Electronic Colloquium on Computational Complexity, Report No. 142 (2015)

A Compression Algorithm for $\mathrm{AC}^0[\oplus]$ circuits using Certifying Polynomials

Srikanth Srinivasan*

Abstract

A recent work of Chen, Kabanets, Kolokolova, Shaltiel and Zuckerman (CCC 2014, Computational Complexity 2015) introduced the *Compression problem* for a class $\mathcal{C}$ of circuits, defined as follows. Given as input the truth table of a Boolean function $f : \{0,1\}^n \to \{0,1\}$ that has a small (say size s) circuit from $\mathcal{C}$, find in time $2^{O(n)}$ *any* Boolean circuit for f of size less than trivial, i.e. much smaller than $2^n/n$.

The work of Chen et al. gave compression algorithms for many classes of circuits including AC^0 (the class of constant-depth unbounded fan-in circuits made up of AND, OR, and NOT gates) and Boolean formulas of size nearly n^2. They asked if similar results can be obtained for the circuit class $\mathrm{AC}^0[\oplus]$, the class of circuits obtained by augmenting AC^0 with unbounded fan-in parity gates.

We answer the question positively here, using techniques from work of Kopparty and the author (FSTTCS 2012).

1 Introduction

We recall the notion of the *Compression problem* for a circuit class $\mathcal{C}$ from the work of Chen et al. [3]. The input to the problem is the truth table of a Boolean function $f : \{0,1\}^n \to \{0,1\}$ which is promised to have a 'small' circuit from the class $\mathcal{C}$. The desired output is a general Boolean circuit C (not necessarily from the class $\mathcal{C}$) of small size that computes the function f; the size of C should be smaller than the trivial $2^n/n$ that is achievable for any Boolean function. Moreover, we require the algorithm that constructs C to run in time polynomial in the size of its input, which is in time $\mathrm{poly}(2^n)$.

The aforementioned paper of Chen et al. [3] that introduced this problem showed that there is a polynomial time compression algorithm for AC^0 in the following sense: given as input the truth table of $f : \{0,1\}^n \to \{0,1\}$ which has an AC^0 circuit of size s and depth $d = O(1)$, the algorithm outputs a circuit computing f of size at most $2^{n-n/(O(\log s))^{d-1}}$. Similar compression algorithms were also obtained for functions that have de Morgan formulas of size at most $n^{2.5-\Omega(1)}$, Boolean formulas (over the complete basis) of size $n^{2-\Omega(1)}$ and read-once branching programs of size $2^{n(\frac{1}{2}-\Omega(1))}$: we refer the reader to [3] for the compression obtained in these cases.

Chen et al. asked if similar compression algorithms could be obtained for $\mathrm{AC}^0[\oplus]$. We resolve this question here, though with slightly weaker parameters.

Theorem 1. *There is a polynomial time algorithm which, when given as input the truth table of a function $f : \{0,1\}^n \to \{0,1\}$ and parameters s and $d = O(1)$ such that f has an $\mathrm{AC}^0[\oplus]$ circuit of size s and depth d, outputs a circuit C of size $2^{n-n/(O(\log s))^{2(d-1)}}$ computing f.*

*Department of Mathematics, IIT Bombay, Mumbai, India. srikanth@math.iitb.ac.in

ISSN 1433-8092

We begin by formally defining our key technical tool, the notion of *certifying polynomials* from [5]. Throughout the paper, we identify $\{0,1\}$ with $\mathbb{F}_2$. Given any function $g : \{0,1\}^n \to \mathbb{F}_2$, we use $\mathrm{Supp}(g)$ to denote the set of x such that $g(x)$ is non-zero.

Definition 2 (Certifying polynomial)**.** *A* non-zero *polynomial* $P(X_1, \ldots, X_n) \in \mathbb{F}_2[X_1, \ldots, X_n]$ *is a certifying polynomial for a function* $f : \{0,1\}^n \to \{0,1\}$ *if* f *is constant on* $\mathrm{Supp}(P)$*. We say that* P *is* b*-certifying, for* $b \in \{0,1\}$*, if* $f|_{\mathrm{Supp}(P)} = b$*.*

This definition is very similar to the notions of *weak*-2 *degree* and *algebraic immunity*, that already appear in the literature [4, 1, 2].

