SIAM J. MATRIX ANAL. APPL.
Vol. 39, No. 1, pp. 83–98

PRIMITIVE SETS OF NONNEGATIVE MATRICES AND SYNCHRONIZING AUTOMATA*

BALÁZS GERENCSÉR†, VLADIMIR V. GUSEV‡, AND RAPHAËL M. JUNGERS§

Abstract. A set of nonnegative matrices $\mathcal{M} = \{M_1, M_2, \ldots, M_k\}$ is called *primitive* if there exist possibly equal indices $i_1, i_2, \ldots, i_m$ such that $M_{i_1} M_{i_2} \cdots M_{i_m}$ is entrywise positive. The length of the shortest such product is called the *exponent* of $\mathcal{M}$. Recently, connections between synchronizing automata and primitive sets of matrices were established. In the present paper, we strengthen these links by providing equivalence results, both in terms of combinatorial characterization and computational complexity. We pay special attention to the set of matrices without zero rows and columns, denoted by $\mathcal{NZ}$, due to its intriguing connections to the Černý conjecture. We rely on synchronizing automata theory to derive a number of results about primitive sets of matrices. Making use of an asymptotic estimate by Rystsov [*Cybernetics*, 16 (1980), pp. 194–198], we show that the maximal exponent $\exp(n)$ of primitive sets of $n \times n$ matrices satisfy $\lim_{n\to\infty} \frac{\log \exp(n)}{n} = \frac{\log 3}{3}$ and that the problem of deciding whether a given set of matrices is primitive is PSPACE-complete, even in the case of two matrices. Furthermore, we characterize the computational complexity of different problems related to the exponent of $\mathcal{NZ}$ matrix sets and present a bound of $2n^2 - 5n + 5$ on the exponent when considering the subclass of matrices having total support.

Key words. nonnegative matrices, primitive sets of matrices, the exponent of a matrix set, carefully synchronizing automata, the Černý conjecture

AMS subject classifications. 15B34, 15B48, 05A05, 68R05, 68Q45

DOI. 10.1137/16M1094099

1. Introduction.

A nonnegative matrix M of size $n \times n$ is called *primitive* if M^k is positive (i.e., has all its entries larger than zero) for a positive integer k. This notion was introduced by Frobenius in 1912 during the development of so-called Perron–Frobenius theory. This theory has found numerous applications since then: in the theory of Markov chains, economics, population modeling, and centrality measures in networks; see [30, Chapter 8] for an introduction to the topic. Motivated by various applications Protasov and Voynov introduced the following generalization of this notion to sets of matrices [36]: a finite set of (entrywise) nonnegative matrices $\mathcal{M} = \{M_1, M_2, \ldots, M_k\}$ is called *primitive* if $M_{i_1} M_{i_2} \ldots M_{i_m}$ is (entrywise) positive for possibly equal indices $i_1, i_2, \ldots, i_m \in [1, k]$. The length of the shortest such product is called the exponent $\exp(\mathcal{M})$ of $\mathcal{M}$. We will denote the value of the largest exponent

*Received by the editors September 14, 2016; accepted for publication (in revised form) by M. Benzi September 26, 2017; published electronically January 16, 2018.
http://www.siam.org/journals/simax/39-1/M109409.html
Funding: This work was supported by the French Community of Belgium, ARC 13/18-054, and by the IAP network DYSCO. The work of the first author was also supported by the Hungarian Academy of Sciences. The work of the second author was supported by the Russian Foundation for Basic Research (RFBR) (grant 16-01-00795), Ministry of Education and Science of the Russian Federation (Minobrnauka) (project 1.3253.2017), and the Competitiveness Program of Ural Federal University. The third author is an F.R.S./F.N.R.S. (Fonds De La Recherche Scientifique) research associate and a Fulbright fellow.
†Alfréd Rényi Institute of Mathematics, Hungarian Academy of Sciences, Budapest H-1053, Hungary (gerencser.balazs@renyi.mta.hu).
‡ICTEAM Institute, Université Catholique de Louvain, Louvain-la-Neuve 1348, Belgium, and Institute of Natural Sciences and Mathematics, Ural Federal University, Ekaterinburg, Russia (vl.gusev@gmail.com).
§ICTEAM Institute, Université Catholique de Louvain, Louvain-la-Neuve 1348, Belgium (raphael.jungers@uclouvain.be, http://perso.uclouvain.be/raphael.jungers/).

$$\mathcal{M} = \left\{ M_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}, M_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \right\}$$

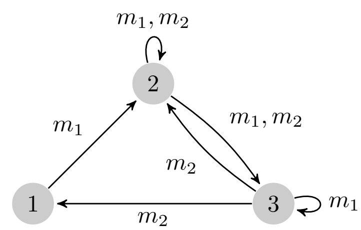

Fig. 1. *The matrix set $\mathcal{M}$ and the corresponding automaton $\mathscr{F}$.*

among all primitive sets of $n \times n$ matrices by $\exp(n)$. For example, the matrix set $\mathcal{M}$ in Figure 1 is primitive, since the product $M_1M_2M_1M_2$ is entrywise positive, and its exponent is equal to 4. Since the actual values of positive entries of matrices in $\mathcal{M}$ do not influence the exponent, in the rest of our paper we will implicitly assume that entries of all matrices are equal to 0 or 1. Moreover, we assume that the product AB of two matrices of size $n \times n$ is also a $(0, 1)$-matrix that is defined as follows:[1] $AB[i, j] = 1$ if $\sum_k A[i, k]B[k, j] > 0$, and $AB[i, j] = 0$ otherwise. Other concepts of primitivity were introduced in [11, 14].

Primitive sets of matrices received a lot of attention for different reasons. We refer the reader to the introduction of [6] for the account of applications of primitive matrix sets to stochastic control theory and to the consensus problem. Primitive sets of matrices further arise in the study of infinite matrix products [21] and are of importance in mathematical ecology [27]. Moreover, primitive matrices are tightly connected to scrambling matrices. A nonnegative square matrix M is *scrambling* if MM^t is entrywise positive. The scrambling index of M is the smallest k such that M^k is scrambling. Note that the scrambling index of a primitive matrix is well-defined and bounded by the exponent. These notions and their generalizations have recently received a lot of attention independently and also in relation with primitivity; see, e.g., [1, 13] and references therein. The classical concept of contractivity in the theory of time-inhomogeneous Markov chains can be stated in terms of scrambling matrices as well: a family of row-stochastic matrices is contractive if there exists k such that every product of k matrices is scrambling [33]. In particular, a primitive row-stochastic matrix is contractive. We refer the reader to [36, section 5] for a discussion on relations between primitivity and contractivity. Furthermore, primitive sets of matrices are tightly related to Boolean networks, which are widely used in biology to model gene regulatory networks. A special class of Boolean networks—disjunctive networks—can be seen as a set of matrices over the Boolean semiring. While researchers in theoretical biology are mostly interested in the attractors and the limit cycles for different types of update schedules (see, for example, [19]), we are mainly interested whether it is possible and how fast one can achieve the all-one state.

The subfamily of nonnegative matrices that have no zero rows and columns, denoted by $\mathcal{NZ}$, will be of major interest to us for the following reasons. A matrix M is called *irreducible* if for every i, j there exists a positive integer m such that $M^m[i, j] > 0$. A set of $k \geq 1$ matrices $\mathcal{M}$ is irreducible if the matrix $\sum_{i=1}^k M_i$ is irreducible. As usual, we will denote by e_i the ith vector of the *canonical basis* in $\mathbb{R}^n$, the ith entry of e_i is one and all the others are zeros. We say that a matrix M acts as a permutation on a partition $V_1, V_2, \ldots, V_m$ of the vectors of the canonical basis if there exists a permutation σ such that for all i, M maps all the elements of V_i into the subspace spanned by $V_{\sigma(i)}$. A classical theorem of Perron–Frobenius theory

[1]Formally speaking, we consider the matrices over the Boolean semiring.

states that an irreducible matrix M is primitive if and only if there is no partition $V_1, \dots, V_m$ of the canonical basis vectors for $m > 1$ such that M acts as a permutation on $V_1, \dots, V_m$. Protasov and Voynov generalized this theorem to sets of matrices belonging to $\mathcal{NZ}$ [36]: an irreducible set of matrices belonging to $\mathcal{NZ}$ is primitive if and only if there is no partition $V_1, \dots, V_m$ for $m > 1$ such that every $M \in \mathcal{M}$ acts as a permutation on $V_1, \dots, V_m$; see [2, 6, 42] for alternative proofs. Thus, the class of primitive matrices belonging to $\mathcal{NZ}$ can be seen as the right class for the Perron–Frobenius-type theory of matrix sets. This characterization also leads to an efficient algorithm that decides whether a set of matrices belonging to $\mathcal{NZ}$ is primitive.

