ELSEVIER

Theoretical Computer Science 295 (2003) 205–222

Theoretical Computer Science

www.elsevier.com/locate/tcs

The complexity of the characteristic and the minimal polynomial ☆

Thanh Minh Hoang, Thomas Thierauf*

Abt. Theoretische Informatik, Universität Ulm, 89069 Ulm, Germany

Received 21 August 2001; received in revised form 14 November 2001; accepted 20 December 2001

Abstract

We investigate the complexity of (1) *computing* the characteristic polynomial, the minimal polynomial, and all the invariant factors of an integer matrix, and of (2) *verifying* them, when the coefficients are given as input.

It is known that each coefficient of the characteristic polynomial of a matrix A is computable in **GapL**, and the constant term, the determinant of A, is complete for **GapL**. We show that the verification of the characteristic polynomial is complete for complexity class $\mathbf{C_=L}$ (*exact counting logspace*).

We show that each coefficient of the minimal polynomial of a matrix A can be computed in $\mathbf{AC^0(GapL)}$, the $\mathbf{AC^0}$-closure of **GapL**, and there is a coefficient which is hard for **GapL**. Furthermore, the verification of the minimal polynomial is in $\mathbf{AC^0(C_=L)}$ and is hard for $\mathbf{C_=L}$. The hardness result extends to (computing and verifying) the system of all invariant factors of a matrix.

Keywords: Computational complexity; Logspace counting classes; Linear algebra; Characteristic polynomial; Minimal polynomial; Invariant factors

1. Introduction

The motivation for our work is twofold: (1) we want to understand the computational complexity of some classical problems in linear algebra, (2) by locating such problems

☆ Supported by the Deutsche Forschungsgemeinschaft.

* Corresponding author.

E-mail addresses: hoang@informatik.uni-ulm.de (T.M. Hoang), thierauf@informatik.uni-ulm.de (T. Thierauf).

0304-3975/03/$ - see front matter
PII: S0304-3975(02)00404-8

in small space complexity classes we want to clarify the inclusion relationship of such classes.

The *characteristic polynomial* and the *minimal polynomial* of a matrix play an important role in matrix theory. In our work we want to study the computational complexities of these problems.

Valiant [21,22] initiated the study of the computational complexity of counting problems. He introduced the counting class #**P** that, intuitively, counts the number of solutions of **NP**-problems. An example for a complete problem for this class is the permanent of a matrix.

Since counting is restricted to nonnegative integers, Fenner, Fortnow, and Kurtz [8] extended #**P** to the class **GapP**, the closure of #**P** under subtraction. It follows that computing the permanent of integer matrices is **GapP**-complete.

In contrast, the determinant of a matrix is complete for **GapL** [7,20,23,25], the class corresponding to **GapP** in the logspace setting. This huge difference in the complexity of the two problems [1] is somewhat surprising since the permanent and the determinant have almost the same cofactor expansion; the only difference comes with the sign. **GapL** turns out to capture the complexity of many other natural problems: computing

- the powers of a matrix,
- iterated matrix multiplication,
- the inverse of a matrix,
- the characteristic polynomial of a matrix.

There are also graph theoretic problems related to counting the number s-t-paths in a graph.

Interesting decision problems can be derived from the above problems. For example, instead of computing the inverse of a matrix, it often suffices to decide whether the inverse *exists*. That is, to decide whether the determinant is zero or not. More generally, this motivates the complexity class $\mathbf{C}_{=}\mathbf{L}$ where one has to *verify* the value of a **GapL** function. Problems that are complete for **GapL** yield verification problems that are complete for $\mathbf{C}_{=}\mathbf{L}$. For example, the determinant is **GapL** complete and checking singularity is complete for $\mathbf{C}_{=}\mathbf{L}$. In case the result is a matrix or a tuple of numbers there is a subtlety one has to be careful about: for example when we say that matrix powering is in **GapL**, what we mean is that each entry of the resulting matrix can be computed within **GapL**. I.e., for a $n \times n$ matrix A this yields n^2 **GapL**-functions, one for each entry of A^m, and each of which is complete for **GapL**. Now there are two variants of the verification version: in the first version we have to verify *one* entry, say $(A^m)_{i,j}$ for given i and j. In the second version, we have to verify *all* the entries, i.e., A^m. Both versions are complete for $\mathbf{C}_{=}\mathbf{L}$.

But the situation can be different. An example is provided by the inverse of a matrix (if it exists). Again we have two variants of the verification problem.

- Verify *one entry* of the inverse: given matrix A, a, i and j, decide whether $(A^{-1})_{i,j} = a$.

[1] Note however that there is no proof yet that $\mathbf{GapL} \neq \mathbf{GapP}$.

This problem is complete for $\mathbf{C}_=\mathbf{L}$. The second variant is as follows.

- Verify the inverse of a matrix: given matrices A and B, check whether $A^{-1}=B$.

This problem can be solved by computing the product AB and comparing it with the identity matrix. Hence this can be solved in $\mathbf{NC}^1$, a subclass of $\mathbf{C}_=\mathbf{L}$. In other words, verifying *one* entry of the inverse is a harder problem than verifying *all* elements.[2] In the latter problem, we put too much information in the input.

We consider the following problem.

- Verify the characteristic polynomial of a matrix: given a matrix A and the coefficients of a polynomial p, check whether $\chi_A = p$.

It follows from a theorem of Berkowitz [3] that this problem is in $\mathbf{C}_=\mathbf{L}$, and Santha and Tan [17] asked whether it is complete for this class.

Recall that the determinant is the constant term in the characteristic polynomial of a matrix and that verifying the determinant is complete for $\mathbf{C}_=\mathbf{L}$. Now, with the different complexities of the above two inverse problems in mind, the question is: is it easier to verify *all* the coefficients of the characteristic polynomial than to verify just *one* of them? We show that this is *not* the case: verifying the characteristic polynomial is still complete for $\mathbf{C}_=\mathbf{L}$.

The minimal polynomial of a matrix is one of the factors of the characteristic polynomial of the matrix. Algorithms to compute the minimal polynomial have been studied for a long time. The best known deterministic algorithm to compute the minimal polynomial of an $n\times n$ matrix makes $O(n^3)$ field operations [18]. The Smith normal form of a polynomial matrix can be computed by a randomized $\mathbf{NC}^2$-circuit, i.e., in $\mathbf{RNC}^2$ [12]. Therefore the rational canonical form of a matrix and the minimal polynomial of a matrix can be computed in $\mathbf{RNC}^2$ as well. In the case of integer matrices there are even $\mathbf{NC}^2$-algorithms [24].

