final vertex of the path d. The number p will be called the length of the path d and denoted by $|d|$.

2.5 Proposition (compare [[5], [6], [7], [10])

Let M be transition matrix of a Markov chain and let $\mathcal{G}_M$ be the transition graph of the chain. Let $c =< s_0, s_1, s_2, s_3, ..., s_{k-1}, s_k >$ be a sequence of states of the chain. The following conditions are equivalent:

- *the sequence c is a computation of the chain such that $pr(c)$ is positive,*
- *the sequence c is a path of the graph $\mathcal{G}_M$.*

□

In the sequel the symbol $\otimes$ will be used to denote logical multiplication of Boolean matrices; if A is a $p \times q$ Boolean matrix, B is a $q \times r$ Boolean matrix, then $C = A \otimes B$ is a $p \times r$ Boolean matrix satisfying

$$C[i,j] = (A[i,1] \wedge B[1,j]) \vee (A[i,2] \wedge B[2,j]) \vee ... \vee (A[i,q] \wedge B[q,j]),$$

for $i = 1,2,...,p,\ j = 1,2,...,r$, where $\wedge$, $\vee$ denote conjunction and disjunction operations. The matrix $\underbrace{G \otimes ... \otimes G}_{p\ times}$ will be called the p-th power of the matrix G and denoted by G^p.

Similarly, for A and B being $p \times q$ Boolean matrices, we define $A \otimes B$ as a $p \times q$ Boolean matrix C satisfying $C[i,j] = (A[i,j] \vee B[i,j])$, for $i = 1,2,...,p,\ j = 1,2,...,q$.

The following fact is analogous to (4) and belongs to the folklore of the graph theory:

2.6 Proposition (cf. [1], [2])

$G_M)^p[i,j] = 1$ *if and only if there exists a path in the graph $\mathcal{G}_M$ of the length p, with the initial vertex i and the final vertex j.*

□

Propositions 2.4, 2.6 can be written together in a more readable form and can be treated as an equivalent form of 2.5.

2.7 Theorem (compare [5], [6], [1], [2])

Let M be the transition $n \times n$ matrix of a Markov chain and let G_M be the adjacency matrix of the graph $\mathcal{G}_M$. Then for all $i, j = 1,2,...,n$, the following equivalence holds:

$$M^p[i,j] > 0 \ \ if\ and\ only\ if \ \ (G_M)^p[i,j] = 1.$$

□

Matrices with all positions being positive will be called a positive matrices.

2.8 Corollary

Let M be the transition $n \times n$ matrix of a Markov chain and let G_M be the adjacency matrix of the graph $\mathcal{G}_M$. Then for all $i, j = 1, 2, ..., n$, the following equivalence holds:

$$M^p \; is \; a \; positive \; matrix \;\; if \; and \; only \; if \;\; (G_M)^p = E.$$

□

This fact indicates that the problem, whether a Markov chain is ergodic or regular, that concern the powers of the transition matrix M, can be transformed into questions concerning the powers of the Boolean matrix $G = G_M$:

(e) a Markov chain is ergodic if and only if the following condition is satisfied:
$(\forall i \in S)(\forall j \in S)(\exists p \in N)((G_m)^p[i, j] = 1)$
i.e., for all $i, j \in \{1, 2, ..., n\}$ there exists a positive natural number k such that the $[i, j]$ position of the matrix $(G_M)^p$ is equal to 1.

(r) a Markov chain is regular if and only if the following condition is satisfied:
$(\exists p \in N)(\forall i \in S)(\forall j \in S)((G_m)^p[i, j] = 1)$
i.e., for all $i, j \in \{1, 2, ..., n\}$ there exists a positive natural number k such that all the position of the matrix $(G_M)^p$ are equal to 1.

