Theoretical Computer Science 2 (1976) 305–315.

A LOWER BOUND ON THE NUMBER OF ADDITIONS IN MONOTONE COMPUTATIONS

C. P. SCHNORR

Fachbereich Mathematik, Universität Frankfurt, Frankfurt, Federal Republic of Germany

Communicated by Maurice Nivat
Received October 1974
Revised August 1975

Abstract. A computation of rational polynomials that only uses variables, positive rational numbers and the operations addition and multiplication is called a monotone, rational computation. We prove a general lower bound on the minimal number of additions in monotone rational computations. This lower bound implies that any monotone rational computation of the nth degree convolution at least requires n^2-2n+1 additions. $\binom{n}{k}-1$ is the minimal number of additions in any monotone computation of the polynomial that is asociated with the k-clique problem for graphs with n nodes.

1. Introduction

There is an increasing interest in the minimal cost of computations of polynomials and sets of polynomials. Up to now very little is known of these minimal costs even for fundamental problems such as matrix multiplication and Boolean convolution. Here we consider a more restrictive problem: The minimal costs of monotone, rational computations. In monotone, rational computations we only allow the operations addition and multiplication and we start from variables and positive rational numbers. A possible motivation for this decisive restriction are the favorable stability properties of monotone rational computations with respect to rounding errors in computers. An operation $+,\cdot$ that is applied on positive values can at most add the relative errors of the operands. Therefore, the monotonicity of the computation eliminates the possibility that small relative errors may produce high relative errors by subtracting two numbers that are approximately equal.

Let $L_+(f)$ be the minimal number of additions in any monotone, rational computation for the monotone polynomial f. We prove a general lower bound on L_+ which is sharp in a number of cases. For instance, this bound implies that n^2-2n+1 additions are necessary in any monotone, rational computation of nth degree convolution, n^3-n^2 additions are necessary in any monotone, rational computation of nth degree matrix product; this also follows from a more powerful recent result of M. Paterson concerning the Boolean matrix product.

We shall consider the rational polynomials $\mathrm{CL}_{n,k}$ that describe the k-clique problem for graphs with n nodes. We prove $L_+(\mathrm{CL}_{n,k})=\binom{n}{k}-1$. This shows that our method yields exponentially lower bounds such as $L_+(\mathrm{CL}_{2n,n}) = \binom{2n}{n} - 1 \approx 2^{\sqrt{n'}}$ for polynomials with $2n$ variables.

Let $\alpha(\mathrm{CL}_{2n,n})$ be the Boolean polynomial which corresponds to $\mathrm{CL}_{2n,n}$. If we could prove that the network complexity of $\alpha(\mathrm{CL}_{2n,n})$ increases faster than n^k for every fixed k then this would solve the famous P = NP? problem in the sense that P ≠ NP. Hereby P and NP are the classes of decision problems that are solvable on deterministic, resp. non-deterministic Turing machines in polynomial time (see [1, 3]). Observe that the clique-problem is in NP and that the network complexity of the Boolean function f yields a lower bound on the running time of every Turing program for f (see [2, 5]).

This raises the question on the size of the gap between $L_+(f)$ and the network complexity of the corresponding Boolean polynomial $\alpha(f)$. How much can Boolean identities help in Boolean computations? It is an open problem whether our lower bound also holds for monotone Boolean computations. Obviously our lower bound does not hold for general rational computations and general Boolean computations (i.e. logical networks). This follows from well-known fast computations for the matrix product and for convolution. But we do not know how much subtraction can help in rational computations of monotone polynomials.

2. Monotone computations of monotone rational polynomials

Let $\mathbf{Q}$ be the field of rational numbers. Let $V = \{x_i \mid i \in \mathrm{N}\}$ be a countable set of rational variables. Let Ω be the set of all polynomials with coefficients in $\mathbf{Q}$ and variables in V.

$f \in \mathbf{Q}$ is called totally *monotone* iff all coefficients of f are positive rational numbers. Let $\Omega_+ \subset \Omega$ be the set of all totally monotone, rational polynomials. Let $\mathbf{Q}_+ \subset \mathbf{Q}$ be the set of all positive rational numbers.

The following operations are used in monotone computations:

(1) all positive rationals in $\mathbf{Q}_+$,

(2) all variables in V,

(3) addition $+$ and multiplication $\cdot$ applied on functions.

A monotone computation is a finite, directed, acyclic labelled graph β such that

(1) Each node ν is labelled with some operation in $\mathbf{Q}_+ \cup V \cup \{\cdot, +\}$.

(2) If ν is labelled with $+$ or $\cdot$ then ν has exactly two entering edges that correspond to the entries of $+$ and $\cdot$, resp.

(3) If ν is labelled with a constant or a variable then ν has no entering edge. In this case ν is called an entry of β.

In an obvious way β associates with every node ν a rational polynomial $\mathrm{res}_{\beta,\nu} \in \Omega_+$ that is obtained by applying the operation of node ν to the results of the directly preceding nodes.