1.1. Automata viewpoint. A classical and convenient way to analyze nonnegative matrices is to represent them as digraphs. With every set $\mathcal{M} = \{M_1, M_2, \dots, M_k\}$ we associate the edge-labeled digraph $\mathscr{F}$ defined as follows (see Figure 1): the set of vertices is equal to $[1, n]$ and there is a (directed) edge from s to t labeled by m_i if and only if $M_i[s, t] > 0$. Recall that a *path* is a sequence of edges $p_1 p_2 \dots p_\ell$ such that the final vertex of p_j is equal to the initial vertex of p_{j+1} for all $j \in [1, \ell - 1]$. The label of a path is the sequence of labels associated with its edges. The following simple observation is very important for us: for all $s, t \in [1, n]$ and every choice of indices $i_1, i_2 \dots, i_m \in [1, k]$ we have $M_{i_1} M_{i_2} \dots M_{i_m}[s, t] > 0$ if and only if there is a path in $\mathscr{F}$ from s to t labeled by $m_{i_1} m_{i_2} \dots m_{i_m}$. In particular, $\mathcal{M}$ is primitive if and only if there exists a word w over the alphabet $m_1, \dots, m_k$ such that for every pair of vertices there is a path between them labeled by w.

Since our paper heavily relies on results from automata theory we will adopt terminology and notation of this field [22]. An *automaton* $\mathscr{A}$ is an edge-labeled digraph formally defined as a triple[2] $\langle Q, \Sigma, \delta \rangle$, where Q is the set of vertices traditionally called *states*, Σ is the set of labels called *the alphabet*, elements of Σ are called *letters*, and sequences of letters are called *words*. The *transition function* $\delta : Q \times \Sigma \to 2^Q$ is encoding the edges as follows: a state f belongs to $\delta(q, m_i) \subseteq Q$ if and only if there is an edge going from q to f labeled by m_i.

Except for Theorem 2, all automata appearing in the paper will belong to the following special class. An automaton $\mathscr{A}$ is *partial* if the cardinality of $\delta(q, m_i)$ is at most one for all $q \in Q$ and $m_i \in \Sigma$. We say that the transition at $q \in Q$ labeled by $m_i \in \Sigma$ is *undefined* if $\delta(q, m_i) = \emptyset$. An automaton is *complete* if there are no undefined transitions. Note that complete automata satisfy the following property: for a given state q and a word w the final state of a path starting at q and labeled by w is uniquely defined. We denote this final state by $q \cdot w$. We keep the same notation for partial automata allowing $q \cdot w$ to be undefined.

1.2. Synchronizing automata. A complete automaton $\mathscr{A}$ is called *synchronizing* if there exist a word w and a state f such that for every state q we have $q \cdot w = f$. Any such word is called a *synchronizing* or *reset* word for $\mathscr{A}$. The length of the shortest such word is called *the reset threshold* $\mathrm{rt}(\mathscr{A})$ of $\mathscr{A}$. Synchronizing automata naturally appear in different areas of research. For example, they were used to model sensorless parts orienting problems: given a part and a set of available actions that can change its spatial orientation, find a sequence of actions that would bring the part to a desired orientation independently of the initial position [31]. Clearly, if we consider an automaton $\mathscr{A}$ with the set of spatial orientations as the set of states, and the available actions as letters, then the "orienting sequence" corresponds to a

[2] There is a large number of automata models which can also include initial and final states, stacks, various outputs, etc. In our paper we will utilize one of the most basic models.

synchronizing word of $\mathscr{A}$. We refer the reader to [40] for the survey of main results and other applications. A recent account of applications of synchronizing automata in group theory can be found in [4]. Persisting interest of the research community to the topic is also driven by one of the most famous open problems in automata theory. Namely, *the Černý conjecture* states that the reset threshold of an n-state automaton is at most $(n-1)^2$ [9, 10]. This bound is reached by the n-state Černý automaton $\mathscr{C}_n$ (see [40, p. 18]), but despite intensive efforts of researchers, the best upper bound $\frac{n^3-n}{6}$ was obtained more than 30 years ago in [15, 35] and independently in [26].

The notion of a synchronizing automaton can be generalized in different ways to incomplete automata [24]. We will focus our attention on the most relevant for us. A partial automaton is *carefully synchronizing* if there exist a word w and a state f such that $q \cdot w$ is defined and equal to f for every state q. Any such word is called a *carefully synchronizing* word. The length of the shortest such word we will denote by $\mathrm{car}(\mathscr{A})$. We will denote by $\mathrm{car}(n)$ the maximum of $\mathrm{car}(\mathscr{A})$ among all carefully synchronizing n-state partial automata. Essentially, carefully synchronizing automata model the problem of bringing a simple finite-state device to a known state with a single input sequence, while avoiding undefined transitions, which are undesirable or can break the device. In matrix terms, it amounts to considering a set of matrices with *at most* one 1-entry per row, and to ask for a product with one (entrywise) positive column.

1.3. Our contributions. Our results can be informally arranged into three different groups. The contributions of the first group significantly improve the understanding of the relationships between primitive sets of matrices and synchronizing automata. The work within this framework started in [3], where well-known examples of primitive matrices with large exponent were used to construct series of automata with relatively large reset thresholds, so-called slowly synchronizing automata. In [6] it was shown that a $f(n)$ bound on the reset threshold of n-state automata implies a $2f(n)+n-1$ bound on the exponent of $\mathcal{NZ}$ matrix sets. We significantly improve these results. We show that the growth rate of $\exp(n)$ is equal to $\Theta(\mathrm{car}(n))$, i.e., there are $k_1, k_2, n_0 > 0$ such that $k_1 \,\mathrm{car}(n) \le \exp(n) \le k_2 \,\mathrm{car}(n)$ for all $n > n_0$. Thus, in a certain sense, the study of the exponents of sets of matrices is equivalent to the study of carefully synchronizing automata. We also formulate an analogous result for primitive $\mathcal{NZ}$ matrix sets. Namely, we introduce a special class of automata $\mathcal{C}$ such that the growth rate of the reset thresholds of automata in this class is equivalent to the growth rate of the exponents of $\mathcal{NZ}$ matrix sets. We propose and formalize a new open question of whether a quadratic bound on $\exp_{\mathcal{NZ}}(n)$ leads to a breakthrough on the Černý conjecture.

The contributions of the second group are of combinatorial nature. Our first result is $\lim_{n\to\infty} \frac{\log \exp(n)}{n} = \frac{\log 3}{3}$. It is the immediate consequence of $\exp(n) = \Theta(\mathrm{car}(n))$ stated above and a major result of Rystsov [37] showing $\lim_{n\to\infty} \frac{\log \mathrm{car}(n)}{n} = \frac{\log 3}{3}$. This contribution can be seen as an extension of a classical theorem by Wielandt that the exponent of a single matrix is at most $(n-1)^2+1$; see, for example, [23, Corollary 8.5.9]. It also answers the question of establishing the growth rate of $\exp(n)$ posed in [6]. Another contribution in this group is a partial result for $\mathcal{NZ}$ matrix sets. Recall that a matrix M has *total support* if every nonzero element $m_{i,j}$ of M lies on a positive diagonal, i.e., for every $i, j \in [1, n]$ such that $m_{i,j} > 0$ there exists a permutation σ with the following properties: $\sigma(i) = j$ and for every $k \in [1, n]$ we have $m_{k,\sigma(k)} \neq 0$. We prove that the exponent of a set of matrices having total support is bounded by $2n^2 - 5n + 5$. In the proof we utilize the well-known theorem by Kari that the reset threshold of an Eulerian automaton is bounded by $n^2 - 3n + 3$. This result suggests that the bounds for other classes of synchronizing automata might be used to obtain upper bounds on the exponent in the special classes of $\mathcal{NZ}$ matrix sets.

The contributions of the last group are related to the computational complexity of basic problems about primitive sets of matrices. We show that the problem of deciding whether a given set of two matrices is primitive is PSPACE-complete. Furthermore, given a set of two matrices belonging to $\mathcal{NZ}$ and possibly an integer k encoded in binary, we establish the exact computational complexity of the following problems:

1. the problem of deciding whether $\exp(\mathcal{M}) \leq k$ is NP-complete;
2. the problem of deciding whether $\exp(\mathcal{M}) = k$ is DP-complete;
3. the problem of computing $\exp(\mathcal{M})$ is $FP^{NP[\log]}$-complete.