Therefore, the question of regularity of a Markov chain, represented by a Boolean $n \times n$ matrix G_M can be formulated as a question, whether the sequence

$$G_M, (G_M)^2, (G_M)^3, ..., (G_M)^i, (G_M)^{i+1}, ...$$

contains the matrix with all positions equal to 1. Let us note that the number of different $n \times n$ Boolean matrices is finite; there is $2^{(n \cdot n)}$ different Boolean $n \times n$ matrices. This means that the above sequence is periodic (cyclic), and enables us to formulate a "brute force" algorithm solving the regularity of Markov chains:

2.9 Algorithm 1

a) the data:
$G = G_M$, a Boolean $n \times n$ matrix G_M representing transition graph of a Markov chain,

b) the result:
the answer, whether it is a matrix of a regular Markov chain,

c) the idea of the algorithm:

```
begin {Algorithm 1}
{first module - begin}
{this is a modification of  Warshall Floyd algorithm}
{----------------------------------------}
  for (i=1;i<=n;i++)
    for (j=1;j<=n;j++)
      tr_cl_G[i][j] := G[i][j];

  for (k=1;k<=n;k++)
    for (i=1;i<=n;i++)
      for (j=1;j<=n;j++)
        if ((tr_cl_G[i][k] = 1)&(tr_cl_G[k][j] = 1))
        then tr_cl_G[i][j] := 1;

  test_erg = 1;
  for (i=1;i<=n;i++)
    for (j=1;j<=n;j++)
       if (tr_cl_G[i][j] = 0) test_erg := 0;
{----------------------------------------}
{first module - end}

{----------------------------------------}
{second module - begin}
  if (test_erg = 0) then test_reg = 0
  else
    begin
      H := G;
      p := 1;
      u:= exp(2,n*n);
      if (H = E) then test_reg := 1
      else test_reg := 0;
        while ((test_reg = 0) & (p<=u)) do
          begin
            H := H ⊗ G;
            if (H = E) then test_reg := 1;
            p = p + 1;
          end;
    end;
```

```
{---------------------------------------}
{second module - end}
end.
```

d) the worst–case complexity: $O(2^{(n \cdot n)})$ matrix multiplications,

e) the upper estimate for analyzed powers of the matrix G:
$U(n) = 2^{(n \cdot n)}$,

f) the idea of the correctness of the algorithm:
the correctness of the algorithm is a simple corollary related to remarks made after points (e), (r) above.

Because of its complexity, Algorithm 1 is of any use in practice. As we have mentioned above, the main problem for speed up this algorithm, is to find, for a given n, an upper estimate $U(n)$, such that if there exists q satisfying $(G_M)^q = E$, then there exist $p \leq U(n)$ such that $(G_M)^p = E$. However, another question related to quality of the algorithm, is to minimize the number of matrix multiplication related to verification whether $(G_M)^p = E$.

To construct effective algorithms solving regularity of Markov chains we need some facts related to Boolean matrix multiplication (Section 3) and the length of cycles, containing all vertices, for transition graphs of ergodic Markov chains (Section 4).

3. Remarks on Boolean matrix multiplication

We shall start with several simple facts, concerning Boolean matrix multiplication. In the authors opinion, these simple facts are rather well known and are elements of the practice of matrix multiplication.

A notation will be used: assume natural ordering between logical values: $0 < 1$. We define an ordering in sets of Boolean matrices of the same dimension: for two Boolean $p \times q$ matrices A, B we shall write

$A \leq B$ if and only if $(A[i,j] < B[i,j] or A[i,j] = B[i,j], for i = 1,2,...,p, j = 1,2,...,q)$.

By I_n we shall denote the $n \times n$ Boolean matrix defined as follows:

$$I_n[i,j] = \begin{cases} 1 & \text{if } i \neq 1 \\ 0 & \text{if } i = 1 \end{cases}, \text{ for } i,j = 1,2,...,n.$$

By E_n we shall denote the $n \times n$ Boolean matrix defined as follows:

$$E_n[i,j] = 1, \text{ for } i,j = 1,2,...,n.$$

We shall often write I and E instead of I_n and E_n if it does not lead to any misunderstanding.

We shall start with a simple remark:

3.1 Remark

Let H, $K \geq I$ be $n \times n$ Boolean matrices. Then $H \otimes K \geq H$.

□

3.2 Corollary

Let $K \geq I$ be an $n \times n$ Boolean matrix.
Then

$$K \leq K^2 \leq K^3 \leq K^4 \leq \dots$$

is a non-decreasing sequence of matrices.