Let $F \subset \Omega_+$ then we say "β computes F" iff "$\forall f \in F$: $\exists$ node ν in β: $\mathrm{res}_{\beta,\nu} = f$".

Obviously the monotone rational computations exactly compute all totally monotone rational polynomials. For $F \subset \Omega_+$ let $L_+(F)$ be the minimal number of additions in any monotone computation that computes F.

$f \in \Omega$ is called a *monomial* if either f is the constant 1 or if f is a product of variables. Let $\text{mon} \subset \Omega_+$ be the set of all monomials.

With $f \in \Omega_+$ we associate a set $\text{mon}(f) \subset \text{mon}$ of monomials by requiring $\exists r$: $\text{mon}(f) \to \mathbb{Q}_+$ such that

$$f = \sum_{m \in \text{mon}(f)} r(m)m.$$

With $f \in \Omega$ we associate the set $V(f) \subset V$ of variables of f.

Our method for proving lower bounds is based on the following theorem that describes the method of inductive substitution.

Theorem 2.1. *Every function* $\# : \Omega_+ \to \mathbb{N} \cup \{\infty\}$ *that satisfies* (1)–(5) *is a lower bound on* L_+, *i.e.* $\forall f \in \Omega_+$: $\#(f) \leq L_+(f)$.

(1) $\forall x_i \in V$: $\#(x_i) = 0$,

(2) $\bar{f} = f_{x_i := x_\nu + x_\mu}$ *and* $x_\nu, x_\mu \notin V(f)$ *implies* $\#(\bar{f}) \leq \#(f) + 1$,

(3) $\bar{f} = f_{x_i := x_\nu \cdot x_\mu}$ *and* $x_\nu, x_\mu \notin V(f)$ *implies* $\#(\bar{f}) \leq \#(f)$,

(4) $\bar{f} = f_{x_\nu := x_\mu}$ *implies* $\#(\bar{f}) \leq \#(f)$,

(5) $\bar{f} = f_{x_\nu := q}$ *and* $q \in \mathbb{Q}_+$ *implies* $\#(\bar{f}) \leq \#(f)$.

Clauses (2)–(5) describe a number of substitution steps. In each of these clauses $\bar{f}$ is obtained from f by substituting a new rational function for (each occurrence) of some variable of f.

Proof. Let β be any computation of f, i.e. $\text{res}_{\beta,r} = f$. By inserting one additional operation, addition or multiplication at an entry of β we obtain a new computation $\bar{\beta}$ that computes $\bar{f}$. $\bar{f}$ can be obtained from f by one substitution step of either clause (2) or clause (3) and two following substitution steps of clause (4) and clause (5) that identify the new variables x_ν, x_μ (occurring in clauses (2), (3)) with some old variable of f or with a rational constant $q \in \mathbb{Q}_+$. In the case that an addition $+$ is inserted at an entry of β clauses (2), (4), (5) imply $\#(\bar{f}) \leq \#(f) + 1$. In the case that a multiplication $\cdot$ is inserted clauses (3), (4), (5) imply $\#(\bar{f}) \leq \#(f)$.

Since every computation is obtained from an initial computation that consists only of variables and constants by successively inserting additions and multiplications at an entry of the preceding computation it follows by induction on the number of arithmetical operations in β that $\#(\text{res}_{\beta,r}) \leq$ "number of additions in β". □

3. A general lower bound on the number of additions in monotone, rational computations

At the first glance we might think that those polynomials $f \in \Omega_1$ are hard to compute which consist of a large set $\text{mon}(f)$ of monomials. However, this idea fails since one additional multiplication can increase the number of monomials considerably. For instance a multiplication of two sums $f = \sum_{i=1}^{n} a_i$ and $g = \sum_{j=1}^{n} b_j$ with $2n$ monomials a_i and b_j yields a product

$$f \cdot g = \sum_{1 \leq i,j \leq n} a_i b_j$$

with n^2 monomials $a_i b_j$. However, in this case a characteristic relation holds for the monomials of $f \cdot g$. This relation can be described by using the following ordering relation $\leq$ on mon:

$$s \leq t \Leftrightarrow \exists r \in \text{mon}: s = t \cdot r.$$

Then the following relation holds for different monomials $a_i b_j$, $a_i b_\mu$ and $a_\nu b_j$ of $f \cdot g$:

$$a_i b_j \geq a_i b_\mu\, a_\nu b_j \wedge a_i b_j \neq a_i b_\mu \wedge a_i b_\mu \neq a_\nu b_j.$$

The following concept of a separated subset $B \subset \text{mon}(f)$ will exclude this type of relation.

