Electronic Colloquium on Computational Complexity, Report No. 24 (2004)

On Closure Properties of **GapL** *

Thanh Minh Hoang
Abt. Theoretische Informatik
Universität Ulm
89069 Ulm, Germany

Thomas Thierauf
FB Elektronik und Informatik
FH Aalen
73430 Aalen, Germany

{hoang,thierauf}@informatik.uni-ulm.de

Abstract

We show necessary and sufficient conditions that certain algebraic functions like the rank or the signature of an integer matrix can be computed in **GapL**.

1 Introduction

Valiant [Val79b, Val79a] introduced the counting class #**P** based on polynomial-time, nondeterministic Turing machines. It characterizes the computational complexity to compute the number of perfect matchings in a graph or the permanent of matrix over the natural numbers. Fenner, Fortnow, and Kurtz [FFK94] extended #**P** to the class **GapP** that can handle negative numbers. The permanent of an integer matrix is a complete problem for **GapP**. Closure properties of #**P** and **GapP** have been investigated in many papers, see for example [BRS91, FR96, HO93, OTTW96, TTW94].

By altering polynomial time to logspace computations, Allender and Ogihara [AO96] defined the counting class **GapL**. It characterizes the computational complexity to compute powers of an integer matrix or, most prominently, the determinant of an integer matrix [Dam91, Tod91b, Vin91, Val92].

The motivation for this paper comes from the fact that some related interesting problems with respect to integer matrices are *not known* to be computable in **GapL**. Examples are the rank or the signature of a matrix. These problems are just known to be computable with several queries to a **GapL**-oracle [ABO99, HT02b]. We investigate the question, whether these functions can be computed in **GapL**. Our main results are that these questions are equivalent to the collapse of certain complexity classes.

Complexity classes based on **GapL** are $\mathbf{C_=L}$ (*exact counting in logspace*) and **PL** (*probabilistic logspace*). Complete problems for these classes are to

*Supported in part by DFG grants Th 472/3-2 and Scho 302/7-1.

ISSN 1433-8092

decide whether the determinant of an integer matrix is zero (singularity), or greater than zero, respectively. Another class is **SPL** which is based on 0-1-valued **GapL**-functions. Intuitively, this is a very small class. Nonetheless, Allender, Reinhardt, and Zhou [ARZ99] showed that the perfect matching problem is located in a nonuniform version of **SPL**.

We show in Section 3 that the rank of a matrix can be computed in **GapL** if and only if $\mathbf{C_=L} = \mathbf{SPL}$. In Section 4 we show that the signature of a matrix can be computed in **GapL** if and only if $\mathbf{PL} = \mathbf{SPL}$.

Note that $\mathbf{NL} \subseteq \mathbf{C_=L}$ and $\mathbf{SPL} \subseteq \oplus\mathbf{L}$. Hence, as a corollary of our results we get: if the rank or the signature of a matrix can be computed in **GapL** then $\mathbf{NL} \subseteq \oplus\mathbf{L}$ and $\mathbf{C_=L}$ is closed under complement. Both consequences are famous open problems right now.

We also consider a relaxed version of the above question. **GapL** is not known to be closed under division. Hence it is natural to ask whether we can write the rank or the signature of a matrix as a quotient of two **GapL**-functions. We show in Section 3 that this is true for the rank of a matrix if and only if $\mathbf{C_=L} = \mathbf{coC_=L}$. In Section 4 we show that this is true for the signature of a symmetric matrix if and only if $\mathbf{PL} = \mathbf{C_=L}$.

Finally, in Section 5 we characterize the case that the absolute value of any **GapL**-function can be computed in **GapL** too.