We will also need the notion of a *probabilistic polynomial.*

Definition 3 (Probabilistic polynomials)**.** *An* ε*-error probabilistic polynomial of degree* D *for a function* $f : \{0,1\}^n \to \{0,1\}$ *is a random polynomial* $\mathbf{P}$ *of degree at most* D *(chosen according to some distribution over polynomials of degree at most* D*) such that for any* $x \in \{0,1\}^n$*, we have* $\Pr_{\mathbf{P}}[f(x) = \mathbf{P}(x)] \geq 1 - \varepsilon$*.*

The following lemma was proved in [5] by building on classical circuit approximation techniques of Razborov [8].

Lemma 4 ([5])**.** *For any* $\varepsilon \in (0, 1/2)$*, any* $\mathrm{AC}^0[\oplus]$ *circuit* C *of size* s *and depth* d *has an* ε*-error probabilistic polynomial of degree at most* $(c \log s)^{d-1} \cdot (\log(1/\varepsilon))$ *for some absolute constant* $c > 0$*. In particular, there is a polynomial* $P \in \mathbb{F}_2[X_1, \ldots, X_n]$ *of degree at most* $(c \log s)^{d-1} \cdot (\log(1/\varepsilon))$ *such that*

$$\Pr_{x \sim \{0,1\}^n}[f(x) = P(x)] \geq 1 - \varepsilon.$$

2 Compression algorithms for $\mathrm{AC}^0[\oplus]$ circuits

Proof Idea. Our starting point is the proof of a theorem from [5] which shows that the input function f has a certifying polynomial of degree at most $D = n/2 - n/(O(\log s))^{2(d-1)}$. We sketch the idea here. Let $\varepsilon = \exp(-n/(c \log s)^{2(d-1)})$ for a constant $c > 0$ yet to be chosen. To construct such a certifying polynomial, we start with a polynomial P (given by Lemma 4) of degree at most $d = (c_1 \log s)^{d-1} \cdot \log(1/\varepsilon)$ that computes f on all but an ε fraction of inputs. Let $\mathcal{E}_P$ be the set of inputs where P does not compute f correctly. We then construct a non-zero polynomial Q of degree $D_0 = \frac{n}{2} - c_2\sqrt{n \log(1/\varepsilon)}$ that vanishes on $\mathcal{E}_P$: to be able to do this, c_2 is chosen so that the number of monomials of degree at most D_0 is greater than $\varepsilon \cdot 2^n \geq |\mathcal{E}_P|$, which implies that there is a non-zero Q as above; the polynomial $Q \cdot P$ is then a 1-certifying polynomial for f of degree at most $D_0 + d$.[1] By now choosing c large enough, we obtain a certifying polynomial of the required degree, which finishes the proof.

Note that the above idea also gives us an efficient algorithm for *constructing* such a certifying polynomial: formally, given the truth table of f, we can efficiently find a certifying polynomial for f of degree at most $D_0 + d$, since the problem of finding a 1-certifying polynomial polynomial for f is equivalent to finding a non-zero solution to a system of homogeneous linear equations over $\mathbb{F}_2$ where the variables correspond to coefficients of monomials of degree at most $D_0 + d$.

This gives us a hint of how to go about compressing the function f. We can try to find a 1-certifying polynomial for f of degree at most $D_0 + d$. Note that (for a suitable choice of c)

[1] There is actually a slight subtlety here since $Q \cdot P$ might be the zero polynomial. In this case, the polynomial $Q \cdot (1 - P)$ is a 0-certifying polynomial for f. For simplicity, we assume for now that this issue does not arise. In the actual algorithm, this will not happen unless f has very few 1s and can thus be easily computed by a brute force circuit.

the number of monomials in such a polynomial is $2^{n-n/(\log s)^{O(d)}}$, and hence this polynomial can be represented as a depth-2 $\mathrm{AC}^0[\oplus]$ circuit of this size (alternately, since the parity function on m bits has a an circuit over the de Morgan basis of size $O(m)$, we can also represent this polynomial as a circuit over the de Morgan basis of size $2^{n-n/(\log s)^{O(d)}}$). Hence, the certifying polynomial gives us a 'small' circuit that computes f correctly on a certain subset of inputs (and in particular is never wrong on inputs of $f^{-1}(0)$).