Definition 3.1. Let $f \in \Omega_+$ then $B \subset \text{mon}(f)$ is called *separated* iff

$$\forall r \in \text{mon}(f): \forall s, t \in B: r \geq s \cdot t \Rightarrow [r = s \quad \text{or} \quad r = t].$$

Our object is to prove that every separated subset $B \subset \text{mon}(f)$ implies $\|B\| - 1 \leq L_+(f)$. This means that the lower bound

$$\bar{\#}(f) = \max\{\|B\| - 1: B \subset \text{mon}(f) \text{ is separated}\}$$

measures the power of addition in monotone computations. There are two features that are expressed by large separated subsets $B \subset \text{mon}(f)$: (1) there exist many monomials in mon(f), and (2) the separatedness condition eliminates many monomials from being in mon(f).

For technical reasons we shall first generalise the bound $\bar{\#}$. This will simplify our proofs. Let $f \in \Omega_+$ and let σ be a map $\sigma: V(f) \rightarrow \text{mon}$. Let $f^\sigma = f_{(x_j := \sigma(x_j) | x_j \in V(f))} \in \Omega_+$. f^σ is obtained from f by substituting the monomials $\sigma(x_j)$ for the variables x_i of f, f^σ is called a substitution function of f. Let Sub(f) be the set of all substitution functions of f. We set

$$\#(f) = \max\{\bar{\#}(f^\sigma): f^\sigma \in \text{Sub}(f)\}.$$

Main Theorem 3.2. $\forall f \in \Omega_+$: $\#(f) \leq L_+(f)$.

The following lemma describes the behaviour of monomials in our substitution steps.

Lemma 3.3.

(1) $$\text{mon}(f^\sigma) = \bigcup_{t \in \text{mon}(f)} \text{mon}(t^\sigma),$$

(2) $$\mathrm{mon}(f_{x_i:=x_\nu+x_\mu}) = \{bx_\nu^r x_\mu^s \mid bx_i^k \in \mathrm{mon}(f), x_i \notin V(b), r+s=k \in \mathbb{N}\}.$$

Proof. Observe that $t \in \mathrm{mon}(f) \Rightarrow t^\sigma \in \mathrm{mon}(f^\sigma)$,

$$\mathrm{mon}(f_1+f_2) = \mathrm{mon}(f_1) \cup \mathrm{mon}(f_2),$$
$$\mathrm{mon}(f_1 \cdot f_2) = \mathrm{mon}(f_1) \cdot \mathrm{mon}(f_2). \quad \square$$

Proof of Theorem 3.2. We apply Theorem 2.1 and prove that $\#$ satisfies (1)–(5) in 2.1.

(1) Let x_i be any variable. Then $\mathrm{mon}(x_i^\sigma) = \{\sigma(x_i)\}$. Hence $\|\mathrm{mon}(x_i^\sigma)\| = 1$. Thus $\#(x_i) = 0$.

(2) Let $\bar{f} = f_{x_i:=x_\nu+x_\mu}$ and $x_\nu, x_\mu \notin V(f)$. Let $\bar{f}^{\bar{\sigma}} \in \mathrm{Sub}(\bar{f})$, let $\bar{B} \subset \mathrm{mon}(\bar{f}^{\bar{\sigma}})$ be separated such that $\|\bar{B}\| = \#(\bar{f})+1$. Then we construct a corresponding $f^\sigma \in \mathrm{Sub}(f)$ and a separated subset $B \subset \mathrm{mon}(f^\sigma)$ such that $\|B\| \geq \|\bar{B}\| - 1$. We define $\sigma_\tau : V(f) \to \overline{\mathrm{mon}}$ as follows:

$$\sigma_\tau(x_j) = \begin{cases} \bar{\sigma}(x_j), & x_j \neq x_i, \\ \bar{\sigma}(x_\tau), & x_j = x_i, \end{cases} \quad \text{for} \quad \tau = \nu, \mu. \quad \square$$

Lemma 3.4. *Suppose* $tx_\nu \in \mathrm{mon}(\bar{f})$, $(tx_\nu)^{\bar{\sigma}} \in \bar{B} - \mathrm{mon}(f^{\sigma_\mu})$ *and* $(sx_\mu)^{\bar{\sigma}} \in \bar{B}$. *This implies* $(tx_\mu)^{\bar{\sigma}} = (sx_\mu)^{\bar{\sigma}}$.

Proof. Obviously $(tx_\mu)^{\bar{\sigma}} \in \mathrm{mon}(\bar{f}^{\bar{\sigma}})$ and $(tx_\mu)^{\bar{\sigma}} \geq (tx_\nu)^{\bar{\sigma}}(sx_\mu)^{\bar{\sigma}}$. Since $\bar{B} \subset \mathrm{mon}(\bar{f}^{\bar{\sigma}})$ is separated, it follows that

$$(tx_\mu)^{\bar{\sigma}} = (tx_\nu)^{\bar{\sigma}} \quad \text{or} \quad (tx_\mu)^{\bar{\sigma}} = (sx_\mu)^{\bar{\sigma}}.$$

