Abstract

A method for establishing lower bounds on the number of multiplications and divisions has been developed by Pan, Winograd and Strassen. A similar method is developed for establishing lower bounds on the number of additions and subtractions. The results obtained partially overlap those of Belaga, Winograd and Kirkpatrick.

Section 1

In this report we present a method for establishing lower bounds on the number of additions and subtractions required to compute rational functions over a field. The algorithms considered will be "straight line," and for the purpose of establishing lower bounds, there will be no loss of generality.

For a field F, a subset H of F and $a_1,\ldots,a_n$ indeterminates over F, an algorithm over $(F(a_1,\ldots,a_n),$ H $\cup$ $\{a_1,\ldots,a_n\})$ consists of a sequence of instructions which compute elements in $F(a_1,\ldots,a_n)$ (the set of rational functions in $a_1,\ldots,a_n$ over F) by applying chains of algebraic operations (from the set $\{+, -, \times, : \}$) to elements of H $\cup$ $\{a_1,\ldots,a_n\}$.

For example, if $F = R(x)$ (the set of real rational functions in x) and $H = R \cup \{x,x^2\}$ then the following is an algorithm computing $a_1x^2 + a_2x^3$ over $(R(x,a_1,a_2),$ $R \cup \{x,x^2,a_1,a_2\})$:

$$1. \quad a_2x \leftarrow (a_2) \times (x)$$
$$2. \quad a_1 + a_2x \leftarrow (a_1) + (a_2x)$$
$$3. \quad a_1x^2 + a_2x^3 \leftarrow (a_1 + a_2x) \times x^2$$

We have not defined the notion of an algorithm formally; for more formal definitions see for example [W67], [W70a], [K74].

Section 2

Our method for establishing lower bounds will be similar to those used by Pan [P66], Winograd [W67], [W70a] and Strassen [S72] for multiplications and divisions. Our results overlap partially results which were obtained before by Belaga [B61], Winograd [W70b] and Kirkpatrick [K72], but we feel that our method is useful, as it can be easily applied and permits

1

certain extensions which other methods do not. The fact that similar approaches may be used for both multiplications and divisions, and additions and subtractions, seems to indicate that a general principle exists for extablishing lower bounds on the number of operations in computations over algebraic structures.

Section 3

We shall start with an almost trivial example which will help to explain the motivation behind the method. We shall take a _particular_ algorithm, computing a _particular_ function and describe a sequence of reductions of this algorithm. Let $F = H = \mathbb{R}$ and let the following be an algorithm over $(F(a_1, a_2, a_3), F \cup \{a_1, a_2, a_3\})$ computing $a_1 + a_2 + a_3$:

1. $a_1 + a_2 \leftarrow (a_1) + (a_2)$
2. $a_1 + a_2 + a_3 \leftarrow (a_1 + a_2) + (a_3)$

This algorithm uses 2 AS's (additions or subtractions) to compute the sum of three algebraically independent elements. We shall show that if a_1, a_2, a_3 instead of being algebraically independent satisfy certain equations, then this algorithm reduces to another one, which has no AS's left.

We first note that without AS's only monomials, namely elements of the form

$$f \prod_1^3 a_i^{\alpha(i)}, \quad f \in F, \quad \alpha(i) \in \mathbb{Z} \tag{1}$$

can be computed. This statement can be proved very easily by (say) induction on the length of algorithms without AS's.

Thus, the first AS's in the algorithm <u>must</u> be of the form

$$f_1 \prod_1^3 a_1^{\alpha(1,i)} \;\pm\; f_2 \prod_1^3 a_i^{\alpha(2,i)} \tag{2}$$

and indeed $(a_1) + (a_2)$ is of this form. If we write the equation $a_1 + a_2 = \sqrt{2}\, a_2$ and assume that a_1 and a_2 satisfy it ($\sqrt{2}$ was chosen completely arbitrarily), then the algorithm computes

$$a_1 + a_2 + a_3 = \sqrt{2}\, a_2 + a_3 \quad \text{by}$$

1. $\sqrt{2}\, a_2 \leftarrow (\sqrt{2}) \times (a_2)$

2. $\sqrt{2}\, a_2 + a_3 \leftarrow (\sqrt{2}\, a_2) + a_3$

This algorithm has only one AS, and this being a first one in an algorithm is of the form (2).