Furthermore, we show that unless $P = NP$, for every positive ε there is no polynomial-time algorithm that computes the exponent of an $\mathcal{NZ}$ matrix set with the approximation ratio $n^{1-\varepsilon}$, even in the case of only three matrices in the set. These results are based on a single relatively simple reduction from synchronizing automata to sets of matrices belonging to $\mathcal{NZ}$. Moreover, for every positive integer c we also present a polynomial time algorithm that approximates the exponent of a given set of $n \times n$ matrices belonging to $\mathcal{NZ}$ within a factor $\frac{n}{c}$.

The paper is organized as follows. Section 2 deals with the primitive sets of matrices in the general case. In subsection 2.1 we show that $\exp(n) = \Theta(\mathrm{car}(n))$ and $\lim_{n\to\infty} \frac{\log \exp(n)}{n} = \frac{\log 3}{3}$. In subsection 2.2 we show that the problem of deciding whether a given set of matrices is primitive is PSPACE-complete. Section 3 is devoted to the $\mathcal{NZ}$ matrix sets. In subsection 3.1 we introduce the class $\mathcal{C}$ such that $\exp_{\mathcal{NZ}}(n) = \Theta(\mathrm{rt}_{\mathcal{C}}(n))$. In subsection 3.2 we present a quadratic bound on the exponent of a set of matrices having total support. In subsection 3.3 we deal with the complexity issues related to computation and approximation of the exponent of $\mathcal{NZ}$ matrix sets.

1.4. Notation.

We denote by $[1, n]$ the set of positive integers i such that $1 \leq i \leq n$. The row vector of all ones is denoted by e, and e_s stands for the row vector with a single nonzero entry equal to 1 at position s. For a matrix A we denote by $A[i, j]$ the element at the ith row and jth column. Also, $AB[i, j]$ should be interpreted as the element at the ith row and jth column of the product AB. For two matrices A, B of the same size we write $A \leq B$ if $A[i, j] \leq B[i, j]$ for all possible i, j. For two arbitrary functions $f, g : \mathbb{N} \to \mathbb{N}$ we write $f(n) = O(g(n))$ if there are positive real numbers k, n_0 such that $f(n) \leq kg(n)$ for all integers $n > n_0$; $f(n) = \Omega(g(n))$ if there are positive real numbers k, n_0 such that $kg(n) \leq f(n)$ for all integers $n > n_0$; and $f(n) = \Theta(g(n))$ if $f(n) = O(g(n))$ and $f = \Omega(g(n))$.

2. The general case.

2.1. Bounds on the exponent.

The growth rate of $\exp(n)$ is one of the most basic questions one can ask about the primitive sets of matrices. Furthermore, an upper bound on $\exp(n)$ gives a bound on the running time of the straightforward algorithm that decides whether a given set of matrices is primitive: we iterate through all the possible products of length up to $\exp(n)$ and check whether they contain a positive matrix. Since the problem is NP-hard [6, Theorem 6], such a simple algorithm might be the best we can hope for. The best known bounds on $\exp(n)$ are summarized in the following theorem.

THEOREM 1 (see [6, Theorem 10]). *If $\mathcal{M}$ consists of m matrices of size $n \times n$, then $\exp(\mathcal{M}) \leq 2^{n^2}$. Moreover, if $m \geq 4$, then for all $\varepsilon > 0$ there exists a sequence of positive integers $n_1, n_2, \ldots,$ tending to infinity such that $((1-\varepsilon)e)^{\sqrt{n_k/2}} \leq \exp(n_k)$.*

The first part of this theorem is actually well-known in the literature and easily follows from the fact that the number of $(0,1)$-matrices of size $n \times n$ is 2^{n^2}. The second part requires a nontrivial construction. In the upcoming theorem we are going to show that $\exp(n)$ grows asymptotically as $\operatorname{car}(n)$. Thus, we can utilize the known bounds on $\operatorname{car}(n)$ to infer the bounds on $\exp(n)$. Theorem 2 extends [6, Theorem 16] from primitive sets of matrices belonging to $\mathcal{NZ}$ to all primitive sets of matrices. Furthermore, we make the construction deterministic, which will be crucial for Theorem 8. Nevertheless, the first part of the proof essentially remains the same as in [6] and is presented here for the convenience of the reader.

Before stating the theorem we require one last definition. Given an automaton $\mathscr{A} = \langle [1,n], \Sigma, \delta \rangle$, the *adjacency matrix* M_ℓ of a letter $\ell \in \Sigma$ is defined as follows: $M_\ell[i,j] = 1$ if $j \in \delta(i,\ell)$, and $M_\ell[i,j] = 0$ otherwise. In Figure 1 the matrix M_1 is the adjacency matrix of the letter m_1.

THEOREM 2. *Let* $\exp(n)$ *be the maximum value of the exponent among all primitive sets of* $n \times n$ *matrices and* $\operatorname{car}(n)$ *be the maximum value of* $\operatorname{car}(\mathscr{A})$ *among all carefully synchronizing* n*-state partial automata* $\mathscr{A}$*. It holds that* $\exp(n) = \Theta(\operatorname{car}(n))$.

Proof. Part I: $\exp(n) = O(\operatorname{car}(n))$. Let us consider an arbitrary primitive set of matrices $\mathcal{M} = \{M_1, \ldots, M_k\}$. We are going to show that $\exp(\mathcal{M}) \leq 2\operatorname{car}(n) + n - 1$. We will achieve this by presenting products P, T, R of matrices in $\mathcal{M}$ with the following properties:

1. the ith column of P is positive for some i and the length of P is at most $\operatorname{car}(n)$;
2. the jth row of R is positive for some j and the length of R is at most $\operatorname{car}(n)$;
3. $T[i,j] > 0$ and the length of T is at most $n-1$.

These properties clearly imply that PTR is positive and the length of PTR is at most $2\operatorname{car}(n) + n - 1$. Thus, $\exp(\mathcal{M}) \leq 2\operatorname{car}(n) + n - 1$.

In order to construct the product P we will consider a partial automaton $\mathscr{A} = \langle [1,n], \Sigma, \delta_{\mathscr{A}} \rangle$. The elements of the alphabet Σ are all possible functions $\varphi : D \to [1,n]$ with $D \subseteq [1,n]$ such that there exists a matrix $M \in \mathcal{M}$ with the following properties: $i \in D$ if and only if the ith row of M has a positive entry; and for all $i \in D$ we have $M[i, \varphi(i)] > 0$. The transition function is defined naturally as $\delta_{\mathscr{A}}(i, \varphi) = \varphi(i)$. First, we are going to show that $\mathscr{A}$ is carefully synchronizing, and then we will use the shortest carefully synchronizing word of $\mathscr{A}$ to obtain the matrix product P.

Claim: $\mathscr{A}$ is carefully synchronizing. Let $\mathscr{F}$ be the automaton associated with $\mathcal{M}$. As was noted in section 1.1 there exists a word w such that for every pair of states of $\mathscr{F}$ there is a path between them labeled by w. We will show now that the word w can be transformed into a carefully synchronizing word of $\mathscr{A}$. Let us fix a state $t \in [1,n]$. There are paths $\pi_1, \ldots, \pi_n$ in $\mathscr{F}$ labeled by w and for every s the path π_s goes from the state s to the state t. Furthermore, we can impose an additional property on these paths. Namely, if after a certain number of steps paths π_x and π_y are in the same state, then their continuations should coincide. Observe now that the paths $\pi_1, \ldots, \pi_n$ can be easily treated as paths leading to the state t in the partial automaton $\mathscr{A}$: by construction for each letter m of $\mathscr{F}$ and states $v_{i_1}, v_{i_2}, \ldots v_{i_p}$ with the property $v_{i_x} \in \delta_{\mathscr{F}}(i_x, m)$ for all $x \in [1,p]$, there exists a letter ℓ of $\mathscr{A}$ such that $\delta_{\mathscr{A}}(i_x, \ell) = v_{i_x}$ for each $x \in [1,p]$. Therefore, there exists a word w' such that for every state s in $\mathscr{A}$ there is a path from s to t labeled by w'. Thus, $\mathscr{A}$ is carefully synchronizing and the claim holds.