However $(tx_\mu)^{\bar{\sigma}} = (tx_\nu)^{\bar{\sigma}}$ implies $\bar{\sigma}(x_\nu) = \bar{\sigma}(x_\mu)$ and therefore yields a contradiction:

$$(tx_\nu)^{\bar{\sigma}} \in \mathrm{mon}(f^{\sigma_\mu}).$$

This proves $(tx_\mu)^{\bar{\sigma}} = (sx_\mu)^{\bar{\sigma}}$. $\square$

Lemma 3.5. *Suppose* $tx_\nu \in \mathrm{mon}(\bar{f})$, $(tx_\nu)^{\bar{\sigma}} \in \bar{B} - \mathrm{mon}(f^{\sigma_\mu})$ *and* $(s_1x_\mu)^{\bar{\sigma}}, (s_2x_\mu)^{\bar{\sigma}} \in \bar{B}$. *This implies* $(s_1x_\mu)^{\bar{\sigma}} = (s_2x_\mu)^{\bar{\sigma}}$.

Proof. It follows from 3.4 that

$$(s_1x_\mu)^{\bar{\sigma}} = (tx_\mu)^{\bar{\sigma}} = (s_2x_\mu)^{\bar{\sigma}}. \quad \square$$

Lemma 3.6. *Either* (1) *or* (2) *or* (3) *holds.*

(1) $\bar{B} \subset \mathrm{mon}(f^{\sigma_\nu})$,

(2) $\bar{B} \subset \mathrm{mon}(f^{\sigma_\mu})$,

(3) $\|\bar{B} \cap \mathrm{mon}(f^{\sigma_\nu})\| = \|\bar{B}\| - 1$.

Proof. Suppose $\neg(1) \wedge \neg(2)$. For every $g \in \bar{B} - \mathrm{mon}(f^{\sigma_\nu})$ there exists $(sx_\nu)^{\bar\sigma} \in \mathrm{mon}(f^{\bar\sigma})$ such that $g = (sx_\mu)^{\bar\sigma}$. Therefore, 3.5 implies that g is uniquely determined. This proves 3.6. $\square$

Set $B_\tau = \bar{B} \cap \mathrm{mon}(f^{\sigma_\tau})$ for $\tau = \nu, \mu$. Then $B_\tau \subset \mathrm{mon}(f^{\sigma_\tau})$ is separated for $\tau = \nu, \mu$. Therefore, 3.6 implies $\#(f^{\sigma_\nu}) \geq \#(\bar{f}^{\bar\sigma}) - 1$ or $\#(f^{\sigma_\mu}) \geq \#(\bar{f}^{\bar\sigma}) - 1$. Hence $\#(f) \geq \#(\bar{f}) - 1$.

(3) Let $\bar{f} = f_{x_i := x_\nu \cdot x_\mu}$ and $x_\nu, x_\mu \notin V(f)$. This implies $\bar{f} \in \mathrm{Sub}(f)$ and $\mathrm{Sub}(\bar{f}) \subset \mathrm{Sub}(f)$. Hence $\#(\bar{f}) \leq \#(f)$.

(4) Let $\bar{f} = f_{x_\nu := x_\mu}$. This implies $\bar{f} \in \mathrm{Sub}(f)$ and $\mathrm{Sub}(\bar{f}) \subset \mathrm{Sub}(f)$. Hence $\#(\bar{f}) \leq \#(f)$.

(5) Let $\bar{f} = f_{x_\nu := r}$ and $r \in \mathbf{Q}_+$. Suppose $\#(\bar{f}) = \bar{\#}(\bar{f}^{\bar\sigma})$. Then $g := (f_{x_\nu := 1})^{\bar\sigma} \in \mathrm{Sub}(f)$ and $\mathrm{mon}(\bar{f}^{\bar\sigma}) = \mathrm{mon}(g)$. Hence $\#(\bar{f}) \leq \bar{\#}(g) \leq \#(f)$.

4. Applications of the main theorem

Our first example is n-degree convolution C_n. Let $a_0, a_1, \ldots, a_{n-1}, b_0, b_1, \ldots, b_{n-1}$ be $2n$ different variables. Let C_n be given as follows:

$$c_k = \sum_{\nu+\mu=k} a_\nu b_\mu, \qquad k = 0, 1, \ldots, 2n-2;$$

$$\mathrm{C}_n = \{c_k \mid k = 0, 1, \ldots, 2n-2\}.$$

We first consider the "sum–function" of C_n. Let $f_1, \ldots, f_m$ be m polynomials and let $z_1, \ldots, z_m$ be variables such that $\forall i, j: z_i \notin V(f_j)$. Then

$$\sum_{i=1}^{m} z_i f_i$$

is called the sum-function of $f_1, \ldots, f_m$. Let

$$\mathrm{SC}_n = \sum_{k=0}^{2n-2} z_k \sum_{\nu+\mu=k} a_\nu b_\mu$$

be the sum-function of n-degree convolution C_n.

Theorem 4.1. *Evey monotone, rational computation of* SC_n *requires* $n^2 - 1$ *additions; moreover* $L_+(\mathrm{SC}_n) = n^2 - 1$.