We take a different approach to compute the minimal polynomial of an integer matrix: we show that the problem can be reduced to matrix powering and solving a system of linear equations. Therefore it is in the class $\mathbf{AC}^0(\mathbf{GapL})$, a subclass of $\mathbf{NC}^2$. With respect to the hardness of the problem we show that matrix powering can be reduced to the minimal polynomial of a matrix. Therefore the latter problem is hard for $\mathbf{GapL}$. With respect to the verification of the minimal polynomial, we have a similar situation as for the characteristic polynomial: verifying whether the constant term c_0 of the minimal polynomial of a matrix A is zero is complete for $\mathbf{C}_=\mathbf{L}$, because $c_0=0$ iff A is singular. We show that verifying *all* the coefficients is still hard for $\mathbf{C}_=\mathbf{L}$.

The system of all *invariant factors* of a matrix A completely determines the structure of A, i.e., these factors are invariant under similarity transformations. Note that the minimal polynomial of A is the first polynomial in its system of all invariant factors. For integer matrices, the invariant factors can be computed in $\mathbf{NC}^2$ [24]. We extend our results and techniques to the *verification* of all the invariant factors of a given integer matrix: it is in $\mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$ (the $\mathbf{AC}^0$-closure of $\mathbf{C}_=\mathbf{L}$) and is hard for $\mathbf{C}_=\mathbf{L}$.

[2] Note however that we do not know whether $\mathbf{NC}^1 \neq \mathbf{C}_=\mathbf{L}$.

One goal of our research is to determine the complexity of algebraic problems as described above, i.e., in the ideal case, to show them complete for some complexity class. Another goal we have in mind is to clarify the relationship of these complexity classes. The may be most challenging open problem here is whether $\mathbf{C}_=\mathbf{L}$ is closed under complement. Many related classes have this property:

- The most popular one is nondeterministic logspace, **NL**, shown by Immerman [11] and Szelepcsényi [19].
- For symmetric logspace, **SL**, this was shown by Nisan and Ta-Shma [15].

Also, for probabilistic logspace, **PL**, it is trivial. For unambiguous logspace, **UL**, it is open as well. For the latter class, however, Reinhardt and Allender [16] showed that the *nonuniform* version of it, $\mathbf{UL}/poly$, is closed under complement. This motivates the conjecture that **UL** might be closed under complement too.

One possible way of proving $\mathbf{C}_=\mathbf{L}$ to be closed under complement is to reduce the singularity problem to the nonsingularity problem. That is, given a matrix A, construct a matrix B (in logspace) such that A is singular if and only if B is nonsingular. It is well known that one does not need to consider an *arbitrary* matrix A: one can assume that A is an upper triangular matrix except for the entry in lower left corner (see [2]). To prove our hardness result for the characteristic polynomial, the minimal polynomial, and the invariant factors we manipulate such matrices. We think that it is quite interesting to see such transformations, because this can give some hints on how to come up with a reduction as above to solve the complementation problem for $\mathbf{C}_=\mathbf{L}$. Therefore the methods we use are interesting in their own right. For more background and interesting results we recommend the paper of Allender, Beals, and Ogihara [2].

The paper is organized as follows. After some definitions in the next section, we present all the upper bounds, i.e., inclusions in complexity classes, of the above-mentioned problems in Section 3. Our main results are the lower bounds, i.e., the hardness results, in Section 4. The reason for this organization is that we obtain the hardness results via a reduction that is successively extended from one problem to the next one. That way, this line of arguments is not interrupted.

2. Preliminaries

Complexity Classes: For a nondeterministic logspace bounded Turing machine M, we denote the number of accepting paths on input x by $acc_M(x)$, and by $rej_M(x)$ the number of rejecting paths. The difference of these two numbers is $gap_M(x) = acc_M(x) - rej_M(x)$.

For the counting classes, we have $\#\mathbf{L}$, the class of functions $acc_M(x)$ for some nondeterministic logspace bounded Turing machine M, and **GapL** based analogously on functions $gap_M(x)$. Based on counting, we consider the class $\mathbf{C}_=\mathbf{L}$: a set L is in $\mathbf{C}_=\mathbf{L}$, if there exists a $f \in \mathbf{GapL}$ such that for all x:

$$x \in L \quad \Leftrightarrow \quad f(x) = 0.$$

Since it is open whether $\mathbf{C}_=\mathbf{L}$ is closed under complement, it makes sense to consider the *Boolean closure of* $\mathbf{C}_=\mathbf{L}$. i.e., the class of sets that can be expressed as a Boolean combination of sets in $\mathbf{C}_=\mathbf{L}$. For our purposes, it suffices to consider the following two classes:

- $\mathbf{co}\,\mathbf{C}_=\mathbf{L}$ is the class of complement sets $\bar{L}$, where $L \in \mathbf{C}_=\mathbf{L}$,
- $\mathbf{C}_=\mathbf{L} \wedge \mathbf{co}\,\mathbf{C}_=\mathbf{L}$ [2] is defined as the class of intersections of sets in $\mathbf{C}_=\mathbf{L}$ with sets in $\mathbf{co}\,\mathbf{C}_=\mathbf{L}$. Formally,

$$L \in \mathbf{C}_=\mathbf{L} \wedge \mathbf{co}\,\mathbf{C}_=\mathbf{L} \;\Leftrightarrow\; \exists L_1 \in \mathbf{C}_=\mathbf{L},\ L_2 \in \mathbf{co}\,\mathbf{C}_=\mathbf{L}:\ L = L_1 \cap L_2.$$

For sets A and B, A is $\mathbf{AC}^0$-*reducible to* B, if there is a logspace uniform circuit family of polynomial size and constant depth that computes A with unbounded fan-in and-, or-gates, not-gates, and oracle-gates for B. In particular, we consider the classes $\mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$ and $\mathbf{AC}^0(\mathbf{GapL})$ of sets that are $\mathbf{AC}^0$-reducible to a set in $\mathbf{C}_=\mathbf{L}$, respectively a function in $\mathbf{GapL}$. Cook [5] defined $\mathbf{DET}$ as the class of functions that are $\mathbf{NC}^1$-reducible to the determinant, i.e., the class $\mathbf{NC}^1(\mathbf{GapL})$ (see [5] for a precise definition). The known inclusion relations of these classes is as follows:

$$\mathbf{NL} \subseteq \mathbf{C}_=\mathbf{L} \subseteq \mathbf{C}_=\mathbf{L} \wedge \mathbf{co}\,\mathbf{C}_=\mathbf{L} \subseteq \mathbf{AC}^0(\mathbf{C}_=\mathbf{L}) \subseteq \mathbf{PL} \subseteq$$

$$\mathbf{AC}^0(\mathbf{GapL}) \subseteq \mathbf{DET} \subseteq \mathbf{TC}^1 \subseteq \mathbf{NC}^2.$$

A set A is $\mathbf{AC}^0$ many-one reducible to a set B, in symbols: $A \leqslant_m^{AC^0} B$, if there is a function $f \in \mathbf{AC}^0$ such that for all x we have $x \in A \Leftrightarrow f(x) \in B$. All reductions used in this paper are $\mathbf{AC}^0$ many-one reductions.