Let $w = w_1 w_2 \ldots w_h$ be the shortest carefully synchronizing word of $\mathscr{A}$. It is easy to see that the product $W_1 W_2 \ldots W_h$ contains a column of ones, where W_x is the adjacency matrix of w_x for $x \in [1, h]$. Since by construction for each matrix W_x there exists $A_x \in \mathcal{M}$ such that $W_x \leq A_x$ we obtain a product $P = A_1 A_2 \dots A_h$ with the properties: P has a column of ones and its length is bounded by $\mathrm{car}(n)$.

The product R is constructed in the same manner by applying the reasoning of the previous paragraphs to the matrix set $\mathcal{M}^T = \{M^T \mid M \in \mathcal{M}\}$. The resulting product R^T has a column of ones and is of length at most $\mathrm{car}(n)$.

The existence of the product T easily follows from the fact that $\mathcal{M}$ is irreducible.

Part II: $\exp(n) = \Omega(\mathrm{car}(n))$. For every carefully synchronizing n-state automaton $\mathscr{A}$ we will construct a primitive set of $n \times n$ matrices $\mathcal{M}$ such that $\mathrm{car}(\mathscr{A}) \leq \exp(\mathcal{M})$. Let $\mathcal{E} = \{e_k^T e \mid k \in [1, n]\}$. The set of matrices $\mathcal{M}$ is defined as the union $\mathcal{M}' \cup \mathcal{E}$, where $\mathcal{M}'$ is the set of the adjacency matrices of letters of the partial automaton $\mathscr{A}$. Since $\mathscr{A}$ is carefully synchronizing, there is a product P of matrices in $\mathcal{M}'$ such that the ith column is positive for some i. If we multiply P by the matrix $e_i^T e \in \mathcal{E}$ on the right, we obtain a positive matrix product. Thus, $\mathcal{M}$ is primitive.

It remains to show that $\mathrm{car}(\mathscr{A}) \leq \exp(\mathcal{M})$. Let W be the shortest positive product of matrices in $\mathcal{M}$. Note, that W contains at least one matrix from $\mathcal{E}$, since every product of matrices in $\mathcal{M}'$ contains at most one 1 in each row. Let $W = UEV$, where the product U does not contain matrices from $\mathcal{E}$ and $E = e_k^T e$ for some k. Observe that the kth column of U is positive. Otherwise, W will have a zero row due to the presence of a zero row in UE. Therefore, the length of the product U is at least $\mathrm{car}(\mathscr{A})$ and we obtain the desired inequality. ∎

The following result immediately follows from Theorem 2 and the estimate on $\mathrm{car}(n)$ obtained by Rystsov [37]. Namely, he proved that $\lim_{n\to\infty} \frac{\log \mathrm{car}(n)}{n} = \frac{\log 3}{3}$ (he denoted $\mathrm{car}(n)$ by $T(n)$ in his work). We want also to note that the works [17] and [29] contain a partial rediscovery of Rystsov's result.

THEOREM 3. *Let* $\exp(n)$ *be the maximum value of the exponent among all primitive sets of* $n \times n$ *matrices. It holds that* $\lim_{n\to\infty} \frac{\log \exp(n)}{n} = \frac{\log 3}{3}$.

2.2. Decidability is PSPACE-complete. It was shown in [6, Theorem 6] that the problem of deciding whether a given set of three matrices is primitive is NP-hard. In this subsection we will significantly strengthen this result by showing that the problem of deciding whether a given set of two matrices is primitive is PSPACE-complete. We assume that the reader is familiar with computational complexity theory. All the missing definitions can be found in the book by Arora and Barak [5].

A state s of a (partial or complete) automaton $\mathscr{A} = \langle Q, \Sigma, \delta \rangle$ is a *sink* if for every $a \in \Sigma$ either $\delta(s, a) = s$ or undefined. We will rely on the following theorem by Martyugin [28, Proposition 2] to obtain our result.

THEOREM 4. *Given a partial automaton* $\mathscr{A}$ *with an alphabet of two letters and a sink state, the problem of deciding whether* $\mathscr{A}$ *is carefully synchronizing is PSPACE-space complete.*

The idea of our reduction is given in the following lemma.

LEMMA 5. *Given a partial automaton* $\mathscr{A}$ *with an alphabet of* k *letters and a sink state, a set of* k *matrices* $\mathcal{M}$ *can be constructed in polynomial time such that* $\mathcal{M}$ *is primitive if and only if* $\mathscr{A}$ *is carefully synchronizing.*

Proof. We construct the matrix set $\mathcal{M}$ as follows. If $\mathscr{A}$ has at least two sink states, then $\mathscr{A}$ is not carefully synchronizing. Thus, we can put $\mathcal{M}$ to be an arbitrary nonprimitive set of matrices, and the theorem holds true. Assume now that $\mathscr{A}$ has a unique sink state, which we denote by s. Let $\mathcal{M}'$ be the set of adjacency matrices of letters of $\mathscr{A}$. The matrix set $\mathcal{M}$ consists of matrices in $\mathcal{M}'$ modified in such a way that the sth row of every matrix is positive, i.e., $\mathcal{M} = \{M' + e_s^T e \mid M' \in \mathcal{M}'\}$.

Assume first that the partial automaton $\mathscr{A}$ is carefully synchronizing. We need to show that the set $\mathcal{M}$ is primitive. Since $\mathscr{A}$ is carefully synchronizing, there exists a product P' of matrices from $\mathcal{M}'$ with a positive column. Due to the fact that for every letter ℓ of $\mathscr{A}$ one has $s \cdot \ell = s$, we conclude that the only positive entry in the sth row of P' is located at position s. Thus, the sth column of P' is positive. Altering each matrix M' of the product P' to a matrix $M \in \mathcal{M}$ with the property $M' \leq M$ we obtain a product P of matrices from $\mathcal{M}$ in which column s is positive. Now multiply by any matrix $M \in \mathcal{M}$ to get a positive product PM. Thus, the set $\mathcal{M}$ is primitive.

Assume now that $\mathcal{M}$ is primitive. Let $P = M_1 M_2 \dots M_m$ be a positive product of matrices in $\mathcal{M}$. Let $P' = M_1' M_2' \dots M_m'$ be the corresponding product of matrices in $\mathcal{M}'$, where M_i and M_i' differ only in the sth row. For each row $i \neq s$ there exists the largest h_i such that the ith row of $P_{h_i} = M_1 M_2 \dots M_{h_i}$ contains a unique positive entry. Due to the fact that the ith row of P_{h_i+1} has several positive entries and the structure of matrices in $\mathcal{M}$ we conclude that the unique positive entry occupies the position s in the ith row of P_{h_i}. Thus, the only positive entry in the ith row of P'_{h_i} is located at position s. Therefore, the sth column of the product P' is positive and $\mathscr{A}$ is carefully synchronizing. ∎

THEOREM 6. *Given a set $\mathcal{M}$ of two $n \times n$ matrices, the problem of deciding whether the set $\mathcal{M}$ is primitive is $PSPACE$-complete.*

Proof. Theorem 4 and Lemma 5 imply that the problem of deciding whether $\mathcal{M}$ is primitive is PSPACE-hard.

It remains to show that the problem belongs to PSPACE. The famous result by Savitch states that PSPACE = NPSPACE [38]. Thus, it is enough to present a nondeterministic Turing machine that solves the problem using only polynomial space. It operates as follows. We initialize the matrix P as the identity matrix and $\ell = 0$. On each step we nondeterministically choose a matrix M from the set of matrices $\mathcal{M}$ and put $P = PM$, $\ell = \ell + 1$. If P is positive, then we have found a positive product and accept the input. If $\ell > 2^{n^2}$, then we reject the input. Note that the storage of P, ℓ (in binary) and the necessary operations require only polynomial space. Since a nondeterministic Turing machine accepts the input if and only if there is a sequence of nondeterministic choices leading to acceptance, we conclude that the presented machine accepts $\mathcal{M}$ if and only if there exists a positive product of matrices from $\mathcal{M}$ of length at most 2^{n^2}. By Theorem 1 it is equivalent to primitivity. ∎

3. Sets with no zero rows nor zero columns.

3.1. Bounds on the exponent. A quadratic lower bound on the maximal exponents of sets of matrices belonging to $\mathcal{NZ}$ was obtained in [6, Corollary 20]. A first cubic upper bound $\frac{n^3+n^2-4n+2}{2}$ was given in [41, Theorem 1].[3] The proof relies on linear algebraic techniques. This bound was improved in [6, Corollary 18] to $\frac{n^3+2n-3}{3}$. The proof is based on the following fact: a bound $f(n)$ for the reset thresholds of synchronizing automata implies a bound $O(f(n))$ for the exponents of $\mathcal{NZ}$ matrix sets [6, Theorem 17]. In this subsection we will strengthen this result.