However, we are looking for a small circuit that computes f *everywhere*. To obtain such a circuit, we try to look for many 1-certifying polynomials $R_1, \ldots, R_m$ and try to "cover" all the 1-inputs of f. If we are able to do this with a small m, then $\bigvee_{i=1}^{m} R_i$ computes the function f. But there are two things that could go wrong with such an approach:

- By definition, any 1-certifying polynomial R is forced to vanish at all inputs $x \in f^{-1}(0)$. However, this could also force R to vanish at some inputs $y \in f^{-1}(1)$. Such "forced" inputs y cannot be covered by any 1-certifying polynomial R.
- Each 1-certifying polynomial R that we find might cover very few $y \in f^{-1}(1)$ and hence we might require many 1-certifying polynomials to cover all of $f^{-1}(1)$.

Handling the second of these issues is not too difficult: we can use a simple linear algebraic argument to show that for each y that is not forced in the above sense, a significant fraction of 1-certifying polynomials cover y. Coupled with a covering argument from [3], we can show that there are a few certifying polynomials that cover all such y.

To get around the first issue, we use a beautiful recent result of Nie and Wang [6], which implies that the number of forced y is vanishingly small if the parameters are chosen carefully. We are therefore able to hardcode these y into our circuit without a significant blowup in size. This finishes the proof.

We now state the result of Nie and Wang that we will use. Given a subset $\mathcal{E} \subseteq \{0,1\}^n$ and a parameter $D \leq n$, we define the *degree D closure of $\mathcal{E}$*, denoted $\mathrm{cl}_D(\mathcal{E})$, which is the set of all points $y \in \{0,1\}^n$ such that any polynomial Q of degree at most D_1 that vanishes on $\mathcal{E}$ vanishes on y.

Theorem 5 (Theorem 5.6 in [6])**.** *Let N_D denote the number of multilinear monomials of degree at most D. Then, we have*

$$\frac{|\mathrm{cl}_D(\mathcal{E})|}{2^n} \leq \frac{|\mathcal{E}|}{N_D}.$$

We now prove Theorem 1.

Proof of Theorem 1. We assume that d and ε are as above. The constant $c > 0$ in the definition of $\varepsilon > 0$ will be chosen below. We will assume that for the c we choose, the quantity $(c \log s)^{2(d-1)} < n/100$: otherwise, the compression algorithm can just output a trivial circuit of size $2^n/n$ for f.

Let $D_1 = \frac{n}{2} - c_3\sqrt{n \log(1/\varepsilon)}$ for a constant $c_3 > 0$ that is chosen to so that the number of monomials of degree at most D_1 is $N_{D_1} \geq \sqrt{\varepsilon} 2^n$. We choose c so that $D' = D_1 + d = n/2 - n/(O(\log s))^{(d-1)}$.

We call $y \in f^{-1}(1)$ forced if any polynomial R that vanishes on $f^{-1}(0)$ also vanishes on y. Let $F \subseteq f^{-1}(1)$ be the set of all forced y. We will prove the following two claims:

Claim 6. $|F| \leq 2^{n-n/(O(\log s))^{2(d-1)}}$.

Claim 7. *There is a polynomial-time algorithm $\mathcal{A}_1$ which when given f, outputs the descriptions of at most $m = O(n)$ 1-certifying polynomials $R_1, \ldots, R_m$ such that for each $y \in f^{-1}(1) \setminus F$, there is an $i \in [m]$ such that $y \in \mathrm{Supp}(R_i)$.*

Given the above two claims, the description of the compression algorithm $\mathcal{A}$ is simple: first run $\mathcal{A}_1$ and obtain a collection of 1-certifying polynomials $R_1, \ldots, R_m$ such that $\bigcup_i \mathrm{Supp}(R_i) = f^{-1}(y) \setminus F$. In particular, if C_i is a circuit of size $2^{n-n/(O(\log s))^{2(d-1)}}$ that accepts exactly the inputs in $\mathrm{Supp}(R_i)$, then $C' = \bigvee_i C_i$ is a circuit of the required size that accepts exactly the set $f^{-1}(y) \setminus F$. The algorithm now constructs a DNF C_F of size $O(n \cdot |F|)$ that accepts exactly the inputs in F (the set F is easily inferred from the circuit C'). The circuit C output by the algorithm is $C' \vee C_F$, which computes f by definition and also has the required size.