Proof. We prove that $\mathrm{mon}(\mathrm{SC}_n) \subset \mathrm{mon}(\mathrm{SC}_n)$ is separated. Let $z_k a_\nu b_{k-\nu}, z_{\bar k} a_{\bar\nu} b_{\bar k - \bar\nu} \in \mathrm{mon}(\mathrm{SC}_n)$ and suppose that

$$z_{\bar k} a_{\bar\nu} b_{\bar k - \bar\nu} \geq z_k a_\nu b_{k-\nu} \angle_{\bar k} a_{\bar\nu} b_{\bar k - \bar\nu}.$$

Then obviously either (i) or (ii) holds:

(i) two variables of $z_{\bar k} a_{\bar\nu} b_{\bar k - \bar\nu}$ are variables of $z_k a_\nu b_{k-\nu}$,

(ii) two variables of $z_{\bar k} a_{\bar\nu} b_{\bar k - \bar\nu}$ are variables of $z_{\bar k} a_{\bar\nu} b_{\bar k - \bar\nu}$.

However, every monomial $z_{\bar{k}}a_{\bar{\nu}}b_{\bar{k}-\bar{\nu}}$ of SC_n is uniquely determined by any choice of two of its variables. This implies that

$$z_{\bar{k}}a_{\bar{\nu}}b_{\bar{k}-\bar{\nu}} = z_k a_\nu b_{k-\nu} \quad \text{or} \quad z_{\bar{k}}a_{\bar{\nu}}b_{\bar{k}-\bar{\nu}} = z_{\bar{k}}a_{\bar{\nu}}b_{\bar{k}-\bar{\nu}}.$$

This proves that mon(SC_n) is separated. Therefore our main theorem yields $L_+(SC_n) \geq \|\text{mon}(SC_n)\| - 1 = n^2 - 1$. On the other hand the standard monotone computation of SC_n only needs $n^2 - 1$ additions. □

Corollary 4.2. *The minimal number of additions for monotone, rational computations of n-degree convolution* C_n *is* $n^2 - 2n + 1$.

Proof. In order to compute SC_n from C_n we need at most $2n - 2$ additions that sum all the $2n - 1$ monomials $z_k c_k$, $k = 0, 1, \ldots, 2n - 2$. This proves

$$L_+(SC_n) \leq L_+(C_n) + 2n - 2.$$

Hence

$$L_+(C_n) \geq n^2 - 2n + 1.$$

In this case, too, the standard monotone computation of C_n achieves this bound. □

Our method also applies to matrix multiplication. The following theorem also follows from a recent result of Paterson [2].

Theorem 4.3. *The minimal number of additions for monotone, rational computations of* (n, n)*-matrix product is* $n^3 - n^2$.

Proof. Let $a_{i,k}, b_{i,k}, 1 \leq i, k \leq n$, be the $2n^2$ variables for two (n, n)-matrices. Let the matrix product M_n be given as follows:

$$c_{i,k} = \sum_{j=1}^{n} a_{i,j} b_{j,k},$$

$$M_n = \{c_{i,k} \mid 1 \leq i, k \leq n\}.$$

Let

$$SM_n = \sum_{1 \leq i,k \leq n} z_{i,k} c_{i,k}$$

be the sum-function of the matrix product M_n. The $z_{i,k}$ are n^2 additional variables. We claim that mon(SC_n) is separated. This is proved as in the proof of 5.1. For every monomial, SC_n is uniquely determined by an arbitrary choice of two of its variables. Therefore, our main theorem implies $L_+(SM_n) \geq n^3 - 1$. This clearly proves $L_+(M_n) \geq n^3 - n^2$. Note that only $n^2 - 1$ additions are required in order to compute SM_n from M. These lower bounds both are achieved by the standard monotone computations. □

Finally we give an example of an exponentially increasing lower bound for a sequence f_n of single polynomials. These polynomials f_n are associated with the clique problem that is known to be polynomial complete in NP see [1, 3]).

Let $a_{i,j}$, $1 \leq i, j \leq n$, be n^2 variables. Every binary choice of value $c_{i,j} \in \{0, 1\}$ for these variables is the representation of a directed graph with nodes $1, 2, \ldots, n$. $c_{i,j} = 1$ means that there is an edge from node i to node j.

A k-clique in $(c_{i,j})$ is a complete subgraph with k nodes. Thus a k-clique in $(c_{i,j})$ is a (k, k) submatrix with equal row and column indices that only consists of 1's. Hence every k-clique in $(c_{i,j})$ is given by k indices $1 \leq \nu_1 < \nu_2 < \ldots < \nu_k \leq n$ such that

$$c_{\nu_i, \nu_j} = 1 \quad \text{for} \quad 1 \leq i, j \leq k.$$

The clique problem is the problem of deciding whether there exists a k-clique in $(c_{i,j})$. This problem is represented by the following monotone function

$$\mathrm{CL}_{n,k} = \sum_{1 \leq \nu_1 < \nu_2 < \ldots < \nu_k \leq n} \; \prod_{1 \leq i,j \leq k} a_{\nu_i, \nu_j}.$$

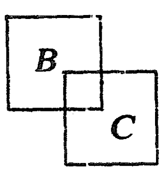
This means for binary inputs $c = (c_{i,j})$ we have $\mathrm{CL}_{n,k}(c) > 0$ iff the graph that is associated with c has an k-clique.