2 Preliminaries

For a nondeterministic Turing machine M on input x, we denote the number of accepting and rejecting computation paths by $acc_M(x)$ and $rej_M(x)$, respectively. The difference of these two quantities is denoted by $gap_M(x)$. That is, $gap_M(x) = acc_M(x) - rej_M(x)$. In the polynomial time setting, complexity classes $\#\mathbf{P}$ and **GapP** are defined via these functions. We are interested in the logspace versions:

$$\begin{aligned} \#\mathbf{L} =& \{\, acc_M \mid M \text{ is a nondeterministic logspace Turing machine} \,\}, \\ \mathbf{GapL} =& \{\, gap_M \mid M \text{ is a nondeterministic logspace Turing machine} \,\}. \end{aligned}$$

In analogy to the polynomial time setting, we define the following counting complexity classes [AO96, ARZ99]:

$$\begin{aligned} \mathbf{C_=L} =& \{\, S \mid \exists f \in \mathbf{GapL},\ \forall x:\ x \in S \Longleftrightarrow f(x) = 0 \,\}, \\ \mathbf{PL} =& \{\, S \mid \exists f \in \mathbf{GapL},\ \forall x:\ x \in S \Longleftrightarrow f(x) > 0 \,\}, \\ \mathbf{SPL} =& \{\, S \mid \chi_S \in \mathbf{GapL} \,\}, \end{aligned}$$

where χ_S is the characteristic function of set S. It is known that

$$\mathbf{SPL} \subseteq \mathbf{C_=L} \subseteq \mathbf{PL} \subseteq \mathbf{TC}^1 \subseteq \mathbf{NC}^2.$$

Also we have $\mathbf{NL} \subseteq \mathbf{C_=L}$.

These classes are interesting because of the complete problems therein. We give some examples. When nothing else is said, by matrices we mean square integer matrices. We use n as the order of the matrices.

Problems complete for **GapL** are to compute (one element of) the m-th power of a matrix and the determinant [Tod91a, Dam91, Vin91, Val92]. It follows that

$$\text{SINGULARITY} = \{ A \mid \det(A) = 0 \}$$

is complete for $\mathbf{C_=L}$. The set

$$\text{POSDETERMINANT} = \{ A \mid \det(A) > 0 \}$$

is complete for **PL**. More general, the sets

$$\begin{aligned} \text{V-POWERELEMENT} =& \{ (A, a, m) \mid (A^m)_{1,n} = a \}, \\ \text{V-DETERMINANT} =& \{ (A, a) \mid \det(A) = a \}, \text{ and} \\ \text{RANK}_< =& \{ (A, r \mid \text{rank}(A) < r \} \end{aligned}$$

are complete for $\mathbf{C_=L}$. Consequently $\text{RANK}_\geq = \{ (A, r \mid \text{rank}(A) \geq r \}$ is complete for $\mathbf{coC_=L}$. The verification of the rank can be written as the intersection of a set in $\mathbf{C_=L}$ and in $\mathbf{coC_=L}$: $\text{V-RANK} = \{ (A, r \mid \text{rank}(A) = r \} = \text{RANK}_< \cap \text{RANK}_\geq$. This means that $\text{V-RANK} \in \mathbf{C_=L} \wedge \mathbf{coC_=L}$.

Allender, Beals, and Ogihara [ABO99] investigated the complexity of computing (one bit of) the rank. That is

$$\text{RANK} = \{ (A, k, b) \mid \text{ the } k\text{-th bit of rank}(A) \text{ is } b \}.$$

They showed that RANK is a complete problem for $\mathbf{AC^0(C_=L)}$, the $\mathbf{AC^0}$-closure of $\mathbf{C_=L}$.

In analogy to the polynomial time setting, Allender, Reinhardt, and Zhou [ARZ99] defined the class **SPL**. They showed that the perfect matching problem is located in a nonuniform version of **SPL**.

GapL possesses some closure properties. In particular, it is closed under exponential summations and polynomial multiplications.

Theorem 2.1 [AO96] *Let any* $f \in \mathbf{GapL}$ *the following functions are in* **GapL** *too:*

1. $f(g(\cdot))$, *for any* $g \in \mathbf{FL}$,
2. $\sum_{i=0}^{2^{|x|^c}} f(x,i)$, *for any constant* c,
3. $\prod_{i=0}^{|x|^c} f(x,i)$, *for any constant* c,
4. $\binom{f(x)}{g(x)}$, *for any* $g \in \mathbf{FL}$ *such that* $g(x) = O(1)$.

The first property has been improved considerably. Essentially, **GapL** is closed under composition:

Theorem 2.2 [AAM03] *The determinant of a matrix having* **GapL***-computable elements can be computed in* **GapL***.*

With respect to the decision problems we have that **PL**, $\mathbf{C_=L}$, and **SPL** are closed under union and intersection. Furthermore, **PL** and **SPL** are closed under complement. For $\mathbf{C_=L}$, closure under complement is an open problem.