□

We end this section with the following two (equivalent) lemmas:

3.3 Lemma

Let M be transition matrix of a Markov chain with n states and let $G = G_M$ be the $n \times n$ Boolean matrix of the transition graph of the chain. If for some natural m, $G^m = E$, then

$$E = G^m = G^{m+1} = G^{m+2} = G^{m+3} = \dots.$$

□

3.4 Lemma

Let M be transition matrix of a Markov chain with n states. If for some natural m, M^m is a positive matrix, then

$$M^{m+1}, M^{m+2}, M^{m+3}, \dots,$$

are also positive matrices.

□

4. Cycles in graphs of ergodic chains

In this section we shall consider transition graphs of ergodic Markov chains. It is easy to argue that in the case of graphs of ergodic chains there exists cycles containing all vertices. Our aim is to estimate the length of such cycles.

The main fact concerning this problem is the following

4.1 Lemma

Let G_M be the transition graph of an ergodic Markov chain with n states. Then there exists a cycle containing all vertices with the length at most $(n^2 - n) = O(n^2)$.

The idea of the proof.

We start with a simple remark: if G_M is the graph of an ergodic chain G_M then there exists a cycle in G_M containing all vertices.

Consider such a cycle $d =< s_0, s_1, s_2, s_3, ..., s_{k-1}, s_k >$, where $s_i \in S = \{1, 2, ..., n\}$, for $i = 1, 2, ..., k$, containing all vertices. Denote by m the number of occurrences of the vertex s_0 in the cycle d. Without loss of generality we can assume, for readability, that $s_0 = 1$. Therefore d can be presented as

$$< 1, s_1^1, s_2^1, ..., s_{r_1}^1, 1, s_1^2, s_2^2, ..., s_{r_2}^2, ..., 1, s_1^m, s_2^m, ..., s_{r_m}^m >$$

where $r_1 + r_2 + ... + r_m + m = k$. Let us assign to each $s \in S \ \{1\} = \{2, 3, ..., n\}$, the number $i(s) \in \{1, 2, ..., m\}$, such that s occurs among $\{s_1^{i(s)}, s_2^{i(s)}, ..., s_{r_{i(s)}}^{i(s)}\}$.

Suppose $m > n\breve{}1$. This means that if we remove from the sequence d all subsequences of the form $< 1, s_1^j, s_2^j, ..., s_{r_j}^j >$, where $j \in \{1, 2, ..., m\} \ \{i(2), i(3), ..., i(n)\}$ then the sequence obtained in this way also forms a cycle of G_M and contains all elements of the set S. Moreover, the number of occurrences of the element $s_0 = 1$ in the obtained sequence does not exceed $n - 1$.

In an analogous manner we can repeat this removing for all remaining elements $s = 2, 3, ..., n$ and argue that each element of S does not occur in the result sequence more than $n - 1$ times.

□

We now formulate a simple remark:

4.2 Remark

If p is the length of a cycle of G containing all vertices of the Markov chain, then $G^p \geq I$.

□

From Lemma 4.1 we obtain:

4.3 Corollary

Let $G = G_M$ be the $n \times n$ Boolean matrix of the transition graph of an ergodic chain. Then, for some $p \leq n^2 - n$, $G^p \geq I$.

□

The following example shows that there exist transition graphs of Markov chains with n states, such that minimal length of a cycle containing all states, is $\Theta(n^2)$.

4.4 Example

Let

$$V = S = \{1,2,3,...,3k-1,3k,3k+1\},\ n = 3k+1,$$

and

$$\begin{aligned}E \subset V \times V = \{<1,3>,<1,6>,<1,9>,...,<1,3(k-1)>,<1,3k>\}\cup\\ \{<3,6>,<3,9>,...,<3,3(k-1)>,<1,3k>\}\cup\\ \{<6,9>,...,<3,3(k-1)>,<1,3k>\}\cup\\ ...\\ \{<3(k-2),3(k-1)>,<3(k-2),3k>\}\cup\\ \{<3(k-1),3k>\}\cup\\ \{<3,2>,<3,4>,<6,5>,<6,7>,...,<3k,3k-1>,<3k,3k+1>\}\cup\\ \{<2,1>,<5,1>,...,<3(k-1)-1,1>,<3k-1,1>\}\cup\\ \{<4,1>,<7,1>,...,<3(k-1)+1,1>,<3k+1,1>\}.\end{aligned}$$

