Algebraic methods for proving lower bounds in circuit complexity

Eric Allender[*]
Department of Computer Science
Rutgers University
New Brunswick, NJ 08903

Abstract

In the limited time available, it will not be possible to give a broad survey of the variety of algebraic techniques that have been used in proving circuit lower bounds. Instead, I will focus narrowly on a body of related results surrounding the complexity class ACC. In particular, I will cover the following:

- Barrington's characterization of NC^1 in terms of bounded-width branching programs [3]. This characterization highlights the importance for circuit complexity of the algebraic notion of solvability.
- The characterizations in [3, 5] of ACC in terms of circuits with MODgates and in terms of solvable algebras.
- The results of [9, 11] giving lower bounds on the size required for circuits with $\wedge$, $\vee$, and MOD_p gates to compute MOD_q for $q \neq p$.
- The results of [14, 6] giving efficient simulations of ACC circuits by circuits with symmetric gates.
- The results of [1] giving lower bounds on the size of "uniform" ACC circuits computing the permanent and recognizing languages in PP.

1 Introduction

In spite of some spectacular progress in different areas of complexity theory over the last several years, it must be admitted that there are still very few nontrivial examples of separations of complexity classes. It is not known if there is any set in $\mathrm{Ntime}(2^{O(n)})$ that cannot be recognized by linear-size circuits of logarithmic depth. The major open questions remain open.

But there *has* been steady progress reported in proving lower bounds for classes of *constant-depth* circuits. In fact, there has been so *much* progress reported, that my AMCoT94 lecture will not attempt to survey all of the work in this area, but instead will tell the story of the complexity class ACC, which in many ways represents the current frontier of research in circuit lower bounds. The main elements of this story are:

[*]Supported in part by NSF grant CCR-9204874.

- The origin. Why was ACC picked as an object for study?
- The milieu. What is known about classes just "slightly" smaller than ACC?
- The story so far. What have we been able to prove about ACC itself?

There is no reason to believe that the story is completed, and further installments are anxiously awaited.

2 The Origin

To understand why ACC is interesting, it is necessary to understand why the algebraic notion of solvability is important in the study of circuit complexity. One of the most important and natural circuit complexity classes is NC^1, the class of languages accepted by fan-in two circuits of $\wedge$, $\vee$, and negation gates, where the circuits have depth $O(\log n)$. NC^1 has many equivalent characterizations in terms of Boolean formulae, alternating Turing machines, and related notions. Most of the natural and important problems that are in Dspace($\log n$) are in NC^1, and since Dspace($\log n$) can be easily characterized in terms of branching programs of polynomial size, it came as quite a shock when Barrington proved in [3] that NC^1 consists of the languages accepted by branching programs of polynomial size and $O(1)$ width (and in fact width 5). The proof in [3] showed that some *regular* sets are complete for NC^1, in particular the word problem for S_5; the proof relied heavily on the fact that S_5 is not a solvable group.

This spawned a new algebraic approach to circuit complexity, building on an already well-established algebraic theory of finite automata, and in [3, 5] more of the details of this approach emerged. These papers characterized NC^1 in terms of programs over monoids. If only solvable monoids are used in this model, one obtains the complexity class ACC. An equivalent way to characterize ACC is in terms of circuits: ACC is the class of languages accepted by polynomial-size circuits of constant depth, with $\wedge$, $\vee$, and MOD_m gates (where the modulus m does not depend on the input length). Every regular set is either in ACC or else it is complete for NC^1; in this way, solvability of a monoid determines complexity.

3 The Milieu

There were other reasons to be interested in constant depth circuits with MODgates. Building on the exciting lower bounds of [14, 7] showing that the MOD2 function requires exponential size on constant depth circuits of $\wedge$, $\vee$, and negation gates, Razborov showed in [9] that, even if MOD_2 gates are allowed, the MOD_3 function remains difficult to compute. Smolensky improved this result in [11] to show that, even with MOD_p gates, the MOD_q function remains difficult to compute, if p is prime, and q is not a power of p.

If Smolensky's result could be further improved by removing the restriction that p is prime, then it would follow that the MAJORITY function is not in ACC, and hence that ACC is not equal to NC^1.

A key part of the proofs in [9, 11] is that unbounded fan-in $\wedge$ and $\vee$ gates can be "simulated" by low-degree polynomials mod p. This "simulation" follows from a simple construction in which $\wedge$ and $\vee$ gates are replaced by probabilistic circuits containing MOD_p gates and small-fan-in $\wedge$ gates. It is important for later applications of this technique that, using the techniques of [13], this construction can be accomplished using a small number of probabilistic bits. These same techniques were used to stunning effect by Toda in [12], and the other applications to constant-depth circuits with MOD_p gates are explored in [2, 8].