It remains to prove Claims 6 and 7, which we do below.

Proof of Claim 6. Let P and $\mathcal{E}_P$ be as above. Note that if $y \notin \mathrm{cl}_{D_1}(\mathcal{E}_P)$, then there is a polynomial Q of degree at most D_1 that vanishes at all points in $\mathcal{E}_P$ but not at y. Hence, the polynomial $Q \cdot P$ is a 1-certifying polynomial for f of degree at most D' that is non-zero at y and thus, y is not forced. Stated in the contrapositive, this argument tells us that $F \subseteq \mathrm{cl}_{D_1}(\mathcal{E}_P)$ and therefore, $|F| \leq |\mathrm{cl}_{D_1}(\mathcal{E}_P)|$.

By Theorem 5, we have

$$\frac{|\mathrm{cl}_{D_1}(\mathcal{E}_P)|}{2^n} \leq \frac{|\mathcal{E}_P|}{N_{D_1}}$$

Since $|\mathcal{E}_P| \leq \varepsilon 2^n$ and $N_{D_1} \geq \sqrt{\varepsilon} 2^n$, we see that the right hand size of the above inequality is bounded by $\sqrt{\varepsilon}$, which implies the claim. □

Proof of Claim 7. Let V denote the vector space of polynomials Q of degree at most D' such that Q vanishes on $f^{-1}(0)$. Note that $F' := f^{-1}(1) \setminus F$ satisfies $F' = \bigcup_{Q \in V} \mathrm{Supp}(Q)$. Let $Q_1, \ldots, Q_N$ be a generating set of V. Note that $N \leq 2^n$. A generic element of V is given by $\sum_{i=1}^{N} \alpha_i Q_i$ for some choice of $\alpha_1, \ldots, \alpha_N \in \mathbb{F}_2$; we denote this element by $Q_{\overline{\alpha}}$, where $\overline{\alpha}$ denotes the vector $(\alpha_1, \ldots, \alpha_N)$.

For any $y \in F'$, we have $Q_{\overline{\alpha}}(y) = \sum_i \alpha_i Q_i(y)$, which is a linear function of $\overline{\alpha}$. Since $y \in F'$, it is not forced to 0 and hence not all the $Q_i(y)$ are 0. Thus, for a random choice of the α_i, the probability that $Q_{\overline{\alpha}}(y) \neq 0$ is $\frac{1}{2}$. We can derandomize this argument using binary error-correcting codes.

Say we have vectors $U = \{u_1, \ldots, u_N\} \subseteq \mathbb{F}_2^M$ (where $M = 2^{O(n)}$) that generate an error-correcting code of distance δM for some constant $\delta > 0$. There are many standard constructions of such sets U in time $\mathrm{poly}(2^n)$ (see, e.g., ??). Let $\mathcal{M}$ be an $M \times N$ matrix with columns $u_1, \ldots, u_N$. Let $\overline{\alpha}^1, \ldots, \overline{\alpha}^M$ denote the rows of $\mathcal{M}$. For any non-zero $\beta_1, \ldots, \beta_N$ we know that $u = \sum_i \beta_i u_i$ has at least δM many non-zero entries. In other words, for any non-zero vector $\overline{\beta} = (\beta_1, \ldots, \beta_N) \in \mathbb{F}_2^N$ and a random $j \in [M]$, the probability that the inner product of $\overline{\beta}$ and $\overline{\alpha}^j$ is non-zero is at least δ.