It is easy to note that the sequence below is an example of a cycle of the graph $<V,E>$, containing all vertices, of minimal length equal to $k \cdot (k+5) = (1/9) \cdot (n-1) \cdot (n+5) = \Theta(n^2)$:

$$\begin{gathered}1,3,2,\ 1,3,4,\ 1,3,6,5,\ 1,3,6,7,\ 1,3,6,9,8,\ 1,3,6,9,10,...,\\ 1,3,6,9,...,3k,3k-1,\ 1,3,6,9,...,3k,3k+1.\end{gathered}$$

□

5. Polynomial algorithms solving regularity of Markov chains

As we have mentioned above, the initial data for such an algorithm is an $n \times n$ Boolean matrix G_M representing transition graph of a Markov chain. The considerations of

Section 2 enables us to formulate the "brute force" algorithm consisting in analyzing powers of the matrix G_M:

$$G_M, (G_M)^2, (G_M)^3, ..., (G_M)^i, (G_M)^{i+1}, ..., (G_M)^{n \cdot n}.$$

Since the sequence of powers of the matrix G_M is periodic (the number of different $n \times n$ Boolean matrices is $2^{(n \cdot n)}$, we have determined an upper estimate $u(n) = 2^{(n \cdot n)}$, such that if there exists q satisfying $(G_M)^q = E$, then there exist $p \leq u(n)$ such that $(G_M)^p = E$.

We shall now use the fact (Corollary 3.3):

$$\text{if } G^m = E, \text{ then } E = G^{m+1} = G^{m+2} = G^{m+3} = ...$$

to minimize the number of matrix multiplication related to verification whether $(G_M)^p = E$, for some $p \leq 2^{(n \cdot n)}$.

5.1 Algorithm 2

a) the data:
$G = G_M$, a Boolean $n \times n$ matrix G_M representing transition graph of a Markov chain,

b) the result:
the answer, whether it is a matrix of a regular Markov chain,

c) the idea of the algorithm:

```
begin {Algorithm 2}
{first module - begin}
{----------------------------------------}
      . . .
{----------------------------------------}
{first module - end}

{second module - begin}
{----------------------------------------}
  if (test_erg = 0) then test_reg = 0
  else
    begin
      H := G;
      p := 1;
      u := n*n;
      if (H = E) then test_reg := 1
      else
        begin
          test_reg := 0;
          while ((test_reg = 0) & (p < u)) do
            begin
              p = 2*p;
              H := H ⊗ H;
              if (H = E) then test_reg := 1;
            end;
        end;
    end;
{----------------------------------------}
{second module - end}
end.
```

d) the worst–case complexity: $O(n^2)$ matrix multiplications,

e) the upper estimate for analyzed powers of the matrix G:
   $U(n) = 2^{(n \cdot n)}$,

f) the idea of the correctness of the algorithm:
   the correctness of the algorithm immediately follows from Lemma 3.3. and the remark formulated below:

□

5.2 Remark (refers to d))

To decide regularity of an ergodic Markov chain suffices n^2 matrix multiplications.

The idea of the proof.

To argue the fact, that for deciding regularity of an ergodic Markov chain suffices $O(n^2)$ matrix multiplications, let us note that the sequence of values of the matrix variable H, produced by the algorithm, is the following sequence of powers of the matrix G:

$$G^1, G^2, G^4, G^8, G^{16}, G^{32}, \ldots$$

This sequence can be also presented as

$$G^{2^0}, G^{2^1}, G^{2^2}, G^{2^3}, G^{2^4}, G^{2^5}, \ldots, G^{2^{(n \cdot n)}}, \ldots, G^{2^m}.$$

and the values of exponents are subsequent values of the variable p. Moreover, the whole number m of repetitions of the loop "while" of the second module of Algorithm 2 is the least number satisfying $2^m \geq 2^{(n \cdot n)}$. According to Lemma 3.3 if the sequence

$$G^1, G^2, G^3, G^4, G^5, ..., G^{2^{(n \cdot n)}},$$

contains the matrix E, $G^k = E$, then $E = G^{k+1} = G^{k+2} = G^{k+3} = ... = G^{2^{(n \cdot n)}}$, and therefore, the sequence

$$G^{2^0}, G^{2^1}, G^{2^2}, G^{2^3}, G^{2^4}, G^{2^5}, ..., G^{2^m}$$

where q is the least number satisfying $2^m \geq 2^{(n \cdot n)}$, also contains the matrix E. Thus $m = O(n^2)$.

