Some Exact Complexity Results for Straight-Line Computations over Semirings

MARK JERRUM AND MARC SNIR

University of Edinburgh, Edinburgh, Scotland

ABSTRACT The problem of computing polynomials in certain semirings is considered. Precise bounds are obtained on the number of multiplications required by straight-line algorithms which compute such functions as iterated matrix multiplication, iterated convolution, and permanent. Using these bounds, it is shown that the use of branching can exponentially speed up computations using the min, + operations, and that subtraction can exponentially speed up arithmetic computations. These results can be interpreted as denying the existence of fast "universal" algorithms for computing certain polynomials.

KEY WORDS AND PHRASES arithmetic complexity, convexity theory, Farkas Lemma, minimax algebra, straight-line algorithm

Categories and Subject Descriptors: F.1.1 [**Computation by Abstract Devices**]: Models of Computation—*computability theory, relations among models*; F.1.3 [**Computation by Abstract Devices**]: Complexity Classes—*relations among complexity classes*; F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems—*computations on polynomials*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Arithmetic complexity, convexity theory, Farkas Lemma, minimax algebra, straight-line algorithm

1. *Introduction*

Finding the number of operations necessary to compute polynomial functions is an old problem in algebraic complexity. Even for seemingly simple computations, such as matrix multiplication, this problem is still unsettled despite long-term efforts. In fact, profound algebraic methods seem necessary to obtain all but most trivial results, and fast algorithms can be built by using nontrivial algebraic properties of the domain of computation. (See [2] for an introduction to this field, [19] for state-of-the-art matrix multiplication.)

An obvious (cowardly?) escape from the difficulties of the general problem is provided by restricted computational models. Thus, in the field of Boolean complexity much work has been done on monotone Boolean computations, which are more tractable than computations using negations (see [7, 10, 15, 16, 28]). Similar work has been done on monotone arithmetic computations, that is, computations using only positive constants, addition, and multiplication [18, 20–22]. In both models it is

Authors' present addresses: M. Jerrum, Department of Computer Science, James Clerk Maxwell Building, The King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, Scotland; M. Snir, Department of Computer Science, New York University, 251 Mercer Street, New York, NY 10012.

relatively easy to prove that matrix multiplication requires n^3 scalar multiplications. Of the same flavor are results concerning regular expressions not using complementation or intersection [4, 6].

Some arguments can be brought in favor of considering restricted computational models. Thus, monotone arithmetic computations have absolute numerical stability [11]; monotone Boolean circuits are easier to test (see [9]). The main argument however is that monotone computation is more "natural," whenever applicable: restriction to monotone arithmetic essentially means restriction to algorithms for which correctness can be deduced from the associativity, commutativity, and distributivity of addition and multiplication (see [21]). It follows that the family of monotone computations is essentially the same in any semiring (i.e., a domain with two binary operations, $\oplus$ and $\otimes$, both associative and commutative, with $\otimes$ distributing over $\oplus$). Algorithms can be built uniformly for different semirings, and lower bounds can be proved uniformly for this class of "universal" algorithms.

Shorter algorithms can be built for most of the arithmetic functions we shall consider by taking into account the existence of an additive inverse in the field of real numbers $\mathbb{R}$. Similarly, the use of the extra axioms of Boolean algebra can yield smaller circuits. We have a trade-off between the simplicity and generality of our algorithms, on the one hand, and their efficiency on the other, that is, between the complexity of the algorithm and the complexity of a validation of it. When a fixed specific function is under consideration, the balance is tilted in favor of the shortest computation. The situation might be different from a system which has to derive from an implicit description of a function an explicit algorithm for its computation, as well as actually computing it (such an ability might be required of the next generation of compilers). It is therefore essential to have a good understanding of the trade-offs incurred when the domain of allowed computations is restricted.

We consider essentially two semirings: the semiring R of nonnegative real numbers with the usual addition and multiplication (monotone arithmetic), and the semiring M (M^+) of (nonnegative) real numbers with the operations of minimum and addition. The latter structure has frequently been used to formulate and solve optimization problems (see [1, p. 195; 3]).

In Section 2 we show that the problem of computing a polynomial function in these semirings is related to the problem of computing a formal polynomial over the semiring. This in turn is as hard as computing a similar formal polynomial over the Boolean semiring B ($\{0, 1\}$ with the two operations or, and). Formal polynomials over B are essentially finite sets of integer-valued vectors, with addition being union and multiplication being componentwise addition. Computations are combinatorial in character, and we develop in Section 3 a combinatorial method which yields lower bounds on the number of multiplications needed. This is achieved essentially by abstracting from the computational task considered a suitable combinatorial optimization problem. Several applications are considered in Section 4: the computation of the product of m $n \times n$ matrices takes $(m - 1)n^3$ multiplications; the wrapped convolution of m n-vectors is computed in $(m - 1)n^2$ multiplications; the computation of the permanent takes $n(2^{n-1} - 1)$ multiplications. All these bounds are tight. Several other functions, which are related to optimization problems, are also considered. A discussion of the results follows in Section 5.

2. *Definitions*

2.1 SEMIRINGS AND POLYNOMIALS. We introduce here the (fairly standard) algebraic terminology we shall subsequently be using.

A *semiring* is a system $(S, \oplus, \otimes, 0, 1)$, where S is a set, $\oplus$ (addition) and $\otimes$ (multiplication) are binary operations on S, and 0 and 1 are elements of S having the following properties:

(i) $(S, \oplus, 0)$ is a commutative monoid, that is, $\oplus$ is associative and commutative and 0 is an identity.
(ii) $(S, \otimes, 1)$ is a commutative monoid.
(iii) $\otimes$ distributes over $\oplus$, that is, $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$.
(iv) $a \otimes 0 = 0$.

We shall subsequently use the following semirings:

(i) The Boolean semiring $B = (\{0, 1\}, \vee, \wedge, 0, 1)$ ($\vee$ being Boolean disjunction and $\wedge$ being conjunction).
(ii) The semiring of nonnegative real numbers with the usual addition and multiplication $R = (\mathbb{R}^+, +, \cdot, 0, 1)$.
(iii) The semiring $M = (\mathbb{R}^*, \min, +, +\infty, 0)$, where $\mathbb{R}^* = \mathbb{R} \cup \{+\infty\}$, min is the binary minimum operation, and + is the usual addition.
(iv) The semiring $M^+ = (\mathbb{R}^{+*}, \min, +, +\infty, 0)$, which is the subsemiring of M obtained by restricting the domain to nonnegative real numbers.

Let S be a semiring and $X = \{x_1, \ldots, x_n\}$ a finite set of *indeterminates*. We denote by $S[X]$ the semiring of (formal) *polynomials* obtained from S by adjunction of the indeterminates $x_1, \ldots, x_n$. Each *monomial* $m = x_1^{i_1} \cdots x_n^{i_n}$ is uniquely determined by the vector of exponents $(i_1, \ldots, i_n)$, so that we can identify monomials with elements of $\mathbb{N}^n$. Each polynomial $p \in S[X]$ may be uniquely written in the form

$$p = \bigoplus_{(i_1, \ldots, i_n) \in \mathbb{N}^n} a_{i_1 \ldots i_n} x_1^{i_1} \cdots x_n^{i_n}, \tag{2.1}$$

where only finitely many *coefficients* $a_{i_1 \ldots i_n} \in S$ are different from zero, so that we can identify polynomials with functions from $\mathbb{N}^n$ to S with finite support. Thus, if $p \in S[X]$, $m \in \mathbb{N}^n$, p_m will denote the value of the coefficient of p with index set m, and eq. (2.1) can be rewritten as

$$p = \bigoplus_{m \in \mathbb{N}^n} p_m m. \tag{2.2}$$

S is embedded in $S[X]$ by identifying each element $s \in S$ with the *constant polynomial* $s x_1^0 \cdots x_n^0$. (For a more elaborate approach see, e.g., [17, Sec. 67].)

We introduce some terminology pertaining to $S[X]$. We assume henceforth that p is a polynomial given by eq. (2.2) and m is a monomial given by $m = (i_1, \ldots, i_n)$.

(i) The *monomial set* of p is

$$\operatorname{mon}(p) = \{m \in \mathbb{N}^n \mid p_m \neq 0\}.$$

(ii) The *degree* of m is

$$\deg(m) = \sum_{j=1}^{n} i_j.$$

(iii) The *degree* of p is

$$\deg(p) = \max\{\deg(m) \mid m \in \operatorname{mon}(p)\}.$$

(iv) p is *homogeneous* if all its monomials have the same degree.
(v) m is *linear* if $m \in \{0, 1\}^n$.
(vi) p is *linear* if all its monomials are linear.

To each polynomial $p \in S[X]$ is associated a *polynomial function* $\nu p: S^n \to S$, the function whose value at $(a_1, \ldots, a_n)$ is obtained by substituting a_i for x_i in p. The map ν is a homomorphism from $S[X]$ to the semiring of functions $[S^n \to S]$, with pointwise addition and pointwise multiplication. We denote by $P_n(S)$ the image of $S[X]$ under ν, that is, the subsemiring of polynomial functions. The map ν need not be injective, as two different polynomials, for example, x and x^2 in $B[x]$, can represent the same function.

2.2 COMPUTATIONS AND COMPLEXITY. Let S be a semiring. A *computation* Γ in S with *input set* $I \subset S$ is a labeled, directed acyclic graph (dag) with the following properties:

(i) Nodes of Γ with indegree 0, termed *input nodes*, are labeled by elements of I.
(ii) The nodes of Γ which are not input nodes, termed *internal nodes*, all have indegree 2 and are labeled either by $\oplus$ or $\otimes$.
(iii) There is a unique node ρ of Γ of outdegree 0, termed the *output node*.

If there is in Γ an edge directed from node α to node β, then α is a *predecessor* of β and β a *successor* of α. The *ancestor* relation is the transitive closure of the predecessor relation; the *descendent* relation is the transitive closure of the successor relation.

A result function, res: nodes$(\Gamma) \to S$, is defined recursively on the nodes of Γ in the following manner:

(i) If α is an input node labeled by $i \in I$, then $\text{res}(\alpha) = i$.
(ii) If α is an internal node labeled by $\oplus$ (a $\oplus$-node) with predecessors β, γ, then $\text{res}(\alpha) = \text{res}(\beta) \oplus \text{res}(\gamma)$.
(iii) If α is a $\otimes$ node with predecessors β, γ, then $\text{res}(\alpha) = \text{res}(\beta) \otimes \text{res}(\gamma)$.