If we write the equation

$$\sqrt{2}\, a_2 + a_3 = 2 a_2$$

and assume that a_2 and a_3 satisfy it (2 was chosen arbitrarily), then the algorithm computes

$$a_1 + a_2 + a_3 = \sqrt{2}\, a_2 + a_3 = 2\, a_2 \quad \text{by}$$

1. $\sqrt{2}\, a_2 \leftarrow (\sqrt{2}) \times (a_2)$

2. $2\, a_2 \leftarrow (2) \times (a_2)$

without any AS's.

Let's summarize: If a_1, a_2, a_3 satisfy the equations

$$a_1 a_2^{-1} = \sqrt{2} - 1$$

$$a_3 a_2^{-1} = 2 - \sqrt{2}$$

then $a_1 + a_2 + a_3$ is equal to a monomial and thus can be computed without any AS's.

We shall use similar procedures to show that _every_ algorithm computing certain rational functions has to have a minimum number of AS's.

Section 4

Let's assume now that we have four (pairwise disjoint) sets of indeterminates over F: $\{a_1,\ldots,a_n\}$, $\{b_1,\ldots,b_n\}$, $\{c_1,\ldots,c_n\}$, $\{d_1,\ldots,d_n\}$. We shall actually analyze algorithms computing elements of $F(a_1,\ldots,a_n)$, but we shall do it by reducing them to algorithms computing functions of other indeterminates. $c_1,\ldots,c_n$, $d_1,\ldots,d_n$ will (intuitively) play the role of parameters, so that we shall (informally) assume that we "do not count" AS's "involving" only elements of $H \cup \{c_1,\ldots,c_n, d_1,\ldots,d_n\}$. This will be formalized in the next definition.

In the sequel we shall frequently use $\underline{a}$ to denote both $\{a_1,\ldots,a_n\}$ and $a_1,\ldots,a_n$; similarly for $\underline{b}$, $\underline{c}$ and $\underline{d}$.

Let's also remark that an algorithm over $(F(\underline{a}),\ H \cup \underline{a})$ is also over $(F(\underline{a},\underline{b},\underline{c},\underline{d}),\ F(\underline{c},\underline{d}) \cup \{\underline{a},\underline{b}\})$ and all the algorithms in the sequel may always be considered to be over $F(\underline{a},\underline{b},\underline{c},\underline{d})$, $F(\underline{c},\underline{d}) \cup \{\underline{a},\underline{b}\}$). Furthermore, $F(\underline{a},\underline{b},\underline{c},\underline{d})$ will also be considered as a linear space over $F(\underline{c},\underline{d})$ so we shall be able to say when δ_1, $\delta_2 \in F(\underline{a},\underline{b},\underline{c},\underline{d})$ are linearly independent.

Section 5

<u>Definition</u>: An AS $\delta \leftarrow \delta_1 \pm \delta_2$ in an algorithm over (E,D) where

$E \subset F(\underline{a},\underline{b},\underline{c},\underline{d})$, $D \subset F(\underline{c},\underline{d}) \cup \{\underline{a},\underline{b}\}$

will be essential, or an EAS, if and only if δ_1 and δ_2 are linearly independent.

Section 6

Example: In the following algorithm

1. $c_1 + a_1 \leftarrow (c_1) + (a_1)$

2. $c_2c_1 + c_2a_1 \leftarrow (c_2) \times (c_1 + a_1)$

3. $(1 + c_2)(c_1 + a_1) = (c_1 + a_1) + (c_2c_1 + c_2a_1)$

1. is an EAS but 3. is not (as $c_1 + a_1$ and $c_2c_1 + c_2a_1$ are linearly dependent over $F(\underline{c},\underline{d})$).

