Complexity Measures and Hierarchies for the Evaluation of Integers, Polynomials, and N-Linear Forms

Richard J. Lipton and David Dobkin

Research Report #38

This paper will be presented at the Seventh Annual ACM Symposium on Theory Computing, to be held in Albuquerque, New Mexico, May, 1975

February 1975

COMPLEXITY MEASURES AND HIERARCHIES FOR THE EVALUATION OF INTEGERS, POLYNOMIALS, AND N-LINEAR FORMS

Richard J. Lipton* and David Dobkin†
Department of Computer Science
Yale University
New Haven, Connecticut 06520

1. Introduction

The difficulty of evaluating integers and polynomials has been studied in various frameworks ranging from the addition-chain approach [5] to integer evaluation to recent efforts aimed at generating polynomials that are hard to evaluate [2,8,10]. Here we consider the classes of integers and polynomials that can be evaluated within given complexity bounds and prove the existence of proper hierarchies of complexity classes. The framework in which our problems are cast is general enough to allow any finite set of binary operations rather than just addition, subtraction, multiplication, and division. The motivation for studying complexity classes rather than specific integers or polynomials is analogous to why complexity classes are studied in automata-based complexity: (i) the immense difficulty associated with computing the complexity of a specific integer or polynomial; (ii) the important insight obtained from discovering the structure of the complexity classes. Thus, we are able to prove that under mild restrictions if

$f(n) > g(n)$ a.e.

where f and g are monotone functions, then there are an infinite number of integers (respectively polynomials) that can be evaluated in f(n) steps but not g(n) steps.

The model used here for polynomial evaluation differs from the model used in Strassen [10] and Paterson and Stockmeyer [8]. The difference lies in their allowing scalar multiplications by constant a at a cost of zero or one, while we charge an amount that is a function of a. This amount essentially reflects the complexity of integer a. The motivation for our model is similar to that of Cook [3]: In a "real" machine model it is realistic to say that the cost of $a \circ p$, where a is an integer and p is some term, is dependent at least on the cost associated with "naming" a. That is, the cost must be at least the amount of information needed to state that the scalar is a and to apply the operation $a\circ$ to p.

The difference between our model and that of Strassen and Paterson and Stockmeyer is most dramatic when one compares the following results:

1) [Strassen] There are polynomials that take $\sim n/\log n$ steps to evaluate.† They have coefficient $\sim 2^{2^{n^3}}$ in size.
2) [Paterson and Stockmeyer] There are 0,1 coefficient polynomials that take $\sim\sqrt{n}$ steps to evaluate.
3) [Theorem 7, section 4] There are 0,1 coefficient polynomials that take $\sim n/\log n$ steps to evaluate.

Thus our result would be an improvement of both (1) and (2) if we had assumed that scalars are of cost 1. An open question is: How much does our assumption affect the complexity of polynomial evaluation?

Since our results are proved for any finite set of binary operations it is not surprising that they follow by counting type arguments. The basic counting tools we use are a number of powerful results from number theory on the density of sequences of integers [7]. These tools allow us to establish our hierarchy results. Moreover, they allow us to refine them so that we can show not only that there are, for example, 0,1 polynomials that take $\sim n/\log n$ steps to evaluation but that "almost all" polynomials take this number of steps.

2. Upper and Lower Bounds

We define an addition chain as a sequence of integers $a_0, a_1, \ldots, a_m$ such that $a_0 = 1$ and, for each i, $a_i = a_j + a_k$ for some $j,k < i$. If $a_m = n$, the chain is said to realize the integer n. We define $C_{\{+\}}(n)$ to be the length of the shortest addition chain realizing n. Brauer [1] obtained the upper bound of

$$\log n + \frac{\log n}{\log \log n} + o\left(\frac{\log n}{\log \log n}\right)$$

on $C_{\{+\}}(n)$ and Erdos [4] showed that for most large n the lower bound of

$$\log n + \frac{\log n}{\log \log n}$$

† The work of this author was supported in part by US Army grant DAHC04-75-6-0037.
* The work of this author was supported in part by National Science Foundation grant GJ-43157.