4 The Story So Far

Yao was the first one to successfully apply any of these techniques to ACC itself [15], and his results were subsequently improved by [6] to show that every set in ACC can be accepted by a depth-two circuit consisting of $\wedge$ gates of fan-in $\log^{O(1)} n$ on the bottom level, and a symmetric gate on the second (output) level. Unfortunately, that is essentially the end of the story thus far; it is still an open question if there is any set in $\mathrm{Ntime}(2^{n^{O(1)}})$ that is not in ACC. (Indeed, it is still open if depth three circuits with MOD_6 gates suffice.)

Surprisingly, however, we know quite a bit more if we impose the restriction that the circuits be "easy to build".

Circuit complexity classes come in two flavors: "uniform" and "nonuniform". All of the results mentioned thus far hold for the "nonuniform" model, meaning that there is no requirement that the circuit C_n (which decides membership for inputs of length n) be constructible from n. Thus many nonrecursive sets (e.g., all unary languages) have trivial nonuniform circuit complexity. The exact notion of "easy to build" that is used in formulating the notion of "uniform" circuits turns out to be somewhat controversial. However, a compelling argument is made in [10] and [4] that a very restrictive notion of "uniformity" is appropriate, especially for very small complexity classes such as ACC.

Simple diagonalization arguments show that uniform NC^1 (and much bigger classes) are properly contained in PSPACE. However, it is not known if uniform NC^1 contains the entire "counting hierarchy": PP, $\mathrm{PP}^{\mathrm{PP}}$, $\mathrm{PP}^{\mathrm{PP}^{\mathrm{PP}}}$,... In [1], we show that the results of [6] hold also in the uniform model, and then we are able to make use of this to show that uniform ACC is properly contained in PP. The same techniques allow us to prove an exponential lower bound on the size required for uniform ACC circuits to compute the permanent of a matrix. Slightly weaker "subexponential" lower bounds are proved on the size of circuits required to recognize sets in PP.

Although the proofs in [1] rely heavily on uniformity, there seems to be nobody who believes that the theorems themselves do not hold in the nonuniform model. An obvious open problem is to remove the hypothesis of uniformity. Another challenge is to prove exponential bounds on the size required to accept PP sets. And of course the most important challenge is to show that ACC is not equal to NC^1.

References

[1] E. Allender and V. Gore, *A uniform circuit lower bound for the permanent*, SIAM J. Comput., 23 (1994), pp. 1026–1049.

[2] E. Allender and U. Hertrampf, *Depth reduction for circuits of unbounded fan-in*, Inform. Comput. 112 (1994), pp. 217–238.

[3] D. Barrington, *Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1*, J. Comput. System Sci., 38 (1989), pp. 150–164.

[4] D. Barrington, N. Immerman, and H. Straubing, *On uniformity within NC^1*, J. Comput. System Sci., 41 (1990), pp. 274–306.

[5] D. Barrington and D. Thérien, *Finite monoids and the fine structure of NC^1*, J. Assoc. Comput. Mach., 35 (1988), pp. 941–952.

[6] R. Beigel and J. Tarui, *On ACC*, in Proc. 32nd Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Washington, DC, 1991, pp. 783–792.

[7] J. Håstad, Computational Limitations for Small Depth Circuits, MIT Press, Cambridge, MA, 1987.

[8] R. Kannan, H. Venkateswaran, V. Vinay, and A. Yao, *A circuit-based proof of Toda's theorem*, Inform. Comput., 104 (1993), pp. 271–276.

[9] A. Razborov, *Lower bounds for the size of circuits of bounded depth with basis $\{\wedge, \oplus\}$*, Math. notes of the Academy of Sciences of the USSR, 41 (1987), pp. 333–338.

[10] W. Ruzzo, *On uniform circuit complexity*, J. Comput. System Sci., 21 (1981), pp. 365–383.

[11] R. Smolensky, *Algebraic methods in the theory of lower bounds for Boolean circuit complexity*, in Proc. 19th Annual ACM Symposium on Theory of Computing, ACM Press, New York, 1987, pp. 77–82.

[12] S. Toda, *PP is as hard as the polynomial-time hierarchy*, SIAM J. Comput., 20 (1991), pp. 865–877.

[13] L. Valiant and V. Vazirani, *NP is as easy as detecting unique solutions*, Theoret. Comput. Sci., 47 (1986), pp. 85–93.

[14] A. Yao, *Separating the polynomial-time hierarchy by oracles*, in Proc. 26th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Washington, DC, 1985, pp. 1–10.

[15] A. Yao, *On ACC and threshold circuits*, in Proc. 31st Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, Washington, DC, 1990, pp. 619–627.