Section 7

Remark: To justify the term "essential" we remark that if $\delta \leftarrow \delta_1 \pm \delta_2$ in an algorithm over $(F(\underline{a},\underline{b},\underline{c},\underline{d}), F(\underline{c},\underline{d}) \cup \{\underline{a},\underline{b}\})$ is not essential, then it can be replaced by a multiplication. Indeed we have

$f_1\delta_1 + f_2\delta_2 = 0$, $f_1, f_2 \in F(c_1,\ldots,c_n,d_1\ldots d_n)$ and w.l.o.g. $f_1 \neq 0$. Then

$$\delta_1 \pm \delta_2 = (f) \times (\delta_2)$$

where $f = -f_2 f_1^{-1} \pm 1$. f, being an element of $F(\underline{c},\underline{d})$ does not have to be computed.

Section 8

Lemma: Any algorithm over $(F(\underline{a},\underline{b},\underline{c},\underline{d}), F(\underline{c},\underline{d}) \cup \{\underline{a},\underline{b}\})$ which has no EAS's computes only elements of the form

$$f\left(\prod_1^n a_i^{\alpha(i)}\right)\left(\prod_1^n b_i^{\beta(i)}\right), \quad f \in F(\underline{c},\underline{d}) \tag{3}$$

Proof: By induction on length of algorithms. □

Section 9

Theorem: Let C be an algorithm over $(F(\underline{a}), H \cup \underline{a})$ which computes $\psi(\underline{a}) \subset F(\underline{a})$ using $m < n$ EAS's. Then there exists an algorithm β over $(F(\underline{c}, b_1, \ldots, b_{n-m}), H \cup \{\underline{c}, b_1, \ldots, b_{n-m}\})$ without any EAS's which computes $\psi(\underline{A})$ where

$$A_i = c_i \prod_{j=1}^{n-m} b_j^{\gamma(i,j)}$$

and $(\gamma(i,j))_{i=1,\ldots,n;\ j=1,\ldots,n-m}$ is a matrix of integers of rank $n-m$.

Proof: Let

$$\delta \leftarrow f_1 \Pi a_j^{\beta(1,j)} \pm f_2 \Pi a_j^{\beta(2,j)}$$

be the first EAS in C. We can replace this instruction by a sequence computing $\Pi a_j^{\beta(1,j)-\beta(2,j)}$, $f_2 f_1^{-1}$, $f_1 \Pi a_j^{\beta(2,j)}$ using multiplications and divisions only, then an EAS of the form

$$\Pi a_j^{\beta(1,j)-\beta(2,j)} \pm f_2 f_1^{-1} \tag{4}$$

and then multiplication of (4) by $f_1 \Pi a_j^{\beta(2,j)}$. Thus we may assume that the first EAS is of the form

$$\Pi a_j^{\alpha(1,j)} \pm f^1, \quad f^1 \in F \tag{5}$$

If we assume that $a_1, \ldots, a_n$ satisfy

$$\Pi a_j^{\alpha(1,j)} = d_1$$

then (5) reduces to $d_1 \pm f^1$ which is not an EAS.

Thus we obtain a new algorithm in which the first EAS can be assumed to be

$$\Pi a_j^{\alpha(2,j)} \pm f^2 \quad , \quad f^2 \in F(d_1) \tag{6}$$

($f^2 \in F(d_1)$ as it may depend on the result of (5)). If $a_1, \ldots, a_n$ satisfy

$$\Pi a_j^{\alpha(2,j)} = d_2$$

then (6) reduces to $d_2 \pm f^2$ which is not EAS.

We repeat this procedure until there are no EAS left. Thus we obtain $m^* \leq m$ equations.

$$\Pi a_j^{\alpha(k,j)} = d_k \qquad k = 1, \ldots, m^* \tag{7}$$

such that if $a_1, \ldots, a_n$ satisfy them, then the algorithm has no EAS's.

The system (7) is a consistent system of $m^* < n$ equations in n variables.

