INFORMATIQUE THÉORIQUE ET APPLICATIONS

CARLO MEREGHETTI
BEATRICE PALANO

Threshold circuits for iterated matrix product and powering

Informatique théorique et applications, tome 34, nº 1 (2000), p. 39-46
<http://www.numdam.org/item?id=ITA_2000__34_1_39_0>

Theoretical Informatics and Applications
Theoret. Informatics Appl. **34** (2000) 39–46

THRESHOLD CIRCUITS FOR ITERATED MATRIX PRODUCT AND POWERING *

CARLO MEREGHETTI[1] AND BEATRICE PALANO[2]

Abstract. The complexity of computing, via threshold circuits, the *iterated product* and *powering* of fixed-dimension $k \times k$ matrices with integer or rational entries is studied. We call these two problems $\mathsf{IMP_k}$ and $\mathsf{MPOW_k}$, respectively, for short. We prove that:

(i) For $k \geq 2$, $\mathsf{IMP_k}$ does not belong to TC^0, unless $\mathrm{TC}^0 = \mathrm{NC}^1$.

(ii) For *stochastic matrices*: $\mathsf{IMP_2}$ belongs to TC^0 while, for $k \geq 3$, $\mathsf{IMP_k}$ does not belong to TC^0, unless $\mathrm{TC}^0 = \mathrm{NC}^1$.

(iii) For any k, $\mathsf{MPOW_k}$ belongs to TC^0.

AMS Subject Classification. 68Q05, 68Q15, 68Q25.

INTRODUCTION

In this work, we study the parallel complexity of performing some matrix operations. As computational model, we use *threshold circuits* [12]. We are interested in solving problems by using threshold circuits of *constant depth.* In this regard, we focus on the class TC^0 [7] of problems solvable by *constant depth* families of (unbounded fan-in) threshold circuits of polynomial size. Several arithmetic and linear algebra operations lie in TC^0: the iterated sum and product of integers and rationals, integer division, matrix multiplication, etc. (see [7,8,16]). We know that TC^0 is contained in NC^1, the class of problems solvable by families of (bounded

Keywords and phrases: Threshold circuits, iterated matrix product, matrix powering.

* *Partially supported by MURST, under the project "Modelli di calcolo innovativi: metodi sintattici e combinatori". Some results in this paper were presented in a preliminary form* [9] *at the* 6th Italian Conference on Theoretical Computer Science (ICTCS 98), *Prato, Italy, November 9–11, 1998.*

[1] Dipartimento di Informatica, Sist. e Com., Università degli Studi di Milano – Bicocca, Via Bicocca degli Arcimboldi 8, 20126 Milano, Italy; e-mail: `mereghetti@disco.unimib.it`

[2] Dipartimento di Informatica, Università degli Studi di Torino, Corso Svizzera 185, 10149 Torino, Italy; e-mail: `beatrice@di.unito.it`

fan-in) AND/OR/NOT-circuits of polynomial size and logarithmic depth [2]. (We do not impose any uniformity condition on circuit families.) It is widely accepted, although still unproved, that $\mathrm{TC}^0 \neq \mathrm{NC}^1$, as the opposite would cause the unlikely collapse of the hierarchy $\{\mathrm{TC}_d^0\}_{d\geq 1}$ (see Sect. 1 for a brief discussion).

The first problem we shall be dealing with is the computation of *the iterated product of fixed-dimension $k \times k$ matrices with integer or rational entries.* We call this problem $\mathsf{IMP_k}$, for short. It can be easily seen that $\mathsf{IMP_k}$ is in NC^1. Here, we investigate whether $\mathsf{IMP_k}$ can be solved in TC^0 as well. By considering the algebraic characterization of regular languages in TC^0 proposed by Barrington *et al.* in [1], we show that: *if $\mathrm{TC}^0 \neq \mathrm{NC}^1$ then, for any $k \geq 2$, $\mathsf{IMP_k}$ does not belong to TC^0.*

We then focus on studying the parallel complexity of $\mathsf{IMP_k}$ on the relevant class of *stochastic matrices.* The interest in such a class of matrices is related to the study of fast parallel algorithms for recognizing *probabilistic languages* [9,10]. We prove a slightly better situation: $\mathsf{IMP_2}$ *for stochastic matrices belongs to* TC^0. On the other hand, *if* $\mathrm{TC}^0 \neq \mathrm{NC}^1$ *then, for any $k \geq 3$,* $\mathsf{IMP_k}$ *for stochastic matrices does not belong to* TC^0.