We denote by $\exp_{\mathcal{NZ}}(n)$ the maximal exponent among all primitive matrix sets belonging to $\mathcal{NZ}$. In order to state a theorem for $\exp_{\mathcal{NZ}}(n)$, analogous to Theorem 2, we will introduce a new class of automata $\mathcal{C}$ defined as follows.

[3]In the discussion of the connections with the Černý conjecture the author refers to a wrong bound; see [20] for a discussion.

An automaton $\mathscr{A}$ with the set of states $[1,n]$ over an alphabet Σ belongs to $\mathcal{C}$ if there exists a partition of Σ into $\Sigma_1, \Sigma_2, \ldots, \Sigma_k$ such that for every $i \in [1,k]$ we have the following:

1. For each state q there exists a state p and a letter $\ell \in \Sigma_i$ such that $p \cdot \ell = q$.
2. Given a list of transformations of states by letters in Σ_i, we can find a letter in Σ_i that performs all the transformations at once, i.e., for every choice of letters $\ell_1, \ell_2, \ldots, \ell_n \in \Sigma_i$ and states $q_1, \ldots, q_n$ such that $j \cdot \ell_j = q_j$ for all $j \in [1,n]$, there exists a letter $\ell \in \Sigma_i$ with the property $j \cdot \ell = q_j$ for all $j \in [1,n]$.

Example 7. The famous Černý automaton $\mathscr{C}_n$ (see [40, p. 18]) equipped with an identity letter c belongs to $\mathcal{C}$. The required partition is $\Sigma_1 = \{a, c\}$ and $\Sigma_2 = \{b\}$. Clearly, the reset threshold is not changed with the addition of the letter c.

THEOREM 8. *Let* $\exp_{\mathcal{NZ}}(n)$ *be the maximum value of the exponent among all primitive sets of* $\mathcal{NZ}$ *matrices of size* $n \times n$ *and* $\mathrm{rt}_{\mathcal{C}}(n)$ *be the largest reset threshold among synchronizing* n*-state automata in* $\mathcal{C}$*. It holds that* $\exp_{\mathcal{NZ}}(n) = \Theta(\mathrm{rt}_{\mathcal{C}}(n))$.

Proof. In order to show that $\exp_{\mathcal{NZ}}(n)$ is $O(\mathrm{rt}_{\mathcal{C}}(n))$ we will reuse the reduction from primitive sets of matrices to partial automata presented in the first part of Theorem 2. Let $\mathcal{M}$ be a primitive set of $\mathcal{NZ}$ matrices. It can be reduced to partial automata $\mathscr{A}$ and $\mathscr{B}$ such that $\exp(\mathcal{M}) \leq \mathrm{car}(\mathscr{A}) + \mathrm{car}(\mathscr{B}) + n - 1$. Since $\mathcal{M}$ is an $\mathcal{NZ}$ matrix set, we conclude that the automata $\mathscr{A}$ and $\mathscr{B}$ are complete. Thus, their carefully synchronizing words are ordinary synchronizing words. Furthermore, the automata $\mathscr{A}$ and $\mathscr{B}$ belong to $\mathcal{C}$. Indeed, every letter of these automata was obtained from a matrix in $\mathcal{M}$ or $\mathcal{M}^T$. In order to obtain the desired partition, we group a pair of letters together if and only if they were derived from the same matrix. It is a straightforward check that both conditions on the partition are satisfied.

For the other direction, given an automaton $\mathscr{A} \in \mathcal{C}$ we will construct a set of $\mathcal{NZ}$ matrices $\mathcal{M}$ such that $\mathrm{rt}(\mathscr{A}) \leq \exp(\mathcal{M})$. It will imply that $\exp_{\mathcal{NZ}}(n) = \Omega(\mathrm{rt}_{\mathcal{C}}(n))$. Let $\Sigma_1, \ldots, \Sigma_k$ be the partition of the letters of $\mathscr{A}$. A set of matrices $\mathcal{M}$ consists of matrices $M_i = \sum_{\ell \in \Sigma_i} A_\ell$, where A_ℓ is the adjacency matrix of the letter ℓ. Clearly, each M_i is an $\mathcal{NZ}$ matrix due to the first property of the partition and the fact that $\mathscr{A}$ is complete. The second property ensures that $\mathrm{rt}(\mathscr{A}) \leq \exp(\mathcal{M})$, since every primitive word of $\mathcal{M}$ can be transformed into a synchronizing word of $\mathscr{A}$ as in the proof of Theorem 2. ∎

PROBLEM 9. *Improve the bounds* $O(n^3)$ *and* $\Omega(n^2)$ *on the growth rate of* $\mathrm{rt}_{\mathcal{C}}(n)$ *and, equivalently,* $\exp_{\mathcal{NZ}}(n)$*. In particular, is there a constant* K *such that* $\mathrm{rt}_{\mathcal{C}}(n) < Kn^2$*?*

This problem can be settled in different ways. On one hand, one can show that Problem 9 is as hard as the Černý conjecture. There are not many natural problems equivalent to it and the problem of bounding $\exp_{\mathcal{NZ}}(n)$ is a good candidate for this purpose. On the other hand, a quadratic bound for $\exp_{\mathcal{NZ}}(n)$ is clearly of interest by itself. We want to emphasize that in the case of a single matrix in a set we have almost complete understanding of the behavior of the exponent; see [7, section 3.5].

3.2. Matrices with total support. In this subsection we will present an upper bound on the exponent of a set of matrices from a special class. A matrix M has *total support* if every nonzero element $m_{i,j}$ of M lies on a positive diagonal, i.e., for every $i, j \in [1,n]$ such that $m_{i,j} > 0$ there exists a permutation σ with the following properties: $\sigma(i) = j$ and for every $k \in [1,n]$ we have $m_{k,\sigma(k)} > 0$. The class of matrices with total support received a lot of attention in the past. For example, it appears in the necessary and sufficient condition for the convergence of the classical Sinkhorn–Knopp method for matrix scaling; see [39]. Another characterization is related to a class of doubly stochastic matrices. A square matrix M is called *doubly stochastic* if the entries are nonnegative and the sum of elements in each row and column is equal to 1. A matrix M is said to have a *doubly stochastic pattern* if there exists a doubly stochastic matrix D such that for all i, j it holds that $D[i, j] > 0$ if and only if $M[i, j] > 0$. A famous result of Perfect and Mirsky [34] states that a matrix M has total support if and only if M has a doubly stochastic pattern; see [8, Theorem 9.2.1] for a modern and much more general treatment of the problem. Now we are ready to state our result.

Theorem 10. *If each matrix of a primitive set $\mathcal{M}$ has total support, then the exponent of $\mathcal{M}$ is at most $2n^2 - 5n + 5$, where $n \times n$ is the size of matrices in $\mathcal{M}$.*

Proof. We will modify the reduction presented in the first part of Theorem 2 from a primitive set of matrices $\mathcal{M}$ to partial automata $\mathscr{A}, \mathscr{B}$ in order to prove the statement. By the aforementioned result of Perfect and Mirsky we conclude that for every matrix $M \in \mathcal{M}$ there exists a doubly stochastic matrix D_1 such that for every i, j we have $M[i, j] > 0$ if and only if $D_1[i, j] > 0$. It is not hard to see that we can further assume that D_1 has only rational entries. Therefore, there exists h and a matrix D_M with nonnegative integer elements such that the sum of entries in each row and column is equal to h, and $M[i, j] > 0$ if and only if $D_M[i, j] > 0$ for all i, j. We will define the automaton $\mathscr{A}$ with the set of states $[1, n]$ as follows. For each matrix $M \in \mathcal{M}$ we add a function $\varphi : [1, n] \to [1, n]$ as a letter of $\mathscr{A}$ multiple times. Namely, we treat $\varphi(\cdot)$ as $\prod_{i=1}^{n} D_M[i, \varphi(i)]$ different letters of the automaton $\mathscr{A}$. Let $\mathscr{B}$ be an automaton obtained from a matrix set $\mathcal{M}^T$ in the same manner. Similarly to the proofs of Theorems 2 and 8 we can conclude that the automata $\mathscr{A}$ and $\mathscr{B}$ are complete and synchronizing, and moreover, $\exp(\mathcal{M}) \le \mathrm{rt}(\mathscr{A}) + \mathrm{rt}(\mathscr{B}) + n - 1$.