We are now ready to describe the algorithm $\mathcal{A}_1$. The algorithm needs to finds $m = O(n)$ elements $R_1, \ldots, R_m$ from V such that $F' \subseteq \bigcup_i \mathrm{Supp}(R_i)$. The algorithm goes through m iterations, the ith iteration producing a polynomial $R_i \in V$. After each iteration, we ensure that the number of elements in F' left uncovered thus far drops by the constant factor $(1 - \delta)$; thus, at the end of $m = 2n \log(1/\delta)$ iterations, all the elements of F' will be covered.

Let $F'_i = F' \setminus \bigcup_{p<i} \mathrm{Supp}(R_p)$ be the set of elements of F' left uncovered after $i-1$ iterations. In the ith iteration, the algorithm looks at each of the rows of $\mathcal{M}$ and picks the j such that $s_j = |\mathrm{Supp}(\sum_{i \in [N]} \overline{\alpha}_i^j Q_i) \cap F'_i|$ is maximized. We know that $v_y := (Q_1(y), \ldots, Q_N(y))$ is a non-zero vector for any choice of $y \in F'_i$. Hence, for a random $j \in [M]$, the probability that the inner product of $\overline{\alpha}^j$ and v is non-zero is at least δ. By averaging, there must be a $j \in [M]$ such that the inner product of $\overline{\alpha}^j$ and v_y is non-zero for at least a δ-fraction of the $y \in F'_i$. Thus, $|F'_{i+1}| \leq (1 - \delta)|F'_i|$. □

□

3 Extension to the MOD_p case

The compression algorithms extend fairly straightforwardly to the setting of $\mathrm{AC}^0[p]$ circuits. The right definition of certifying polynomials is obtained by simply replacing 2 by p in Definition 2 (where $\mathrm{Supp}(P)$ is the set of points x s.t. $P(x) \neq 0$). The only missing links in the proof is an extension of Lemma 4 to the setting of $\mathrm{AC}^0[p]$ and the theorem of Nie and Wang [6] in this setting. The former appears in the work of Oliveira and Santhanam [7]. For the latter, it turns out that Theorem 5 holds over *any* field. For fields other than $\mathbb{F}_2$, this is a slightly different statement than the one that appears in the work of Nie and Wang, who only consider the closure over the larger domain $\mathbb{F}^n$, where $\mathbb{F}$ is any finite field. However, a straightforward modification of their argument also gives the result for closure over $\{0,1\}^n \subseteq \mathbb{F}^n$, where $\mathbb{F}$ can be any field (possibly even infinite).

Acknowledgements

The author would like to thank Abhishek Bhrushundi, Prahladh Harsha, Valentine Kabanets, Antonina Kolokolova and Swastik Kopparty for useful discussions, feedback, and encouragement.

References

[1] Michael Alekhnovich and Alexander A. Razborov. Lower bounds for polynomial calculus: Non-binomial case. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 190–199, 2001.

[2] Claude Carlet, Deepak Kumar Dalai, K. C. Gupta, and Subhamoy Maitra. Algebraic immunity for cryptographically significant boolean functions: Analysis and construction. *IEEETIT: IEEE Transactions on Information Theory*, 52, 2006.

[3] Ruiwen Chen, Valentine Kabanets, Antonina Kolokolova, Ronen Shaltiel, and David Zuckerman. Mining circuit lower bound proofs for meta-algorithms. *Computational Complexity*, 24(2):333–392, 2015.

[4] Frederic Green. A complex-number fourier technique for lower bounds on the mod-m degree. *Computational Complexity*, 9(1):16–38, 2000.

[5] Swastik Kopparty and Srikanth Srinivasan. Certifying polynomials for acˆ0(parity) circuits, with applications. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, pages 36–47, 2012.

[6] Zipei Nie and Anthony Y. Wang. Hilbert functions and the finite degree zariski closure in finite field combinatorial geometry. *Journal of Combinatorial Theory, Series A*, 134:196 – 220, 2015.

[7] Igor Carboni Oliveira and Rahul Santhanam. Majority is incompressible by acˆ0[p] circuits. In *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, pages 124–157, 2015.

[8] Alexander A. Razborov. Lower bounds on the size of constant-depth networks over a complete basis with logical addition. *Mathematicheskie Zametki*, 41(4):598–607, 1987.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de