□

5.3 Corollary

If we use the ordinary procedure of Boolean matrix multiplication then the worst–case complexity of Algorithm 2 is $\Theta(n^5)$.

□

5.4 Corollary (cf. also Corollary 5.12)

If we use the method of four Russian or the Strassen method for matrix multiplication then the worst–case complexity of Algorithm 2 is of the order less than $O(n^5)$.

□

We shall now use the facts presented in Section 3 (Boolean matrix multiplication) and Section 4 (cycles in transition graphs of ergodic chains) to improve Algorithm 1 in another manner.

5.5 Algorithm 3

a) the data:
$G = G_M$, a Boolean $n \times n$ matrix G_M representing transition graph of a Markov chain,

b) the result:
the answer, whether it is a matrix of a regular Markov chain,

c) the idea of the algorithm:

```
begin {Algorithm 3}
{first module - begin}
{-----------------------------------------}
      . . .
{-----------------------------------------}
{first module - end}

{second module - begin}
{-----------------------------------------}
  if (test_erg = 0) then test_reg = 0
  else
    begin
      H := G;
      p := 1;
      u := n*(n-1);
      if (H >= I) then test_I := 1
      else test_I := 0;
      {first loop - begin}
      while (test_I = 0) do
        begin
          H := H ⊗ G;
          if (H >= I) then test_I := 1;
          p = p + 1;
        end;
      {first loop - end}
      {first loop is repeated at most (n^2 - n) times,
       because the chain is ergodic}
      {this loop ends with test_I = 1 and H >= I}
      K := H;
      p := 1;
      u := n*n - n;
      if (H = E) then test_reg := 1
      else test_reg := 0;
      {second loop - begin}
      while ((test_reg = 0) & (p < u))
        begin
          H := H ⊗ K;
          if (H = E) then test_reg := 1;
          p = p + 1;
        end;
      {second loop - end}
      {second loop is repeated
       ??????? NO MORE ??????? than (n^2 - n) times}
      {if the chain is regular then this loop
       ends with test_reg = 1 and H = E}
    end;
{---------------------------------------}
{second module - end}
end.
```

d) the worst–case complexity: $O((n^2 - n) + (n^2 - n)) = O(n^2)$ matrix multiplications,

e) the upper estimate for analyzed powers of the matrix G:
$U(n) = (n^2 - n) \cdot (n^2 - n) = n^2 \cdot (n-1)^2$,

f) the idea of the correctness of the algorithm:
the correctness of the algorithm immediately follows from Lemma 4.1 and Corollary 4.3.

5.6 Remark (refers to f) and e))

To decide regularity of an ergodic Markov chain suffices to consider the powers G^q of the matrix G, representing transition graph of the chain, for $q \leq n^2 \cdot (n-1)^2$.

The idea of the proof.

Let us note that the second module is realized, provided that the chain is ergodic. In this case (cf. Corollary 4.3) the first loop ends its computation with the value of variable `test_I` being 1 and the value of the variable `H` being a matrix K, $K \geq I$. The second loop starts with the initial value of the variable `K` being the value of the variable `H` at the end of first loop. Then the value of the variable `K` (at the start of the second loop) is the power G^p of the matrix G, where $p \leq n \cdot (n-1)$ (cf. Corollar 4.3) and $G^p > I$. This means that the number of positions of this matrix that are equal to 1 is at least n. Denote by $up(L)$, for an $n \times n$ Boolean matrix L, the number of positions equal to 1. Therefore, according to Corollary 4.3, for the subsequent values of the variable `H`,