Recall that the in-degree of a state i of $\mathscr{A}$ is the total number of edges leading to i, i.e., the number of different letters ℓ such that $i = \delta_{\mathscr{A}}(j, \ell)$ for some $j \in [1, n]$. Now we are going to show that the automaton $\mathscr{A}$ is Eulerian, i.e., the in-degree of each state is equal to its out-degree (which equals to the total number of letters in $\mathscr{A}$). Let Σ_M be the set of letters generated from a matrix $M \in \mathcal{M}$ and h be the row (and column) sum of D_M. It is not hard to see that by the definition of $\mathscr{A}$ the size of Σ_M is h^n. Furthermore, the number of letters from Σ_M that move a state i to a state j is equal to $D_M[i, j]h^{n-1}$. Thus, the number of incoming edges to j labeled by a letter from Σ_M is equal to $\sum_i D_M[i, j]h^{n-1} = h^n$, which is equal to the size of Σ_M. Since the alphabet of $\mathscr{A}$ is $\cup_{M \in \mathcal{M}} \Sigma_M$ and for every Σ_M the number of incoming and outgoing edges labeled by Σ_M coincide, we conclude that the automaton $\mathscr{A}$ is Eulerian. The same reasoning allows us to conclude that the automaton $\mathscr{B}$ is also Eulerian. By the famous result of Kari about the reset thresholds of Eulerian automata [25] we have $\mathrm{rt}(\mathscr{A}), \mathrm{rt}(\mathscr{B}) \le n^2 - 3n + 3$. Thus, the exponent of the matrix set $\mathcal{M}$ is at most $2(n^2 - 3n + 3) + n - 1 = 2n^2 - 5n + 5$. ∎

3.3. Computation and approximation of the exponent. In this subsection we will focus on the problem of computing the exponent of a set of $\mathcal{NZ}$ matrices. Our results rely on the following lemma.

Lemma 11. *For every synchronizing automaton $\mathscr{A}$ with a sink state there exists a set of $\mathcal{NZ}$ matrices $\mathcal{M}$ constructible in polynomial time such that $\exp(\mathcal{M}) = \mathrm{rt}(\mathscr{A}) + 1$.*

Proof. If the number of states of $\mathscr{A}$ is equal to 1, then $\mathcal{M}$ can be an arbitrary set of matrices with the exponent equal to 2. For example, $\mathcal{M} = \{\left(\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right)\}$. In all the other cases we construct the matrix set $\mathcal{M}$ as follows. Since $\mathscr{A}$ is synchronizing, it has a unique sink state, which we denote by s. Let $\mathcal{M}'$ be the set of adjacency matrices of letters of $\mathscr{A}$. The matrix set $\mathcal{M}$ consists of matrices in $\mathcal{M}'$ modified in such a way that the sth row of every matrix is positive, i.e., $\mathcal{M} = \{M' + e_s^T e \mid M' \in \mathcal{M}'\}$. Since the automaton $\mathscr{A}$ is complete, we conclude that every matrix $M \in \mathcal{M}$ has no zero rows. Furthermore, each column of M has a positive element at position s. Thus, the set of matrices $\mathcal{M}$ belongs to $\mathcal{NZ}$. Now we will demonstrate that the set $\mathcal{M}$ is primitive. Since the automaton $\mathscr{A}$ is synchronizing there exists a product P' of matrices from $\mathcal{M}'$ with a positive column. Due to the fact that for every letter ℓ of $\mathscr{A}$ one has $s \cdot \ell = s$, we conclude that the only positive entry in the sth row of P' is located at position s. Thus, the sth column of P' is positive. Altering each matrix M' of the product P' to a matrix $M \in \mathcal{M}$ with the property $M' \leq M$ we obtain a product P of matrices from $\mathcal{M}$ in which column s is positive. Now multiply by any matrix $M \in \mathcal{M}$ to get a positive product PM. Thus, the set $\mathcal{M}$ is primitive and $\exp(\mathcal{M}) \leq \mathrm{rt}(\mathscr{A}) + 1$.

It remains to show that $\exp(\mathcal{M}) \geq \mathrm{rt}(\mathscr{A}) + 1$. Let $P = M_1 M_2 \dots M_m$ be the shortest positive product of matrices in $\mathcal{M}$. Let $P' = M_1' M_2' \dots M_m'$ be the corresponding product of matrices in $\mathcal{M}'$, where M_i and M_i' differ only in the sth row. For each row $i \neq s$ there exists the largest h_i such that the ith row of $P_{h_i} = M_1 M_2 \dots M_{h_i}$ contains a unique positive entry. Due to the fact that the ith row of P_{h_i+1} has several positive entries and the structure of matrices in $\mathcal{M}$ we conclude that the unique positive entry occupies the position s in the ith row of P_{h_i}. Therefore, the only positive entry in the ith row of P'_{h_i} is located at position s. Since for every row the value of h_i is strictly less than m, we conclude that the product P'_{m-1} has a positive column s. It implies that $\mathrm{rt}(\mathscr{A}) \leq m - 1$. Rearranging, we obtain the desired inequality $\exp(\mathcal{M}) \geq \mathrm{rt}(\mathscr{A}) + 1$. ∎

The computational complexities of problems related to the computation of reset thresholds and synchronizing words were extensively studied. Now we will leverage Lemma 11 to easily obtain a large body of results on the computation and approximation of the exponents of $\mathcal{NZ}$ matrix sets. Recall that DP stands for the class of all languages of the form $L = L_1 \setminus L_2$ with $L_1, L_2 \in NP$. Note that DP is a superclass of both NP and $coNP$. Thus, the hardness result that we are going to obtain applies for both classes. The class DP is contained in $P^{NP[log]}$—the class of problems solvable by a deterministic polynomial-time Turing machine that can use a logarithmic number of queries to an oracle for an NP-complete problem. It is generally believed that the inclusion is proper. We will denote by $FP^{NP[log]}$ the functional analogue of $P^{NP[log]}$. For a function $f(n) : \mathbb{N} \to \mathbb{R}$ we say that an algorithm approximates the exponent within a factor $f(n)$ if for every set $\mathcal{M}$ of matrices of size $n \times n$ the value V returned by the algorithm satisfies $\exp(\mathcal{M}) \leq V \leq f(n) \exp(\mathcal{M})$.

THEOREM 12. *Given a set $\mathcal{M}$ of three $n \times n$ matrices belonging to $\mathcal{NZ}$ and possibly a positive integer k encoded in binary the following hold:*
1. *The problem of deciding whether* $\exp(\mathcal{M}) \leq k$ *is NP-complete.*
2. *The problem of deciding whether* $\exp(\mathcal{M}) = k$ *is DP-complete.*
3. *The problem of computing* $\exp(\mathcal{M})$ *is $FP^{NP[\log]}$-complete.*
4. *For every constant $\varepsilon > 0$ it is NP-hard to approximate* $\exp(\mathcal{M})$ *within a factor $n^{1-\varepsilon}$.*

Proof. The hardness results for these problems follow from the corresponding statements about the reset thresholds of synchronizing automata with a sink state and Lemma 11. Eppstein [12] proved that it is NP-hard to decide whether $\mathrm{rt}(\mathscr{A}) \leq k$, where k is given in binary. Olschewski and Ummels [32, Theorem 1] showed that it is DP-hard to decide whether $\mathrm{rt}(\mathscr{A}) = k$. The same authors [32, Theorem 4] also proved that the problem of computing $\mathrm{rt}(\mathscr{A})$ is $FP^{NP[\log]}$-hard. A recent breakthrough by Gawrychowski and Straszak [16, Theorem 16] states that for every constant $\varepsilon > 0$ it is NP-hard to approximate $\mathrm{rt}(\mathscr{A})$ within a factor $n^{1-\varepsilon}$. Moreover, only automata with a sink state were utilized in all reductions presented by the aforementioned authors. In the proof presented by Gawrychowski and Straszak [16, Theorem 16] the number of letters $\mathscr{A}$ is equal to three, while in all the other reductions automata with only two letters were utilized.