Theorem 4.4. *The minimal number of additions in any monotone, rational computation of* $\mathrm{CL}_{n,k}$ *is* $\binom{n}{k} - 1$.

Proof. It satisfies to prove that $\mathrm{mon}(\mathrm{CL}_{n,k}) \subset \mathrm{mon}(\mathrm{CL}_{n,k})$ is separated. Observe that $\|\mathrm{mon}(\mathrm{CL}_{n,k})\| = \binom{n}{k}$. The separatedness of $\mathrm{mon}(\mathrm{CL}_{n,k})$ immediately follows from the following:

Fact. *Let A, B, C be any three (k, k)-submatrices of an (n, n)-matrix. Suppose that the set of row indices and the set of column indices coincides for each of the matrices A, B, C. Let the set of positions $\mathrm{P}(A)$ of A be contained in the union of the corresponding sets $\mathrm{P}(B)$ and $\mathrm{P}(C)$. Then it follows that $A = B$ or $A = C$.*

The following picture illustrates this fact:



This fact and our main theorem implies $L_+(\mathrm{CL}_{n,k}) \geq \binom{n}{k} - 1$. Obviously the standard monotone computation of $\mathrm{CL}_{n,k}$ achieves this bound.

Observe that Theorem 4.4 yields an exponentially increasing lower bound.

$$L_+(\mathrm{CL}_{2n,n}) = \binom{2n}{n} - 1 \geqslant 2^{2n}/2n$$

for the polynomials $\mathrm{CL}_{2n,n}$ that depend on n^2 variables.

5. Some generalizations of the concept of separatedness

The concept of separatedness well applies to homogeneous polynomials such as $\mathrm{CL}_{n,k}$, SC_n and SM_n. However, it does not apply to the non-homogeneous polynomials such as $\mathrm{CL}_{n,k} + (\mathrm{CL}_{n,k})^2$, $\mathrm{SC}_n + (\mathrm{SC}_n)^2$, $\mathrm{SM}_n + (\mathrm{SM}_n)^2$.

For instance neither the set $\mathrm{mon}(\mathrm{CL}_{n,k})$ nor the set $\{t \cdot t \mid t \in \mathrm{mon}(\mathrm{CL}_{n,k})\}$ is separated in $\mathrm{mon}(\mathrm{CL}_{n,k} + (\mathrm{CL}_{n,k})^2)$. This difficulty can be handled by the following modification of the concept of separatedness.

Definition 5.1. A subset $B \subset \mathrm{mon}(f)$ is called *1-separated* if (S1), (S2) hold.

(S1) $\forall r \in \mathrm{mon}(f)$: $\forall s, t \in B$: $r \geqslant s \cdot t \Rightarrow [r \leqslant s \quad \text{or} \quad r \leqslant t]$,

(S2) $\forall r \in \mathrm{mon}(f)$: $\forall s \in B$: $r \geqslant s \Rightarrow r = s$.

Observe that $\mathrm{mon}(\mathrm{CL}_{n,k})$ is 1-separated in $\mathrm{mon}(\mathrm{CL}_{n,k} + (\mathrm{CL}_{n,k})^2)$, $\mathrm{mon}(\mathrm{SC}_n)$ is 1-separated in $\mathrm{mon}(\mathrm{SC}_n + (\mathrm{SC}_n)^2)$ a.s.o.

We define the corresponding lower bound on L_+ as follows:

$$\bar{\#}_1(f) = \max\{\|B\| - 1 \mid B \subset \mathrm{mon}(f) \quad \text{is} \quad \text{1-separated}\},$$

$$\#_1(f) = \max\{\bar{\#}(f^\sigma) \mid f^\sigma \in \mathrm{Sub}(f)\}.$$

Theorem 5.2. $\forall f \in \Omega_+$: $\#_1(f) \leqslant L_+(f)$.

Proof. We apply Theorem 2.1 and prove that $\#_1$ satisfies (1)–(5) in 2.1. Clauses (1), (3), (4), (5) are trivial. It remains to consider the crucial clause (2).

(2) Let $\bar{f} = f_{x_i := x_\nu + x_\mu}$ and $x_\nu, x_\mu \notin V(f)$. Let $\bar{f}^{\bar{\sigma}} \in \mathrm{Sub}(\bar{f})$, let $\bar{B} \subset \mathrm{mon}(\bar{f}^{\bar{\sigma}})$ be 1-separated such that $\|\bar{B}\| = \#_1(\bar{f}) + 1$. Then we construct a corresponding $f^\sigma \in \mathrm{Sub}(f)$ and a (1)-separated set $B \subset \mathrm{mon}(f^\sigma)$ such that $\|B\| \geqslant \|\bar{B}\| - 1$.