Γ *computes* s if $\text{res}(\rho) = s$, where ρ is the output node of Γ.

The $\otimes$ ($\oplus$) *-complexity* of Γ is simply the number of $\otimes$ ($\oplus$) -nodes of Γ. The $\otimes$ ($\oplus$) *-complexity of* $s \in S$ *with respect to* $I \subset S$ is the minimal $\otimes$ ($\oplus$) -complexity of a computation with input set I computing s. Of particular interest to us will be computations of polynomials in $S[X]$ and polynomial functions in $P_n(S)$. For computations in $S[X]$ the input set will always be assumed to be $S \cup X$, and for computations in $P_n(S)$ it will accordingly consist of the constant functions and projection functions. Thus the $\otimes$ ($\oplus$) -complexity of a polynomial (polynomial function) will be understood to mean the $\otimes$ ($\oplus$) -complexity with respect to these sets.

Whenever an algebraic structure is homomorphic to another, computations in the first structure are related to computations in the second, and so complexity results for the second structure translate into results for the first. Indeed we have

LEMMA 2.1. *Let S, S' be semirings and $\tau: S \to S'$ a homomorphism. Let Γ compute $s \in S$ with input set $I \subset S$. Let Γ' be obtained from Γ by relabeling each input node with label $i \in I$ by $\tau(i)$. Then Γ' is a computation in S' with input set $\tau(I)$; for each node α of Γ, if $r = res(\alpha)$, then $\tau(r)$ is the result at α in Γ'. In particular, Γ' computes $\tau(s)$.*

COROLLARY 2.2. *Let S, S' be semirings, $\tau: S \to S'$ a homomorphism.*

(i) The $\otimes$ ($\oplus$) -complexity of $s \in S$ with respect to $I \subset S$ is no smaller than the $\otimes$ ($\oplus$) -complexity of $\tau(s)$ with respect to $\tau(I)$.
(ii) If τ is surjective, then the $\otimes$ ($\oplus$) -complexity of $s' \in S'$ with respect to $I \subset S'$ is equal to the minimal $\otimes$ ($\oplus$) -complexity of an element $s \in \tau^{-1}(s')$ with respect to $\tau^{-1}(I)$.

As an important application of Corollary 2.2. we obtain, having in mind the mapping from polynomials onto polynomial functions,

COROLLARY 2.3. *The $\otimes$ ($\oplus$) -complexity of a polynomial function f is equal to the minimal $\otimes$ ($\oplus$) -complexity of a polynomial representing f.*

The above result is very useful in semirings in which each polynomial function is represented by a unique polynomial (indeed, in such cases, it is customary to blur the distinction between them). Such is the case for the semiring R. Thus, in this semiring the $\otimes$ ($\oplus$) -complexity of a polynomial p equals the $\otimes$ ($\oplus$) -complexity of the polynomial function it represents. This is not true in general for the semirings M, M^+, and B in which there is no unique representation of polynomial functions. Section 2.3 will deal with this problem.

Our complexity results will be derived in the first instance for polynomials in $B[X]$. These results can be extended, using Corollary 2.2, to any other polynomial semiring $S[X]$, provided that we can exhibit a homomorphism from $S[X]$ to $B[X]$, mapping $S \cup X$ into $B \cup X$. But any homomorphism $\tau: S \to B$ extends naturally to a homomorphism $\tau: S[X] \to B[X]$ which maps S into B and x_i onto itself. For all three semirings R, M, M^+ such a homomorphism exists and is given by

$$\tau(a) = \begin{cases} 0_B & \text{if} \quad a = 0_S \quad (0_S \text{ is } 0 \text{ in } R \text{ and } +\infty \text{ in } M, M^+), \\ 1_B & \text{if} \quad a \neq 0_S. \end{cases}$$

It should be mentioned that τ maps polynomials with 0–1 coefficients into formally identical polynomials, and thus any lower bound obtained for the $\otimes$ ($\oplus$) -complexity of a polynomial $p \in B[X]$ yields immediate lower bounds on the $\otimes$ ($\oplus$) -complexity of the formally identical polynomials in $R[X]$, $M[X]$, and $M^+[X]$.

As has been remarked, in the case of $M[X]$ and $M^+[X]$ the canonical homomorphism ν from formal polynomials to polynomial functions is not an isomorphism. The remainder of Section 2—which is self-contained and can be omitted—establishes the machinery required to deal with this problem.

2.3 ENVELOPES AND COMPUTATIONS IN MIN, +. As will be seen in Section 3, our methods are better suited to handle homogeneous polynomials. We can however extract, from any polynomial, homogeneous components which are simpler to compute. Let $p \in S[X]$ be given by eq. (2.2) and $k = \min\{\deg(m) \mid m \in \operatorname{mon}(p)\}$. The *lower envelope* of p is given by

$$\operatorname{le}(p) = \bigoplus_{\deg(m)=k} p_m m.$$

Similarly, if $K = \max\{\deg(m) \mid m \in \operatorname{mon}(p)\}$, then the *higher envelope* of p is given by

$$\operatorname{he}(p) = \bigoplus_{\deg(m)=K} p_m m.$$

Thus $\operatorname{le}(p)$ ($\operatorname{he}(p)$) is obtained from p by preserving only the terms of minimal (maximal) degree. Now assume that the function τ of the previous section is indeed a homomorphism.

If $p = q_1 \oplus q_2$, then

$$\begin{array}{lll} \text{if} \quad \deg(\operatorname{le}(q_1)) = \deg(\operatorname{le}(q_2)), & \text{then} & \operatorname{le}(p) = \operatorname{le}(q_1) \oplus le(q_2), \\ \text{if} \quad \deg(\operatorname{le}(q_1)) < \deg(\operatorname{le}(q_2)), & \text{then} & \operatorname{le}(p) = \operatorname{le}(q_1), \\ \text{if} \quad p = q_1 \otimes q_2, & \text{then} & \operatorname{le}(p) = \operatorname{le}(q_1) \otimes \operatorname{le}(q_2). \end{array}$$

Similar relations hold for the higher envelope.

It is thus obvious that any computation of $p \in S[X]$ can restructured, by appropriately discarding some of its additions, into a computation of le(p) (he(p)). We thus have

THEOREM 2.4. *Let $p \in S[X]$. The $\oplus$ ($\otimes$) -complexity of p is no smaller than the $\oplus$ ($\otimes$) -complexity of le(p) (he(p)).*

Let us now turn to the semirings M and M^+. We shall investigate how the structure of a polynomial is determined by the function it represents. We assume $p \in M[X]$ ($p \in M^+[X]$) is given by

$$p = \bigoplus_{i=1}^{k} a_i m_i, \tag{2.3}$$

where $a_i \neq +\infty$, $m_i \in \mathbb{N}^n$. The function f represented by p is

$$f(\bar{u}) = f(u_1, \ldots, u_n) = \min_i \langle m_i \cdot \bar{u} \rangle + a_i,$$

where $\langle \bar{u} \cdot \bar{v} \rangle$ denotes scalar product. We associate with f the set Gr(f) $\subseteq \mathbb{R}^{n+1}$, which is bounded above by the graph of f.

$$\begin{aligned} \mathrm{Gr}(f) &= \{(u_1, \ldots, u_n, v) \mid v \leq f(\bar{u})\} \\ &= \{(\bar{u}, v) \mid v \leq \langle m_i \cdot \bar{u} \rangle + a_i \text{ for } i = 1, \ldots, k\}. \end{aligned}$$

Gr(f) is the intersection of k closed half-spaces corresponding to the k terms of p and has nonempty interior (unless $p = -\infty$). There is a unique minimal family of half-spaces whose intersection yields Gr(f), each half-space being bounded by a hyperplane which contains one of the n-dimensional faces of the $(n + 1)$-dimensional polyhedron Gr(f). It follows that there is a unique set of terms of p which appears in any polynomial representing f. We have also a nice characterization of the remaining (redundant) terms.

THEOREM 2.5. *Let $f \in P_n(M)$ be a polynomial function over M. There exists a unique set of terms $T = \{a_i m_i\}$ such that if p represents f in $M[X]$, then*

(*i*) *each term of T occurs in p;*
(*ii*) *if $a'm'$ is a term of p, then there exist $\lambda_1, \ldots, \lambda_n$ such that*

$$\lambda_i \geq 0, \qquad \forall i, \tag{2.4}$$

$$\sum \lambda_i = 1, \tag{2.5}$$

$$\mathrm{m}' = \sum \lambda_i m_i, \tag{2.6}$$

$$a' \geq \sum \lambda_i a_i. \tag{2.7}$$

(Each monomial of p is a convex combination of the *essential monomials*, with its coefficient bounded below by the convex combination of their coefficients.)

PROOF. See the appendix. □

The characterization of Theorem 2.5 yields a unique representation theorem for certain functions.

THEOREM 2.6. *Let $p, q \in M[X]$ represent the same function. Then*

(*i*) *if p is linear, then $p = q$;*
(*ii*) *if le(p) (he(p)) is linear, then le(p) = le(q) (he(p) = he(q)).*

PROOF

(i) Let $T = \{a_i m_i\}$ be the set of essential terms occurring both in p and q. We claim that no other term occurs in p or q. Indeed, let $a'm'$ be a term of p or q. Then $m' = \sum \lambda_i m_i$, with $\lambda_i \geq 0$, $\sum \lambda_i = 1$. But m_i are 0–1 valued vectors, and no nontrivial convex combination of them can yield an integer-valued vector (the interior of the unit cube does not contain lattice points). Thus the monomial m' occurs in T, and so $a'm' \in T$.

(ii) Let $k = \min \deg(m_i)$. We claim that the terms of le(p) (le(q)) are precisely the minimal degree terms of T. If $\deg(m_i) = k$, then $a_i m_i$ occurs in le(p). On the other hand, let $a'm'$ be a term of le(p) or le(q). Then $\deg(m') = k$, and $m' = \sum \lambda_i m_i$, with $\lambda_i \geq 0$, $\sum \lambda_i = 1$. But $\deg(m') = \sum \lambda_i \deg(m_i) \geq \min \deg(m_i) = k$, and equality can occur only if $\lambda_i = 0$ whenever $\deg(m_i) > k$. Thus m' is a convex combination of the minimal degree monomials in T, and by the same argument used in (i) it follows that $a'm' \in T$. The proof for higher envelopes is similar. □

COROLLARY 2.7. *Let $p \in M[X]$ represent the function $f \in P_n(M)$. Then*

(*i*) *if p is linear, then the $\oplus$ ($\otimes$) -complexity of f is equal to the $\oplus$ ($\otimes$) -complexity of p;*
(*ii*) *if le(p) (he(p)) is linear, then the $\oplus$ ($\otimes$) -complexity of f is no smaller than the $\oplus$ ($\otimes$) -complexity of le(p) (he(p)).*

PROOF. Use Corollary 2.3 and Theorems 2.4 and 2.6. □

When the domain of computation is restricted to nonnegative numbers, there is greater freedom in choosing representations for functions.