The rank of the solution space of the homogenous system

$$\Pi a_j^{\alpha(k,j)} = 1$$

is at least $n-m^* \geq n-m$.

Therefore, for suitable $\varphi_1, \ldots, \varphi_n \in F(d_1, \ldots, d_n)$ and $(\gamma(i,j))_{i=1,\ldots,n; \ j=1,\ldots,n-m}$ over $\mathbb{Z}$ of rank $n-m$ (7) is solved by

$$A_i = \varphi_i \prod_{j=1}^{n-m} b_j^{\gamma(i,j)}$$

Thus we obtain an algorithm over $F(d_1, \ldots, d_{m^*}, b_1, \ldots, b_{n-m})$, $F \cup \{d_1, \ldots, d_{m^*}, b_1, \ldots, b_{n-m}\}$) computing $\psi(\varphi_1 \Pi b_j^{\gamma(1,j)}, \ldots, \varphi_n \Pi b_j^{\gamma(n,j)})$ without EAS's. Similarly, we can construct an algorithm $\mathcal{B}$ over $(F(\underline{c}, b_1, \ldots, b_{n-m}), H \cup \{\underline{c}, b_1, \ldots, b_{n-m}\})$ which satisfies the requirement of the theorem. It is also necessary to note that assuming that in $\mathcal{G}$ there were no attempts to divide by zero, then in $\mathcal{B}$ too there are no attempts to divide by zero. $\qquad \square$

Section 10

Example: Before stating a general theorem, we shall prove that if G computes $a_1 + a_2 + \ldots + a_n$ over $(F(\underline{a}), H \cup \underline{a})$, then G has at least n-1 EAS's. Indeed, assume that G uses at most n-2 EAS's. Then the corresponding B computes

$$q = \Sigma \; c_1 \overset{2}{\underset{1}{\Pi}} b_j^{\gamma(i,j)}$$

and the rank of $(\gamma(k,j))$ is 2. Let k be the number of distinct rows of $\gamma(i,j)$ and thus $k \geq 2$.

W.l.o.g. let $i(1) < i(2) < \ldots < i(k) < i(k + 1)$ be such that $i(1) = 1$, $i(k + 1) = n + 1$, and for every $\ell \leq k$ the rows $i(\ell)$ through $i(\ell + 1) - 1$ are equal.

Therefore,

$$q = \overset{k}{\underset{\ell=1}{\Sigma}} \left(\overset{i(\ell+1)-1}{\underset{i=i(\ell)}{\Sigma}} c_i \right) \overset{2}{\underset{j=1}{\Pi}} b_j^{\gamma(i(\ell),j)}$$

which is not a monomial of the form (3).

Section 11

From here on we shall use the notation $\nu_E(S)$ to denote the minimum number of EAS's required to compute $S \subset F(\underline{a})$ over $(F(\underline{a}), H \cup \underline{a})$ when we assume that it is clear what F and H are.

Lemma: Let $P_1, \ldots, P_m \in F[\underline{a}]$ ($F[\underline{a}]$ denotes the set of polynomials in $\underline{a}$ over F). Let $r_1, \ldots, r_m$ be monomials over F, namely elements of the form

$$f \overset{n}{\underset{1}{\Pi}} a_i^{\alpha(i)} \quad , \quad f \in F$$

Then $\nu_E(P_1, \ldots, P_m) = \nu_E(P_1 r_1, \ldots, P_m r_m)$.

Proof: Trivial. ⊓

Section 12

Let $P = \sum_{j=1}^{k} f_j \prod_{\ell=1}^{n} a_\ell^{\alpha(j,\ell)}$ be in $F[\underline{a}]$. (When we write such an

equation, we shall always assume that

$$j_1 \neq j_2 \Rightarrow \prod a_j^{\alpha(j_1,\ell)} \neq \prod a_j^{\alpha(j_2,\ell)}.)$$