It remains to show that the stated problems belong to the corresponding classes:

1. Since $\exp(\mathcal{M})$ is bounded by a low-degree polynomial $p(n)$, e.g., $p(n) = \frac{n^3+2n-3}{3}$ by [6, Corollary 18], it suffices to guess a positive product of matrices from $\mathcal{M}$ of length $\min\{p(n), k\}$. Verification for every such product can be done in polynomial time. Thus, the problem belongs to NP.
2. It is easy to see that $\exp \mathcal{M} = k$ if and only if $\exp \mathcal{M} \leq k$ and $\exp \mathcal{M} \leq k-1$ does not hold. Since the problem of deciding whether $\exp(\mathcal{M}) \leq k$ belongs to NP, we conclude that the problem of deciding whether $\exp(\mathcal{M}) = k$ belongs to DP.
3. The algorithm from $FP^{NP[\log]}$ that computes $\exp(\mathcal{M})$ is a simple binary search algorithm that uses logarithmic number of queries to the oracle for the NP-complete problem $\exp(\mathcal{M}) \leq k$. Recall that the exponent of $\mathcal{M}$ is bounded by a polynomial $p(n) = \frac{n^3+2n-3}{3}$. Thus, in $\log p(n)$ iterations we can establish a precise value of $\exp(\mathcal{M})$. ∎

We will conclude this subsection with an approximation algorithm for the computation of the exponent whose performance guarantee is close to the bound given by Theorem 12.

THEOREM 13. *For every positive integer c there is a polynomial time algorithm that approximates the exponent of a given set $\mathcal{M}$ of $n \times n$ matrices belonging to $\mathbb{NZ}$ within a factor $\frac{n}{c}$.*

Proof. Let us fix a positive integer c. Let $\mathscr{A}, \mathscr{B}$ be deterministic synchronizing n-state automata such that $\exp(\mathcal{M}) \leq \mathrm{rt}(\mathscr{A}) + \mathrm{rt}(\mathscr{B}) + n - 1$ and $\mathrm{rt}(\mathscr{A}), \mathrm{rt}(\mathscr{B}) \leq \exp(\mathcal{M})$. Their existence follows from the proof of Theorem 8. We will utilize the algorithms presented in [18] to approximate the reset thresholds of $\mathscr{A}$ and $\mathscr{B}$ in order to approximate $\exp(\mathcal{M})$. To be precise, the authors of [18, Theorem 1] have shown that for any constant c the reset threshold of a given n-state automaton can be approximated in polynomial time within a factor $\frac{n}{c}$. Keeping this in mind, we describe Algorithm 1.

Let us first show that the proposed algorithm approximates the exponent within a factor $\frac{3n}{c}$. Since c is an arbitrary constant, the statement of the theorem will follow. If $\exp(\mathcal{M}) \leq c$, then we will output the exact value of the exponent at step 4. Assume now that $\exp(\mathcal{M}) > c$. Since $\mathrm{rt}(\mathscr{A}), \mathrm{rt}(\mathscr{B}) \leq \exp(\mathcal{M})$ we have the following inequality:

$$\ell_1 + \ell_2 + n - 1 \leq \mathrm{rt}(\mathscr{A})\frac{n}{c} + \mathrm{rt}(\mathscr{B})\frac{n}{c} + n\frac{\exp(\mathcal{M})}{c} \leq \frac{3n}{c}\exp(\mathcal{M}).$$

The lower bound follows from the definition of $\mathscr{A}, \mathscr{B}$:

$$\exp(\mathcal{M}) \leq \mathrm{rt}(\mathscr{A}) + \mathrm{rt}(\mathscr{B}) + n - 1 \leq \ell_1 + \ell_2 + n - 1.$$

Algorithm 1 Approximation of $\exp(\mathcal{M})$

1: **for** $\ell = 1$ **to** c **do**
2:   **for all** products P of length ℓ of matrices from $\mathcal{M}$ **do**
3:     **if** P is entrywise positive **then**
4:       **return** the length of P
5:     **end if**
6:   **end for**
7: **end for**
8: **Compute** an approximation ℓ_1 of $\mathrm{rt}(\mathscr{A})$ within a factor $\frac{n}{c}$
9: **Compute** an approximation ℓ_2 of $\mathrm{rt}(\mathscr{B})$ within a factor $\frac{n}{c}$
10: **return** $\ell_1 + \ell_2 + n - 1$

It remains to show that the algorithm can be implemented to run in polynomial time. Steps 1–7 involve computation of at most $(k+1)^c$ matrix products, where k is the cardinality of $\mathcal{M}$. It can be done in polynomial time.

The major obstacle for the remaining steps is that automata $\mathscr{A}$ and $\mathscr{B}$ can have exponentially many letters. Thus, the naive approach does not lead to a polynomial time algorithm. Our idea is to avoid computation of these automata and immediately obtain the approximate value of the reset threshold using the auxiliary automaton (or table) utilized in [18, Lemma 1]; see Subroutine 2.

Claim: Steps 8 and 9 can be implemented to run in polynomial time. Let us consider the case of the automaton $\mathscr{A}$ with the set of states $[1, n]$. The case of the automaton $\mathscr{B}$ can be done in the same manner. To proceed, we need to define an auxiliary automaton $\mathcal{G}_c$ and a special operation $*$.

Construction of $\mathcal{G}_c$. Recall that every letter of $\mathscr{A}$ is derived from a matrix of $\mathcal{M}$. We define an auxiliary automaton $\mathcal{G}_c$ as follows: the set of states is equal to all possible nonempty subsets of $[1, n]$ of cardinality at most c; each matrix from $\mathcal{M}$ is treated as a letter of $\mathcal{G}_c$; and there is an edge from a subset S_1 to a subset S_2 labeled by $M_i \in \mathcal{M}$ if and only if $S_1 \cdot x = S_2$ for some letter x associated with M_i of the automaton $\mathscr{A}$, i.e., for all $s \in S_1$ there exists $t \in S_2$ such that $M_i[s, t] > 0$.

We will show now that $\mathcal{G}_c$ can be constructed in polynomial time. Note that for a given pair of subsets S_1, S_2 and a matrix $M_i \in \mathcal{M}$ we can check in polynomial time whether a letter associated with M_i brings S_1 to S_2. Indeed, the number of possible actions from S_1 to S_2 is at most c^c. Thus, we can iterate through all of them and check whether they can be obtained from the matrix M_i. Therefore, by successively iterating through all matrices of $\mathcal{M}$ we can establish in polynomial time whether there is an edge from S_1 to S_2. Since the number of vertices of $\mathcal{G}_c$ is bounded by $(n+1)^c - 1$, we conclude that $\mathcal{G}_c$ can be indeed constructed in polynomial time.

The results of [18] imply that given $\mathcal{G}_c$ one can approximate $\mathrm{rt}(\mathscr{A})$ in polynomial time within a factor $\lceil \frac{n-1}{c-1} \rceil$. For completeness, we briefly describe the main ideas below; see Subroutine 2.

The * *operation.* Let u be a path in $\mathcal{G}_c$ from a state S'_0 to S'_ℓ labeled by $u_1 u_2 \dots u_\ell$, and $S'_{i+1} = S'_i \cdot u_{i+1}$. By construction of $\mathcal{G}_c$ it implies that there exists a word x in $\mathscr{A}$ that brings the subset $S'_0 \subseteq [1, n]$ to S'_ℓ. Simply speaking, we define $S * u$ to be the image of S under the action of x in the automaton $\mathscr{A}$ for every $S \subseteq [1, n]$. To be more precise, let $S_0 = S$ and assume that $S_i = S_0 * u_1 \dots u_i$ is already defined. By the construction of $\mathcal{G}_c$ we conclude that there exists a letter x_{i+1} associated with u_{i+1} such that $S'_{i+1} = S'_i \cdot x_{i+1}$ in $\mathscr{A}$. (If there are many such letters, we fix one of them.) By definition we put $S_{i+1} * u_{i+1} := S_{i+1} \cdot x_{i+1}$. Note that $S_\ell = S * u = S \cdot x_1 \dots x_\ell$

in $\mathscr{A}$ and $S'_\ell = S'_0 * u = S'_0 \cdot x_1 \ldots x_\ell$. In other words, $S * u$ is the image of S under the action of some word that brings S'_0 to S'_ℓ in $\mathscr{A}$.