THEOREM 2.8. *Let $f \in P_n(M^+)$ be a polynomial function over M^+. There exists a unique set of terms $T = \{a_i m_i\}$ such that if p represents f in $M^+[X]$, then*

(*i*) *each term of T occurs in p;*
(*ii*) *if $a'm'$ is a term of p, then there exist $\lambda_1, \ldots, \lambda_n$ such that*

$$\lambda_i \geq 0, \qquad \forall i, \tag{2.8}$$

$$\sum \lambda_i = 1, \tag{2.9}$$

$$m' \geq \sum \lambda_i m_i, \tag{2.10}$$

$$a' \geq \sum \lambda_i a_i. \tag{2.11}$$

(*$a'm'$ is bounded below by a convex combination of the terms in T.*)

PROOF. See the appendix. □

For M^+ we have the following unique representation theorem.

THEOREM 2.9. *Let $p, q \in M^+[X]$ represent the same function. Then*

(*i*) *if le(p) is linear, then le(p) = le(q);*
(*ii*) *if p is linear and homogeneous, then p = le(q).*

PROOF

(i) It is easy to check that the argument used in proving Theorem 2.6(ii) can be carried over using inequality (2.10) instead of equality (2.6). (There is no analogous argument for higher envelopes.)
(ii) If p is homogeneous, then p = le(p) and (ii) follows from (i). □

COROLLARY 2.10. *Let $p \in M^+[X]$ represent the function $f \in P_n(M^+)$. Then*

(*i*) *if $le(p)$ is linear, then the $\oplus$ ($\otimes$) -complexity of f is no smaller than the $\oplus$ ($\otimes$) -complexity of $le(p)$;*
(*ii*) *if p is linear and homogeneous, then the $\oplus$ ($\otimes$) -complexity of f is equal to the $\oplus$ ($\otimes$) -complexity of p.*

PROOF. Use Corollary 2.3 and Theorems 2.4 and 2.9. □

3. *The Lower Bound Argument*

3.1 COMPUTATIONS IN $B[X]$. In this section we restrict our attention to computations in $B[X]$. Extensions of results obtained here to other polynomial semirings will be immediate from the considerations introduced in Section 2. Throughout the following, Γ will denote an arbitrary computation in $B[X]$ with result node ρ and $\mathrm{res}(\rho) = p \in B[X]$. Γ_{opt} will be such a computation in which the total number of $\otimes$-nodes is minimized.

We extend our previous notation and introduce some concepts which aid the study of computations in $B[X]$.

If $\alpha \in \mathrm{nodes}(\Gamma)$, then *mon*($\alpha$) is the monomial set of res(α), and *deg*(α) the degree of res(α). *pred*(α) denotes the set of predecessors of α. Γ is said to be *linear* (*homogeneous*) if res(α) is a linear (homogeneous) polynomial for all $\alpha \in \Gamma$.

We may as well assume that 0 is not an input of Γ (we lose no computational power by this), in which case it is easy to check.

LEMMA 3.1

(*i*) *Γ is linear if and only if p is.*
(*ii*) *Γ is homogeneous if and only if p is.*
(*iii*) *If α, β are nodes of Γ, β is a descendant of α, and $m \in mon(\alpha)$, then $mon(\beta)$ contains a monomial of the form mm'.*

Lemma 3.1 captures that property of computation in $B[X]$ which makes it amenable to treatment in the style of [20] or of the present paper. Stated informally, once a monomial has been created, it must find its way into the final result; this "conservation of monomials" ensures that no "invalid" monomials are formed and severely limits the rate at which monomials may be accumulated in the computation.

If $\alpha \in \mathrm{nodes}(\Gamma)$, then the *complement* of α is the set

$$\mathrm{complement}(\alpha) = \{m = x_1^{i_1}x_2^{i_2} \cdots x_n^{i_n} \mid \forall m' \in \mathrm{mon}(\alpha),\ mm' \in \mathrm{mon}(\rho)\},$$

and the *content* of α is the set

$$\mathrm{content}(\alpha) = \{mm' \mid m \in \mathrm{complement}(\alpha),\ m' \in \mathrm{mon}(\alpha)\}.$$

We remark that $\mathrm{content}(\alpha) \subseteq \mathrm{mon}(\rho)$.

3.2. PARSE-TREE. At the crux of our argument is the concept of parse-tree, which has meaning for all linear computations, and which we now elaborate. If $\alpha \in \mathrm{nodes}(\Gamma)$ and $m \in \mathrm{mon}(\alpha)$, $m \neq 1$ (the unit monomial), then the *parse-tree induced by α and m*, PT(α, m), is a labeled subtree of Γ, rooted at α, and defined recursively on the nodes of Γ as follows:

(i) If $\deg(m) = 1$, then PT(α, m) is simply the subgraph of Γ formed by α labeled by m. Otherwise:

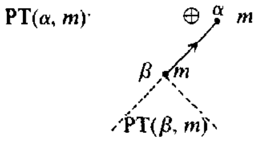

FIGURE 1

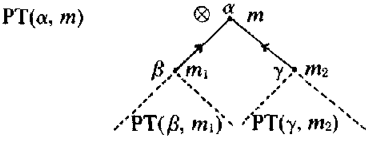

FIGURE 2

(ii) If α is a $\oplus$-node, then let pred(α) = $\{\beta, \gamma\}$. Since $m \in$ mon(α), we may deduce that either $m \in$ mon(β) or $m \in$ mon(γ) (or both). Without loss of generality we may suppose the former. Then define PT(α, m) to be the subtree of Γ formed by augmenting PT(β, m) with the node α, labeled by m, and the edge $\beta\alpha$ (see Figure 1). Note that we have a certain freedom in choosing between β and γ, but we can make our procedure deterministic by, say, ordering the predecessors of α.

(iii) If α is a $\otimes$-node, then again let pred(α) = $\{\beta, \gamma\}$. Since $m \in$ mon(α), there must exist $m_1 \in$ mon(β) and $m_2 \in$ mon(γ) such that $m = m_1 m_2$. (Again, m_1 and m_2 are not uniquely defined, but we can provide a rule for choosing such a pair.) Dismissing first the degenerate case when one of m_1, m_2, say m_2, is 1 (the unit monomial), we set PT(α, m) to be the subtree of Γ consisting of PT(β, m) augmented with the node α, labeled by m, and the edge $\beta\alpha$. In general we define PT(α, m) to be the subtree of Γ consisting of the union of PT(β, m_1) and PT(γ, m_2) augmented with the node α, labeled by m, and the edges $\beta\alpha$, $\gamma\alpha$ (see Figure 2). For this to make sense we require PT(β, m_1) and PT(γ, m_2) to be disjoint. But this is ensured by the linearity of Γ, for if δ were a common node and $m_3 \in$ mon(δ), $m_3 \neq 1$, we could deduce by Lemma 3.1(iii) that mon(β) contains a monomial of the form $m_3 n_1$ and mon(γ) one of the form $m_3 n_2$. Hence mon(α) would contain a monomial of the form $n_1 n_2 m_3^2$, which is a contradiction.

It is hoped that the parse-tree is an intuitively appealing construct; essentially it is a family tree which charts the generation of a particular monomial in the final result. Those familiar with [4] will note the similarity between our notion and the "parse function" which is defined there on elements of regular sets.

THEOREM 3.2. *Let m be an element of mon(ρ). Then $\alpha \in$ nodes[PT(ρ, m)] implies $m \in$ content(α).*

PROOF. Suppose that $\alpha \in$ nodes[PT(ρ, m)]. Denote the monomial which labels α by m_α. We are done if we can show that for each α $\exists m'_\alpha$ such that

$$m_\alpha m'_\alpha = m, \tag{3.1}$$

$$m'_\alpha n \in \text{mon}(\rho) \qquad \forall n \in \text{mon}(\alpha). \tag{3.2}$$

For if (3.1) and (3.2) are satisfied, we have $m'_\alpha \in$ complement(α), $m_\alpha \in$ mon(α), and hence $m = m_\alpha m'_\alpha \in$ content(α). We establish the existence of m'_α satisfying (3.1) and (3.2) by induction on the nodes of PT(ρ, m). First we note:

(i) Our hypothesis is true for the root node ρ. Take $m'_\rho = 1$; then (3.1) and (3.2) are trivially true.

(ii) Assume the hypothesis is true for $\oplus$-node β labeled by monomial m_β. Let γ be the predecessor of β in PT(ρ, m), and let m'_β satisfy (3.1) and (3.2). We show that the hypothesis holds for γ. By construction, m_β also labels γ, that is, m_γ =

m_β. We satisfy (3.1) by taking $m'_\gamma = m'_\beta$ and note that since

$$\begin{aligned}\{m'_\gamma n \mid n \in \text{mon}(\gamma)\} &\subseteq \{m'_\gamma n \mid n \in \text{mon}(\beta)\}\\ &= \{m'_\beta n \mid n \in \text{mon}(\beta)\}\\ &\subseteq \text{mon}(\rho),\end{aligned}$$

(3.2) is also satisfied.

(iii) Assume the hypothesis is true for ⊗-node β, labeled by monomial m_β. Let $\text{pred}(\beta) = \{\gamma, \delta\}$, m_γ, m_δ be the monomials which label γ and δ, and m'_β satisfy (3.1) and (3.2). We show that the hypothesis holds for γ. Set $m'_\gamma = m'_\beta m_\delta$. We observe that $m_\gamma m'_\gamma = m_\gamma m'_\beta m_\delta = m'_\beta m_\beta = m$, and so (3.1) is satisfied. Additionally,

$$\begin{aligned}\{m'_\gamma n \mid n \in \text{mon}(\gamma)\} &= \{m'_\beta m_\delta n \mid n \in \text{mon}(\gamma)\}\\ &\subseteq \{m'_\beta n' \mid n' \in \text{mon}(\beta)\}\\ &\subseteq \text{mon}(\rho) \qquad \text{by the induction hypothesis,}\end{aligned}$$

and so 3.2 is also satisfied. □

We may capitalize on the previous result in a straightforward way. Γ contains $|\text{mon}(\rho)|$ parse-trees corresponding to distinct monomials of p. Distinct parse-trees may share nodes of Γ, but the amount of sharing that takes place is limited by Theorem 3.2. We hope to obtain from this a lower bound on the complexity of Γ. In order to make this qualitative argument precise we introduce a weight function for parse-trees.