The second problem we shall consider is *powering fixed-dimension $k \times k$ matrices with integer or rational entries.* We call this problem $\mathsf{MPOW_k}$, for short. By using notions from linear algebra plus fast mathematics on polynomials, we are able to show that, *for any k,* $\mathsf{MPOW_k}$ *belongs to* TC^0. So, in sharp contrast with $\mathsf{IMP_k}$, fixing matrix dimension leads to a full feasibility result for $\mathsf{MPOW_k}$.

The paper is organized as follows: Section 1 contains basic definitions and results concerning circuits and algebraic automata theory. In Section 2, we evaluate the hardness of computing the iterated product of fixed-dimension matrices, both in the case of integer or general rational matrices, and in the particular case of stochastic matrices. In Section 3, we exhibit an algorithm to power in TC^0 fixed-dimension integer or rational matrices.

The results presented here are contained in a preliminary form in [9], where related issues concerning the possibility of accepting in TC^0 regular and probabilistic languages are addressed, too (see also [10] for this latter topic).

1. Preliminaries

We assume some familiarity with the complexity classes defined via traditional and threshold circuits [2,7,12,15]. We recall that TC_d^0 is the class of problems solvable by families of (unbounded fan-in) threshold circuits of polynomial weights and size, and constant depth d. Then, the class $\mathrm{TC}^0 = \bigcup_d \mathrm{TC}_d^0$, introduced in [7], contains the problems solvable by families of threshold circuits of polynomial size and *constant depth.* Typical problems in TC^0 are: the iterated sum of integers (in TC_2^0), integer division (in TC_3^0), iterated product of integers (in TC_4^0), iterated sum and product of rationals, matrix multiplication, and modular arithmetics (the reader is referred to [7,8,16] where he can find a deep study on the exact number of layers in threshold circuits for several tasks).

It is easy to see that $\mathrm{TC}^0 \subseteq \mathrm{NC}^1$, where NC^1 is the class of problems that can be solved in logarithmic depth by families of (bounded fan-in) AND/OR/NOT-circuits of polynomial size [2]. It is still an open problem to decide whether such an inclusion is proper. Indeed, $\mathrm{TC}^0 = \mathrm{NC}^1$ would imply the collapse, from a certain level on, of the hierarchy $\mathrm{TC}^0_1 \subset \mathrm{TC}^0_2 \subset \mathrm{TC}^0_3 \subseteq \mathrm{TC}^0_4 \subseteq \ldots$ (see, *e.g.* [14], Th. IX.1.6) which is a hardly believed event (notice that the first three levels of the hierarchy have already been separated [7]). Thus, it is customarily and reasonably assumed that $\mathrm{TC}^0 \neq \mathrm{NC}^1$.

A problem f is TC^0*-reducible* to a problem g whenever f can be solved by a family of TC^0 circuits with *oracle gates* for g. It is easy to see that $g \in \mathrm{TC}^0$ implies $f \in \mathrm{TC}^0$ as well.

Here, we impose no uniformity condition on our circuit families. As it turns out, this makes no difference to our conclusions (see [1] and [14], Sect. VIII.2 for a discussion).

Let us now briefly review some elementary notions from algebraic automata theory. For more details, we refer the reader to [4, 6]. Given an alphabet Σ, Σ^* denotes the free monoid of all strings on Σ. Given a language $L \subseteq \Sigma^*$, the *syntactic monoid* $\mathscr{M}(L)$ is the quotient monoid $\Sigma^*/\sim_L$, where $\sim_L \subseteq \Sigma^* \times \Sigma^*$ is the congruence defined as: $x \sim_L y$ whenever $vxw \in L$ if and only if $vyw \in L$, for any $v, w \in \Sigma^*$.

A deterministic (this attribute will always be understood) *automaton* $A = (Q, \Sigma, \delta, q_0, F)$ consists of the finite set Q of states, the input alphabet Σ, the initial state q_0, the set $F \subseteq Q$ of final states, and the transition function $\delta : Q \times \Sigma \to Q$ that extends to strings as usual. The *language recognized by* A is the set $L(A) = \{x \in \Sigma^* \mid \delta(q_0, x) \in F\}$.