Each such P can be written as

$$P = f_s \prod a_j^{\alpha(s,\ell)} \left(\sum_{1}^{k} f_j f_s^{-1} \prod_{\ell=1}^{n} a_\ell^{\alpha(j,\ell)-\alpha(s,\ell)} \right) = r(1 + q) \text{ where}$$

$$r = f_s \prod a_j^{\alpha(s,\ell)}, \quad q = \sum_{1}^{k} g_j \prod a_\ell^{\beta(j,\ell)} \text{ for suitable } \beta(j,\ell) \text{ and } g_j.$$

We have assumed that $f_s \neq 0$ and of course $g = 0$. Such a $1 + q$

will be called a __normal form__ of P. We shall say that $1 + q_1, \ldots, 1 + q_m$

are a normal form for $P_1, \ldots, P_m$ if each $1 + q_i$ is a normal form for P_i.

Let $q_1, \ldots, q_m$ be like above. $M_1, \ldots, M_t$ of the form

$$M_j = \prod_{1}^{n} a_\ell^{\beta(j,\ell)}$$

will be called the __basis__ for $q_1, \ldots, q_m$ if and only if for every q_i there

exist $g_1^i, \ldots, g_t^i \in F$ such that

$$q_i = \sum_{j=1}^{t} g_j^i M_j$$

and $\{M, \ldots, M_t\}$ is a minimal set satisfying these requirements. Such a set

is unique.

The matrix $((P(i, j)))_{j=1,\ldots,\,i=1}$ is called the

defining matrix for $q_1,\ldots,q_m$.

Section 13

Example: Let $F = R$, $n = 2$

$$P_1 = a_1^2 a_2 + 2a_1 - 3a_2$$

$$P_2 = 5a_1 a_2^2 + a_1^2 a_2$$

$$P_3 = a_2 + a_2^2 + a_2^3$$

Then e.g.

$$P_1 = 2a_1\left(\frac{a_1}{2a_2} + 1 - \frac{3a_2}{2a_1}\right)$$

$$P_2 = a_1^2 a_2\left(\frac{5a_2}{a_1} + 1\right)$$

$$P_3 = a_2\left(1 + a_2 + a_2^2\right)$$

$$1 + q_1 = 1 + \frac{a_1}{2a_2} - \frac{3a_2}{2a_1}$$

$$1 + q_2 = 1 + \frac{5a_2}{a_1}$$

$$1 + q_3 = 1 + a_2 + a_2^2$$

$$M_1 = a_1 a_2^{-1}$$

$$M_2 = a_1^{-1} a_2$$

$$M_3 = a_2$$

$$M_4 = a_2^2$$

and we may also write

$$
\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & -\frac{3}{2} & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_1 a_2^{-1} \\ a_1^{-1} a_2 \\ a_2 \\ a_2^2 \end{bmatrix}
$$

We also have

$$
\beta(j,\ell) = \begin{bmatrix} 1 & -1 \\ -1 & 1 \\ 0 & 1 \\ 0 & 2 \end{bmatrix}
$$

Section 14

Theorem: Let $P_1,\ldots,P_m \in F[\underline{a}]$. Let $1 + q,\ldots,1 + q_m$ be a normal form

for $P_1,\ldots,P_m$ and let $(\beta(j,\ell))_{j=1,\ldots,t;\ \ell=1,\ldots,n}$ be the defining matrix

for the basis of $q_1,\ldots,q_m$. Then

$$
\nu_E(P_1,\ldots,P_m) \geq \mathrm{rank}(\beta(j,\ell)).
$$

Proof: By the lemma in Section 11, it is enough to show that if G computes

$1 + q_1,\ldots,1 + q_m$, it has at least $\rho = \mathrm{rank}(\beta(j,\ell))$ EAS's. Assume that G

has at most $\rho - 1$ EAS's. Then by the theorem in Section 9, there exists

an algorithm β computing

$$\{1 + \sum_{j=1}^{t} g_j^i \hat{M}_j \mid 1 \le i \le m\} \quad \text{without EAS's where}$$

$$M_j = \prod_{\ell=1}^{n} (c_\ell \prod_{e=1}^{n-\rho+1} b_e^{\gamma(\ell,e)})^{\beta(j,\ell)}$$

$(c_\ell \prod\limits_{e=1}^{n-\rho+1} b_e^{\gamma(\ell,e)}$ was substituted for a_ℓ).