Linear Algebra: Let $A \in \mathscr{F}^{n\times n}$ be a matrix over the field $\mathscr{F}$. The *characteristic polynomial* of A is the polynomial $\chi_A(x) = det(xI - A)$. A nonzero polynomial $p(x)$ over $\mathscr{F}$ is called an *annihilating polynomial* of A if $p(A) = \mathbf{0}$. The Cayley–Hamilton Theorem states that $\chi_A(x)$ is an annihilating polynomial. The characteristic polynomial is a *monic polynomial*: its highest coefficient is one. The *minimal polynomial* of A, denoted $\mu_A(x)$, is the unique monic annihilating polynomial of A with minimal degree.

Let polynomial $d_k(x)$ be the greatest common divisor of all sub-determinants of $(xI - A)$ of order k. For example $d_n(x) = \chi_A(x)$. It is known that d_k divides d_{k+1} for each index $0 \leqslant k < n$. Define $d_0(x) \equiv 1$. The *invariant factors* of $(xI - A)$ (or A, for short) are defined as the following (monic) polynomials:

$$i_1(x) = \frac{d_n(x)}{d_{n-1}(x)}, \quad i_2(x) = \frac{d_{n-1}(x)}{d_{n-2}(x)}, \ldots, i_n(x) = \frac{d_1(x)}{d_0(x)}.$$

The characteristic polynomial of A is the product of all the invariant factors, that is $\chi_A(x) = i_1(x) \cdots i_n(x)$. Note that the minimal polynomial of A is the first invariant factor, i.e., $\mu_A(x) = i_1(x)$. The $n \times n$ polynomial diagonal matrix that has the invariant factors of A as its diagonal entries (starting with $i_n(x)$) and zero elsewhere is the *Smith normal form* of $xI - A$.

We decompose the invariant factors into irreducible divisors over the given number field $\mathscr{F}$:

$$\begin{aligned} i_1(x) &= [e_1(x)]^{j_{1,1}}[e_2(x)]^{j_{1,2}}\cdots[e_s(x)]^{j_{1,s}},\\ i_2(x) &= [e_1(x)]^{j_{2,1}}[e_2(x)]^{j_{2,2}}\cdots[e_s(x)]^{j_{2,s}},\\ &\vdots\\ i_n(x) &= [e_1(x)]^{j_{n,1}}[e_2(x)]^{j_{n,2}}\cdots[e_s(x)]^{j_{n,s}}, \end{aligned}$$

where $j_{1,k} \geqslant j_{2,k} \geqslant \cdots \geqslant j_{n,k} \geqslant 0$ for $k=1,\ldots,s$. The irreducible divisors $e_1(x), e_2(x), \ldots, e_s(x)$ are distinct (with highest coefficient 1) and occur in $i_1(x), i_2(x), \ldots, i_n(x)$. All powers $[e_1(x)]^{j_{1,1}}, \ldots, [e_s(x)]^{j_{n,s}}$, which are different from 1, are called the *elementary divisors* of A in $\mathscr{F}$.

Note that the coefficients of the characteristic polynomial and the invariant factors of an integer matrix are all integers. Furthermore, the set of eigenvalues of A is the same as the set of all roots of $\chi_A(x)$ which, in turn, is the set of all roots of $\mu_A(x)$.

Problems: Next, we define some natural problems in linear algebra we are looking at. If nothing else is said, our domain for the algebraic problems are the integers.

(1) PowerElement
Input: an $n\times n$ matrix A and i, j, and m, $(1 \leqslant i,j,m \leqslant n)$.
Output: $(A^m)_{i,j}$, the (i,j)th element of A^m.

(2) Determinant
Input: an $n\times n$ matrix A.
Output: $\det(A)$, the determinant of A.

(3) CharPolynomial
Input: an $n\times n$ matrix A and $i \leqslant n$.
Output: c_i, the ith coefficient of the characteristic polynomial $\chi_A(x)=x^n + c_{n-1}x^{n-1} + \cdots + c_0$ of the matrix A.

(4) MinPolynomial
Input: an $n\times n$ matrix A and $i \leqslant n$.
Output: c_i, the ith coefficient of the minimal polynomial $\mu_A(x)=x^m + c_{m-1}x^{m-1} + \cdots + c_1x + c_0$ of the matrix A.

(5) InvSystem
Input: an $n\times n$ matrix A and j, $k \leqslant n$.
Output: the kth coefficient of the jth invariant factor of the matrix A.

The functions PowerElement, Determinant and CharPolynomial are complete for **GapL** [3,7,20,23,25]. MinPolynomial and InvSystem are in $\mathbf{RNC}^2$ [12], and in $\mathbf{NC}^2$ for integer matrices [24].

For each of them, we define the *verification problem* as the graph of the corresponding function: for a fixed function $f(x)$, define v-f as the set all pairs (x,y) such that $f(x)=y$. This yields the verification problems v-PowerElement and v-Determinant. With respect to v-CharPolynomial, v-MinPolynomial and v-InvSystem, we take the tuple of *all* coefficients of a polynomial as the underlying function. I.e., for example in v-CharPolynomial, we have given A and $c_{n-1},\ldots,c_0$, and have to decide whether $1, c_{n-1},\ldots,c_0$ are the coefficients of $\chi_A(x)$.

A **GapL**-complete function yields a $\mathbf{C}_=\mathbf{L}$-complete verification problem. Hence v-PowerElement and v-Determinant are complete for $\mathbf{C}_=\mathbf{L}$. We note that a special case of v-Determinant is singularity, where one has to decide whether the determinant of a matrix A is zero. singularity is complete for $\mathbf{C}_=\mathbf{L}$ as well. In case of v-CharPolynomial we have a tuple of n underlying **GapL**-functions. The constant term, c_0, is complete for **GapL** (because $c_0=(-1)^n\det(A)$). But not all coefficients are complete for **GapL**: for example c_{n-1} is the *trace* of A (the sum of all elements on the main diagonal). Therefore c_{n-1} can be computed in $\mathbf{NC}^1$. It was an open problem whether v-CharPolynomial is complete for $\mathbf{C}_=\mathbf{L}$ [17]. We show that this is indeed the case.

A similar comment can be made for v-MinPolynomial. The characteristic and the minimal polynomial of a matrix A have the same set of roots, namely, the eigenvalues of A, and their respective constant terms are the products of these roots. Therefore, A is singular iff the constant term of the minimal polynomial of A is zero, and hence the zero-test of the constant term is complete for $\mathbf{C}_=\mathbf{L}$. We show that also v-MinPolynomial, where we have to verify *all* the coefficients, is hard for $\mathbf{C}_=\mathbf{L}$. The same hardness result holds for v-InvSystem.