The *recognizing group-like automaton on a group*[1] $(G, \cdot)$ is defined as $\mathfrak{G}_\mathfrak{i} = (G, G, \cdot, \mathfrak{i}, \{\mathfrak{i}\})$ with $\mathfrak{i}$, the identity of $(G, \cdot)$, being both the initial and the unique final state. It is easy to see the relation

$$\mathscr{M}(L(\mathfrak{G}_\mathfrak{i})) \cong G, \tag{1}$$

where "$\cong$" stands for "isomorphic to".

A seminal result in [1] relates the possibility for a regular language to be recognized in TC^0 with the "group structure" of its syntactic monoid. To see this, we first need some terminology. Given a class $\mathscr{K}$ of algebras, we let the classes: $\mathbf{H}(\mathscr{K})$ of all homomorphic images, and $\mathbf{S}(\mathscr{K})$ of all subalgebras, of algebras in $\mathscr{K}$. A group is called *simple* whenever it has no other normal subgroups but the trivial ones. Thus, we have:

Theorem 1.1. ([1], Th. 8 (b)) *Let L be a regular language. If* $\mathrm{TC}^0 \neq \mathrm{NC}^1$ *then* $L \in \mathrm{TC}^0$ *if and only if each* simple group *in* $\mathbf{HS}(\mathscr{M}(L))$ *is Abelian.*

[1] We will always be considering *finite* groups.

This theorem can be rewritten for recognizing group-like automata as:

Proposition 1.2. *Let $\mathfrak{G}_{\mathfrak{i}}$ be the recognizing group-like automaton on a group $(G,\cdot)$. If* $\mathrm{TC}^0 \neq \mathrm{NC}^1$ *then* $L(\mathfrak{G}_{\mathfrak{i}}) \in \mathrm{TC}^0$ *if and only if each* simple group *in* $\mathbf{HS}(G)$ *is Abelian.*

Proof. Just observe that $\mathscr{M}(L(\mathfrak{G}_{\mathfrak{i}})) \cong G$, as Relation (1) shows. Thus, the claimed result follows at once from Theorem 1.1. □

Proposition 1.2 will be our main tool in the next section, where we inspect the possibility of performing iterated matrix multiplications with constant depth threshold circuits.

2. The complexity of iterated matrix multiplication

We begin by studying the complexity of computing the iterated product of fixed-dimension integer matrices. Formally, this problem can be stated as

- ITERATED $k \times k$ MATRIX PRODUCT (IMP_k)
  INPUT: Integer matrices $M_1, M_2, \ldots, M_n$ of dimension $k \times k$, with n-bit entries.
  OUTPUT: The iterated product $M_1 \cdot M_2 \cdot \ldots \cdot M_n$.

We are going to give a fine – in terms of k – evaluation of the difficulty of IMP_k.

To this purpose, for each prime power $p^m > 3$, consider *the set* $\mathrm{LF}(2,p^m)$ *of* 2×2 *matrices of determinant unity, with entries in the Galois field* $\mathrm{GF}(p^m)$. Moreover, let "$\cdot$" be the usual row-column product with arithmetics performed in $\mathrm{GF}(p^m)$. It is a very well-known fact in group theory (see, *e.g.* [3], Chap. I (Second Part)) that:

Theorem 2.1. $(\mathrm{LF}(2,p^m),\cdot)$ *is a* simple nonabelian group *of order* $\frac{p^m(p^{2m}-1)}{2\,;1}$ *(*$2\,;1$ *depending on* $p>2\,;\,p=2$*).*

Now, let us consider the group $(\mathrm{LF}(2,5),\cdot)$ whose matrices have entries in $\mathrm{GF}(5)$ which actually is $\mathbf{Z}_5$. We can show that

Theorem 2.2. *If* $\mathrm{TC}^0 \neq \mathrm{NC}^1$ *then* IMP_2 *on* $\mathrm{LF}(2,5)$ *does not belong to* TC^0.

Proof. Let $\mathfrak{L}\mathfrak{F}_{\mathfrak{i}}$ be the recognizing group-like automaton on the group $(\mathrm{LF}(2,5),\cdot)$ in which arithmetics is performed "mod 5" ($\mathfrak{i}$ denotes the 2×2 identity matrix). If IMP_2 on $\mathrm{LF}(2,5)$ was in TC^0, then membership in $L(\mathfrak{L}\mathfrak{F}_{\mathfrak{i}})$ could be checked in TC^0 as well.