But also $\hat{M}_j = (\prod_\ell c_\ell^{\beta(j,\ell)}) \prod\limits_{e=1}^{n-\rho+1} b_e^{\sum\limits_{\ell=1}^{n} \beta(j,\ell)\gamma(\ell,e)}$

If we define

$$\mathcal{E}(j,e) = \sum_{\ell=1}^{n} \beta(j,\ell)\gamma(\ell,e),$$

$$j = 1,\ldots,t; \quad e = 1,\ldots, n - \rho + 1$$

we see that $\mathcal{E}(j,e)$ is the product of matrices $\beta(j,\ell)$ and $\gamma(\ell,e)$ of ranks ρ and $n - \rho + 1$ respectively. Therefore,

$$\text{rank}(\mathcal{E}(j,e)) \ge 1$$

and let j_0 be a row which has a nonzero element (and thus $M_{j_0} \in F(\underline{c}, b_1, \ldots, b_{n-\ +1}) - F(\underline{c})$). Let i_0 be such that $g_{j_0}^{i_0} \ne 0$. Then, using arguments similar to those of the example in Section 10 it follows that $1 + \hat{q}_i \doteq 1 + \sum_j g_j^i \hat{M}_j$ is not a monomial and the theorem is proved by contradiction. $\square$

Before stating several corollaries we can remark that a similar

theorem holds for rational functions.

Section 15

Corollary: At least 2 EAS's are required to compute both $a_1 a_3 - a_2 a_4$ and

$a_1 a_4 + a_2 a_3$ (the real and the imaginary parts of "$(a_1 + a_2 i) \times (a_3 + a_4 i)$." $\Box$

Section 16

In the following corollaries the variables ξ and η, with indices,

will range over $a_1, \ldots, a_n$. We shall assume that variables having

different indices are distinct. Furthermore $\xi \cap \eta = \emptyset$.

Section 17

Corollary: Let $\xi \triangleq (\xi(i,j))_{i=1,\ldots,m; \ j=1,\ldots,n}$ and

$\eta \triangleq (\eta(j,k))_{j=1,\ldots,n; \ k=1,\ldots,p}$ be matrices. Then the product of ξ and η

cannot be computed with less than $(m+p-1)(n-1)$ EAS's. $\Box$

Section 18

Corollary: Let $F = G(x)$ and $H = G\{x\}$ for some field G and an indeterminate

x over F. Let $P_i = \sum_0^{n(i)} \xi(i,j) x^j$ for $i = 1, \ldots, m$. Then $P_1, \ldots, P_m$ cannot

be computed with less than $\sum_{i=1}^{t} n_i$ EAS's. $\Box$

Section 19

Corollary: $\displaystyle\sum_{j=0}^{n(i)}\sum_{k=0}^{n(i)}\xi(i,j,k)x^{j}y^{k}$, $i = 1,\ldots t,$, cannot be computed

with less than $\displaystyle\sum_{1}^{t}(n(i) + 1)^{2}$ EAS's where $F = G(x,y)$, $H = G \cup \{x,y\}$. $\square$

Section 20

Our method is capable of establishing lower bounds which at most equal the number of indeterminates, and thus may also be referred to as a dimensionality method (dimension = number of indeterminates). If now $P \in G(x)$ is to be computed over $(G(x), G \cup \{x\})$ then we should be able to find lower bound on the number of AS's required to compute P. Some results in this direction have been recently obtained by Borodin and Cook [B&C 74]. Their methods belong to the class of rate-of-growth methods.

Using arguments similar to those of [K74] it is sometimes possible to combine these two methods. We shall describe briefly a weaker version of a more general theorem.

Section 21

We shall assume that $F = G(x)$, $H = G \cup \{x\}$ and we introduce the notion of a weakly-essential AS (WEAS).

