A new characterization of ACC^0 and probabilistic CC^0

Kristoffer Arnsfelt Hansen
Department of Computer Science
Aarhus University
Århus, Denmark
Email: arnsfelt@cs.au.dk

Michal Koucký
Institute of Mathematics
Academy of Sciences of Czech Republic
Prague, Czech Republic
Email: koucky@math.cas.cz

***Abstract*—Barrington, Straubing and Thérien (1990) conjectured that the Boolean AND function can not be computed by polynomial size constant depth circuits built from modular counting gates, i.e., by CC^0 circuits. In this work we show that the AND function can be computed by uniform probabilistic CC^0 circuits that use only $O(\log n)$ random bits. This may be viewed as evidence contrary to the conjecture.**

As a consequence of our construction we get that all of ACC^0 can be computed by probabilistic CC^0 circuits that use only $O(\log n)$ random bits. Thus, if one were able to derandomize such circuits, we would obtain a collapse of circuit classes giving $\mathrm{ACC}^0 = \mathrm{CC}^0$. We present a derandomization of probabilistic CC^0 circuits using AND and OR gates to obtain $\mathrm{ACC}^0 = \mathrm{AND} \circ \mathrm{OR} \circ \mathrm{CC}^0 = \mathrm{OR} \circ \mathrm{AND} \circ \mathrm{CC}^0$. AND and OR gates of sublinear fan-in suffice.

Both these results hold for uniform as well as non-uniform circuit classes. For non-uniform circuits we obtain the stronger conclusion that $\mathrm{ACC}^0 = \mathrm{rand}-\mathrm{ACC}^0 = \mathrm{rand}-\mathrm{CC}^0 = \mathrm{rand}(\log n)-\mathrm{CC}^0$, i.e., probabilistic ACC^0 circuits can be simulated by probabilistic CC^0 circuits using only $O(\log n)$ random bits.

As an application of our results we obtain a characterization of ACC^0 by constant width planar nondeterministic branching programs, improving a previous characterization for the quasipolynomial size setting.

I. Introduction

Bounded depth circuits are a natural computational model introduced in early 80's as a restriction of general Boolean circuits. Despite the almost 30 years of study we still do not know the model quite well. The celebrated results of Furst, Saxe and Sipser [22] also proven independently by Ajtai [1] show that the PARITY (MOD_2) function cannot be computed by polynomial size constant depth circuits consisting of AND and OR gates of unbounded fan-in — AC^0 circuits. This result was further improved by Yao and Håstad [29], [48] to show that exponential size is necessary to compute PARITY by AC^0 circuits. Razborov [37] and Smolensky [39] extended this result to show that exponential size is necessary to compute MOD_q by constant depth circuits consisting of AND, OR and MOD_p gates of unbounded fan-in — ACC^0 circuits — if p is a prime co-prime with q. Since then our understanding did not expand much further as far as lower bounds are concerned. Indeed, we cannot rule out that all functions in NP are computable by depth 3 circuits consisting of MOD_6 gates with the number of gates being linear in the input size. Such a possibility seems highly implausible, though.

In fact, Barrington, Straubing and Thérien [13] conjectured that even the Boolean AND function cannot be computed by polynomial size bounded depth circuits consisting entirely of MOD_q gates — CC^0 circuits. This seems to be a natural conjecture dual to the fact that MOD_q cannot be efficiently computed by AND and OR. The conjecture is indeed true when $q = p^k$ is a prime power [13]: any constant depth circuit consisting of only MOD_q gates, $q = p^k$, is equivalent to a constant degree polynomial over $\mathbf{Z}_p$ and therefore cannot compute the AND function *at all* regardless of its size. On the other hand, when q is not a prime power exponentially large CC^0 circuits can compute any Boolean function [13]. Improving on results of Barrington [10] and Smolensky [40], Thérien [44] shows that CC^0 circuits computing AND require at least $\Omega(n)$ gates at the bottom level. To date no better lower bound on the number of gates has been obtained for CC^0 circuits computing any explicit function.

In this paper we show that *probabilistic* CC^0 circuits can compute AND. These circuits can be constructed to use only $O(\log n)$ random bits. As a consequence, the entire class ACC^0 can be computed by probabilistic CC^0 circuits using only $O(\log n)$ random bits. We suggest that this fact may be viewed as evidence that AND can in fact be computed by small CC^0 circuits.