3.3 Weight Function. Suppose T is a parse-tree in Γ. Define the *weight* of T, $w(T)$, to be

$$w(T) = \sum_{\alpha \in \otimes\text{-nodes}(T)} |\text{content}(\alpha)|^{-1}.$$

Theorem 3.3

$$\sum_{m \in mon(\rho)} w(PT(\rho, m)) \leq |\otimes\text{-}nodes(\Gamma)|.$$

Proof

$$\begin{aligned}&\sum_{m \in \text{mon}(\rho)} w(\text{PT}(\rho, m))\\ &= \sum_{m \in \text{mon}(\rho)} \sum_{\alpha \in \otimes\text{-nodes}(\text{PT}(\rho, m))} |\text{content}(\alpha)|^{-1}\\ &= \sum_{\alpha \in \otimes\text{-nodes}(\Gamma)} \frac{|\{m \mid \alpha \in \otimes\text{-nodes}(\text{PT}(\rho, m))\}|}{|\text{content}(\alpha)|}\\ &\leq \sum_{\alpha \in \otimes\text{-nodes}(\Gamma)} \frac{|\{m \mid m \in \text{content}(\alpha)\}|}{|\text{content}(\alpha)|} \quad \text{(by Theorem 3.2)}\\ &= |\otimes\text{-nodes}(\Gamma)|.\end{aligned}$$

□

Now suppose that for the specific homogeneous, linear polynomial p we have some bound on the content of nodes in the computation. Specifically, we assume the existence of a function $\bar{c}(r, d)$, $2 \leq r \leq \deg(p)$, $1 \leq d \leq \lfloor r/2 \rfloor$ which satisfies

$$\begin{aligned}\bar{c}(r, d) \geq \max\{\,|\text{content}(\alpha)| \mid \alpha \in \otimes\text{-nodes}(\Gamma),\ &\deg(\alpha) = r,\\ &\deg(\text{pred}(\alpha)) = \{d, r - d\}\},\end{aligned}$$

irrespective of our choice of Γ. We use $\bar{c}$ to construct a lower bound $\underline{w}$ on w, which depends only on the degree of the root.

THEOREM 3.4. *If $\underline{w}$ is defined by*

$$\underline{w}(1) = 0, \tag{3.3}$$

$$\underline{w}(r) = \min_{1 \le d \le \lfloor r/2 \rfloor} \left\{ \underline{w}(d) + \underline{w}(r-d) + \frac{1}{\bar{c}(r, d)} \right\}, \tag{3.4}$$

then $w(PT(\alpha, m)) \ge \underline{w}(deg(\alpha))$ for all $\alpha \in nodes(\Gamma)$, $m \in mon(\alpha)$. In particular, $w(PT(\rho, m)) \ge \underline{w}(deg(p))$.

PROOF. We proceed by induction on tree structure. The result is clearly true when α is a leaf, and the induction step is trivial when α is a $\oplus$-node or $\otimes$-node with a predecessor of degree 0. Assume, therefore, that α is a $\otimes$-node with $\deg(\alpha) = r$, $\text{pred}(\alpha) = \{\beta, \gamma\}$, and $\deg(\beta) \le \deg(\gamma)$, in which case

$$\begin{aligned} w(\text{PT}(\alpha, m)) &= w(\text{PT}(\beta, m_\beta)) + w(\text{PT}(\gamma, m_\gamma)) + |\text{content}(\alpha)|^{-1} \\ &\ge \underline{w}(\deg(\beta)) + \underline{w}(\deg(\gamma)) + \frac{1}{\bar{c}(r, \deg(\beta))} \quad \text{(by induction hypothesis)} \\ &\ge \min_{1 \le d \le \lfloor r/2 \rfloor} \left\{ \underline{w}(d) + \underline{w}(r-d) + \frac{1}{\bar{c}(r, d)} \right\} \\ &= \underline{w}(r). \end{aligned}$$

□

It may be remarked that the theorem remains true if the equalities of (3.3) and (3.4) are replaced by inequalities ($\le$). This observation can be useful if an exact solution to the original equations is hard to obtain.

COROLLARY 3.5. *For linear, homogeneous p,*

$$|mon(p)| \cdot \underline{w}(deg(p)) \le \otimes\text{-}complexity\ of\ p.$$

PROOF. Applying Theorem 3.3 to Γ_{opt}, we have

$$\sum_{m \in \text{mon}(\rho)} w(\text{PT}(\rho, m)) \le \otimes\text{-complexity of } p,$$

and applying Theorem 3.4, we obtain

$$\sum_{m \in \text{mon}(\rho)} \underline{w}(\deg(p)) \le \otimes\text{-complexity of } p.$$

□

In the next section we compute content bounds for specific polynomials and derive the corresponding weight bounds. We show that for several polynomials the lower bound implied by Corollary 3.5 is tight. In order to help us solve the recurrences (3.3) and (3.4) we introduce a final lemma.

LEMMA 3.6. *If for all $4 \le r \le n$, $1 \le d \le \lfloor r/2 \rfloor - 1$,*

$$\frac{1}{\bar{c}(r, d+1)} + \frac{1}{\bar{c}(d+1, 1)} - \frac{1}{\bar{c}(r, d)} - \frac{1}{\bar{c}(r-d, 1)} \ge 0 \tag{3.5}$$

is satisfied, then the solution to (3.3) and (3.4) is

$$\underline{w}(r) = \sum_{i=2}^{r} \frac{1}{\bar{c}(i, 1)}. \tag{3.6}$$

PROOF. By induction on r. The lemma is trivially true for $r = 2, 3$; otherwise,

$$\underline{w}(r) = \min_{1 \le d \le \lfloor r/2 \rfloor} \left\{ \sum_{i=2}^{d} \frac{1}{\bar{c}(i, 1)} + \sum_{i=2}^{r-d} \frac{1}{\bar{c}(i, 1)} + \frac{1}{\bar{c}(r, d)} \right\}$$

$$= \min_{1 \le d \le \lfloor r/2 \rfloor} g(d).$$

The observation is that g is a monotonically increasing function in the range $1 \le d \le \lfloor r/2 \rfloor$, since

$$g(d+1) - g(d) = \frac{1}{\bar{c}(d+1, 1)} + \frac{1}{\bar{c}(r, d+1)} - \frac{1}{\bar{c}(r-d, 1)} - \frac{1}{\bar{c}(r, d)}$$

$$\ge 0 \qquad \text{by stated condition.}$$

Thus

$$\underline{w}(r) = g(1) = \sum_{i=2}^{r} \frac{1}{\bar{c}(i, 1)}. \qquad \square$$

4. *Complexity of Specific Polynomials*

4.1 ITERATED MATRIX MULTIPLICATION. Suppose $X^{(1)}, X^{(2)}, \ldots, X^{(t)}$ are $n \times n$ matrices; $X^{(k)} = x_{ij}^{(k)}$ $(1 \le i, j \le n)$. We are interested in the number of multiplications required to compute the product

$$(X^{(1)}X^{(2)} \cdots X^{(t)})_{ij} = \bigoplus_{1 \le i_j \le n} x_{i_1 i_2}^{(1)} x_{i_2 i_3}^{(2)} x_{i_3 i_4}^{(3)} \cdots x_{i_t j}^{(t)}.$$

We note that any computation for the above can be converted into a computation for the related polynomial

$$p = \bigoplus_{1 \le i_k \le n} x_{i_1 i_2}^{(1)} x_{i_2 i_3}^{(2)} \cdots x_{i_t i_{t+1}}^{(t)} x_{i_{t+1} i_1}^{(t+1)},$$

with the addition of at most n^2 ⊗-nodes. The number of multiplications necessary for matrix multiplication is thus no smaller than (⊗-complexity of p) $- n^2$.

The first step in establishing a bound on the complexity of p is to compute a suitable content bound $\bar{c}(r, d)$. Suppose q is a polynomial with indeterminates $x_{ij}^{(k)}$. Define the index set I_q to be the set of superscripts of the indeterminates occuring in q. Now consider polynomials a, b, and c of degrees d, $r - d$, and $t - r + 1$, respectively, with the property that $\text{mon}(abc) \subseteq \text{mon}(p)$. Looking at the form of monomials of p, we see immediately that I_a, I_b, I_c are disjoint, and, moreover, $|I_a| \ge d$, $|I_b| \ge r - d$, $|I_c| \ge t - r + 1$. Hence $\{I_a, I_b, I_c\}$ is a partition of $\{1, 2, \ldots, t + 1\}$. Define the set A of *articulations* to be

$$A = \{k \mid (2 \le k \le t + 1,\ k \text{ and } k - 1 \text{ are in distinct index sets})$$
$$\vee\ (k = 1,\ 1 \text{ and } t + 1 \text{ are in distinct index sets})\}.$$

Next consider a general element of $\text{mon}(abc)$,

$$x_{i_1 i_2}^{(1)} x_{i_2 i_3}^{(2)} \cdots x_{i_{t+1} i_1}^{(t+1)}.$$

Observe that if k is an articulation $(k \in A)$, then i_k is necessarily fixed by the condition $\text{mon}(abc) \subseteq \text{mon}(p)$; otherwise i_k is free to assume the n possible values. Hence,

$$|\text{mon}(abc)| \le n^{t+1-|A|}.$$

If $r < t + 1$, then $I_a, I_b, I_c \neq \varnothing$, which implies $|A| \geq 3$; if $r = t + 1$, then $I_a, I_b \neq \varnothing$, $I_c = \varnothing$, and $|A| \geq 2$. Consequently, we take as our content bound,

$$\bar{c}(r, d) = \begin{cases} n^{t-2} & (r < t + 1), \\ n^{t-1} & (r = t + 1). \end{cases}$$

The recurrence relations (3.3) and (3.4) are easily solved in this case, where $\bar{c}$ is essentially a constant. Condition (3.5) is trivially satisfied, and so, invoking Lemma 3.6, we obtain

$$\underline{w}(t + 1) = \sum_{i=2}^{t+1} \frac{1}{\bar{c}(i, 1)} = (t - 1)n^{(2-t)} + n^{(1-t)}.$$

Hence, by Corollary 3.5,

$$\begin{aligned} \otimes\text{-complexity of } p &\geq [(t - 1)n^{(2-t)} + n^{(1-t)}]\,|\mathrm{mon}(p)| \\ &= (t - 1)n^3 + n^2, \end{aligned}$$

and by our initial observation, the number of multiplications required for matrix multiplication is $(t - 1)n^3$. (For the case $t = 2$, our result is implied by a stronger one, obtained in [10, 15, 16], for the monotone Boolean matrix product.) The obvious algorithm, derived from the definition of matrix multiplication, yields an upper bound of $(t - 1)n^3$ and illustrates that our bound is tight. Note that since p is linear and homogeneous, the conditions of Corollaries 2.7 and 2.10 are satisfied, and our lower bound is valid for matrix multiplication over R, M, and M^+.