Definition: An AS $\delta \leftarrow \delta_{1} \pm \delta_{2}$ in an algorithm over $(G(x,\underline{a},\underline{b},\underline{c},\underline{d})$, $G \cup \{x,\underline{a},\underline{b},\underline{c},\underline{d}\})$ will be called weakly essential if and only if

1. $\delta_{1},\delta_{2} \in G(x,\underline{c},\underline{d})$ and,

2. δ_{1} and δ_{2} are linearly independent as elements of $G(x,\underline{c},\underline{d})$ when considered as a linear space over $G(\underline{c},\underline{d})$.

(Such AS $\sigma \leftarrow \delta_1 \overset{+}{-} \delta_2$ is obviously not an EAS.)

For a set $\psi(x,\underline{a}) \subset G(x,\underline{a})$ we shall denote by $\nu(\psi(x,\underline{a}))$ the minimum number of both EAS's and WEAS's required to compute $\psi(x,\underline{a})$ over $(G(x,\underline{a}), G \cup \{x,\underline{a}\})$.

Section 22

Theorem: Let G compute $\psi(x,\underline{a})$ over $(G(x,\underline{a}), G \cup \{x,\underline{a}\})$ using $\nu(G)$ EAS's and WEAS's. Then there exists B over $(G(x,\underline{c}), G \cup \{x,\underline{c}\})$ which computes $A_1(x,\underline{c}),\ldots,A_n(x,\underline{c})$, $\psi(x,\underline{A}) \in G(x,\underline{c})$ using at most $\nu(G) - \nu_E(\psi(x,\underline{a}))$ WEASs.   ⊡

Section 23

This theorem may be used in the following way: if for a certain $\psi(x,\underline{a})$, $A_1(x,\underline{c}),\ldots,A_n(x,\underline{c})$, $\psi(x,\underline{A})$ can be shown to require always at least (say) m WEAS's to be computed, then G has at least $\nu_E(\psi(x,\underline{a})) + m$ EAS's and WEASs.

Section 24

Corollary: Let G compute $\sum_1^n a_i x^{i-1}$ and $P(x) \in G(x)$ over $(G(x,\underline{a}), G \cup \{x,\underline{a}\})$ and let m be the minimum number of WEASs required to compute $P(x)$ over $(G(x), G \cup \{x\})$. Then as $\nu_E(\sum_1^n a_i x^{i-1}) = n - 1$ it follows that G has at least $m + n - 1$ ASs.

(Such AE $\sigma = \delta_1 \pm \delta_2$ is obviously not an EAS.)

For a set $\#(x,\underline{s}) \subset G(x,\underline{s})$ we shall denote by $v(\#(x,\underline{s}))$ the minimum number of both EAS's and WEAS's required to compute $\#(x,\underline{s})$ over $(G(x,\underline{s}), \subset \cup \{x,\underline{s}\})$.

Section 22

Theorem: Let G compute $\#(x,\underline{s})$ over $(G(x,\underline{s}), \subset \cup \{x,\underline{s}\})$ using $v(1)$ EAS's and WEAS's. Then there exists G over $(G(x,\underline{s}), \subset \cup \{x,\underline{s}\})$ which computes $A_1(x,\underline{s}),\ldots,A_n(x,\underline{s}), \#(x,\underline{s}) \in G(x,\underline{s})$ using at most $v(1) - v_{\underline{s}}(\#(x,\underline{s}))$ WEAS's. $\square$

Section 23

This theorem may be used in the following way: if for a certain $\#(x,\underline{s})$, $A_1(x,\underline{s}),\ldots,A_n(x,\underline{s}), \#(x,\underline{s})$ can be shown to require always at least (any) m WEAS's to be computed, then G has at least

$$v_{\underline{s}}(\#(x,\underline{s})) + m \quad \text{EAS's and WEAS's.}$$

Section 24

Corollary: Let G compute $\sum_{1}^{m} c_i x^{i-1}$ and $P(x) \in G(x)$ over $(G(x,\underline{s}),(x), \subset \cup \{x,\underline{s}\})$ and let m be the minimum number of WEAS's required to compute $P(x)$ over $(G(x), \subset \cup \{x\})$. Then as $v_{\underline{s}}(\sum_{1}^{m} c_i x^{i-1}) = n - 1$ it follows that G has at least $m + n - 1$ AS's.