Obviously, if our probabilistic CC^0 circuits could be derandomized we would obtain a family of CC^0 circuits computing the AND function. However, we do not know how to do this. (Here by derandomization of probabilistic CC^0 circuits we mean finding deterministic CC^0 circuits computing the same functions, or more generally, by derandomization of probabilistic CC^0 circuits to a complexity class $\mathcal{C}$ we mean showing that the functions computed by the probabilistic CC^0 circuits are from the class $\mathcal{C}$.) Indeed, we observe as a consequence to our results that derandomizing probabilistic CC^0 circuits to CC^0 circuits is in fact *equivalent* to constructing CC^0 circuits for the AND function. This leaves us with the following two exciting possibilities: Either we have a collapse of circuit classes giving $\mathrm{ACC}^0 = \mathrm{CC}^0$, or the conjecture of Barrington, Straubing and Thérien is true and CC^0 constitutes an intriguing computational model that *cannot* be derandomized.

Using the technique of Ajtai and Ben-Or [4], and Allender and Hertrampf [6] we can derandomize probabilistic CC^0 circuits to the (possibly) larger class of $\textsc{And} \circ \textsc{Or} \circ \mathrm{CC}^0$ circuits — circuits consisting of a single $\textsc{And}$ gate at the top fed by $\textsc{Or}$ gates that in turn take as inputs the output of CC^0 circuits. $\textsc{And}$ and $\textsc{Or}$ gates of sublinear fan-in suffice for this. Hence, arbitrarily complex ACC^0 circuits can be converted into such $\textsc{And} \circ \textsc{Or} \circ \mathrm{CC}^0$ circuits. This is rather intriguing when one considers the fact that there is a proper hierarchy of functions computed by AC^0 circuits of different depths. Thus, it is evident that there must be a considerable computational power hidden in $\textsc{Mod}_q$ gates.

A final application of this derandomization and characterization of ACC^0 is to provide the missing piece of a complete characterization of the classes AC^0, ACC^0 and NC^1 in terms of constant width nondeterministic branching programs under *geometric* restrictions. This line of research originates with the surprising result due to Barrington [14] that constant width polynomial size branching programs as well as Boolean circuits compute exactly NC^1. His proof proceeds by showing that the *word problem* over the group S_5 of permutations on 5 elements is complete for NC^1. Barrington and Thérien [17] extended this to a complete *algebraic* characterization of the classes AC^0, ACC^0 and NC^1 showing that word problems over different classes of finite monoids are complete for them; AC^0 corresponds to the class of aperiodic monoids, ACC^0 to the class of solvable monoids, and NC^1 to all finite monoids. A *geometric* characterization of the classes AC^0, ACC^0 and NC^1 in terms of constant width circuits was obtained by Barrington et al. and Hansen. Namely, AC^0 is precisely the class of functions computed by constant width upward planar circuits of polynomial size [16], and ACC^0 is precisely the class of functions computed by constant width planar circuits of polynomial size [25] . Turning to the model of branching programs, Barrington et al. [15] and Vinay [45] gave a characterization of AC^0 by constant width upward planar nondeterministic branching programs. Recently Hansen [27] gave a characterization of ACC^0 by constant width planar nondeterministic branching programs, but only in the quasipolynomial size setting — it was left as an open question whether such a characterization could be obtained in the polynomial size setting. We can now answer this in the affirmative.

The techniques that we use to obtain our results are not new. They rest mainly on the approximation method introduced by Razborov [37] and its extensions. As such one may expect that further ideas will be needed in order to derandomize CC^0 circuits. We do feel however that our results shed new light on the power of such circuits. Our understanding of CC^0 circuits is still very limited. Despite that fact, it is now well established that counting modulo a composite number can have surprising computational power [11], [26]. Indeed, our current understanding of this power has been sufficient to exploit it in exciting constructions such as set systems with restricted intersections and Ramsey graphs [24] as well as the recent 3-query locally decodable codes [21].

A. Boolean circuits

For a function $f : \{0,1\}^* \rightarrow \{0,1\}$ and an integer n, f_n denotes f restricted to the inputs of size n. We say that f is in AC^0 if there is a family of circuits $\{C_n\}_{n\geq 1}$ such that the circuit C_n is of size polynomial in n and constant depth, it consists of negation gates and unbounded fan-in $\textsc{And}$ and $\textsc{Or}$ gates, and it computes f_n. We call such a circuit family AC^0 *circuits*. Similarly, for fixed integer $m > 1$, we say that f is in $\mathrm{CC}^0[m]$ if there is a family of constant depth, polynomial size circuits consisting of negation gates and $\textsc{Mod}_m$ gates computing f. We call such circuits $\mathrm{CC}^0[m]$ circuits. A $\textsc{Mod}_m$ gate is a gate that evaluates to one on an input $x \in \{0,1\}^n$ iff the number of ones in x is not divisible by m. The class CC^0 of functions is defined by $\mathrm{CC}^0 = \bigcup_{m>1} \mathrm{CC}^0[m]$. Finally, f is in ACC^0 if for some fixed m, f is computable by a family of circuits of constant depth and polynomial size consisting of negation gates and unbounded fan-in $\textsc{And}, \textsc{Or}$ and $\textsc{Mod}_m$ gates.