3 Matrix rank

Assume that the rank of a matrix could be computed in **GapL**. Then the verification of the rank, v-Rank, would be in $\mathbf{C_=L}$. On the other hand v-Rank is complete for $\mathbf{C_=L} \wedge \mathbf{coC_=L}$. Hence this would imply $\mathbf{C_=L} = \mathbf{coC_=L}$.

The following theorem strengthens this collapse considerably.

Theorem 3.1 $\mathbf{C_=L} = \mathbf{SPL} \iff \text{rank} \in \mathbf{GapL}$.

Proof. Assume that $\mathbf{C_=L} = \mathbf{SPL}$. Then v-Rank $\in \mathbf{SPL}$. Hence, there is a function $g \in \mathbf{GapL}$ such that for a given matrix A of order n and a number r we have

$$\begin{aligned} \text{rank}(A) = r &\implies g(A,r) = 1, \\ \text{rank}(A) \neq r &\implies g(A,r) = 0. \end{aligned}$$

It follows that $\text{rank}(A) = \sum_{r=1}^{n} r\, g(A,r)$, and therefore rank $\in \mathbf{GapL}$.

Conversely, suppose that rank $\in \mathbf{GapL}$. We consider Singularity which is complete for $\mathbf{C_=L}$. Let A be a matrix. Define matrix B of order 1 by $B = (\det(A))$. Now $\text{rank}(B) = 0$, if A is singular, and $\text{rank}(B) = 1$, otherwise. Since **GapL** is closed under composition, $\text{rank}(B)$ is computable in **GapL**. Therefore, $\text{rank}(B)$ is the characteristic function of Singularity, i.e. Singularity $\in \mathbf{SPL}$ □

Next we weaken the assumption for the rank-function: instead of one **GapL**-function that computes the rank directly, suppose there are two **GapL**-functions g and h such that the rank can be written as the quotient of g and h, i.e., $\text{rank}(A) = g(A)/h(A)$. We show that this is a necessary and sufficient condition for $\mathbf{C_=L}$ being closed under complement.

Theorem 3.2 $\mathbf{C_=L} = \mathbf{coC_=L} \iff \exists g, h \in \mathbf{GapL} \quad \text{rank} = g/h$.

Proof. Assume that $\mathbf{C_=L} = \mathbf{coC_=L}$. Then the problem of verifying the rank of a matrix, v-Rank, is in $\mathbf{coC_=L}$. That is, there is a function $f \in \mathbf{GapL}$ such that for any symmetric matrix A and any r,

$$\text{rank}(A) = r \iff f(A,r) \neq 0.$$

Define functions $g(A) = \sum_{r=0}^{n} r f(A,r)$ and $h(A) = \sum_{r=0}^{n} f(A,r)$. Then we have $g, h \in$ **GapL** and rank $= g/h$ as claimed.

Conversely, let $g, h \in$ **GapL** such that rank $= g/h$. For a given symmetric matrix A and an integer $k \geq 0$, define $f(A,k) = g(A) - k\, h(A)$. Then $f \in$ **GapL** and we have

$$\text{rank}(A) = r \iff f(A,r) = 0.$$

It follows that the rank of a matrix can be verified in $\mathbf{C_=L}$. Hence $\mathbf{C_=L} = \mathbf{coC_=L}$. □

It was shown in [HT02a] that the degree of the minimal polynomial is computationally equivalent to matrix rank. Therefore, we can formulate the above theorems also in terms of the degree of the minimal polynomial.

There is an interesting alternative way of representing the rank of a matrix. Consider an $n \times n$ symmetric matrix A and let its characteristic polynomial be

$$\chi_A(x) = x^n + c_{n-1}x^{n-1} + \cdots + c_1 x + c_0.$$

It is well known from linear algebra that $\text{rank}(A) = k \iff c_{n-k} \neq 0$ and $c_{n-k-1} = c_{n-k-2} = \cdots = c_0 = 0$. Furthermore, all coefficients c_i are computable in **GapL** [Ber84].