Acknowledgement

The author wishes to thank S. Winograd, A. Borodin, and D. Kirkpatrick for the valuable conversations he had with them.

<u>References</u>

[B61]   Belaga, E.C., On computing polynomials in one variable with initial preconditioning of coefficients, Problemi Kibernetiki <u>5</u> (1961), 7-15.

[B & C74]   Borodin, A. and S. Cook, On the number of additions to compute specific polynomials, <u>Proceedings of the Sixth Annual Symposium in the Theory of Computing</u>, (1974), 342-347.

[K72]   Kirkpatrick, D., On the additions necessary to compute certain functions, <u>Proceedings of the Fourth Annual Symposium on the Theory of Computing</u>, 1972, 94-101.

[K74]   Kedem, Z.M., Combining dimensionality and rate of growth arguments for establishing lower bounds on the number of multiplications preliminary report: <u>Proceedings of the Sixth Annual Symposium on the Theory of Computing</u>, 1974, 334-341; final report: to be issued by Project MAC (M.I.T.), June 1974, 1-38.

[P66]   Pan, V. Ya., Methods of computing values of polynomials, Russian Math. Surveys <u>21</u> (1966), 105-136.

[S72]   Strassen, V., Evaluation of rational functions, "Complexity of Computer computations" ed. by R.E. Miller and J.W. Thatcher, Plenum Press, New York (1972), 1-10.

[W67]   Winograd, S. On the number of multiplications required to compute certain functions, Proc. Nat. Acad. Sci. U.S.A. <u>58</u> (1967), 1840-1842.

[W70a]   Winograd, S., On the number of multiplications necessary to compute certain functions, Comm. Pure Appl. Math. <u>23</u> (1970), 165-179

[W70b]   Winograd, S., On the algebraic complexity of functions, Actes Congres intern. Math. <u>3</u> (1970), 283-288.

BIBLIOGRAPHIC DATA SHEET	1. Report No. NSF-OCA-GJ34671 - TM - 48	2.	3. Recipient's Accession No.

4. Title and Subtitle

The reduction Method for Establishing Lower Bounds on the Number of Additions

5. Report Date : Issued July 1974

6.

7. Author(s)
Zvi M. Kedem

8. Performing Organization Rept. No. MAC TM-48

9. Performing Organization Name and Address

PROJECT MAC; MASSACHUSETTS INSTITUTE OF TECHNOLOGY:

545 Technology Square, Cambridge, Massachusetts 02139

10. Project/Task/Work Unit No.

11. Contract/Grant No.

GJ34671

12. Sponsoring Organization Name and Address
Associate Program Director
Office of Computing Activities
National Science Foundation
Washington, D. C. 20550

13. Type of Report & Period Covered: Interim Scientific Report

14.

15. Supplementary Notes

16. Abstracts

A method for establishing lower bounds on the number of multiplications and divisions has been developed by Pan, Winograd and Strassen. A similar method is developed for establishing lower bounds on the number of additions and subtractions. The results obtained partially overlap those of Belaga, Winograd and Kirkpatrick.

17. Key Words and Document Analysis. 17a. Descriptors
Additions
Algebraic operations
Algorithms
Analysis of algorithms
Computational complexity
Dimensionality
Lower bounds
Optimality
Polynomials
Rate of growth - Rational function - Reduction

17b. Identifiers/Open-Ended Terms

17c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Approved for Public Release; Distribution Unlimited	UNCLASSIFIED	22
	20. Security Class (This Page) UNCLASSIFIED	22. Price

FORM NTIS-35 (REV. 3-72)   THIS FORM MAY BE REPRODUCED   USCOMM-DC 14952-P72