$$K, K^2, K^3, K^4, \ldots$$

during the realization of the second loop, we have

$$up(K) \leq up(K^2) \leq up(K^3) \leq up(K^4) \leq \ldots$$

Since, for each position (K^m) of this sequence of matrices, $up(K^m) \leq n^2$, this sequence is periodic (cyclic) and the length of the period is less than $(n^2 - n)$.
This means that to decide, whether this sequence contains the matrix E, it is sufficient to analyze only $(n^2 - n - 1)$ first positions of the sequence.
Thus, the final value of the variable `H` at the end of second loop is the power G^q of the matrix G, where $q \leq (n^2 - n) \cdot (n^2 - n)$.
Recapitulation: Algorithm 3 ends his computation with the value of the variable `H` being E and the value of the variable `test_reg` being 1, if and only if, $G^m = E$ for some m, such that $m \leq (n^2 - n) \cdot (n^2 - n) = n^2 \cdot (n-1)^2 = n^4 - 2 \cdot n^3 + n^2$.

□

From Lemma 3.3 immediately follows

5.7 Corollary

A Markov chain with the matrix G is regular if and only if the matrix $G^{n^2 \cdot (n-1)^2}$ is equal to E.

□

Now, we shall improve Algorithm 3 using the same method that we have used in passing from Algorithm 1 to Algorithm 2:

5.8 Algorithm 4

a) the data:
$G = G_M$, a Boolean $n \times n$ matrix G_M representing transition graph of a Markov chain,

b) the result:
the answer, whether it is a matrix of a regular Markov chain,

c) the idea of the algorithm:

```
begin {Algorithm 4}
{first module - begin}
{---------------------------------------}
      . . .
{---------------------------------------}
{first module - end}
{second module - begin}
{---------------------------------------}
if (test_erg = 0) then test_reg = 0
else
```

```
  begin
    H := G;
    p := 1;
    u := n*n*(n-1)*(n-1);
    if (H = E) then test_reg := 1
    else test_reg := 0;
    while ((test_reg = 0) & (p < u))
      begin
        p = 2*p;
        H := H ⊗ H;
        if (H = E) then test_reg := 1
        else test_reg := 0;
      end;
  end;
{second module - end}
{--------------------------------------}
end.
```

d) the worst–case complexity: $O(\log_2((n^2-n)\cdot(n^2-n))) = O(\log_2 n)$ matrix multiplications,

e) the upper estimate for analyzed powers of the matrix G:
$U(n) = (n^2-n)\cdot(n^2-n) = n^2\cdot(n-1)^2$,

f) the idea of the correctness of the algorithm:
the correctness of Algorithm 4 is an immediate consequence of Remark 5.9 formulated below:

5.9 Remark (refers to f))

To decide regularity of an ergodic Markov chain suffices to consider the powers of the matrix G representing transition graph of the chain:

$$G^{2^0}, G^{2^1}, G^{2^2}, G^{2^3}, G^{2^4}, G^{2^5}, \ldots, G^{2^q}, \ldots$$

for least q satisfying $2^q \geq n^2\cdot(n-1)^2$, i.e., $q \geq \log_2(n^2\cdot(n-1)^2) = O(\log_2 n)$.
The idea of the proof.

From Remark 5.6 it follows that to solve regularity of a Markov chain it is sufficient to analyze the sequence of powers of the matrix G representing transition graph of the chain:

$$G^1, G^2, G^4, G^8, G^{16}, G^{32}, \ldots, G^m, \ldots, G^{n^2\cdot(n-1)^2},$$

for $m \leq n^2 \cdot (n-1)^2$. From Lemma 3.3 it follows, that if this sequence contains the matrix E, $G^m = E$, then $E = G^{m+1} = G^{m+2} = G^{m+3} = ... = G^{n^2 \cdot (n-1)^2}$. Therefore, the sequence

$$G^{2^0}, G^{2^1}, G^{2^2}, G^{2^3}, G^{2^4}, G^{2^5}, ..., G^{2^q},$$

where q is the least number satisfying $2^q \geq U(n) = n^2 \cdot (n-1)^2$, also contains the matrix E. It is easy to note that $q = O(\log_2 n)$.