3. Upper bounds

The Characteristic Polynomial. Berkowitz [3] showed that for a given matrix A one can construct in logspace a sequence of matrices such that all the coefficients of $\chi_A(x)$ appear in the iterated product of these matrices. Since each element of an iterated matrix product can be computed in **GapL**, it follows that each coefficient of $\chi_A(x)$ can be verified in $\mathbf{C}_=\mathbf{L}$. Since $\mathbf{C}_=\mathbf{L}$ is closed under logspace conjunctive reductions, also v-CharPolynomial can be solved in $\mathbf{C}_=\mathbf{L}$.

Theorem 1 (Berkowitz [3]). v-CharPolynomial $\in \mathbf{C}_=\mathbf{L}$.

The Minimal Polynomial. We mentioned in the previous section that the minimal polynomial $\mu_A(x)$ of an integer matrix A can be computed in $\mathbf{NC}^2$ [24]. We take a different approach (see [10], Section 3.3, problem 5) and show that MinPolynomial is in $\mathbf{AC}^0(\mathbf{GapL})$, a subclass of $\mathbf{NC}^2$.

Let $p(x)=x^m+c_{m-1}x^{m-1}+\cdots+c_0$ be a monic polynomial and A be a matrix. Then $p(x)=\mu_A(x)$, iff

(i) $p(A)=A^m+c_{m-1}A^{m-1}+\cdots+c_0I=\mathbf{0}$, i.e., $p(x)$ is an annihilating polynomial of A, and

(ii) for every monic polynomial $q(x)$ of degree smaller than $p(x)$, we have $q(A)\neq\mathbf{0}$.

Define vectors $\boldsymbol{a}_i=vec(A^i)$ for $i=0,1,2,\ldots,n$, where $vec(A^i)$ is the vector of length n^2 obtained by putting the columns of A^i below each other. The equation $p(A)=\mathbf{0}$ can be rewritten as

$$\boldsymbol{a}_m+c_{m-1}\boldsymbol{a}_{m-1}+\cdots+c_0\boldsymbol{a}_0=\mathbf{0}. \tag{1}$$

In other words, the vectors $\boldsymbol{a}_m,\ldots,\boldsymbol{a}_0$ are linearly dependent. Consequently, for some monic polynomial q with degree $k<m$, the inequation $q(A)\neq\mathbf{0}$ means that the vectors $\boldsymbol{a}_k,\ldots,\boldsymbol{a}_0$ are linearly independent.

In summary, the coefficients $c_{m-1},\ldots,c_0$ of $\mu_A(x)$ are the (unique) solution of system (1), for the smallest m where this system has a solution. Hence we have the following algorithm to compute $\mu_A(x)$.

MinPolynomial(A)
1. $\boldsymbol{a}_i \leftarrow vec(A^i)$ for $i=0,\ldots,n$
2. determine m such that $\boldsymbol{a}_{m-1},\ldots,\boldsymbol{a}_1,\boldsymbol{a}_0$ are linearly independent **and** $\boldsymbol{a}_m,\ldots,\boldsymbol{a}_1,\boldsymbol{a}_0$ are linearly dependent
3. solve the linear system $\boldsymbol{a}_m + c_{m-1}\boldsymbol{a}_{m-1} + \cdots + c_0\boldsymbol{a}_0 = \mathbf{0}$
4. **return** $(1, c_{m-1},\ldots,c_0)$, the coefficients of $\mu_A(x)$.

In step 1 in the above algorithm, each element of $\boldsymbol{a}_i$ can be computed in **GapL**. In step 2, checking linear independence of given vectors is in $\mathbf{co}\,\mathbf{C}_=\mathbf{L}$ and linear dependence is in $\mathbf{C}_=\mathbf{L}$ (see [2]). In step 3, we have to solve a linear system of equations. Since the vectors $\boldsymbol{a}_{m-1},\ldots,\boldsymbol{a}_0$ are linearly independent and $\boldsymbol{a}_m,\boldsymbol{a}_{m-1},\ldots,\boldsymbol{a}_0$ are linearly dependent, the system of linear equations in step 3 has a unique solution. Let C be the $n^2\times m$ matrix with columns $\boldsymbol{a}_{m-1},\ldots,\boldsymbol{a}_0$, i.e., $C=(\boldsymbol{a}_{m-1}\cdots\boldsymbol{a}_0)$. In step 3 we have to solve the system $C\boldsymbol{c}=-\boldsymbol{a}_m$ in the unknown $\boldsymbol{c}=(c_{m-1},\ldots,c_0)^{\mathrm{T}}$. Define the $m\times m$ matrix B and vector $\boldsymbol{b}$ of length m as

$$B = C^{\mathrm{T}}C \quad \text{and} \quad \boldsymbol{b} = -C^{\mathrm{T}}\boldsymbol{a}_m.$$

Since C has full column rank, matrix B is nonsingular. Therefore

$$C\boldsymbol{c} = -\boldsymbol{a}_m \;\Leftrightarrow\; B\boldsymbol{c} = \boldsymbol{b}.$$

Hence we obtain the unique solution in step 3 as $\boldsymbol{c}=B^{-1}\boldsymbol{b}$. The inverse of a given matrix can be computed in **GapL**. When m is known after step 2, each entry of B and $\boldsymbol{b}$ is computable in **GapL**, and therefore each entry of $B^{-1}\boldsymbol{b}$ is in **GapL** as well [1]. In summary, each coefficient c_i of $\mu_A(x)$ can be computed in $\mathbf{AC}^0(\mathbf{GapL})$.

Theorem 2. MinPolynomial $\in \mathbf{AC}^0(\mathbf{GapL})$.

In the corresponding *verification version* we have given A and the coefficients of a monic polynomial, and have to decide whether these coefficients represent in fact the minimal polynomial of A.

To verify the minimal polynomial we can simplify the above algorithm for MinPolynomial as follows:

v-MinPolynomial($A, c_{m-1},\ldots,c_0$)
1. $\boldsymbol{a}_i \leftarrow vec(A^i)$ for $i=0,\ldots,m$
2. **if** $\boldsymbol{a}_m + c_{m-1}\boldsymbol{a}_{m-1} + \cdots + c_0\boldsymbol{a}_0 = \mathbf{0}$ **and** $\boldsymbol{a}_{m-1},\ldots,\boldsymbol{a}_1,\boldsymbol{a}_0$ are linearly independent
3. **then** accept **else** reject.

Since the components of vectors $\boldsymbol{a}_i$ can be computed in **GapL** (line 1), the first condition in line 2 can be decided in $\mathbf{C}_=\mathbf{L}$. For the second condition, let B be the symmetric $m \times m$ matrix defined above, i.e.,

$$B = (\boldsymbol{a}_{m-1} \cdots \boldsymbol{a}_1 \boldsymbol{a}_0)^{\mathrm{T}} (\boldsymbol{a}_{m-1} \cdots \boldsymbol{a}_1 \boldsymbol{a}_0).$$

Now, $\boldsymbol{a}_{m-1}, \ldots, \boldsymbol{a}_1, \boldsymbol{a}_0$ are linearly independent iff B is nonsingular.