In fact, to decide whether a string (of $\mathrm{LF}(2,5)$ matrices) $\mu_1\mu_2\cdots\mu_n$ belongs to $L(\mathfrak{L}\mathfrak{F}_{\mathfrak{i}})$, we could compute in TC^0 the iterated product $\mu_1\cdot\mu_2\cdot\ldots\cdot\mu_n$, and accept if and only if the resulting matrix is $\mathfrak{i}$.

So, we would get that $L(\mathfrak{L}\mathfrak{F}_{\mathfrak{i}}) \in \mathrm{TC}^0$ but this, under the assumption $\mathrm{TC}^0 \neq \mathrm{NC}^1$, would contradict Proposition 1.2, since $\mathbf{HS}$(LF(2,5)) contains $\mathrm{LF}(2,5)$ itself which is a *simple nonabelian group*, as pointed out in Theorem 2.1. □

Thus, IMP_2 turns out to be hard for a *finite* group (namely, LF(2, 5) which has exactly 60 elements) of *integer* matrices with very small entries. Hence, *a fortiori, the general* IMP_2 *is hard*, since otherwise we could apply (in TC^0, see Sect. 1) the "mod 5" transformation and solve IMP_2 on LF(2, 5) in TC^0. The unique easy instance of IMP_k is then the trivial IMP_1, *i.e.*, the iterated product of integers which is in TC^0, as observed in Section 1.

A brief remark is in order. It is quite obvious that the complexity analysis so far exhibited would remain unchanged if IMP_k referred to $k \times k$ matrices with *rational* entries expressed as pairs of n-bit integers $\langle$numerator, denominator$\rangle$. Thus, it is fair to use IMP_k even to denote the problem of performing iterated multiplications of fixed-dimension rational matrices.

Summing up, we have that: *if* $\mathrm{TC}^0 \neq \mathrm{NC}^1$ *then* IMP_k *for rational matrices belongs to* TC^0 *if and only if* $k = 1$.

Let us now focus on a relevant subclass of rational matrices: the *stochastic matrices*, *i.e.*, matrices whose entries are rational numbers in the interval $[0, 1]$, and where each row sum equals 1. Our interest in stochastic matrices comes also from the fact that fast algorithms for computing their iterated product would imply fast recognition of *probabilistic languages*, a topic that is investigated in [9, 10].

We soon discover that IMP_k for stochastic matrices turns out to be a slightly more feasible problem. In fact, contrary to Theorem 2.2, we can show that

Theorem 2.3. *For stochastic matrices,* IMP_2 *belongs to* TC^0.

Proof. First, notice that any 2×2 stochastic matrix can be written as $\begin{pmatrix} a & 1-a \\ b & 1-b \end{pmatrix}$, where a and b are rational numbers in $[0, 1]$. Thus, it is not hard to see that the iterated product $P_1 \cdot P_2 \cdot \ldots \cdot P_n$ of 2×2 stochastic matrices, with $P_i = \begin{pmatrix} a_i & 1-a_i \\ b_i & 1-b_i \end{pmatrix}$, yields the 2×2 stochastic matrix $P = \begin{pmatrix} \alpha_1 & 1-\alpha_1 \\ \alpha_2 & 1-\alpha_2 \end{pmatrix}$ where, for $\ell = 1, 2$, we have

$$\alpha_\ell = \sum_{i=1}^{n} \sigma_{i\ell} \prod_{j=i+1}^{n} (a_j - b_j) \quad \text{with} \quad \sigma_{i\ell} = \begin{cases} a_1 & \text{if } i = \ell = 1 \\ b_i & \text{otherwise.} \end{cases}$$

We stipulate that the inner product yields 1 whenever the lower index exceedes the upper one. Hence, computing the entries of P reduces to sum a linear amount of products each one involving a linear amount of rationals. All this can be done in TC^0, as seen in Section 1. □

Unfortunately, this is the only case of feasible iterated product for stochastic matrices, as witnessed by the following:

Theorem 2.4. *There exists a finite set* $\mathscr{B}$ *of* 3×3 *stochastic matrices for which the iterated product does not belong to* TC^0, *unless* $\mathrm{TC}^0 = \mathrm{NC}^1$.

Proof. We are to show that IMP_2 on the *finite* group LF(2, 5) is TC^0-reducible to the iterated product on a set $\mathscr{B}$ of 3×3 stochastic matrices, having the same cardinality as LF(2, 5). Then, by Theorem 2.2, we get the claimed result.