We also consider functions computable by probabilistic circuits. For circuits of type $\mathcal{C}$ we say that f is in $\mathrm{rand}(r(n)){-}\mathcal{C}$ if f is computable by a family of circuits $\{C_n\}_{n\geq 1}$ of type $\mathcal{C}$ where each circuit C_n takes in addition to its actual input of size n also $O(r(n))$ many random bits and for each input $x \in \{0,1\}^n$ it outputs $f_n(x)$ with probability $\geq 2/3$ where the probability is taken over the random bits. By $\mathrm{rand}{-}\mathcal{C}$ we denote $\bigcup_{k>0} \mathrm{rand}(n^k){-}\mathcal{C}$.

We will consider *uniform* as well as *non-uniform* circuits. We say that a circuit family $\{C_n\}$ is uniform if the connectivity language of the circuit family is computable in linear space and polynomial time in the size of gate indices. One could call this *DPLOGTIME-uniformity* whereas the usual DLOGTIME-uniformity [12], [38] corresponds to the case of the connectivity language being computable in *linear* time in the size of the gate indices. One may wonder why we use DPLOGTIME uniformity instead of the more usual DLOGTIME uniformity. Several of our proofs use walks on expanders and outcome of the expander walks is essentially built into the circuit. Hence a procedure that construct the circuit or decides its connectivity language has to be able to calculate those outcomes. This can certainly be done in time polynomial in the length of the expander walks but we do not know how to do it in linear time. In the last section we will use the weaker notion of logspace-uniformity.

By *the size* of a circuit we will understand the number of wires (see [33].)

II. Probabilistic construction

We use the technique of Razborov and Smolensky [37], [39] to show the following lemma:

Lemma 1 (Main Lemma). *If $p, q \geq 2$ are co-prime integers, then $\textsc{Or}_n$ can be computed with error $n^{-\log n}$ by uniform probabilistic polynomial size constant depth circuits consisting of $\textsc{Mod}_{pq}$ gates.*

To prove the lemma we use the following proposition that appears implicitly in [13] and can be found stated explicitly in [42] (see also [41, Theorem VIII.3.1]).

Proposition 2 (Barrington, Straubing, Thérien). *If $p, q \geq 2$ are co-prime integers, then $\textsc{Or}_{\log n}$ can be computed by uniform polynomial size constant depth circuits consisting of $\textsc{Mod}_{pq}$ gates.*

Proof: (Main Lemma) Let $\ell = \log^2 n$. Pick independently at random ℓ sets $S_1, S_2, \ldots, S_\ell \in \{1, \ldots, n\}$. We claim that for any $x \in \{0,1\}^n$ with probability at least $1 - \frac{1}{n^{\log n}}$ the following circuit computes the $\textsc{Or}$ of x correctly:

$$\bigvee_{j=1}^{\ell} \textsc{Mod}_q\{x_i;\ i \in S_j\} \ .$$

Indeed, let $x \neq 0^n$ and j be fixed. Clearly, $\textsc{Mod}_q\{x_i;\ i \in S_j\}$ evaluates to one with probability $\geq 1/2$. As $S_1, \ldots, S_\ell$ are chosen independently, the probability that for all j, $\textsc{Mod}_q\{x_i;\ i \in S_j\}$ evaluates to zero is at most $2^{-\ell} = n^{-\log n}$. For $x = 0^n$ the circuit clearly evaluates to zero always. Since $\textsc{Mod}_q$ can be computed using $\textsc{Mod}_{pq}$ gates and $\textsc{Or}_{\log^2 n}$ can be computed by a depth two tree of $\textsc{Or}_{\log n}$ gates, using the previous proposition we obtain a probabilistic distribution of deterministic CC^0 circuits computing $\textsc{Or}_n$ of a given input with high probability. (For each choice of sets $S_1, \ldots, S_\ell$ we have one CC^0 circuit.) To obtain a probabilistic CC^0 circuit rather than the probability distribution of deterministic circuits we use $\ell \times n$ random bits $r_{1,1}, r_{1,2}, \ldots, r_{\ell,n}$ in addition to the input x. Bits $r_{j,1}, \ldots, r_{j,n}$ determine the characteristic sequence of set S_j. Thus the computation of the final probabilistic circuit proceeds according to the following formula:

$$\bigvee_{j=1}^{\ell} \textsc{Mod}_q\{x_i \wedge r_{j,i};\ i \in \{1, \ldots, n\}\} \ .$$