4.2 ITERATED WRAPPED CONVOLUTION. Suppose $\bar{x}^{(1)}, \bar{x}^{(2)}, \ldots, \bar{x}^{(t)}$ are n-vectors, $\bar{x}^{(k)} = x_i^{(k)}$ $(0 \leq i \leq n - 1)$. The *wrapped convolution* of these vectors is the n-vector $\bar{y}$ whose components are given by

$$y_j = \bigoplus_{i_1+i_2+\cdots+i_t \equiv j \pmod{n}} x_{i_1}^{(1)} x_{i_2}^{(2)} \cdots x_{i_t}^{(t)}.$$

As before, we define a related polynomial,

$$p = \bigoplus_{i_1+i_2+\cdots+i_{t+1} \equiv 0 \pmod{n}} x_{i_1}^{(1)} x_{i_2}^{(2)} \cdots x_{i_t}^{(t)} x_{i_{t+1}}^{(t+1)},$$

where $\bar{x}^{(t+1)}$ is an n-vector, and remark that the number of multiplications required to compute $\bar{y}$ is at least ($\otimes$-complexity of p) $- n$.

Consider polynomials a, b, and c of degrees d, $r - d$, and $t - r + 1$, respectively, with the property that $\mathrm{mon}(abc) \subseteq \mathrm{mon}(p)$. As before, define the index set I_q of a polynomial q to be the set of all superscripts occurring in the indeterminates which form q. Again, I_a, I_b, I_c form a partition of $\{1, 2, \ldots, t + 1\}$. If we now consider a general monomial,

$$m_a m_b m_c = x_{i_1}^{(1)} x_{i_2}^{(2)} \cdots x_{i_{t+1}}^{(t+1)},$$

of $\mathrm{mon}(abc)$, we see from the definition of p that

$$\sum_{k\in I_a} i_k + \sum_{k\in I_b} i_k + \sum_{k\in I_c} i_k \equiv 0 \pmod{n}$$

and, letting m_a range over $\mathrm{mon}(a)$ while holding m_b, m_c fixed, we deduce that $\sum_{k\in I_a} i_k$ is congruent to a constant, modulo n. Similar arguments apply for I_b, I_c, and hence $|\mathrm{mon}(abc)|$ is bounded by the number of assignments which can be made to $i_1, i_2, \ldots, i_{t+1}$ and which fix the above three sums. If $r < t + 1$, then I_a, I_b, I_c are all nonempty, and the number of assignments which can be made is $n^{(t-2)}$; if $r = t + 1$, then $I_c = \varnothing$, and there are $n^{(t-1)}$ possible assignments. Our content bound is thus

$$\bar{c}(r, d) = \begin{cases} n^{(t-2)} & (r < t + 1), \\ n^{(t-1)} & (r = t + 1). \end{cases}$$

Observing that this bound is identical to that derived in the previous section, we can immediately write down

$$\underline{w}(t + 1) = (t - 1)n^{(2-t)} + n^{(1-t)},$$

and so, by Corollary 3.5,

$$\begin{aligned} \otimes\text{-complexity of } p &\geq [(t - 1)n^{(2-t)} + n^{(1-t)}]\,|\text{mon}(p)| \\ &= (t - 1)n^2 + n. \end{aligned}$$

The number of multiplications required to compute the wrapped convolution is thus at least $(t - 1)n^2$. That this bound is tight may be seen by considering the algorithm derived from the definition. Again the bound is valid for R, M and M^+.

4.3 PERMANENT. Suppose X is an $n \times n$ matrix of indeterminates, x_{ij} $(1 \leq i,j \leq n)$. The *permanent function* on X is defined to be

$$\text{per}_{n\times n}(X) = p = \bigoplus_{\pi \in S(n)} x_{1,\pi(1)} x_{2,\pi(2)} \cdots x_{n,\pi(n)},$$

where $S(n)$ is the set of all permutations of the first n natural numbers. The permanent is of great significance in combinatorial mathematics, and a comprehensive account of it is available in [12]. If we associate indeterminate x_{ij} with edge ij in the complete graph K_n on the nodes $\{1, 2, \ldots, n\}$, we see that monomials of $\text{per}_{n\times n}(X)$ correspond to cycle covers of K_n. Over R, taking X to be the adjacency matrix of an arbitrary graph G on n nodes, the permanent can be interpreted as enumerating such cycle covers in G; over M^+, regarding X as a weight function on the edges of G, the interpretation is of finding the minimal weight cover. Again, if x_{ij} is associated with edge ij' in the complete bipartite graph B_n on the $2n$ nodes $\{1, \ldots, n, 1', \ldots, n'\}$, then the monomials of $\text{per}_{n\times n}(X)$ correspond to perfect matchings in B_n. Over R and M^+ the permanent has the interpretation of counting perfect matchings and finding the minimal matching, respectively, in a bipartite graph.

Despite its formal similarity with the determinant function, the permanent appears, in general, to be much more difficult to compute, for whereas the determinant can be computed in $O(n^{2.52})$ multiplications/divisions [19], the evaluation of the permanent for 0, 1 matrices is known to be #P-complete [25], and the permanent function itself is algebraically complete [26]. However, the min, + interpretation of the permanent as minimal matching in a bipartite graph (the so-called "assignment problem") is tractable, and an $O(n^3)$ algorithm can be found in [8].

To study the complexity of monotone computation of the permanent, we first determine a content bound. Suppose a, b, and c are polynomials of degrees d, $r - d$, and $n - r$, respectively, with $\text{mon}(abc) \subseteq \text{mon}(p)$. If q is a polynomial with indeterminates x_{ij}, we denote by I_q and J_q the sets

$$\begin{aligned} I_q &= \{i \,|\, x_{ij} \text{ occurs in } q\}, \\ J_q &= \{j \,|\, x_{ij} \text{ occurs in } q\}. \end{aligned}$$

If we consider a general element of $\text{mon}(abc)$,

$$m_a m_b m_c = x_{1,\pi(2)} x_{2,\pi(2)} \cdots x_{n,\pi(n)},$$

we can see that the sets I_a, I_b, and I_c are disjoint and

$$|I_a| = d, \qquad |I_b| = r - d, \qquad |I_c| = n - r,$$

so that $\{I_a, I_b, I_c\}$ is a partition of $\{1, 2, \ldots, n\}$. Since π is a permutation, the same argument yields that $\{J_a, J_b, J_c\}$ is also a partition. Elements of mon(abc) correspond to permutations π which observe the restrictions

$$\pi(I_a) = J_a, \qquad \pi(I_b) = J_b, \qquad \pi(I_c) = J_c.$$

The total number of such permutations is clearly $d!(r - d)!(n - r)!$, and so we may take as our content bound,

$$\bar{c}(r, d) = d!(r - d)!(n - r)!$$

We claim that this bound satisfies the condition (3.5). In order to show this we need the following, easily validated lemma.

LEMMA 4.1. *If $s \geq 2$, then $\binom{s+r}{r} \geq r(s + 1)$.*

Using the substitutions $s = r - 2d$ and $t = n - r$, condition (3.5) becomes

$$\frac{1}{(d+1)!(s+d-1)!t!} + \frac{1}{d!(s+d+t-1)!} - \frac{1}{d!(s+d)!t!} - \frac{1}{(s+d-1)!(t+d)!} \geq 0$$

for $d \geq 1$, $s \geq 2$, $t \geq 0$. Equivalently, by multiplying through by $(d + 1)!(s + d - 1)!t!$ we obtain

$$f(t, d, s) = 1 + \frac{d+1}{\binom{d+s+t-1}{t}} - \frac{d+1}{d+s} - \frac{d+1}{\binom{t+d}{t}} \geq 0$$

for $d \geq 1$, $s \geq 2$, $t \geq 0$. We remark that $f(t, d, s)$ is a monotonically increasing function of s, for

$$f(t, d, s+1) - f(t, d, s) = (d+1)\left[\frac{1}{(d+s+1)(d+s)} - \frac{t}{(d+s+t)\binom{d+s+t-1}{t}}\right],$$

and we have two cases:

(1) $t = 0$: the difference is clearly positive.
(2) $t > 0$:

$$\begin{aligned} f(t, d, s+1) - f(t, d, s) &\geq \frac{(d+1)[1/(d+s) - t/\binom{d+s+t-1}{t}]}{d+s+1} \\ &\geq \frac{(d+1)[1/(d+s) - 1/(d+s)]}{d+s+1} \qquad \text{(by Lemma 4.1)} \\ &= 0. \end{aligned}$$

Hence it is sufficient to show that $f(t, d, s) \geq 0$ when $s = 2$. However,

$$\begin{aligned} f(t, d, 2) &= 1 + \frac{d+1}{\binom{d+t+1}{t}} - \frac{d+1}{d+2} - \frac{d+1}{\binom{t+d}{t}} \\ &= \frac{1}{d+2} + \frac{(d+1)^2}{(d+t+1)\binom{t+d}{t}} - \frac{d+1}{\binom{t+d}{t}} \\ &= \frac{1}{d+2} - \frac{t(d+1)}{(d+t+1)\binom{t+d}{t}}. \end{aligned}$$

We consider three cases

(1) $t = 0$: $f(0, d, 2) = 1/(d + 2) \geq 0$.
(2) $d = 1$: $f(t, 1, 2) = 1/3 - 2t/(t + 1)(t + 2) \geq 0$.
(3) $d > 1, t > 0$: $f(t, d, 2) \geq 1/(d + 2) - 1/(d + t + 1) \geq 0$ (by Lemma 4.1).