Since each entry of B can be computed in **GapL**, the determinant of B can be computed in **GapL** as well [1]. Thus the latter test can be done in $\mathbf{co}\,\mathbf{C}_=\mathbf{L}$. Therefore v-MinPolynomial can be decided by a $\mathbf{C}_=\mathbf{L}$ predicate in conjunction with a $\mathbf{co}\,\mathbf{C}_=\mathbf{L}$ predicate.

Corollary 3. v-MinPolynomial $\in \mathbf{C}_=\mathbf{L} \wedge \mathbf{co}\,\mathbf{C}_=\mathbf{L}$.

The Invariant Factors. The system of all invariant factors of an integer matrix can be computed in $\mathbf{NC}^2$ [24]. We show that the invariant factors can be verified in $\mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$.

Theorem 4. v-InvSystem $\in \mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$.

Proof. Let $\mathcal{S} = \{i_1(x), \ldots, i_n(x)\}$ be the system of n given monic polynomials and let A be an $n \times n$ matrix. The algorithm exploits a result from linear algebra (see [9]): we construct the companion matrices that correspond to the non-constant polynomials in $\mathcal{S}$. Let D denote the diagonal block matrix of all these companion matrices. Then $\mathcal{S}$ is the system of all invariant factors of A iff A is similar to D. Testing similarity can be done in $\mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$ [17], therefore v-InvSystem is in $\mathbf{AC}^0(\mathbf{C}_=\mathbf{L})$ too. □

4. Lower bounds

The characteristic polynomial is known to be hard for **GapL**. In this section we show that the same holds for the minimal polynomial and the invariant factors. We show that all the corresponding verification problems are hard for $\mathbf{C}_=\mathbf{L}$.

A problem known to be complete for **GapL** is PowerElement where one has to compute the entry (i, j) of A^m, for an $n \times n$ integer matrix A. W.l.o.g. we can focus on entry $(1, n)$ of A^m, i.e. $(A^m)_{1,n}$. Consequently, v-PowerElement is complete for $\mathbf{C}_=\mathbf{L}$. We take PowerElement and v-PowerElement as the reference problems to show our hardness results. Since the construction of the graph G below in this section can be done in $\mathbf{AC}^0$, all reductions here are $\mathbf{AC}^0$ many-one reductions.

4.1. Verifying the characteristic polynomial

The reduction from v-PowerElement to v-CharPolynomial builds on techniques from Toda [20] and Valiant [23] to reduce iterated matrix multiplication to the determinant. In parts of our presentation we follow [2].

Theorem 5. v-PowerElement $\leqslant_m^{AC^0}$ v-CharPolynomial.

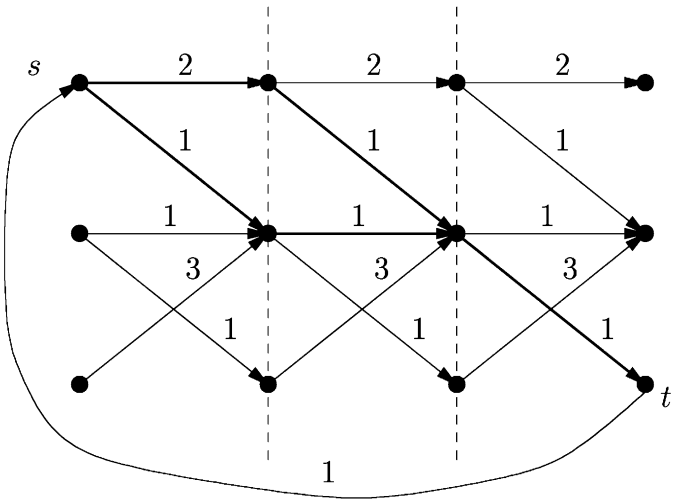

Fig. 1. The graph G constructed from matrix A for $m=3$. The three columns are indicated by the dashed lines. The edge labels are the corresponding entries of A. The thicker edges indicate the two paths from s to t. The weights of these two paths sum up to 3, which is the value of $(A^3)_{1,3}$. For the characteristic polynomial of the adjacency matrix B we get $\chi_B(x)=x^{12}-3x^8$. As we will see in Section 4.2, for the minimal polynomial we get $\mu_B(x)=x^8-3x^4$.

Proof. Let A be an $n\times n$ matrix and $1\leqslant m\leqslant n$. We will construct a matrix B such that the value $(A^m)_{1,n}$ occurs as one of the coefficients of $\chi_B(x)$.

Interpret A as representing a directed bipartite graph on $2n$ nodes and e edges. That is, the nodes are arranged in two columns of n nodes each. In both columns, nodes are numbered from 1 to n. If entry $a_{k,l}$ of A is not zero, then there is an edge labeled $a_{k,l}$ from node k in the first column to node l in the second column. The number of non-zero entries in A is exactly e.

Now, take m copies of this graph, put them in a sequence and identify each second column of nodes with the first column of the next graph in the sequence. Call the resulting graph G'. Graph G' has $m+1$ columns of nodes, and each column has exactly n nodes. The *weight* of a path in a graph is the product of all labels on the edges of the path. The crucial observation now is that the entry at position $(1,n)$ in A^m is the sum of the weights of all paths in G' from node 1 in the first column to node n in the last column. Call these two nodes s and t, respectively. Add an edge labeled 1 from t to s, and call the resulting graph G. An example for the above construction of G for

$$A=\begin{pmatrix}2&1&0\\0&1&1\\0&3&0\end{pmatrix}$$

is shown in Fig. 1.

Let B be the adjacency matrix of G. So B is an $N \times N$ matrix, where $N = (m+1)n$ is the number of nodes of G. Let the characteristic polynomial of B have the form

$$\chi_B(x) = \det(xI_N - B) = x^N + \sum_{i=0}^{N-1} c_i x^i,$$

where I_N is the $N \times N$ identity matrix. We give two ways how to compute the coefficients c_i in $\chi_B(x)$:

(1) one way is to use elementary linear transformations and bring the polynomial matrix $xI_N - B$ into triangular block form. Then the characteristic polynomial of B can be computed from the resulting polynomial matrix.

(2) a very elegant proof is provided by combinatorial matrix theory. From there we know that the coefficients of the characteristic polynomial can be expressed as cycle covers in the graph G (see [4,6,13,14,26]).

We start by giving the combinatorial argument which is much shorter than the algebraic argument.