Note, for $m > 2$, a binary $\textsc{Or}$ can be computed by feeding the two inputs into a $\textsc{Mod}_m$ gate. A single $\textsc{Mod}_m$ gate can also be used to compute $\textsc{Not}$. Hence using DeMorgan's rule, each binary $\textsc{And}$ can be computed with four $\textsc{Mod}_{pq}$ gates. ∎

Theorem 3 (Main Theorem). *Both uniformly and non-uniformly we have*

$$\mathrm{ACC}^0 \subseteq \mathrm{rand} - \mathrm{CC}^0 \ .$$

Proof: Any ACC^0 circuit of size $O(n^k)$ built out of $\textsc{Not}, \textsc{And}, \textsc{Or}$ and $\textsc{Mod}_q$ gates can be transformed into an ACC^0 circuit of size $O(n^k)$ built out of $\textsc{Not}, \textsc{Or}$ and $\textsc{Mod}_{6q}$ gates. As noted in the previous proof, $\textsc{Not}$ gates can be replaced by $\textsc{Mod}_{6q}$ gates. Since $\textsc{Or}$ can be computed by probabilistic CC^0 circuits using $\textsc{Mod}_6$ gates it can also be computed by probabilistic CC^0 circuits using $\textsc{Mod}_{6q}$ gates. By replacing each $\textsc{Or}$ gate in the ACC^0 circuit by the probabilistic CC^0 circuit consisting of $\textsc{Mod}_{6q}$ gates we obtain a probabilistic CC^0 circuit computing the same function as the original ACC^0 circuit. The success probability of this circuit will be at least $1 - O\left(\frac{n^k}{n^{\log n}}\right) \geq 1 - n^{-O(\log n)}$. ∎

We note that the simulation can be done efficiently in size. Using the downward self-reducibility of $\textsc{Or}_n$ [5], [32] one can prove that for any fixed $\epsilon > 0$, $\textsc{Or}_n$ has $\mathrm{rand} - \mathrm{CC}^0$ circuits of size $O(n^{1+\epsilon})$. This implies that ACC^0 circuits of size $O(n^k)$ can be simulated by $\mathrm{rand} - \mathrm{CC}^0$ circuits of size $O(n^{k+\epsilon})$ for an arbitrary small $\epsilon > 0$.

The above proof of the main lemma requires polynomially many random bits. Although we do not know how to derandomize these circuits we can at least reduce the required number of random bits. Allender et al. [7] provide a more randomness efficient construction of $\textsc{Or}_n$.

Proposition 4 (Allender et al. [7], Lemma 4.4). *For each $c \geq 1$, $\textsc{Or}_n$ can be computed with error $\leq 1/n^c$ by probabilistic polynomial size constant depth circuits consisting of $\textsc{Mod}_2$ and $\textsc{And}_{O(\log n)}$ gates and taking $O(\log n)$ random bits.*

The proof of Allender et al. is based on randomness optimal isolation lemma of Chari et al. [20] and random walks on expanders [31]. We note that the original Valiant-Vazirani isolation lemma [46] based on pair-wise independent hash functions together with randomness efficient hash functions based on convolution could also be used to prove the lemma. Allender et al. [7] claim that their construction is logspace-uniform. One can easily verify that it is DPLOGTIME-uniform and can be generalized to arbitrary $\textsc{Mod}_q$ gates. We state the following corollary.

Corollary 5. *For all $c \geq 1$ and co-prime integers $p, q \geq 2$, $\textsc{Or}_n$ can be computed with error $\leq 1/n^c$ by uniform probabilistic polynomial size constant depth circuits consisting of $\textsc{Mod}_{pq}$ gates and taking $O(\log n)$ random bits.*

Proof: We provide a brief sketch of the proof. The random bits in the Main lemma were used to select sets $S_1, \ldots, S_\ell$. It turns out that one does not need to use fully independent random bits but rather one can use somewhat correlated bits. In particular, one can use Valiant-Vazirani isolation lemma [46] to select each set S_j, and instead of using fully independent sets S_j one can use sets $S_1, \ldots, S_\ell$ determined by a random walk on an expander. We elaborate on this little bit more. Denote by $k = \lceil \log_2 n \rceil$. Let H_k^m be a

2-universal family of hash functions from $\{0,1\}^k$ to $\{0,1\}^m$ [19]. The following fact is well known, see e.g. [9].