Thus we have shown that condition (3.5) holds and deduce from (3.6) that

$$\underline{w}(n) = \sum_{i=2}^{n} \frac{1}{(i-1)!(n-i)!} = \frac{\left[\sum_{i=2}^{n} \binom{n-1}{i-1}\right]}{(n-1)!} = \frac{2^{n-1}-1}{(n-1)!}.$$

By Corollary 3.5,

$$\otimes\text{-complexity of } p \geq \frac{n!(2^{n-1}-1)}{(n-1)!} = n(2^{n-1}-1).$$

This lower bound is in fact attained by the permanental equivalent of Laplace's expansion rule for determinants. Essentially, we form the permanents of all the $r \times r$ submatrices contained in the first r rows of X (the "subpermanents" of the first r rows) recursively from all the $(r-1) \times (r-1)$ subpermanents of the first $r-1$ rows. The upper bound of $n(2^{n-1}-1)$ is then implied by the recurrence $C(r) = C(r-1) + r\binom{n}{r}$, $C(1) = 0$. Clearly we can obtain variants of this algorithm by permuting rows of X and transposing X; what is more interesting is that the optimal algorithm lacks uniqueness in a nontrivial way, this stemming from the observation that several "shapes" of parse-tree all have optimal weight. More precisely, when $r = n - 1$, the value of $\underline{w}(d) + \underline{w}(r-d) + 1/\bar{c}(r-d)$ is independent of d, which leads to the following family of optimal algorithms for $1 \leq t \leq n-2$:

(i) Evaluate all $t \times t$ subpermanents of the first t rows using Laplace's expansion.
(ii) Evaluate the $(n-t-1) \times (n-t-1)$ subpermanents of the rows $t+1$, $t+2, \ldots, n-1$ in the same way.
(iii) Use the results of (i) and (ii) to compute all $(n-1) \times (n-1)$ subpermanents of the first $n-1$ rows.
(iv) From (iii) compute per(X) by Laplace's expansion.

We note that, once more, the lower bound is valid for R, M, and M^+.

4.4 Hamiltonian Circuit Polynomial. Suppose again that X is an $n \times n$ matrix of indeterminates, x_{ij} $(1 \leq i, j \leq n)$. The *Hamiltonian circuit polynomial* is

$$\mathrm{HC}_{n\times n} = p = \bigoplus_{\pi \in C(n)} x_{1,\pi(1)} x_{2,\pi(2)} \cdots x_{n,\pi(n)},$$

where $C(n)$ is the set of all *cyclic* permutations of the first n natural numbers. If we regard each indeterminate x_{ij} as representing the ij edge of the complete graph K_n on the n nodes $\{1, 2, \ldots, n\}$, we note that the monomials of p correspond to Hamiltonian circuits in K_n. Over R, the polynomial can be viewed as enumerating Hamiltonian circuits on a graph with n nodes, while the corresponding interpretation for M^+ is that of finding the shortest circuit which covers all nodes of a graph—the so-called Traveling Salesman Problem.

In the usual way we let a, b, and c be polynomials of degrees d, $r-d$, and $n-r$, respectively, with $\mathrm{mon}(abc) \subseteq \mathrm{mon}(p)$. Using the same reasoning as for the permanent, a, b, and c define two partitions of $\{1, 2, \ldots, n\}$, namely,

$$\{I_a, I_b, I_c\} \qquad \text{and} \qquad \{J_a, J_b, J_c\}.$$

If we consider a general monomial of abc,

$$m_a m_b m_c = x_{1,\pi(1)} x_{2,\pi(2)} \cdots x_{n,\pi(n)},$$

we have that

$$\pi(I_a) = J_a, \qquad \pi(I_b) = J_b, \qquad \text{and} \qquad \pi(I_c) = J_c,$$

and so $|\mathrm{mon}(abc)|$ is bounded by the number of cyclic permutations π which satisfy these constraints. Suppose we fix m_b, m_c, that is, fix π on $I_b \cup I_c$; we wish to know the number of possible choices for m_a, that is, the number of ways of extending π to I_a. Define

$$\pi^*: I_a \to I_a, \qquad \pi^*(i) = \pi^\alpha(i),$$

where α is the smallest positive number such that $\pi^\alpha(i) \in I_a$ (such an α exists since π is cyclic). Informally, viewing π as a circle, π^* is the circle obtained by deleting vertices not in I_a. Note that π is completely determined by π^* and the restriction of π to $I_b \cup I_c$. We observe that π^* is a cyclic permutation; hence the number of extensions of π to I_a is bounded by the number of cyclic permutations on d objects; that is,

$$|\mathrm{mon}(a)| \leq (d-1)!.$$

Similarly,

$$|\mathrm{mon}(b)| \leq (r-d-1)!$$

and

$$|\mathrm{mon}(c)| \begin{cases} \leq (n-r-1)! & (r < n), \\ = 1 & (r = n), \end{cases}$$

the second case being the degenerate one when $I_c = \varnothing$. Consequently, we take as our content bound

$$\bar{c}(r, d) = \begin{cases} (d-1)!(r-d-1)!(n-r-1)! & (r < n), \\ (d-1)!(n-d-1)! & (r = n). \end{cases}$$

By an argument completely analogous to the case of the permanent, we can show that this bound satisfies condition (3.5), and hence applying (3.6) we obtain

$$\begin{aligned} \underline{w}(n) &= \sum_{i=2}^{n-1} \frac{1}{(i-2)!(n-i-1)!} + \frac{1}{(n-2)!} \\ &= \sum_{i=2}^{n-1} \frac{\binom{n-3}{i-2}}{(n-3)!} + \frac{1}{(n-2)!} \\ &= \frac{2^{(n-3)}(n-2)+1}{(n-2)!}. \end{aligned}$$

By Corollary 3.5,

$$\begin{aligned} \otimes\text{-complexity of } p &\geq \frac{(n-1)![2^{(n-3)}(n-2)+1]}{(n-2)!} \\ &= (n-1)[(n-2)2^{(n-3)}+1]. \end{aligned}$$

Again this bound is valid for R, M, and M^+, and is attainable. Let $p_{i,I,j}$ for $i, j \in \{1, 2, \ldots, n\}$, $I \subseteq \{1, 2, \ldots, n\} \backslash \{i, j\}$, be the polynomial corresponding to all Hamiltonian paths in K_n which start at node i, end at node j, and pass through all the nodes in I. We may form $p_{1,I,j}$ recursively as follows:

$$\begin{aligned} p_{1,\varnothing,j} &= x_{1j}, \\ p_{1,I,j} &= \sum_{i \in I} p_{1,I\backslash i,i}\, x_{ij} \qquad (I \neq \varnothing). \end{aligned}$$

Generating all $p_{1,I,j}$ ($|I| = r$) given $p_{1,I',j'}$($|I'| = r-1$) can be achieved using

$r(n-1)\binom{n-2}{r}$ multiplications, so we can compute $p_{1,\{2,3,\ldots,n\}\setminus j,j}$ in

$$\sum_{r=1}^{n-2} r(n-1)\binom{n-2}{r} = (n-1)(n-2)\sum_{r=1}^{n-2}\binom{n-3}{r-1}$$
$$= (n-1)(n-2)2^{(n-3)}$$

multiplications. Now

$$\mathrm{HC}_{n\times n} = \bigoplus_{j=2}^{n} p_{1,\{2,3,\ldots,n\}\setminus j,j}x_{j1},$$

so it can be computed in

$$(n-1)(n-2)2^{(n-3)} + (n-1)$$

multiplications. We highlight a close connection which exists between the Hamiltonian circuit polynomial and self avoiding walks in a graph. Define the *self avoiding walk polynomial* to be

$$\mathrm{SAW}_{n\times n} = \bigoplus_{k=1}^{n-1} \bigoplus_{\substack{i_0,i_1,\ldots,i_k \text{ distinct}\\ i_0=1,\, i_k=n}} x_{i_0i_1}x_{i_1i_2}\cdots x_{i_{k-1}i_k},$$

so that monomials correspond to simple paths from 1 to n in the complete graph K_n. We remark that

$$\otimes\text{-complexity of } \mathrm{HC}_{(n-1)\times(n-1)} \le \otimes\text{-complexity of } p_{1,\{2,3,\ldots,n-1\},n},$$

since a computation for the latter may be transformed into one for the former by changing the inputs $x_{i,n}$ to x_{i1}. Hence our lower bound for $\mathrm{HC}_{n\times n}$ implies a lower bound of $(n-2)[(n-3)2^{(n-4)}+1]$ multiplications for $p_{1,\{2,3,\ldots,n-1\},n}$. However, $p_{1,\{2,3,\ldots,n-1\},n} = \mathrm{he}(\mathrm{SAW}_{n\times n})$, and so by Theorem 2.4 and Corollary 2.7 we obtain a lower bound of $(n-2)[(n-3)2^{(n-4)}+1]$ multiplications for $\mathrm{SAW}_{n\times n}$ when we are working with the semirings R and M (but, as will be shown, not M^+). Again we can attain the lower bound; we simply compute $p_{1,I,j}$ for all $j \ne n$, $n \notin I$ and note that

$$\mathrm{SAW}_{n\times n} = x_{1,n} \oplus \left[\bigoplus_{j\ne 1,n}\left\{\left(\sum_{n\notin I} p_{1,I,j}\right)x_{jn}\right\}\right].$$

4.5 Spanning Tree Polynomial. Suppose X is an $n \times n$ matrix of indeterminates x_{ij} $(1 \le i, j \le n)$. Define the spanning tree polynomial $\mathrm{ST}_{n\times n}$ to be

$$\mathrm{ST}_{n\times n} = p = \bigoplus_{t\in T(n)} x_{2,t(2)}x_{3,t(3)}\cdots x_{n,t(n)},$$

where $T(n) = \{t:\{2,3,\ldots,n\} \to \{1,2,\ldots,n\} \mid \forall i \exists k\, t^k(i) = 1\}$. Note that monomials correspond to directed trees spanning K_n and rooted at node 1. All our lower bounds so far have been attainable; in this case we are unable to obtain a precise lower bound. We therefore content ourselves with a crude bound on the content of a node, which, however, is good enough to yield an exponential lower bound on the $\otimes$-complexity of $\mathrm{ST}_{n\times n}$. Let a, b, and c be polynomials of degrees d, $r-d$, and $n-r-1$, respectively, satisfying $\mathrm{mon}(abc) \subseteq \mathrm{mon}(p)$. In the usual way we define the index set I_q of a polynomial q to be

$$I_q = \{i \mid x_{ij} \text{ is an indeterminate of } q\},$$

and note that $\{I_a, I_b, I_c\}$ is a partition of $\{2, 3, \ldots, n\}$. Let

$$X_i = \{x_{ij} \mid x_{ij} \text{ is an indeterminate of } a, b \text{ or } c\}.$$