□

The theorem below summarizes all the observations:

5.10 Theorem

M is the matrix of a regular Markov chain if and only if all elements of the sequence of powers of the matrix G,

$$G^{U(n)}, G^{U(n)+1}, G^{U(n)+2}, ...$$

are equal to the matrix E.

□

In the case, where we shall use efficient algorithms for matrix multiplication, e.g., four Russian algorithm or Strassen algorithm we finally obtain the worst-case complexity of the second module at most $\Theta(n^3)$. The worst–case complexity of four Russian algorithm (cf. [1], [2]) is $O(n^3/\log_2 n)$. The worst–case complexity of Strassen algorithm (cf. [1], [2]) is $O(n^{\log_2 7})$.

5.11 Corollary

Consider the case, where the Boolean matrix multiplication $\otimes$ is realized in Algorithm 4 by means of four Russian algorithm (cf. [1], [2]). Then the worst–case complexity of this version of Algorithm 4 is $O(n^3)$.

□

5.12 Corollary

Consider the case, where the Boolean matrix multiplication $\otimes$ is realized in Algorithm 4 by means of Strassen algorithm (cf. [1], [2]). Then the worst–case complexity of this version of Algorithm 4 is less than $O(n^3)$.

□

6. Final remarks

It was a little surprising for the authors that the modules corresponding respectively to investigations of ergodity and regularity of Markov chains are of comparable complexity.

We would like to stress that the proof of the correctness of the algorithms proposed in the paper do not make any use of facts characterizing Markov chains in terms of eigenvalues of matrices.

6.1 Remark

Algorithms presented in the paper were implemented and tested by Krystian Moraś in [10].

The implemented (in C++) version of Algorithm 4 one cane find at: p.wi.pb.edu.pl/wiktor–danko .

References

[1] Aho A.V., Hopcroft J.E., Ullman J.D.: Design and Analysis of Computer Algorithms, Addison-Vesley, 1981.

[2] Cormen T., Leiserson C., Riverst R., Stein C.: Introduction to Algorithms, Massachusetts Institute of Technology, 2001.

[3] Dańko A., Dańko W.: Improving Pseudo-Random Generators, International Conference on Biometrics and Kansei Engineering 24-28 June, Cieszyn, Poland, pp. 163-166, 2009.

[4] Dańko W.: The Set of Probabilistic Algorithmic Formulas Valid in a Finite Structure is Decidable with Respect to Its Diagram, Fundamenta Informaticae, vol. 19 (3-4), pp. 417-431, 1993.

[5] Feller W.: An Introduction to Probability Theory and Its Applications, John Wiley and Sons, Inc., New York, London, 1961.

[6] Josifescu M.: Finite Markov Processes and Their Applications, John Wiley and Sons, New York, London 1988.

[7] Kubik L., Krupowicz A.: Introduction to Probability Theory and Its Applications, PWN, Warszawa, 1982, (in Polish).

[8] Mirkowska G., Salwicki A.: Algorithmic Logic, D.Reidel Publ. Co. & PWN, Warsaw, 1987.

[9] Modica G., Poggiolini L.: A First Course in Probability and Markov Chains, John Wiley and Sons, 2013.

[10] Moraś K.: Markov Chains. Analysis of Regular Chains, bachelor thesis, Bialystok University of Technology, 2015, (in Polish).

[11] Ross S. M.: Introduction to Probability Models Academic Press, 2003.

[12] Rubino G., Sericola B.: Markov Chains and Dependability Theory, Cambridge University Press, 2014.

[13] Snell Laurie J.: Introduction to Probability, Random House Inc., New York, 1988.

EFEKTYWNE ALGORYTMY ROZSTRZYGANIA REGULARNOŚCI ŁAŃCUCHÓW MARKOWA

Streszczenie W pracy proponujemy algorytmy rozstrzygające regularność łańcuchów Markowa o macierzy przejść rozmiaru $n \times n$. Najniższa złożoność takiego algorytmu może być nie większa niż $O(n^3)$ i podana jest argumentacja, że nie można jej istotnie obniżyć.

Słowa kluczowe: łańcuch Markowa, ergodyczny łańcuch Markowa, regularny łańcuch Markowa