Define a vector $\boldsymbol{w} = (w_n, w_{n-1}, \cdots, w_1, w_0)^T$, where $w_j = \sum_{i=0}^{j} c_i^2$, for $j = 0, 1, \ldots, n$. Hence every element of $\boldsymbol{w}$ is computable in **GapL**. Furthermore we have $\text{rank}(A) = k$ if and only if

(i) $\boldsymbol{w}$ has precisely $k+1$ positive elements, $w_n, w_{n-1}, \ldots, w_{n-k}$, and

(ii) precisely $n-k$ zero elements, $w_{n-k-1} = w_{n-k-2} = \cdots = w_0 = 0$.

Conversely, for a given nonnegative **GapL**-vector v, the number of its positive elements is exactly the rank of the diagonal matrix whose diagonal is v.

In summary, the problem of determining the rank of a matrix is (logspace) equivalent to the problem of determining the number of consecutive zeros at the right end in a **GapL**-vector.

4 Matrix inertia

The *inertia* of a symmetric $n \times n$ matrix A is defined to be the triple $i(A) = (i_+(A), i_-(A), i_0(A))$, where $i_+(A)$, $i_-(A)$, and $i_0(A)$ are the number of eigenvalues of A, counting multiplicities, which are positive, negative, and zero, respectively. Hoang and Thierauf [HT02b] used the Routh-Hurwitz Theorem to show that (bits of) the inertia of a symmetric matrix can be computed in **PL**.

The *signature of* A is defined as $\mathrm{sig}(A) = i_+(A) - i_-(A)$. Note that $i_+(A) + i_-(A) + i_0(A) = n$, and $\mathrm{rank}(A) = i_+(A) + i_-(A) = n - i_0(A)$.

The rank can be reduced to the signature and to i_+, because

$$\mathrm{rank}(A) = \mathrm{rank}(A^T A) = \mathrm{sig}(A^T A) = i_+(A^T A).$$

Functions i_+ and i_- are computationally equivalent, because $i_+(A) = i_-(-A)$. Therefore the signature can be reduced to either i_+ or i_-, $\mathrm{sig}(A) = i_+(A) - i_+(-A) = i_-(-A) - i_-(A)$.

The notions above are closely related to an equivalence relation on symmetric matrices: two symmetric matrices A and B are *congruent*, if there exists a nonsingular (real) matrix S such that $A = SBS^T$. We denote the set of congruent symmetric matrices by Congruence. It is known that

$$A \text{ is congruent to } B \iff i(A) = i(B).$$

Congruence is contained in **PL** [HT02b] and hard for $\mathbf{AC}^0(\mathbf{C_=L})$ [HT00].

The following theorem characterizes the case that the upper bound for the inertia or the signature can be improved from **PL** to **GapL**.

Theorem 4.1 *The following conditions are equivalent.*

(i) $\mathbf{PL} = \mathbf{SPL}$,

(ii) Congruence $\in \mathbf{SPL}$,

(iii) $\mathrm{sig} \in \mathbf{GapL}$,

(iv) $i_+ \in \mathbf{GapL}$.

Proof. From the outline above, it follows that *(i)* implies *(ii)*.

We show that *(ii)* implies *(iii)*. Assume that Congruence $\in$ **SPL**. That is, there is a function $f \in \mathbf{GapL}$ such that for any two symmetric matrices B and C we have

$$\begin{aligned}(B, C) \in \textsc{Congruence} &\implies f(B, C) = 1,\\ (B, C) \notin \textsc{Congruence} &\implies f(B, C) = 0,\end{aligned}$$

Now let A be a symmetric matrix of order n. We show that the signature of A can be computed in **GapL**. For $k = 0, \dots, n$ define matrices B_k, C_k, D_k, E_k of order $n + k$ as follows:

$$B_k = \begin{pmatrix} A & & & \\ & -1 & & \\ & & \ddots & \\ & & & -1 \end{pmatrix} \quad \text{and} \quad D_k = \begin{pmatrix} A & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix}$$

and $C_k = -B_k$ and $E_k = -D_k$. Then we have

$$\begin{array}{rcl} i(B_k) & = & (i_+(A), \quad i_-(A)+k, \quad i_0(A)), \\ i(C_k) & = & (i_-(A)+k, \quad i_+(A), \quad i_0(A)), \end{array}$$

and

$$\begin{array}{rcl} i(D_k) & = & (i_+(A)+k, \quad i_-(A), \quad i_0(A)), \\ i(E_k) & = & (i_-(A), \quad i_+(A)+k, \quad i_0(A)). \end{array}$$