We define $\sigma_\tau : V(f) \to \mathrm{mon}$, $\tau = \nu, \mu$, as in the proof of 3.2.

Lemma 5.3. *Let* $(tx_\mu)^{\bar{\sigma}}, (sx_\nu)^{\bar{\sigma}} \in \mathrm{mon}(f^{\bar{\sigma}})$, $(tx_\nu)^{\bar{\sigma}} \in \bar{B} - \mathrm{mon}(f^{\sigma_\mu})$, $(sx_\mu)^{\bar{\sigma}} \in \bar{B} - \mathrm{mon}(f^{\sigma_\nu})$. *This implies* $t^{\bar{\sigma}} = s^{\bar{\sigma}}$.

Proof. Obviously $(tx_\mu)^{\bar{\sigma}} \geqslant (tx_\nu)^{\bar{\sigma}} (sx_\mu)^{\bar{\sigma}}$. Since $(tx_\mu)^{\bar{\sigma}} \in \mathrm{mon}(\bar{f}^{\bar{\sigma}})$ and $\bar{B} \subset \mathrm{mon}(\bar{f}^{\bar{\sigma}})$ is 1-separated it follows from (S1) that $(tx_\mu)^{\bar{\sigma}} \leqslant (tx_\nu)^{\bar{\sigma}}$ or $(tx_\mu)^{\bar{\sigma}} \leqslant (sx_\mu)^{\bar{\sigma}}$.
Suppose $(tx_\mu)^{\bar{\sigma}} \leqslant (tx_\nu)^{\bar{\sigma}}$. This implies $\bar{\sigma}(x_\mu) \leqslant \bar{\sigma}(x_\nu)$. Hence $(sx_\nu)^{\bar{\sigma}} \geqslant (sx_\mu)^{\bar{\sigma}}$. Since $(sx_\nu)^{\bar{\sigma}} \in \mathrm{mon}(\bar{f}^{\bar{\sigma}})$, (S2) implies $(sx_\nu)^{\bar{\sigma}} = (sx_\mu)^{\bar{\sigma}}$. It follows $\bar{\sigma}(x_\nu) = \bar{\sigma}(x_\mu)$. This im-

plies $\text{mon}(f^{\sigma}) = \text{mon}(f^{\sigma_\nu}) = \text{mon}(f^{\sigma_\mu})$ and therefore contradicts our assumption $(tx_\nu)^{\bar\sigma} \notin \text{mon}(f^{\sigma_\mu})$. Hence $(tx_\mu)^{\bar\sigma} \leq (sx_\mu)^{\bar\sigma}$. This implies $t^{\bar\sigma} \leq s^{\bar\sigma}$. By permuting the role of tx_ν and sx_μ the same argument implies $t^{\bar\sigma} \geq s^{\bar\sigma}$. This proves $t^{\bar\sigma} = s^{\bar\sigma}$. □

Lemma 5.4. *Either* (1) *or* (2) *or* (3) *holds*

(1) $\bar{B} \subset \text{mon}(f^{\sigma_\nu})$,

(2) $\bar{B} \subset \text{mon}(f^{\sigma_\mu})$,

(3) $\|\bar{B} \cap \text{mon}(f^{\sigma_\nu})\| = \|\bar{B} \cap \text{mon}(f^{\sigma_\mu})\| = \|\bar{B}\| - 1$.

Proof. Suppose $\neg(1) \wedge \neg(2)$. For every $g \in \bar{B} - \text{mon}(f^{\sigma_\nu})$ there exists $(sx_\nu)^{\bar\sigma} \in \text{mon}(f^{\bar\sigma})$ such that $g = (sx_\mu)^{\bar\sigma}$. For every $h \in \bar{B} - \text{mon}(f^{\sigma_\mu})$ there exists $(tx_\mu)^{\bar\sigma} \in \text{mon}(f^{\bar\sigma})$ such that $h = (tx_\nu)^{\bar\sigma}$. It follows from Lemma 5.3 that $t^{\bar\sigma} = s^{\bar\sigma}$. Therefore $h \in \bar{B} - \text{mon}(f^{\sigma_\mu})$ and $g \in \bar{B} - \text{mon}(f^{\sigma_\nu})$ are uniquely determined. This proves (3). □

Obviously $\bar{B} \cap \text{mon}(f^{\sigma_\tau}) \subset \text{mon}(f^{\sigma_\tau})$ is 1-separated for $\tau = \nu, \mu$. Therefore Lemma 5.4 implies

$$\#_1(f^{\sigma_\nu}) \geq \#_1(\bar{f}) - 1 \quad \text{or} \quad \#_1(f^{\sigma_\mu}) \geq \#_1(\bar{f}) - 1.$$

This implies $\#_1(f) \geq \#_1(\bar{f}) - 1$. Hence (2) in 2.1 holds and this proves 5.2. □