The combinatorial way. It is known that the coefficient c_i in $\chi_B(x)$ equals the sum of the disjoint weighted cycles that cover $N-i$ nodes in G, with appropriate sign (see [4] or [6] for more details). In the graph G, all edges go from a layer to the next layer. The only exception is the edge (t,s). So any cycle in G must use precisely this edge (t,s), and then trace out a path from s to t. Therefore each cycle in G has exactly the length $m+1$, and the weighted sum of all these cycles is precisely $(-1)^{m+1}(A^m)_{1,n}$ (for the sign, recall that we consider $xI_N - B$). The sign of the cycle (as a permutation) is $(-1)^m$. Hence

$$c_{N-(m+1)} = (-1)^{m+1}(-1)^m (A^m)_{1,n} = -(A^m)_{1,n}$$

and all other coefficients must be zero. That is, for $a = (A^m)_{1,n}$,

$$\chi_B(x) = x^N - ax^{N-(m+1)}$$

is the characteristic polynomial of B.

The algebraic way. We consider the adjacency matrix B of the graph G. Except for the edge from t to s, graph G is acyclic. Thus we can put the nodes of G in such an order, that adjacency matrix B is upper triangular for the first $N-1$ rows with zeros along the main diagonal. The last row of B has a one in the first position (representing edge (t,s)), and the rest is zero.

Now we can write B as a $(m+1) \times (m+1)$ block matrix as follows

$$B = \begin{pmatrix} & A & & \\ & & \ddots & \\ & & & A \\ L & & & \end{pmatrix}.$$

Matrix A occurs m-times on the upper sub-diagonal of B. L is the $n \times n$ matrix with a one at position $(n,1)$ and zero elsewhere. The empty places in B are all zero (matrices).

Therefore $xI_N - B$ has the form

$$xI_N - B = \begin{pmatrix} xI_n & (-A) & & \\ & \ddots & \ddots & \\ & & xI_n & (-A) \\ -L & & & xI_n \end{pmatrix}.$$

To compute $\chi_B(x)$ we transform $xI_N - B$ into an upper triangular block matrix. Note that it already is upper triangular except for matrix L in the lower left corner. We want to eliminate this block.

The first step is to multiply the last block row by xI_n, and add to it the first block row multiplied by L (from right). This transforms the last block row into

$$\mathbf{0}, -AL, \mathbf{0}, \ldots, \mathbf{0}, x^2 I_n.$$

In the second step, we multiply the last block row again by xI_n, and add to it the second block row multiplied by AL (from right). This transforms the last block row into

$$\mathbf{0}, \mathbf{0}, -A^2 L, \mathbf{0}, \ldots, \mathbf{0}, x^3 I_n.$$

Continuing that way for m iterations, we bring the last block row into

$$\mathbf{0}, \ldots, \mathbf{0}, x^{m+1} I_n - A^m L.$$

Let $D(x)$ be the resulting upper triangular matrix. The diagonal of $D(x)$ is

$$xI_n, \ldots, xI_n, x^{m+1} I_n - A^m L.$$

The determinant of $D(x)$ is the product of the determinants of diagonal blocks, that is

$$\det(D(x)) = x^{N-n} \det(x^{m+1} I_n - A^m L).$$

It remains to compute the determinant of $x^{m+1} I_n - A^m L$. Recall the form of matrix L: the only non-zero entry is a 1 in the lower left corner. Therefore $A^m L$ has the last column of A^m as its first column and 0 elsewhere. Hence $x^{m+1} I_n - A^m L$ is an $n \times n$ lower triangular matrix with the diagonal

$$x^{m+1} - (A^m)_{1,n}, x^{m+1}, \ldots, x^{m+1},$$

that has determinant

$$\det(x^{m+1} I_n - A^m L) = x^{(n-1)(m+1)} (x^{m+1} - a),$$

where $a = (A^m)_{1,n}$. Thus

$$\det(D(x)) = x^{N-n} x^{(n-1)(m+1)} (x^{m+1} - a).$$

Note, however, that this is not the same as $\chi_B(x)$: we changed $\chi_B(x)$ with each multiplication of the last block row by xI_n, and we did this m-times. Therefore

$$\begin{aligned}\chi_B(x) &= \det(D(x))/\det(x^m I_n)\\ &= x^{N-n}\, x^{(n-1)(m+1)}\,(x^{m+1}-a)\,x^{-mn}\\ &= x^N - ax^{N-(m+1)}.\end{aligned}$$

In summary, both methods explicitly yield the coefficients of $\chi_B(x)$ and we have

$$(A^m)_{1,n} = a \;\Leftrightarrow\; \chi_B(x) = x^N - ax^{N-(m+1)}.$$

This proves the theorem. □

Corollary 6. v-CharPolynomial *is complete for* $\mathbf{C}_=\mathbf{L}$.

4.2. The minimal polynomial

We show in this section that the minimal polynomial of a matrix is hard for **GapL**. To do so, we extend the reduction from v-PowerElement to v-CharPolynomial to a reduction from PowerElement to MinPolynomial. Namely, we show that the minimal polynomial of the matrix B above has the value $(A^m)_{1,n}$ as one of its coefficients.

Theorem 7. PowerElement $\leqslant_m^{AC^0}$ MinPolynomial.

Proof. We consider the $N \times N$ matrix B from the previous section. The characteristic polynomial of B is $\chi_B(x)=x^N - ax^{N-(m+1)}$. We claim that the minimal polynomial of B is $\mu_B(x)=x^{2m+2} - ax^{m+1}$.

Recall that polynomial $d_{N-1}(x)$ is the greatest common divisor of all sub-determinants of $(xI_N - B)$ of order $N-1$. We observe that the sub-determinant at position $(1,1)$ is x^{N-1}. Hence $d_{N-1}(x)=x^l$ for some l. Therefore the minimal polynomial must have the form

$$\mu_B(x) = \frac{\chi_B(x)}{d_{N-1}(x)} = x^{N-l} - ax^{N-(m+1)-l}$$

for some $l \geqslant 0$.

Define polynomials $p_k(x)=x^{(m+1)+k} - ax^k$ for $0 \leqslant k \leqslant N-(m+1)$. We claim that $\mu_B = p_{m+1}$. To prove our claim, we have to show that $p_{m+1}(B)=\mathbf{0}$ and $p_k(B)\neq\mathbf{0}$ for all $k<m+1$. To do so, we explicitly compute all the powers of B, i.e., B^i for

$i=2,\ldots,m+1$. We get

$$B^2=\left(\begin{array}{cc|c} & & A^2 \\ & & \ddots \\ & & A^2 \\ \hline AL & & \\ & LA & \end{array}\right), \quad B^3=\left(\begin{array}{cc|c} & & A^3 \\ & & \ddots \\ & & A^3 \\ \hline A^2L & & \\ & ALA & \\ & & LA^2 \end{array}\right).$$

The general form of B^i for $i \leqslant m$ is as follows

$$B^i=\begin{array}{c} \begin{array}{cccc} & & i & i+1 \\ & & \downarrow & \downarrow \end{array} \\ \left(\begin{array}{ccc|c} & & & A^i \\ & & & \ddots \\ & & & A^i \\ \hline A^{i-1}L & & & \\ & A^{i-2}LA & & \\ & & \ddots & \\ & & & LA^{i-1} \end{array}\right) \end{array} \begin{array}{l} \leftarrow 1 \\ \\ \leftarrow m+1-i \\ \leftarrow m+2-i \\ \\ \\ \leftarrow m+1 \end{array}.$$