† n = degree of the polynomial.

is valid. In the current paper we extend addition chains to B-chains as follows:

Definition: Let B be a finite set of binary operations over N. A B-chain is a sequence $\alpha_0,\ldots,\alpha_m$ such that $\alpha_0 = 1$ and for each i $\alpha_i = \alpha_j \circ \alpha_k$ where $j,k < i$ and $\circ$ is an operation of B. If $\alpha_m = n$, the B-chain is said to realize n. The length of the shortest B-chain for n is denoted by $C_B(n)$. By convention, $C_B(0)$ and $C_B(1)$ are defined as zero.

We shall denote the operations addition, subtraction, multiplication, division, and exponentiation by +, -, ×, ÷, ↑ in the current paper, where $a \div b = \lfloor \frac{a}{b} \rfloor$. For this notation, the following lower bounds are obtained.

Theorem 1: For all n,

a) $C_{\{+\}}(n) \geq C_{\{+,-\}}(n) \geq \log n$ *

b) $C_{\{+,-,\times,\div\}}(n) > \log \log n$

c) $C_{\{+,-,\times,\div,\uparrow\}}(n) > \log (G(n))$

where G(n) is the number of times the logarithm of n must be taken to yield a value less than or equal to 1.

Proof: In each case, it suffices to consider the largest number achievable in n steps. □

Extensions of Theorem 1 to other basis sets is possible and fairly standard. For example, if for all operations $\circ \in B$ there exists k such that, for all x and y, $x \circ y$ is of order $x^k y^k$, then $C_B(n)$ grows asymptotically at least as fast as log log n. We can also obtain the upper bounds.

Theorem 2: For all n,

$$C_{\{+,-,\times,\div,\uparrow\}}(n) \leq C_{\{+,-,\times,\div\}}(n) \leq C_{\{+,\times\}}(n) \leq \frac{2 \log n}{\log \log n} + o\left(\frac{\log n}{\log \log n}\right).$$

Proof: Let $n = \sum_{i=0}^{m} \lambda_i \alpha^i$ be the expansion of n in base α for some α; then n can be found by computing $2,\ldots,\alpha-1,\alpha$ and using Horner's rule to evaluate for n. Thus,

$C_{\{+,\times\}}(n) \leq \alpha - 1 + 2m = \alpha - 1 + 2[\log_\alpha n]$.

The choice of $\alpha = \frac{\log n}{(\log \log n)^2}$ yields the desired result. □

Next, we study cumulative lower bounds. Rather than consider the complexity of reaching n by a B-chain, we define as $H_B(n)$ the maximum value of $C_B(k)$ for any $k \leq n$. This measure is actually more natural than $C_B(n)$ since $C_B(n)$ may fluctuate greatly. Then we can achieve the surprising result that $H_B(n)$ is asymptotically independent of B if $+,\times \in B$.

Theorem 3: For any choice of B,

$H_B(n) \approx O(\frac{\log n}{\log \log n})$. †

Proof: A simple counting argument shows that the number of B-chains of length $\leq m$ is $\leq |B|^m((m-1)!)^2$. By the definition of $H_B(n)$, we see that a growth rate asymptotic to $\frac{\log n}{\log \log n}$ is necessary. □

Let $h(n) = \frac{\log n}{\log \log n}$ and observe that, for all B, $H_B(n) \approx h(n)$.

3. Complexity Classes on N

The results of the previous section pave the way for some interesting questions. We observe that there are constants K_1 and K_2 such that for each n $C_B(n) \leq K_1 h(n)$ and, for some $p \leq n$, $C_B(p) \geq K_2 h(n)$. This leads to interesting questions on the complexity classes into which the integers can be partitioned by B-chains for varied bases B. Before studying such questions, we make contact with some results from elementary number theory.

Definition: For A a subset of N such that $0,1 \in A$, the Schnirelmann density d(A) is

$$d(A) = \inf_{n \geq 1} \frac{A(n)}{n}$$

where A(n) is the number of elements of A less than or equal to n.

We observe that $d(A) = 1$ if and only if $A = \mathbb{N}$ and present the following result on d(A).

Theorem 4: (α-β Theorem [7])

If $C = A + B \overset{\Delta}{=} \{a + b \mid a \in A, b \in B\}$, then $d(c) \geq \min[1, d(A) + d(B)]$.

As a corollary to this theorem, we have the result that if A is any set of positive density then, if

---

* Throughout this paper, all logarithms are base 2.

† $f(n) \approx g(n)$ if there exist constants m,M such that $0 < m < M < \infty$ with $mf(n) < g(n) < Mf(n)$ for all n.