Subroutine 2 Approximation of $\mathrm{rt}(\mathscr{A})$

1: $S := [1, n], w = \varepsilon$
2: **while** $|S| > 1$ **do**
3: Choose a subset $S' \subseteq S$ of size $\alpha = \min\{c, |S|\}$
4: Compute a shortest path labeled by u from S' to a singleton state in $\mathcal{G}_c$
5: $S := S * u$
6: $w := wu$
7: **end while**
8: **return** $|w|$

Observe that at the end of each iteration 2–8 of Subroutine 2 we have $[1, n]*w = S$. Furthermore, the cardinality of S is strictly decreasing at every iteration until it becomes 1. Therefore, it is not difficult to see that there exists a synchronizing word x for $\mathscr{A}$ of length $|w|$ such that for any i the ith letter of x is associated with the ith letter of w. Moreover, $|x| = |w| \leq \mathrm{rt}(\mathscr{A})\lceil \frac{n-1}{c-1} \rceil$, since at every iteration 2–8 of Subroutine 2 the cardinality of S is decreased by $c - 1$ by a word u of length at most $\mathrm{rt}(\mathscr{A})$; see [18] for a detailed proof. □

4. Conclusion. The goal of our work was to emphasize and leverage the fact that several problems about primitive sets of matrices are in some sense equivalent to problems about synchronizing automata. More precisely, we related the bounds on the exponents and the lengths of the shortest carefully synchronizing words in the general case, and the exponents and reset thresholds in the case of matrices without zero rows and columns. Furthermore, we utilized these connections to easily establish the exact complexity classes of different problems concerning the computation of the exponent of a set of matrices belonging to $\mathcal{NZ}$. Thus, we believe that the joint research effort on both topics at the same time can lead to substantial progress on some of the most difficult problems in both fields at the same time. We left a quadratic upper bound on the exponent of an $\mathcal{NZ}$ matrix set as an open problem, and whether its existence brings any implication for the Černý conjecture. Our future work includes the search for matrix counterparts of special classes of automata, which have quadratically bounded reset threshold. These statements will be analogous to our result on primitive sets of matrices having total support. We plan to apply our ideas to the closely related concept of scrambling matrix and its generalizations to sets of matrices as well.

Acknowledgments. We are sincerely grateful to Pavel Panteleev for bringing the paper [37] to our attention and to the reviewers for valuable comments and suggestions.

REFERENCES

[1] M. Akelbek and S. Kirkland, *Coefficients of ergodicity and the scrambling index*, Linear Algebra Appl., 430 (2009), pp. 1111–1130.

[2] Y. A. Al'pin and V. S. Al'pina, *Combinatorial properties of irreducible semigroups of nonnegative matrices*, J. Math. Sci., 191 (2013), pp. 4–9.

[3] D. S. Ananichev, M. V. Volkov, and V. V. Gusev, *Primitive digraphs with large exponents and slowly synchronizing automata*, J. Math. Sci. 192 (2013), pp. 263–278.

[4] J. Araújo, P. J. Cameron, and B. Steinberg, *Between Primitive and* 2*-transitive: Synchronization and Its Friends*, http://arxiv.org/abs/1511.03184, 2015.

[5] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*, Cambridge University Press, Cambridge, 2009.
[6] V. D. Blondel, R. M. Jungers, and A. Olshevsky, *On primitivity of sets of matrices*, Automatica, 61 (2015), pp. 80–88.
[7] R. Brualdi and H. Ryser, *Combinatorial Matrix Theory*, Cambridge University Press, Cambridge, 1991.
[8] R. A. Brualdi, *Combinatorial Matrix Classes*, Cambridge University Press, Cambridge, 2006.
[9] J. Černý, *Poznámka k homogénnym experimentom s konečnými automatmi*, Matematicko-fyzikálny Časopis Slovenskej Akadémie Vied, 14 (1964), pp. 208–216 (in Slovak).
[10] J. Černý, A. Pirická, and B. Rosenauerová, *On directable automata*, Kybernetica, 7 (1971), pp. 289–298.
[11] J. E. Cohen and P. H. Sellers, *Sets of nonnegative matrices with positive inhomogeneous products*, Linear Algebra Appl., 47 (1982), pp. 185–192.
[12] D. Eppstein, *Reset sequences for monotonic automata*, SIAM J. Comput., 19 (1990), pp. 500–510.
[13] W. Fang, Y. Gao, Y. Shao, W. Gao, G. Jing, and Z. Li, *The generalized competition indices of primitive minimally strong digraphs*, Linear Algebra Appl., 493 (2016), pp. 206–226.
[14] E. Fornasini and M. E. Valcher, *Directed graphs, 2D state models, and characteristic polynomials of irreducible matrix pairs*, Linear Algebra Appl., 263 (1997), pp. 275–310.
[15] P. Frankl, *An extremal problem for two families of sets*, European J. Combin., 3 (1982), pp. 125–127.
[16] P. Gawrychowski and D. Straszak, *Strong inapproximability of the shortest reset word*, in Mathematical Foundations of Computer Science, Lecture Notes in Compt. Sci. 9234, Springer, New York, 2015, pp. 243–255.
[17] Z. Gazdag, S. Iván, and J. Nagy-György, *Improved upper bounds on synchronizing nondeterministic automata*, Inform. Process. Lett., 109 (2009), pp. 986–990.
[18] M. Gerbush and B. Heeringa, *Approximating Minimum Reset Sequences*, in Implementation and Application of Automata, Lecture Notes in Comput. Sci. 6482, Springer, New York, 2011, pp. 154–162.
[19] E. Goles and M. Noual, *Disjunctive networks and update schedules*, Adv. Appl. Math., 48 (2012), pp. 646–662.
[20] F. Gonze, R. Jungers, and A. Trahtman, *A note on a recent attempt to improve the Pin–Frankl bound*, Discrete Math. Theor. Comput. Sci., 17 (2015), pp. 307–308.
[21] D. J. Hartfiel, *Nonhomogeneous Matrix Products*, World Scientific, River Edge, NJ, 2002.
[22] J. Hopcroft, R. Motwani, and J. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Pearson Education, London, 2001.
[23] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, 1995.
[24] B. Imreh and M. Steinby, *Directable nondeterministic automata*, Acta Cybernet., 14 (1999), pp. 105–115.
[25] J. Kari, *Synchronizing finite automata on Eulerian digraphs*, Theoret. Comput. Sci., 295 (2003), pp. 223–232.
[26] A. A. Klyachko, I. K. Rystsov, and M. A. Spivak, *An extremal combinatorial problem associated with the bound on the length of a synchronizing word in an automaton*, Cybernetics, 23 (1987), pp. 165–171.
[27] D. O. Logofet, *Markov chains as succession models: New perspectives of the classic paradigm*, Lesovedenie, 2 (2010), pp. 46–59.
[28] P. Martyugin, *Computational complexity of certain problems related to carefully synchronizing words for partial automata and directing words for nondeterministic automata*, Theory Comput. Syst., 54 (2014), pp. 293–304.
[29] P. V. Martyugin, *A lower bound for the length of the shortest carefully synchronizing words*, Russian Math., 54 (2010), pp. 46–54.
[30] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, SIAM, Philadelphia, 2000.
[31] B. K. Natarajan, *An algorithmic approach to the automated design of parts orienters*, in Proceedings of the Symposium on Foundations of Computer Science, 1986, pp. 132–142.
[32] J. Olschewski and M. Ummels, *The Complexity of Finding Reset Words in Finite Automata*, in Mathematical Foundations of Computer Science, Lecture Notes in Comput. Sci. 6281, Springer, New York, 2010, pp. 568–579.
[33] A. Paz, *Definite and quasidefinite sets of stochastic matrices*, Proc. Amer. Math. Soc., 16 (1965), pp. 634–641.
[34] H. Perfect and L. Mirsky, *The distribution of positive elements in doubly-stochastic matrices*, J. Lond. Math. Soc., s1-40 (1965), pp. 689–698.

[35] J.-E. Pin, *On Two Combinatorial Problems Arising from Automata Theory*, in Graph Theory and Combinatorics, North-Holland Math. Stud. 75, 1983, pp. 535–548.
[36] V. Y. Protasov and A. S. Voynov, *Sets of nonnegative matrices without positive products*, Linear Algebra Appl., 437 (2012), pp. 749–765.
[37] I. K. Rystsov, *Asymptotic estimate of the length of a diagnostic word for a finite automaton*, Cybernetics, 16 (1980), pp. 194–198.
[38] W. J. Savitch, *Relationships between nondeterministic and deterministic tape complexities*, J. Comput. System Sci., 4 (1970), pp. 177–192.
[39] R. Sinkhorn and P. Knopp, *Concerning nonnegative matrices and doubly stochastic matrices*, Pacific J. Math., 21 (1967), pp. 343–348.
[40] M. V. Volkov, *Synchronizing Automata and the Černý Conjecture*, in Language and Automata Theory and Applications, Lecture Notes in Comput. Sci. 5196, Springer, New York, 2008, pp. 11–27.
[41] A. S. Voynov, *Shortest positive products of nonnegative matrices*, Linear Algebra Appl., 439 (2013), pp. 1627–1634.
[42] A. S. Voynov and V. Y. Protasov, *Compact noncontraction semigroups of affine operators*, Sb. Math., 206 (2015), pp. 921–940.