It follows that for $k = 0, \ldots, n$,

$$\mathrm{sig}(A) = k \iff (B_k, C_k) \in \textsc{Congruence}$$
$$\mathrm{sig}(A) = -k \iff (D_k, E_k) \in \textsc{Congruence}.$$

Therefore, we can write

$$\mathrm{sig}(A) = \sum_{k=1}^{n} k \ \left(f(B_k, C_k) - f(D_k, E_k) \right),$$

thus $\mathrm{sig}(A) \in \mathbf{GapL}$.

To show that *(iii)* implies *(iv)*, we express i_+ in terms of the signature and the rank:

$$2 i_+(A) = \mathrm{rank}(A) + \mathrm{sig}(A).$$

From the outline above, sig $\in$ **GapL** implies rank $\in$ **GapL**, and therefore $\mathbf{C_=L} = \mathbf{SPL}$ by Theorem 3.1. Define sets S_k for $k = 0, 1, \ldots n$,

$$S_k = \{\, A \mid \mathrm{rank}(A) + \mathrm{sig}(A) = 2k \,\}.$$

Sets S_k are in $\mathbf{C_=L}$, and therefore in **SPL**. Hence, for each k, there is a function $f_k \in \mathbf{GapL}$ such that $f_k(A) = 1$, if $A \in S_k$, and $f_k(A) = 0$, otherwise. Then we can write

$$i_+(A) = \sum_{k=1}^{n} k \ f_k(A).$$

We conclude that $i_+ \in \mathbf{GapL}$.

To show that *(iv)* implies *(i)*, assume that $i_+ \in \mathbf{GapL}$. Since this implies that rank $\in$ **GapL**, we have that $\mathbf{C_=L} = \mathbf{SPL}$ by Theorem 3.1. Therefore it suffices to show that PosDeterminant $\in \mathbf{C_=L}$.

For a given matrix A define matrix $B = (\det(A))$ of order 1. We have

$$\det(A) > 0 \iff i_+(B) = 1.$$

We conclude that PosDeterminant $\in \mathbf{C_=L} = \mathbf{SPL}$. □

Like for the rank in Section 3, we consider the weaker condition that we can express i_+ or the signature as a quotient of two **GapL**-functions.

Theorem 4.2 *The following conditions are equivalent.*

(i) $\mathbf{PL} = \mathbf{C_{=}L}$,

(ii) CONGRUENCE $\in \mathbf{C_{=}L}$,

(iii) $\exists g, h \in \mathbf{GapL} \;\; \mathrm{sig} = g/h$,

(iv) $\exists g, h \in \mathbf{GapL} \;\; i_+ = g/h$.

Proof. Conditions *(i)* and *(ii)* are equivalent: recall that CONGRUENCE $\in$ $\mathbf{PL}$. Therefore *(i)* implies *(ii)*. For the reverse direction, we show that POSDETERMINANT $\in \mathbf{C_{=}L}$. For a given matrix A define matrices $B = (\det(A))$ and $C = (1)$ of order 1. Then we have

$$\det(A) > 0 \iff (B, C) \in \text{CONGRUENCE}.$$

We conclude that POSDETERMINANT $\in \mathbf{C_{=}L}$.

Conditions *(iii)* and *(iv)* are equivalent, because we can write $\mathrm{sig}(A) = i_+(A) - i_+(-A)$ and $i_+(A) = (\mathrm{rank}(A) + \mathrm{sig}(A))/2 = (\mathrm{sig}(AA^T) + \mathrm{sig}(A))/2$.

To show that *(i)* implies *(iv)*, assume that $\mathbf{PL} = \mathbf{C_{=}L}$. Note that $\mathbf{PL}$ is closed under complement. It follows that we can verify i_+ in $\mathbf{coC_{=}L}$. That is, there is a function $f \in \mathbf{GapL}$ such that for any symmetric matrix A and any k,

$$i_+(A) = k \iff f(A, k) \neq 0.$$

Define functions $g(A) = \sum_{k=0}^{n} k f(A, k)$ and $h(A) = \sum_{k=0}^{n} f(A, k)$. Then we have $g, h \in \mathbf{GapL}$ and $i_+ = g/h$ as claimed.