C is the sum of $\lceil \frac{1}{d(A)} \rceil$ copies of A, $C = \mathbb{N}$. Now we define complexity classes and prove our hierarchy results.

Definition: If $f: \mathbb{N} \to \mathbb{N}$ is a monotone function, then C_f^B, the complexity class of f with respect to B, is defined as $\{n \in \mathbb{N} \mid C_B(n) \leq f(n)\}$.

We will not define C_f^B if f is not monotone.

Recall that h(n) is defined as $\frac{\log n}{\log \log n}$. The following lemma yields the hierarchy theorem.

Lemma: If $+ \in B$, f is monotone, and $\lim_{n\to\infty} \frac{f(n)}{h(n)} = 0$, then $d(C_f^B) = 0$.

Proof: Suppose that $d(C_f^B) > 0$ and $\lim_{n\to\infty} \frac{f(n)}{h(n)} = 0$. Then, since every integer can be expressed as the sum of at most $\lceil 1/d(C_f^B) \rceil$ integers, $H_B(n) \leq 2\lceil 1/d(C_f^B) \rceil f(n)$ for all n, contradicting theorem 3. □

Thus,

Theorem 5: (Hierarchy Theorem)
Suppose that $+ \in B$ and f and g are monotone integer-valued functions such that

1) $f(n) > g(n)$ a.e.
2) $\lim_{n\to\infty} \frac{h(n)}{g(n)} = \infty$ (i.e. $C_g^B \subsetneq \mathbb{N}$) and
3) g grows sufficiently fast such that C_g^B is infinite. Then $C_f^B - C_g^B$ is infinite.

Proof: It is clear that C_g^B has zero density since g grows asymptotically more slowly than h. Also, there is an integer N_0 such that for all $x \geq N_0$ $1 + g(x) \leq f(x)$. Furthermore, there is a choice of $N_1 > N_0$ such that $N_1 \in C_g^B$ and $N_1 + 1 \notin C_g^B$. Now,

$$C_B(N_1 + 1) \leq 1 + C_B(N_1) \leq 1 + g(N_1) \leq f(N_1) \leq f(N_1 + 1)$$

and hence $N_1 + 1 \in C_f^B$. We may extend this method to form a sequence $\{N_i\}$ such that $N_i > N_{i-1}$ and each $N_i + 1 \in C_f^B - C_g^B$. □

Typical of the applications of this hierarchy result are

Corollary 1: If $B_1 = \{+,-,\times,\div\}$, then

$$C^{B_1}_{\log\log n} \subset^{+} C^{B_1}_{(\log\log n)G(n)} \subset^{+} \dots \subset^{+} C^{B_1}_{(\log\log n)G^k(n)} \subset^{+} \dots \subset^{+} C^{B_1}_{(\log\log n)^2} \subset^{+} \dots \subset^{+} C^{B_1}_{h(n)/G(n)} \subset^{+} C^{B_1}_{h(n)} = \mathbb{N}.$$

Corollary 2: If $B_2 = \{+,-,\times,\div,\uparrow\}$, then

$$C^{B_2}_{\log(G(n))} \subset^{+} \dots \subset^{+} C^{B_2}_{\log^2(G(n))} \subset^{+} \dots \subset^{+} C^{B_2}_{\log\dots\log n} \subset^{+} \dots \subset^{+} C^{B_2}_{\log\log n} \subset^{+} \dots \subset^{+} C^{B_2}_{h(n)/G(n)} \subset^{+} C^{B_2}_{h(n)} = \mathbb{N}.$$

These results make contact with some interesting results in number theory.

Fact 1: (Landau [6])
Every integer can be expressed as the sum of 67 or fewer primes.

Fact 2: (Waring's problem [7])
For each integer k, there is a number g(k) such that every integer can be expressed as the sum of g(k) or fewer k^{th} powers.

Using these results, we obtain

Corollary 3: For any B, there is an infinite subsequence $\{P_i\}$ of the sequence of primes such that $C_B(P_i)$ grows as $O(\log P_i/\log\log P_i)$.

Corollary 4: For any B and each integer k, there is an infinite sequence $\{x_i^k\}$ such that $C_B(x_i^k)$ grows as $O(\log x_i^k/\log\log x_i^k) = O(k\log x_i/\log(k\log x_i))$.