Corollary 5.5.

$$L_+(\text{CL}_{n,k} + (\text{CL}_{n,k})^2) \geq \binom{n}{k} - 1,$$

$$L_+(\text{SC}_n + (\text{SC}_n)^2) \geq n^2 - 1.$$

Proof. $\text{mon}(\text{CL}_{n,k}) \subset \text{mon}(\text{CL}_{n,k} + (\text{CL}_{n,k})^2)$ is 1-separated. $\text{mon}(\text{SC}_n) \subset \text{mon}(\text{SC}_n + (\text{SC}_n)^2)$ is 1-separated. Observe that these bounds are rather sharp since

$$L_+(\text{CL}_{n,k} + (\text{CL}_{n,k})^2) \leq \binom{n}{k},$$

$$L_+(\text{SC}_n + (\text{SC}_n)^2) \leq n^2. \quad \square$$

Next we consider the problem whether these bounds apply to the Boolean case. In the Boolean case we substitute $\wedge$ for $\cdot$ and $\vee$ for $+$ and Boolean variables for rational variables. Let Ω_+^b be the set of all monotone Boolean functions. We consider monotone Boolean computations for functions $f \in \Omega_+^b$, i.e. logical networks with the operations $\wedge$ and $\vee$. Let $L_\vee(f)$ be the minimal number of $\vee$-gates in any monotone Boolean computation for $f \in \Omega_+^b$.

There is a natural translation of the concept of separatedness to the Boolean case. Boolean monomials are also called implicants. The relation $t \leq s$ for implicants t, s is defined as $t \leq s \leftrightarrow t \wedge s = t$. Let prime$(f)$ be the set of prime implicants of $f \in \Omega_+^b$. A subset $B \subset \text{prime}(f)$ is called *b-separated* if $\forall r \in \text{prime}(f)$: $\forall s, t \in B$: $r \geq s \wedge t \Rightarrow [r = s \text{ or } r = t]$.

Define

$$\#_b(f) = \max\{\|B\| - 1 \mid B \subset \mathrm{prime}(f) \text{ is } b\text{-separated}\}.$$

A main open problem is to prove or to disprove the following:

Conjecture 5.6. $\forall f \in \Omega_+^b$: $L_\vee(f) \geq \#_b(f)$.

One difficulty in proving 5.6 is that an application of a substitution $\bar{f} := f_{x_i := x_\nu \vee x_\mu}$ can eliminate prime implicants of f which do not depend on x_i; these prime implicants can be absorbed by greater prime implicants that are generated in the same substitution step. This possibly may lead to $\#_b(\bar{f}) \geq \#_b(f)$ in a case where prime implicants of f disappear which prevent certain subsets $B \subset \mathrm{prime}(f)$ to be b-separated.

It should be observed that there are some characteristic connections between $L_\vee$ and L_+. Let $\alpha : \Omega_+ \to \Omega_+^b$ be the natural transformation which is inductively defined as follows:

$$\forall r \in \mathbb{Q}_+: \ \alpha(r) = 1,$$

$$\forall f, g \in \Omega_+: \ \alpha(f + g) = \alpha(f) \vee \alpha(g),$$

$$\forall f, g \in \Omega_+: \ \alpha(f \cdot g) = \alpha(f) \wedge \alpha(g).$$

Moreover α maps rational variables into Boolean variables in a one-one manner. It can easily be seen that

$$L_\vee(f) = \min\{L_+(g) \mid \alpha(g) = f\} \quad \text{for} \quad f \in \Omega_+^b.$$

($\leq$) Every monotone, rational computation for g yields a monotone Boolean computation for $\alpha(g)$ by replacing $+$ by $\vee$ and $\cdot$ by $\wedge$.

($\geq$) Every monotone Boolean computation β yields a monotone rational computation $\bar{\beta}$ by replacing $\vee$ by $+$ and $\wedge$ by $\cdot$. Obviously $\mathrm{res}_\beta^\vee = \alpha(\mathrm{res}_{\bar{\beta}}^\times)$.

Some more details that relate different concepts of separatedness for rational polynomials to the concept of b-separatedness for Boolean polynomials can be found in [6].

References

[1] S. A. Cook, The complexity of theorem-proving procedures, Symposium on Theory of Computing (1971) 151–158.

[2] M. J. Fischer, Lectures on network complexity, Preprint Universität Frankfurt (1974).

[3] R. M. Karp, Reducibility among combinatorial problems, in: *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher (eds.) (Plenum Press, New York, 1972) 85–104.

[4] M. S. Paterson, Complexity of monotone networks for Boolean matrix product, *Theoret. Comput. Sci.* 1 (1975) 13–20.

[5] C. P. Schnorr, The network complexity and the Turing machine complexity of finite functions, *Acta Informat.*, to appear.

[6] C. P. Schnorr, A lower bound on the number of additions in monotone computations of monotone rational polynomials, Preprint Universität Frankfurt (1974).