Finally, matrix B^{m+1} is a diagonal block matrix. Its ith diagonal block is $A^{m+1-i}LA^{i-1}$ for all $1 \leqslant i \leqslant m+1$. Matrix $B^{2m+2}=(B^{m+1})^2$ is therefore a diagonal block matrix too. Its ith diagonal block is the square of the ith diagonal block of B^{m+1}, i.e.,

$$(A^{m+1-i}LA^{i-1})^2=A^{m+1-i}LA^mLA^{i-1}.$$

Now, observe that there occurs the factor LA^mL in each of the diagonal entries of B^{2m+2}. It is easy to verify that $LA^mL=aL$. Therefore we can pull the factor a in front of the matrix and what remains is again B^{m+1}. I.e., we have shown that $B^{2m+2}=aB^{m+1}$. Therefore

$$p_{m+1}(B)=B^{2m+2}-aB^{m+1}=\mathbf{0}.$$

It remains to prove $p_k(B)=B^{m+1+k}-aB^k \neq \mathbf{0}$ for all $k \leqslant m$. Note that it suffices to prove this for $k=m$, because $p_k(B)=\mathbf{0}$ for some k implies $p_{k+1}(B)=\mathbf{0}$.

Assume that $p_m(B)=B^{2m+1}-aB^m=\mathbf{0}$. Then $B^{2m+1}=aB^m$. We consider the blocks at position $(1,m+1)$ in B^{2m+1} and B^m:

- in B^m it is A^m,
- compute B^{2m+1} as the product $B^{m+1}B^m$. Then it is easy to see that the block at position $(1,m+1)$ is A^mLA^m.

Now, if $p_m(B)=\mathbf{0}$, then we must have $A^mLA^m=aA^m$. However, the latter equation cannot hold: by Lemma 9 below we can assume that A is nonsingular. Therefore $\operatorname{rank}(A^mLA^m)=1$, whereas $\operatorname{rank}(aA^m)=n$, for $a\neq 0$, and 0, otherwise. We conclude that $p_m(B)\neq\mathbf{0}$.

In summary, we get $\mu_B(x)=x^{2m+2}-ax^{m+1}$, where $a=(A^m)_{1,n}$. This proves the theorem. □

Corollary 8. (1) MINPOLYNOMIAL *is hard for* **GapL**,
(2) V-MINPOLYNOMIAL *is hard for* $\mathbf{C}_{=}\mathbf{L}$.

It remains to justify that we may assume A to be nonsingular (in the proof of Theorem 7).

Lemma 9. *Suppose A is an $n\times n$ matrix. Then there is a nonsingular upper triangular $p\times p$ matrix C (that can be easily constructed) such that $(C^m)_{1,p}=(A^m)_{1,n}$.*

Proof. Define C as a $(m+1)\times(m+1)$ block matrix

$$C=\begin{pmatrix} I & A & & \\ & \ddots & \ddots & \\ & & I & A \\ & & & I \end{pmatrix},$$

where I is the $n\times n$ identity matrix. Then C is nonsingular and C^m has the following form

$$C^m=\begin{pmatrix} I & mA & mA^2 & \cdots & mA^{m-1} & A^m \\ & I & mA & \cdots & mA^{m-2} & mA^{m-1} \\ & & \ddots & \ddots & \vdots & \vdots \\ & & & \ddots & mA & mA^2 \\ & & & & I & mA \\ & & & & & I \end{pmatrix}$$

and, for $p=(m+1)n$, we have $(C^m)_{1,p}=(A^m)_{1,n}$ as claimed. □

4.3. The invariant factors

Since the minimal polynomial is the first polynomial in the system of all invariant factors, it follows from Theorem 7 that this system is hard for **GapL** as well.

Now we show that the verification of the system of all invariant factors is hard for $\mathbf{C}_{=}\mathbf{L}$.

Theorem 10. V-INVSYSTEM *is hard for* $\mathbf{C}_{=}\mathbf{L}$.

Proof. We continue with the setting from the proof of Theorem 7, in particular with matrix B. Our goal is to determine the system of all invariant factors of B. We have already shown that $i_1(x)=\mu_B(x)=x^{2m+2}-ax^{m+1}$, where $(A^m)_{1,n}=a$. It remains to compute the invariant factors $i_2(x),\ldots,i_N(x)$ of B.

From the proof of Theorem 7 we know that $d_{N-1}(x)=x^{N-(2m+2)}$. Since $d_{N-1}(x)=i_2(x)\cdots i_N(x)$, each of the invariant factors must have the form x^l for some

number l. Note that the non-constant invariant factors of the form x^l are already *elementary divisors* of B. Therefore, it suffices to determine all elementary divisors of B.

Define g_i to be the *number of occurrences of the elementary divisor* x^i, and let r_i denote the rank of B^i. The following formula relates the ranks to numbers g_i (see [9, Chapter VI]:

$$g_i = r_{i-1} + r_{i+1} - 2r_i, \tag{2}$$

for $i=1,\ldots,t$, where $r_0=N$ and t is the smallest index such that $r_{t-1}>r_t=r_{t+1}$. We can actually compute all the ranks r_i from the matrices B^i which we have already computed in the proof of Theorem 7.

By Lemma 9 we may assume that $\operatorname{rank}(A)=n$ and therefore $\operatorname{rank}(A^i)=n$ for all i. Consider the general form of B^i for $1\leqslant i\leqslant m$. The rank of B^i equals the sum of the ranks of the matrices on the lower and upper sub-diagonals.

- Each of the $m+1-i$ blocks on the upper sub-diagonal of B^i has the form A^i, and $\operatorname{rank}(A^i)=n$.
- Each of the i blocks on the lower sub-diagonal of B^i has the form $A^{i-k}LA^{k-1}$ for $1\leqslant k\leqslant i$, and $\operatorname{rank}(A^{i-k}LA^{k-1})=\operatorname{rank}(L)=1$.