4. Polynomial Evaluation

The difficulty of polynomial evaluation has been studied in a variety of settings. Lately, a number of authors have focused on finding polynomials that are difficult to evaluate regardless of how much preconditioning of coefficients is allowed [2,8,10]. The results of these studies are hard to evaluate polynomials that have extremely large coefficients. For example, Strassen [10] shows that the evaluation of

$$P_1(x) = \sum_{\delta=0}^{d} 2^{2^{\delta d^2}} x^{\delta}$$

requires either d/2 - 2 non-scalar multiplication/divisions or at least $d^2/\log_2 d$ total arithmetics and that the evaluation of

$$P_2(x) = \sum_{\delta=0}^{d} 2^{2^\delta} x^\delta$$

requires at least $\sqrt{d/(3\log d)}$ arithmetics. In the terminology of this paper, however, the computation, from a basis of $+,-,\times,\div$, of single coefficients of $P_1(x)$ (respectively $P_2(x)$) requires d^3 (respectively d) operations, and thus the evaluation cost is unimportant relative to this cost. In this manner, we shall diverge from the methodology of previous studies of polynomial evaluation. We shall try to find the chain requiring the least number of operations from a basis B that, starting from inputs 1 and x_0, generates the value of a polynomial p(x) at the point x_0. We have chosen this model because we feel that it addresses some of the issues not considered in previous studies of this problem. A positive result of lower bounds using this model is that such bounds give lower bounds on the sizes of scalars that must be used in previous models. That is, we define

Definition: If $p(x) \in \mathbb{N}(x)$, then $\delta_B(p)$ is the length of the shortest sequence $\alpha_{-1},\ldots,\alpha_k$ (i.e. the least k) such that $\alpha_{-1} = 1$, $\alpha_0 = x$, and, for $1 \le i \le k$, $\alpha_i = \alpha_j \circ \alpha_k$ where $\circ \in B$ and $j,k \le i$.

The following theorem is then immediate.

Theorem 6: $\delta_B(p) \ge C_B(p(n)) - C_B(n)$ for any integer n.

Proof: It is obvious that $\delta_B(p) + C_B(n)$ is an upper bound for $C_B(p(n))$. □

Within our measure, we have the cost for Strassen's polynomials as

Corollary: For $B = \{+,-,\times,\div\}$,

1) $\delta_B(\sum_{i=0}^{d} 2^{2^{id^2}} x^i) \ge d^4/2$

2) $\delta_B(\sum_{i=0}^{d} 2^{2^i} x^i) \ge d^2/2$

We now wish to ask how hard the hardest polynomials are to evaluate in our complexity measure. By defining complexity classes of polynomials, we obtain an extension of a result due to Savage [8] on the complexity of polynomial evaluation.

Theorem 7: Let $D_F^B = \{p \in \mathbb{N}[x] \mid \delta_B(p) \le F(\deg(p))\}$ be a complexity class for polynomials and suppose that D_F^B contains all polynomials with 0,1 coefficients and + is in B; then $F(n) \ge n/\log n$.

Proof: Since D_F^B contains all 0,1 polynomials, we observe that the set $\{p(2) \mid \delta_B(p) \le F(\deg(p))\}$ contains all of $\mathbb{N}$. Now, we define $g(k) = F(\lfloor \log k \rfloor) + 1$ for integer k and claim that $C_g = N$. The proof of this follows since any integers k can be written as $p_k(2)$ where $p_k(x)$ is a polynomial with 0,1 coefficients of degree $\lfloor \log k \rfloor$ and the result of Theorem 6 implies that

$$C_B(k) = C_B(p_k(2)) \le C_B(2) + \delta_B(p_k) \le 1 + \delta_B(p_k)$$
$$\le 1 + F(\lfloor \log k \rfloor) = g(k)$$

for all integers k. Thus, $C_g = \mathbb{N}$. By Theorem 3, however, $g(n) \ge h(n)$ and hence $F(n) \ge n/\log n$. □

The result of this theorem is somewhat surprising, since we have shown the existence of polynomials with 0,1 coefficients whose evaluation by an algorithm using any finite set of basis operations requires at least $O(n/\log n)$ of these operations. For example, as a corollary to this theorem, we have

Corollary: Let $b_1 = +$ and $b_2,\ldots,b_r$ be any set of binary operations. Then there is a family $\{q_n(x)\}$ of polynomials with 0,1 coefficients such that the complexity of evaluating $q_n(x)$ by any algorithm using using the operations $b_1,\ldots,b_r$ grows asymptotically with $O(n/\log n)$.