To show that *(iv)* implies *(i)*, assume that $i_+ = g/h$ for $g, h \in \mathbf{GapL}$. For any symmetric matrix A and any k, define $f(A, k) = g(A) - k\ h(A)$. Then $f \in \mathbf{GapL}$ and we have

$$i_+(A) = k \iff f(A, k) = 0.$$

That is, we can verify i_+ in $\mathbf{C_{=}L}$. Therefore, $\mathbf{PL} = \mathbf{C_{=}L}$. □

5 Absolute value

For any function f mapping to integers, by $\mathrm{abs}(f)$ we denote the function of absolute values of f. That is

$$\mathrm{abs}(f)(x) = \begin{cases} f(x), & \text{if } f(x) \geq 0, \\ -f(x), & \text{otherwise.} \end{cases}$$

Theorem 5.1 $\mathbf{PL} = \mathbf{SPL} \iff \forall f \in \mathbf{GapL} \;\; \mathrm{abs}(f) \in \mathbf{GapL}$.

Proof. Suppose $\mathbf{PL} = \mathbf{SPL}$ and let $f \in \mathbf{GapL}$. Define the set $S = \{\, x \mid f(x) > 0 \,\}$. By definition $S \in \mathbf{PL}$ and therefore $S \in \mathbf{SPL}$, by assumption. That is, there is $g \in \mathbf{GapL}$ such that

$$g(x) = \begin{cases} 1, & \text{if } x \in S, \\ 0, & \text{otherwise.} \end{cases}$$

Then we can write $\text{abs}(f) = (2g - 1)f$, and therefore $\text{abs}(f) \in \mathbf{GapL}$.

Conversely, let $S \in \mathbf{PL}$. That is, for some function $f \in \mathbf{GapL}$, we can write $S = \{\, x \mid f(x) > 0 \,\}$. We define the following functions

$$\begin{aligned} g =& \text{abs}(f) - \text{abs}(f - 1), \\ h =& \binom{g+1}{2}. \end{aligned}$$

We have $g \in \mathbf{GapL}$, by assumption. It follows that $h \in \mathbf{GapL}$ by the closure properties of $\mathbf{GapL}$. Now observe that

$$h(x) = \begin{cases} 1, & \text{if } f(x) > 0, \\ 0, & \text{otherwise.} \end{cases}$$

This shows that $S \in \mathbf{SPL}$, and therefore $\mathbf{PL} = \mathbf{SPL}$. □

Open Problems

In the polynomial time setting it is known that $\mathbf{PP} \subseteq \mathbf{SPP}^{\mathbf{C_=P}}$. The proof is quite easy:

> Let $A = \{\, x \mid f(x) > 0 \,\} \in \mathbf{PP}$, for some $f \in \mathbf{GapP}$. A nondeterministic machine M on input x guesses $k > 0$ and asks its $\mathbf{C_=P}$-oracle whether $f(x) = k$. If the answer is "yes", then M accepts. If the answer is "no", then M branches ones and accepts on one branch and rejects on the other branch. This shows that $A \in \mathbf{SPP}^{\mathbf{C_=P}}$.

Note that this proof doesn't work in the logspace setting: in the Ruzzo-Simon-Tompa model of space-bounded oracle machines, the machine has to be deterministic while writing a query. Hence we ask

- Is $\mathbf{PL} \subseteq \mathbf{SPL}^{\mathbf{C_=L}}$?

Because $\mathbf{SPP}^{\mathbf{SPP}} = \mathbf{SPP}$, the above inclusion implies that $\mathbf{C_=P} = \mathbf{SPP} \implies \mathbf{PP} = \mathbf{SPP}$. In the logspace setting, we also have $\mathbf{SPL}^{\mathbf{SPL}} = \mathbf{SPL}$ [ARZ99], but the above conclusion is open.

- Does $\mathbf{C_=L} = \mathbf{SPL} \implies \mathbf{PL} = \mathbf{SPL}$?