Obviously, $\sum_{i=2}^{n} |X_i| \leq (n-1)^2$, but we may improve on this trivial bound by the following observation. Suppose $i_a \in I_a$ and $i_b \in I_b$; then the indeterminates $x_{i_a i_b}$ and $x_{i_b i_a}$ cannot both appear in $\bigcup_{i=2}^{n} X_i$, for if they did, $x_{i_a i_b}$ would appear in a, $x_{i_b i_a}$ would appear in b, and the invalid monomial $x_{i_a i_b} x_{i_b i_a}$ m would appear in mon(abc). Thus a better restriction is

$$\begin{aligned}\sum_{i=2}^{n} |X_i| &\leq (n-1)^2 - |I_a||I_b| - |I_b||I_c| - |I_c||I_a| \\ &= (n-1)^2 - d(r-d) - (r-d)(n-r-1) - (n-r-1)d \\ &= (n-1)^2 - d(r-d) - r(n-r-1).\end{aligned}$$

The number of monomials in mon(abc) is clearly bounded by the number of functions

$$t: \{2, 3, \ldots, n\} \to \{1, 2, \ldots, n\}$$

which respect $x_{i,t(i)} \in X_i$ for $2 \leq i \leq n$; this number is just $\prod_{i=2}^{n} |X_i|$. The product is maximized when $|X_i|$ is independent of i; thus,

$$|\text{mon}(abc)| \leq \bar{c}(r, d) = \left[\frac{(n-1)^2 - d(r-d) - r(n-r-1)}{n-1}\right]^{(n-1)}.$$

It has been remarked that if a tight lower bound is not required, the equalities (3.3) and (3.4) can be changed to inequalities ($\leq$) without affecting the veracity of Theorem 3.4. We show that, taking $\underline{w}$ to be

$$\begin{aligned}\underline{w}(r) &= 0 & &\left(r < \frac{n}{2}\right), \\ \underline{w}(r) &= \left(\frac{3n}{4}\right)^{(1-n)} & &\left(r \geq \frac{n}{2}\right),\end{aligned}$$

(3.3) and (3.4) are satisfied as inequalities. In fact, (3.3) is trivially satisfied, and (3.4) is satisfied providing we can show that $\bar{c}(r, d) \leq (3n/4)^{(n-1)}$ whenever $n/2 < r \leq n-1$ and $d \leq (r-d) < n/2$. Holding r constant, we find that $\bar{c}(r, d)$ achieves its maximum in the stated range at $d = r - \lceil \frac{1}{2}n \rceil + 1$, and, allowing r to vary, we find the overall maximum to be at $r = n - 1$. Therefore, in the range in question,

$$\begin{aligned}\bar{c}(r, d) &\leq \left[\frac{(n-1)^2 - \frac{1}{2}n(\frac{1}{2}n - 1)}{n-1}\right]^{(n-1)} \\ &= \left[\frac{3n^2/4 - 3n/2 + 1}{n-1}\right]^{(n-1)} \\ &\leq \left(\frac{3n}{4}\right)^{(n-1)}.\end{aligned}$$

But $|\text{mon}(p)| = n^{(n-2)}$ [13], and hence, by Corollary 3.5,

$$\otimes\text{-complexity of } p \geq \left(\frac{3n}{4}\right)^{(1-n)} n^{(n-2)} = n^{-1}\left(\frac{4}{3}\right)^{(n-1)}.$$

In this way we obtain an exponential bound valid for R, M, and M^+, that is, for the problems of counting the number of spanning trees of a graph or of finding in a graph such a tree of minimal weight.

5. *Discussion of Results*

We have obtained lower bounds for the $\otimes$-complexity of a wide range of functions in different semirings. Some of these results, such as the exponential lower bound for the minimal spanning tree computation, stand in stark contrast to the known tractability of the problem and raise questions as to the relevancy of our results to actual computations. Our lower bounds can therefore be interpreted in two complementary ways; on the one hand they deny the existence, for many problems, of fast "combinatorial" algorithms which work independently of the domain of computation, while on the other hand they affirm the power of algorithms which exploit the algebraic idiosyncrasies of a specific problem. In Sections 5.1 and 5.2 we explore the efficiency which can be gained by using less restrictive models of computation.

5.1 The Power of Negation. Our model of computation suffers from two weaknesses. The more obvious is the restriction on the allowed operations. In the arithmetic case we considered only computations not involving subtraction. It was already known that such a restriction could entail an exponential penalty; [27] treats the example of perfect matchings in a planar graph. In the same vein, our results indicate an exponential gap for the spanning tree polynomial. From Section 4.5 we see that any monotone arithmetic computation for the spanning tree polynomial requires at least $n^{-1}(\frac{4}{3})^{(n-1)}$ multiplications; in contrast, if negative constants are allowed, the same polynomial can be expressed as an $n \times n$ determinant whose elements are linear combinations of the indeterminates [13], and this determinant can then be evaluated (without divisions) in $O(n^{3\,52})$ multiplications using the method of Strassen [23]. Even for functions which have polynomial monotone complexity, subtraction is still helpful. From Section 4.1 we have that in the monotone case, multiplication of two $n \times n$ matrices requires at least n^3 multiplications, whereas, allowing negative constants, Schonhage's method [19] computes the product in $O(n^{2\,52})$ multiplications. Similarly, a gain can be achieved for the convolution of Section 4.2 using the fast Fourier transform method [1, p. 257]. A very modest gain can be demonstrated for the permanent function: referring to Section 4.3, any monotone computation requires at least $n(2^{(n-1)} - 1)$ multiplications; however, using a modification of the inclusion-exclusion method of Ryser [14, p. 158], the same computation can be effected using subtraction in only $(n - 1)2^{(n-1)} + 3$ multiplications. The interest in this case is that, although small, the complexity gap is the only one we know for a 0-1 polynomial which is complete in the sense of [26].

All this evidence points to the value of complex algorithms which exploit the particular characteristics of the domain of computation, in this case, the ability to form monomials which cancel out in subtle ways in the final result. Of particular interest is the power of linear algebra to make tractable polynomials whose monotone complexity is exponential. By contrast, it is noteworthy that augmenting our set of operations with division and performing computation over the rational functions is of limited value, as division can be stimulated by truncated power series [23].

5.2 The Power of Branching. The second weakness of our model is less obvious, since it is not usually encountered in algebraic complexity. We used what is essentially a straight-line algorithm (sla) model to measure the complexity of computation, neglecting the additional computational power that branching (test and branch instructions) can provide. It is well known [24] that branching cannot help in the computation of polynomials over an infinite field, so that our model is adequate for R in this respect. The situation is, however, completely different in M or M^+, where branching can yield dramatically shorter computations. To return to the example of the directed spanning tree polynomial from Section 4.5, we learn that $n^{-1}(\frac{4}{3})^{(n-1)}$ additions are necessary to compute the polynomial using an sla, whereas it can be computed in $O(n^2 \log n)$ min, + operations if we allow branching [8, p. 348]. As another demonstration of an exponential gain we might consider the permanent, which over M^+ is connected with the minimal assignment problem. In our model the permanent requires $n(2^{(n-1)} - 1)$ + operations but with branching can be computed in $O(n^3)$ operations [8, p. 205]. Indeed, we can paraphrase [27] and assert that "branching can be exponentially powerful."

5.3 FORMALLY IDENTICAL POLYNOMIALS OVER DIFFERENT SEMIRINGS. Another lesson we may draw is that the algebraic idiosyncrasies of different semirings can cause the same formal polynomial to have radically different complexities. In fact, we have only one consistent relation between the semirings we examined: it was always easier to compute a 0-1 polynomial in B than in M, M^+, or R. (Loosely speaking, checking the existence of a solution to a problem is always easier than finding the minimal solution or counting their number.) This gap can be exponential; the spanning tree polynomial ST has exponential complexity over M, M^+, and R but polynomial complexity over B. Over B, $\mathrm{ST}(X) = 1$ if and only if the graph whose adjacency matrix is X has a spanning tree, that is to say, if and only if it is connected. However, the connectedness function can be monotonically computed in $O(n^3)$ operations by computing the transitive closure X^* of X and AND-ing n^2 coefficients of X^* [1, p. 199]. Another interesting example is provided by the self-avoiding walk polynomial SAW. If x_{ij} is the length of edge ij in K_n, then $\mathrm{SAW}(x_{ij})$ is (in M, M^+) the minimal length of a self-avoiding walk fron node 1 to node n. In M^+ this is equal to the minimal length of a path from 1 to n and can be computed in $O(n^3)$ operations [1, p. 202]. In M, however, the problem has exponential complexity. The same exponential bound is valid in R for the problem of counting the number of self-avoiding walks (where x_{ij} is the adjacency matrix of the graph).

5.4 OTHER SEMIRINGS. The results presented here may be extended to other algebraic structures. We can, for example, consider the semiring of real numbers together with $-\infty$ and the operations max and +. It is dual to the M semiring, and the same results are valid. Thus, computing the maximal weight of a spanning tree, the maximal length of a self-avoiding walk and the maximal weight of a matching all require an exponential number of operations, even when computation is restricted to positive inputs.

One might also consider semirings in which we drop the condition that multiplication is commutative. For example, letting $X = \{x_1, \ldots, x_n\}$, we might consider the semiring $L[X]$ of finite languages over X with the operation $\otimes$ being concatenation and $\oplus$ being union. By viewing $B[X]$ as the quotient of $L[X]$ by the commutative identity $x \otimes y = y \otimes x$, we can see that any lower bounds obtained for $B[X]$ are valid also for $L[X]$. In order to obtain tight bounds, however, we may need to take into account the noncommutativity of $\otimes$ by redefining the content function in an obvious way. Let the complement of a node α be the set of ordered pairs of monomials,

$$\mathrm{complement}(\alpha) = \{(m_1, m_2) \mid \forall m \in \mathrm{mon}(\alpha),\ m_1 m m_2 \in \mathrm{mon}(\rho)\},$$

and define the content of α to be

$$\mathrm{content}(\alpha) = \{m_1 m m_2 \mid m \in \mathrm{mon}(\alpha),\ (m_1, m_2) \in \mathrm{complement}(\alpha)\}.$$

Using this amended definition, the arguments of Section 3 go through as before, and we can achieve nontrivial lower bounds such as $(2^n - 2)$ for the number of concatenations required to compute

$$\bigcup_{\pi \in S(n)} x_{\pi(1)} x_{\pi(2)} \cdots x_{\pi(n)},$$

the language of all strings which are permutations of the symbols $x_1, x_2, \ldots, x_n$ (see [6] for related results).