Therefore $\operatorname{rank}(B^i)=(m+1-i)n+i$ for $1\leqslant i\leqslant m$. Analogously we can compute the ranks of B^{m+1} and B^{m+2}:

$$\operatorname{rank}(B^{m+1}) = \operatorname{rank}(B^{m+2}) = m+1.$$

Therefore we get the general form for $r_i=\operatorname{rank}(B^i)$:

$$r_i = \begin{cases} (m+1-i)n+i & \text{for } i=1,\ldots,m,\\ m+1 & \text{for } i=m+1,m+2. \end{cases}$$

Plugged into Eq. (2), we see that $t=m+1$ because $r_m>r_{m+1}=r_{m+2}$. Furthermore, we get from Eq. (2)

$$g_i = \begin{cases} N-n(m+1) & \text{for } i=1,\\ 0 & \text{for } i=2,\ldots,m,\\ n-1 & \text{for } i=m+1. \end{cases} \tag{3}$$

From Eq. (3) we can deduce the invariant factors: we have $n-2$ factors x^{m+1} (note that one of the $n-1$ elementary divisors x^{m+1} occurs in $i_1(x)$), furthermore $N-n(m+1)$ factors x, and constant 1 as the remaining factors:

$$i_k(x) = \begin{cases} x^{m+1} & \text{for } k=2,\ldots,n-1,\\ x & \text{for } k=n,\ldots,N-nm-1,\\ 1 & \text{for } k=N-nm,\ldots,N. \end{cases} \tag{4}$$

In summary, $(A^m)_{1,n}=a$ iff $i_1(x)=x^{2m+2}-ax^{m+1}$, and $i_2(x),\ldots,i_N(x)$ as defined in (4) are the invariant factors of A. This completes the proof of Theorem 10. □

Corollary 11. InvSystem *is hard for* **GapL**.

Summary and Open Problems

The following table summarizes the lower and upper bounds for the problems considered in this paper.

Problem	hard for	contained in
DETERMINANT	**GapL**	**GapL**
CHARPOLYNOMIAL	**GapL**	**GapL**
V-CHARPOLYNOMIAL	$\mathbf{C_=L}$	$\mathbf{C_=L}$
MINPOLYNOMIAL	**GapL**	$\mathbf{AC^0(GapL)}$
V-MINPOLYNOMIAL	$\mathbf{C_=L}$	$\mathbf{C_=L} \wedge \mathbf{co}\,\mathbf{C_=L}$
INVSYSTEM	**GapL**	$\mathbf{NC^2}$
V-INVSYSTEM	$\mathbf{C_=L}$	$\mathbf{AC^0(C_=L)}$

An obvious task for further research is to close the gaps between the lower and the upper bounds where they do not match.

Another important question is whether $\mathbf{C_=L}$ is closed under complement. In the case of an affirmative answer, $\mathbf{C_=L}$ would equal $\mathbf{AC^0(C_=L)}$. In particular this would close the gap for V-INVSYSTEM and V-MINPOLYNOMIAL (and would solve lots of other problems (see [2])).

Acknowledgements

We wish to thank Eric Allender. He gave many helpful comments on the paper. Also he pointed us to the results in [1] which lead to an improved upper bound for V-MINPOLYNOMIAL (Corollary 3). We thank Meena Mahajan and V. Vinay for pointing out to us the combinatorial proof of Theorem 5. Furthermore, the comments of the anonymous referees helped to improve the presentation of the paper.

References

[1] E. Allender, V. Arvind, M. Mahajan, Arithmetic complexity, Kleene closure, and formal power series, 1999, Available at http://www.cs.rutgers.edu/∼allender/publications.
[2] E. Allender, R. Beals, M. Ogihara, The complexity of matrix rank and feasible systems of linear equations, Comput. Complexity 8 (1999) 99–126.
[3] S. Berkowitz, On computing the determinant in small parallel time using a small number of processors, Inform. Process. Lett. 18 (1984) 147–150.
[4] R. Brualdi, H. Ryser, Combinatorial Matrix Theory, Encyclopedia of Mathematics and its Applications, Vol. 39, Cambridge University Press, Cambridge, 1991.
[5] S. Cook, A taxonomy of problems with fast parallel algorithms, Inform. Control 64 (1985) 2–22.

[6] D. Cvetković, M. Doob, H. Sachs, Spectra of Graphs, Theory and Application, Academic Press, New York, 1980.
[7] C. Damm. DET $= L^{(\#L)}$ (1985), Technical Report Informatik-Preprint 8, Fachbereich Informatik der Humboldt-Universität zu Berlin, 1991.
[8] S. Fenner, L. Fortnow, S. Kurtz, Gap-definable counting classes, J. Comput. System Sci. 48 (1994) 116–148.
[9] F. Gantmacher, The Theory of Matrices, Vols. 1 and 2, AMS Chelsea Publishing, Providence, RI, 1977.
[10] R. Horn, C. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 1985.
[11] N. Immerman, Nondeterministic space is closed under complement, SIAM J. Comput. 17 (1988) 935–938.
[12] E. Kaltofen, B. Saunders, Fast parallel computation of Hermite and Smith forms of polynomial matrices, SIAM Algebraic Discrete Methods 8 (1987) 683–690.
[13] M. Mahajan, V. Vinay, Determinant: combinatorics, algorithms, and complexity, Chicago J. Theoret. Comput. Sci. (5) (1997).
[14] M. Mahajan, V. Vinay, Determinant: old algorithms, new insights SIAM J. Discrete Math. 12 (4) (1999) 474–490.
[15] N. Nisan, A. Ta-Shma, Symmetric logspace is closed under complement, Chicago J. Theoret. Comput. Sci. (1) (1995).
[16] K. Reinhardt, E. Allender, Making nondeterminism unambiguous, in: 38th Symp. on Foundation of Computer Science, IEEE Computer Society Press, Silver Spring, MD, 1997, pp. 244–253.
[17] M. Santha, S. Tan, Verifying the determinant in parallel, Comput. Complexity 7 (1998) 128–151.
[18] A. Storjohann, An $O(n^3)$ algorithm for Frobenius normal form, in: Internat. Symp. on Symbolic and Algebraic Computation (ISSAC), 1998.
[19] R. Szelepcsényi, The method of forced enumeration for nondeterministic automata, Acta Inform. 26 (3) (1988) 279–284.
[20] S. Toda, Counting problems computationally equivalent to the determinant, Technical Report CSIM 91-07, Department of Computer Science and Information Mathematics, University of Electro-Communications, Chofu-shi, Tokyo, Japan, 1991.
[21] L. Valiant, The complexity of computing the permanent, Theoret. Comput. Sci. 8 (1979) 189–201.
[22] L. Valiant, The complexity of enumeration and reliability problems, SIAM J. Comput. 8 (1979) 410–421.
[23] L. Valiant, Why is boolean complexity theory difficult, in: M.S. Paterson (Ed.), Boolean Function Complexity, London Mathematical Society Lecture Notes Series, Vol. 169, Cambridge University Press, Cambridge, 1992.
[24] G. Villard, Fast parallel algorithms for matrix reduction to normal forms, Appl. Algebra Eng. Comm. Comput. (AAECC) 8 (1997) 511–537.
[25] V. Vinay, Counting auxiliary pushdown automata and semi-unbounded arithmetic circuits, in: 6th IEEE Conf. on Structure in Complexity Theory, 1991, pp. 270–284.
[26] D. Zeilberger, A combinatorial approach to matrix algebra, Discrete Math. 56 (1985) 61–72.