We make use of the following transformation Γ, easily seen to be implemented in TC^0: let $P = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ be a 2×2 matrix in LF(2, 5), and hence with entries in $\mathbf{Z}_5$; we define the 3×3 matrix

$$\Gamma(P) = \begin{pmatrix} \frac{a_{11}}{2^3} & \frac{a_{12}}{2^3} & 1 - \frac{a_{11}+a_{12}}{2^3} \\ \frac{a_{21}}{2^3} & \frac{a_{22}}{2^3} & 1 - \frac{a_{21}+a_{22}}{2^3} \\ 0 & 0 & 1 \end{pmatrix}.$$

Next, we let $\mathscr{B} = \{\Gamma(P) \mid P \in \mathrm{LF}(2,5)\}$. Obviously, $\mathscr{B}$ is a set of 3×3 stochastic matrices, with the same cardinality as LF(2, 5). At this point, it is easy to see that, for any given n-tuple $P_1, P_2, \ldots, P_n$ of matrices in LF(2, 5), we have

$$\Gamma(P_1) \cdot \Gamma(P_2) \cdot \ldots \cdot \Gamma(P_n) = \begin{pmatrix} \multicolumn{2}{c}{\multirow{2}{*}{$\frac{P_1 \cdot P_2 \cdot \ldots \cdot P_n}{2^{3n}}$}} & 1 - \frac{\alpha_1}{2^{3n}} \\ & & 1 - \frac{\alpha_2}{2^{3n}} \\ 0 & 0 & 1 \end{pmatrix},$$

which is a 3×3 stochastic matrix, where α_1 (resp. α_2) equals the sum of the entries in the first (resp. second) row of $P_1 \cdot P_2 \cdot \ldots \cdot P_n$. Hence, to compute $P_1 \cdot P_2 \cdot \ldots \cdot P_n$ in LF(2, 5), we first compute $\Gamma(P_i)$ in TC^0, and then use an oracle to evaluate the iterated product $\Gamma(P_1) \cdot \Gamma(P_2) \cdot \ldots \cdot \Gamma(P_n)$. Finally, we read off $P_1 \cdot P_2 \cdot \ldots \cdot P_n$ from the resulting matrix, and transform (in TC^0) the entries "mod 5". □

Indeed, the hardness of IMP_3 on $\mathscr{B}$ implies *the hardness of* IMP_k *for general stochastic matrices, for any* $k \geq 3$.

3. The complexity of matrix powering

Let us now turn to study the parallel complexity of powering fixed-dimension integer matrices. The problem formalizes as:

- $k \times k$ MATRIX POWERING (MPOW_k)
  INPUT: An integer matrix M of dimension $k \times k$, with n-bit entries.
  OUTPUT: The n-th power M^n.

We are going to show that MPOW_k *is in* TC^0, *for any* k.

To this aim, we need to recall a few elementary notions from linear algebra (see, *e.g.* [13]). Let M be a $k \times k$ integer matrix. Its *characteristic polynomial* is defined as $p_M(x) = \det(M - xI) = (-1)^k x^k + c_{k-1}x^{k-1} + \cdots + c_1 x + c_0$, where I is the $k \times k$ identity matrix, while each c_i is known to be the sum of all the principal minors of order $(k-i)$, taken with the sign $(-1)^i$. The *Cayley-Hamilton theorem* states that

$$p_M(M) = \mathbf{0}, \tag{2}$$

with $\mathbf{0}$ being the $k \times k$ zero matrix. Let us see how to use this fact to efficiently compute M^n. If we divide x^n by $p_M(x)$, we obtain the equation

$$x^n = q(x)\, p_M(x) + r_M(x), \tag{3}$$

where the *remainder* $r_M(x)$ *is a polynomial of degree not exceeding* $k-1$. Evaluating equation (3) in M yields $M^n = q(M)\, p_M(M) + r_M(M)$ and, by equation (2), we get

$$M^n = r_M(M).$$

This leads to the following algorithm to compute M^n:

(i) compute $p_M(x)$;
(ii) compute $r_M(x) = x^n \bmod p_M(x)$;
(iii) evaluate $r_M(M)$.

Such an algorithm can be implemented in TC^0.

Here, we will not examine the technical details of the implementation for which we refer the reader to [11] (where the number of neuron layers is also investigated). However, we would briefly argue that each step of the algorithm is in TC^0.