5.5 FUTURE DIRECTIONS. Perhaps the most obvious shortfall in our technique is our inability to produce tight lower bounds of the number of additions required to compute a function which would match our results for multiplications. The best tools at our disposal at present are the "separated sets" of Schnorr [18] and the "fan in" argument of Shamir and Snir [20]. It would be interesting to find good absolute bounds or demonstrate some trade-off between the number of multiplications and the number of additions necessary to compute a function.

We have already mentioned that our results do not extend to unrestricted arithmetic computation; the hope is however that some refinement of the approach might enable us to deal with monotone Boolean computation. At present, lower bounds have been achieved only for linear and bilinear functions, and attempts to extend methods for monotone arithmetic functions to Boolean functions have not been fruitful; a conjecture that the technique of [18] could be applied to monotone Boolean computations has been disproved by Wegener [29].

Another possible extension of the method is to define new "content" and "degree" functions with the same formal properties as the familiar ones used here. This approach is used in [21, 22] to obtain lower bounds for the symmetric function (i.e., that function which contains all the linear monomials of given degree over a fixed set of indeterminates) and can quite possibly be used to extend the results obtained here for wrapped convolution to polynomial multiplication.

Finally, further research on the complexity of algorithms set in formal, weak algebraic structures might yield a classification of problems as "algebraic" versus "combinatorial," that is, between those problems which have fast solutions based on the complex algebraic properties of the domain of computation and those which do not.

Appendix

The two characterization theorems, 2.5 and 2.8, are essentially equivalent forms of the basic separation theorem in convexity theory. We shall use it in the following formulation.

THEOREM A1 [FARKAS]. *Let $f, f_i \in \mathbb{R}^n$, $b, b_i \in \mathbb{R}$, for $i = 1, \ldots, k$. The following two assertions are equivalent.*

(*i*) *The system of inequalities*

$$\langle f_i \cdot \bar{u} \rangle \geq b_i, \qquad i = 1, \ldots, k,$$

implies the inequality

$$\langle f \cdot \bar{u} \rangle \geq b.$$

(*ii*) $\exists \lambda_1, \ldots, \lambda_k$ *such that*

$$\lambda_i \geq 0, \qquad f = \sum \lambda_i f_i, \qquad b \geq \sum \lambda_i b_i.$$

PROOF. See [5, Th. 4]. □

COROLLARY A2. *Let $m, m_i \in \mathbb{R}^n$, $c, a_i \in \mathbb{R}$, for $i = 1, \ldots, k$. The following two assertions are equivalent.*

(*i*) $\forall \bar{u} \in \mathbb{R}^n \ \langle m \cdot \bar{u} \rangle + c \geq min_i \langle m_i \cdot \bar{u} \rangle + a_i$.
(*ii*) $\exists \lambda_1, \ldots, \lambda_k$ *such that*

$$\lambda_i \geq 0, \qquad \sum \lambda_i = 1,$$
$$m = \sum \lambda_i m_i \qquad c \geq \sum \lambda_i a_i.$$

PROOF. (i) is equivalent to the assertion that in $\mathbb{R}^{n+1}$ the set of inequalities

$$\langle m_i \cdot \bar{u} \rangle + a_i \geq v, \qquad i = 1, \ldots, k,$$

implies the inequality

$$\langle m \cdot \bar{u} \rangle + c \geq v.$$

By taking, in Theorem A1, $f_i = (m_i, -1)$, $f = (m, -1)$, $b_i = -a_i$, and $b = -c$, the above assertion is seen to be equivalent to the existence of $\lambda_1, \ldots, \lambda_k$ such that

$$\lambda_i \geq 0, \qquad (m, -1) = \sum \lambda_i (m_i, -1), \qquad -c \leq \sum \lambda_i(-a_i),$$

which yields the desired result. □

Corollary A2 does in fact provide the required proof for Theorem 2.5. The validity of Theorem 2.8 follows from the following corollary.

COROLLARY A3. *Let* $m_i, m \in \mathbb{R}^n$, $a_i, c \in \mathbb{R}$, *for* $i = 1, \ldots, k$. *The following two assertions are equivalent.*

(*i*) $\forall \bar{u} \in \mathbb{R}^n, \bar{u} \geq 0 \Longrightarrow \langle m \cdot \bar{u} \rangle + c \geq min_i \langle m_i \cdot \bar{u} \rangle + a_i$.
(*ii*) $\exists \lambda_1, \ldots, \lambda_k$ *such that*

$$\lambda_i \geq 0, \qquad \sum \lambda_1 = 1,$$
$$m \geq \sum \lambda_i m_i, \qquad c \geq \sum \lambda_i a_i.$$

PROOF. (i) is equivalent to the assertion that in $\mathbb{R}^{n+1}$ the system of inequalities,

$$u_j \geq 0, \qquad j = 1, \ldots, n,$$
$$\langle m_i \cdot \bar{u} \rangle + a_i \geq v, \qquad i = 1, \ldots, k,$$

implies the inequality

$$\langle m \cdot \bar{u} \rangle + c \geq v.$$

Setting, in Theorem 2.5, $f_i = (m_i, -1)$ for $i = 1, \ldots, k$; $f_{k+j} = e_j$, the jth unit vector, for $j = 1, \ldots, n$; $b_i = -a_i$ for $i = 1, \ldots, k$; $b_{k+j} = 0$ for $j = 1, \ldots, n$; $f = (m, -1)$; and $a = -c$, we obtain that the above assertion is equivalent to the existence of $\lambda_1, \ldots, \lambda_{k+n}$ such that

$$\lambda_i \geq 0,$$
$$(m, -1) = \sum_{i=1}^{k} \lambda_i(m_i, -1) + \sum_{j=1}^{n} \lambda_{k+j}(e_j, 0),$$
$$-c \geq \sum_{i=1}^{k} \lambda_i(-a_i).$$

This yields the desired result. □

ACKNOWLEDGMENTS. Many ideas developed here originated from previous work of the first author with Leslie Valiant and of the second author with Eli Shamir. We would like to thank L. Valiant for his helpful comments.

REFERENCES

1 Aho, A V , Hopcroft, J E , and Ullman, J D *The Design and Analysis of Computer Algorithms.* Addison Wesley, Reading, Mass , 1974

2 Borodin, A , and Munro, I *The Complexity of Algebraic and Numerical Problems* American Elsevier, New York, 1974.

3. Cuninghame-Green, R. *Minimax Algebra* Springer-Verlag, Berlin, 1979.

4 Ehrenfeucht, A , and Zeiger, P Complexity measures for regular expressions Proc. 6th ACM Symp on Theory of Computing, Seattle, Wash , 1974, pp 75–79

5 Fan, K On systems of linear inequalities In *Linear Inequalities and Related Systems,* H W. Kuhn and A W Tucker, Eds , Princeton University Press, Princeton, N J , 1956, pp. 99–156.

6 Goodrich, G B , Ladner, R E., and Fisher, M J Straight-line programs to compute finite languages Proc Conf. on Theoretical Computer Science, Waterloo, Ontario, Canada, 1977, pp 221–229.

7 Lamagna, E A , and Savage, J.E Combinatorial complexity of some monotone functions Proc 15th IEEE Symp on Switching and Automata Theory, New Orleans, La , 1974, pp 140–144.

8 Lawler, E L *Combinatorial Optimisation. Networks and Matroids.* Holt, Rinehart and Winston, New York, 1976

9 Lee, S C *Modern Switching Theory* Prentice Hall, Englewood Cliffs, N J , 1978.

10 Mehlhorn, K , and Galil, Z Monotone switching circuits and Boolean matrix product *Comput.* *16* (1976), 99–111

11 Miller, W Computer search for numerical instability *J ACM 22,* 4 (Oct 1975), 512–521.

12 Minc, H *Permanents,* Encyclopedia of Mathematics and its Applications, 6. Addison-Wesley, Reading Mass , 1978

13 Moon, J W *Counting Labelled Trees.* Canadian Mathematical Congress, Montreal, 1970

14. Nijenhuis, A , and Wilf, H S. *Combinatorial Algorithms* Academic Press, New York, 1975.

15 Paterson, M S Complexity of monotone networks for Boolean matrix product *Theor Comput. Sci 1* (1975), 13–20

16. Pratt, V R The power of negative thinking in multiplying Boolean matrices *SIAM J. Comput 4* (1975), 326–330.

17 Redei, L *Algebra, Vol 1* Pergamon Press, Oxford, 1967

18 Schnorr, C P A lower bound on the number of additions in monotone computations *Theor. Comput Sci. 2* (1976), 305–315.

19 Schonhage, A Partial and total matrix multiplication Manuscript, Mathematisches Institut, Universitat Tubingen, Tubingen, Germany

20 Shamir, E., and Snir, M. Lower bounds on the number of multiplications and the number of additions in monotone computations Tech Rep RC 6757, IBM Thomas J. Watson Research Center, Yorktown Heights, N Y , 1977

21 Shamir, E , and Snir, M On the depth complexity of formulas To appear in *Math Syst. Theory.*

22 Snir, M On the size complexity of monotone formulas Tech Rep CSR-46-79, Univ. of Edinburgh, Edinburgh, Scotland, 1979 Also presented at the 7th Int Colloq on Automata, Language and Programming, Noordwijkerhout, Holland, July 1980.

23 Strassen, V Vermeidung von divisionen *J Reine Angew Math 264* (1973), 182–202.

24 Strassen, V Berechnung un program II *Acta Inf 2* (1973), 64–79

25 Valiant, L G The complexity of computing the permanent *Theor Comput Sci 8* (1979), 189–201

26 Valiant, L G Completeness classes in algebra Proc 11th ACM Symp on Theory of Computing, Atlanta, Ga , 1979, pp 249–261

27 Valiant, L G. Negation can be exponentially powerful. Proc. 11th ACM Symp. on Theory of Computing, Atlanta, Ga., 1979, pp 189–196

28. Wegener, I Switching functions whose monotone complexity is nearly quadratic *Theor Comput Sci 9* (1979), 83–87.

29 Wegener, I A counterexample to a conjecture of Schnorr referring to monotone networks *Theor Comput Sci 9* (1979), 147–150

RECEIVED MARCH 1980, REVISED FEBRUARY 1981, ACCEPTED FEBRUARY 1981