If the answer is "yes", the conditions in Theorem 3.1 and 4.1 would all be equivalent. In particular, this question is equivalent to finding a reduction from the signature to the rank of a matrix, and the latter functions don't look very different (in complexity).

Acknowledgments

We wish to thank Eric Allender for reading the earlier draft of this paper. He gives us helpful comments on the topic of the paper.

References

[AAM03] E. Allender, V Arvind, and M. Mahajan. Arithmetic complexity, Kleene closure, and formal power series. *Theory of Computing Systems*, 36(4):303–328, 2003.

[ABO99] E. Allender, R. Beals, and M. Ogihara. The complexity of matrix rank and feasible systems of linear equations. *Computational Complexity*, 8:99 –126, 1999.

[AO96] E. Allender and M. Ogihara. Relationship among PL, #L, and the determinant. *RAIRO-Theoretical Informatics and Applications*, 30:1 –21, 1996.

[ARZ99] E. Allender, K. Reinhardt, and S. Zhou. Isolating, matching, and counting: uniform and nonuniform upper bounds. *Journal of Computer and System Sciences*, 59:164–181, 1999.

[Ber84] S. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Information Processing Letters*, 18:147–150, 1984.

[BRS91] R. Beigel, N. Reingold, and D. Spielman. PP is closed under intersection. In *23rd Annual ACM Symposium on the Theory of Computing*, pages 1–9. ACM Press, 1991.

[Dam91] C. Damm. DET = $L^{(\#L)}$. Technical Report Informatik-Preprint 8, Fachbereich Informatik der Humboldt-Universität zu Berlin, 1991.

[FFK94] S. Fenner, L. Fortnow, and S. Kurtz. Gap-definable counting classes. *Journal of Computer and System Sciences*, 48:116–148, 1994.

[FR96] Lance Fortnow and Nick Reingold. PP is closed under truth-table reductions. *Information and Computation*, 124(1):1–6, 1996.

[HO93] L. A. Hemachandra and M. Ogiwara. Is #P closed under subtraction. In G. Rozenberg and A. Salomaa, editors, *Current Trends in Theoretical Computer Science: Essays and Tutorials*, pages 523–536. World Scientific, 1993.

[HT00] T. M. Hoang and T. Thierauf. The complexity of verifying the characteristic polynomial and testing similarity. In *15th IEEE Conference on Computational Complexity (CCC)*, pages 87–95. IEEE Computer Society Press, 2000.

[HT02a] T. M. Hoang and T. Thierauf. The complexity of the inertia. In *22nd Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, Lecture Notes in Computer Science 2556, pages 206–217. Springer-Verlag, 2002.

[HT02b] T. M. Hoang and T. Thierauf. On the minimal polynomial of a matrix. In *8th Annual International Computing & Combinatorics Conference (COCOON)*, Lecture Notes in Computer Science 2387, pages 37–46. Springer-Verlag, 2002.

[OTTW96] M. Ogihara, T. Thierauf, S. Toda, and O. Watanabe. On closure properties of #P in the context of PF∘#P. *Journal of Computer and System Sciences*, 53(2):171–179, 1996.

[Tod91a] S. Toda. Counting problems computationally equivalent to the determinant. Technical Report CSIM 91-07, Dept. of Computer Science and Information Mathematics, University of Electro-Communications, Chofu-shi, Tokyo 182, Japan, 1991.

[Tod91b] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20:865–877, 1991.

[TTW94] T. Thierauf, S. Toda, and O. Watanabe. On closure properties of gapp. *Journal of Computational Complexity*, 4:242–261, 1994.

[Val79a] L. Valiant. Completeness classes in algebra. In *11th Symposium on Theory of Computing (STOC)*, pages 249–261. ACM Press, 1979.

[Val79b] L. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8:189–201, 1979.

[Val92] L. Valiant. Why is boolean complexity theory difficult. In M.S. Paterson, editor, *Boolean Function Complexity*, London Mathematical Society Lecture Notes Series 169. Cambridge University Press, 1992.

[Vin91] V Vinay. Counting auxiliary pushdown automata and semi-unbounded arithmetic circuits. In *6th IEEE Conference on Structure in Complexity Theory*, pages 270–284, 1991.

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'