While we have shown the existence of such a family, we leave its construction as an open problem. We can extend this result to a hierarchy result analogous to Theorem 5.

Theorem 8: (Hierarchy Theorem for Polynomials) Suppose that $\{+,\times\} \in B$ and F and K are functions such that

1) $F(n) > K(n) + 1$ a.e.

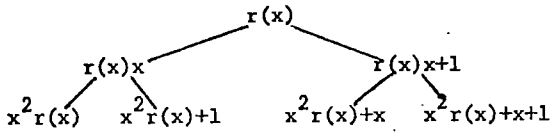
2) $\lim_{n\to\infty} \frac{K(n)}{\log n} \ge 1$ (i.e. D_K^B is infinite)

3) $\lim_{n\to\infty} \frac{K(n)}{n/\log n} = 0$ (i.e. $D_K^B \subset \mathbb{N}[x]$)

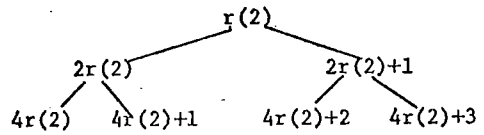
then $D_F^B - D_K^B$ is infinite.

Proof: Define $p(x)x$ and $p(x)x+1$ to be the successors

of $p(x)$. We claim that for any N_0 there is a $p_{N_0}(x) \in D_K^B$ such that one of the successors of $p_{N_0}(x)$ is not in D_G^B and such that $p_{N_0}(x)$ is of degree at least N_0. Assume not and consider the tree $T(r(x))$ for some $r(x) \in D_K^B$



Either our claim is true or every node of the infinite tree $T(r(x))$ belongs to D_K^B. If every node of the tree $T(r(x))$ is in D_K^B, then every node of the tree $T'(r(2))$ is in C_k^B where $k = K(\lfloor \log n \rfloor) + 1$ and $T'(r(2))$ is given by



But the nodes of $T'(r(2))$ are seen to have positive density and this contradicts the hypothesis of the theorem, since if this were true then for some integer n $C_{nk}^B = \mathbb{N}$, contradicting previous results. Thus, either $r(x)x$ or $r(x)x+1$ is not in D_K^B; if we let the successor of $r(x)$ not in D_K^B be $q(x)$, then

$$\delta_B(q(x)) \leq 2 + \delta_B(r(x)) \leq 2 + K(\deg(r(x)))$$
$$\leq F(\deg(r(x)))$$

for large enough N_0. Thus, $q(x) \in D_F^B$. By an argument similar to that used in the proof of Theorem 5, we can show that an infinite sequence of polynomials in $D_F^B - D_K^B$ exists. □

The results of Theorem 7 can be extended to multivariate polynomials and n-linear forms by applying reducibilities to make these problems equivalent to single variable polynomial evaluation.

References

[1] A. Brauer. Bulletin of the AMS 45:736-739, 1939.

[2] A. Borodin and S. Cook. On the number of additions to compute specific polynomials. Conference Record of the Sixth ACM Symposium on the Theory of Computing, Seattle, Washington, May 1974.

[3] S. Cook. Linear time simulation of deterministic two-way push down automata. Proceedings of IFIP Congress 71, TA-2, North-Holland, Amsterdam, 172-179.

[4] P. Erdös. Acta Arithmetica 6:77-81, 1960.

[5] D. Knuth. The Art of Computer Programming, Volume II: Seminumerical Algorithms. Addison-Wesley, Reading, Massachusetts, 1969.

[6] E. Landau. Über einige neuere Fortschritte der additiven Zahlentheorie. Cambridge, 2nd edition, 1937.

[7] I. Niven and H. Zuckerman. An Introduction to the Theory of Numbers. John Wiley and Sons, New York, 2nd edition, 1966.

[8] M. S. Paterson and L. J. Stockmeyer. On the number of nonscalar multiplications necessary to evaluate polynomials. SIAM J. Computing 2(1):60-66, March 1973.

[9] J. Savage. An Algorithm for the computation of linear forms. SIAM J. Computing 3:150-158, 1974.

[10] V. Strassen. Polynomials with rational coefficients which are hard to compute. SIAM J. Computing 3:128-149, 1974.