STEP (i): To get the i-th coefficient of $p_M(x)$, for $0 \le i \le k-1$, we basically have to sum $\binom{k}{k-i}$ determinants of $(k-i) \times (k-i)$ submatrices of M; this can be clearly done in constant depth. Hence, by computing in parallel all such coefficients, we obtain $p_M(x)$ in TC^0.

STEP (ii): We can refer to fast parallel algorithms for dividing polynomials presented, *e.g.*, in [5]. This would suffice to show that $r_M(x)$ can be computed in TC^0. In [11], we have preferred to suitably transform polynomials into integers, and then to operate with such integers.

STEP (iii): The polynomial $r_M(x)$ has degree at most $k-1$. Hence, computing $r_M(M)$ amounts to computing a linear combination of powers M^i, with $i \le k-1$. Even this task is easily seen to be in TC^0.

Thus, we can conclude that:

Theorem 3.1. *For any k, MPOW_k belongs to* TC^0.

We wish to thank Alberto Bertoni, Giovanni Pighizzini, and Sebastiano “seba” Vigna for stimulating discussions. Ingo Wegener is also greatly acknowledged for his continuous encouragement, very helpful remarks, and for pointing out reference [1]. Finally, our thanks go to anonymous referees for their valuable comments.

REFERENCES

[1] D.A. Mix Barrington, K. Compton, H. Straubing and D. Thérien, Regular languages in NC^1. *J. Comp. System Sci.* **44** (1992) 478–499.

[2] S.A. Cook, A taxonomy of problems with fast parallel algorithms. *Inform. and Control.* **64** (1985) 2–22.

[3] L.E. Dickson, *Linear Groups with an Exposition of the Galois Field Theory.* 1901. Reprinted by Dover (1958).

[4] S. Eilenberg, *Automata, Languages, and Machines.* Academic Press (1976).

[5] W. Eberly, Very fast parallel polynomial arithmetic. *SIAM J. Comput.* **18** (1989) 955–976.

[6] F. Gécseg, *Products of Automata.* Springer Verlag, *Monogr. Theoret. Comput. Sci. EATCS Ser.* **7** (1986).

[7] A. Hajnal, W. Maaß, P. Pudlák, M. Szegedy and G. Turán, Threshold circuits of bounded depth, in *Proc. 28th IEEE Symposium on Foundations of Computer Science* (1987) 99–110. Also in *J. Comp. System Sci.* **46** (1993) 129–154.

[8] T. Hofmeister, Depth–efficient threshold circuits for arithmetic functions, edited by V. Roychowdhury, K.-Y. Siu and A. Orlitsky, *Theoretical Advances in Neural Computation and Learning.* Kluwer Academic (1994) 37–84.

[9] C. Mereghetti and B. Palano, Threshold circuits for some matrix operations. Consequences on regular and probabilistic languages. *Theoretical Computer Science – Proceedings of the 6th Italian Conference.* World Scientific (1998) 216–227.

[10] C. Mereghetti and B. Palano, The Parallel Complexity of Deterministic and Probabilistic Automata. Technical Report No. 242-99, Dipartimento di Scienze dell'Informazione. Università degli Studi di Milano (1999). Available at `http://gongolo.usr.dsi.unimi.it/~mereghc/papers/TR_242-99.ps`

[11] C. Mereghetti and B. Palano, Matrix Powering in Constant Depth. Technical Report No. 245-00, Dipartimento di Scienze dell'Informazione. Università degli Studi di Milano (2000). Available at `http://gongolo.usr.dsi.unimi.it/~mereghc/papers/TR_245-00.ps`

[12] V. Roychowdhury, K.-Y. Siu and A. Orlitsky, Neural models and spectral methods, edited by V. Roychowdhury, K.-Y. Siu and A. Orlitsky, *Theoretical Advances in Neural Computation and Learning.* Kluwer Academic (1994) 3–36.

[13] G. Shilov, *Linear Algebra.* Prentice Hall (1971). Reprinted by Dover (1977).

[14] H. Straubing, *Finite Automata, Formal Logic, and Circuit Complexity.* Birkhäuser (1994).

[15] I. Wegener, *The Complexity of Boolean Functions.* Teubner (1987).

[16] I. Wegener, Optimal lower bounds on the depth of polynomial–size threshold circuits for some arithmetic functions. *Inform. Process. Lett.* **46** (1993) 85–87.

Communicated by I. Wegener.
Received January 20, 1999. Accepted March 13, 2000.