Claim 6. *Let $S \subseteq \{0,1\}^k$ be a non-empty set and m be an integer satisfying $2^m/8 \le |S| \le 2^m/4$. Then*

$$\Pr_{h\in H_k^m}[|\{i \in S;\ h(i) = 0^m\}| = 1] \ge 1/16\ .$$

Let $H_k^2, H_k^3, \dots, H_k^{k+2}$ be 2-universal families of hash functions. Pick $h_1^2, h_2^2, \dots, h_\ell^2 \in H_k^2$, $\dots$, $h_1^{k+2}, h_2^{k+2}, \dots, h_\ell^{k+2} \in H_k^{k+2}$ at random and replace each set S_j in the proof of Main lemma by sets $S_j^2, \dots, S_j^{k+2}$ defined as $S_j^m = \{i \in \{1,\dots,n\};\ h_j^m(\bar{i}) = 0^m\}$, where $\bar{i}$ stands for the k-bit binary representation of i.

For fixed $x \in \{0,1\}^n$, if the hash functions h_j^m are picked independently at random from the respective hash families then with probability at least $1 - \left(\frac{15}{16}\right)^\ell$ the following circuit computes the OR of x correctly:

$$\bigvee_{j\in\{1,\dots,\ell\},\ m\in\{2,\dots,k+2\}} \textsc{Mod}_q\{x_i;\ i \in S_j^m\}\ .$$

Since we allow for error $1/n^c$, we set $\ell = O(\log n)$. The number of random bits one needs for the construction depends on how many random bits are needed to specify a single function in any of the 2-universal families. Nisan [36] gives for every $k, m \ge 1$, a 2-universal family H_k^m based on convolution such that each hash function in H_k^m can be uniquely determined by $k + 2m - 1$ bits. Hence, a direct implementation of the above scheme uses only $O(\log^3 n)$ random bits. We need to do somewhat better, though.

One can save a factor of $\log n$ by observing that for each j, $h_j^2, h_j^3, \dots, h_j^{k+2}$ do not have to be independent. Indeed, h_j^m for $m \in \{2,\dots,k+1\}$ can be taken to be the projection of h_j^{k+2} on the first m coordinates of the image. (Projecting all functions in a 2-universal family of hash functions on the same set of coordinates gives a family (multi-set) of hash functions that is again 2-universal.) Thus to fully specify $h_j^2, h_j^3, \dots, h_j^{k+2}$ we need only $3k+3$ bits. The last savings of another $\log n$-factor will be achieved by not taking h_j^{k+2} for different j's fully independent but rather, h_j^{k+2}'s will be given by steps of a random walk on an expander with the vertex set H_k^{k+2}. An analysis similar to the one in the proof of Theorem 12 shows that for suitably chosen $\ell = O(\log n)$ the error in computing OR of a given x will still be bounded by $1/n^c$.

The random walk will require $\log_2 |H_k^{k+2}| = 3k + 3$ random bits to specify the starting vertex and $O(\ell) = O(\log n)$ random bits to specify all the individual steps of the walk of length ℓ. Thus, $O(\log n)$ bits will be needed to specify all the sets S_j^m. It remains to explain how will the probabilistic CC^0 circuit compute. The following formula describes the circuit that takes input $x \in \{0,1\}^n$ and random bits $r \in \{0,1\}^{O(\log n)}$:

$$\bigvee_{j\in\{1,\dots,\ell\},\ m\in\{2,\dots,k+2\}} \textsc{Mod}_q\{x_i \wedge \mathrm{Magic}_{r',h,j,i,0^m}(r);$$
$$i \in \{1,\dots,n\},\ h \in H_k^m,\ r' \in \{0,1\}^{O(\log n)}\}.$$

The $\mathrm{Magic}_{r',h,j,i,0^m}(r)$ predicate is one iff $r' = r$, the j-th step of the random walk given by r' determines hash function h, and $h(\bar{i}) \in 0^m\{0,1\}^{k+2-m}$. Clearly, given $r', h, j, i, 0^m$ we can verify in polynomial time in the length of $r', h, j, i, 0^m$ that the j-th step of the random walk given by r' determines hash function h, and $h(\bar{i}) \in 0^m\{0,1\}^{k+2-m}$. Here we assume that the expander is explicitly constructible. Thus evaluation of $\mathrm{Magic}_{r',h,j,i,0^m}(r)$ reduces to the problem of checking $r = r'$ which can be done by polynomial size CC^0 circuits by Proposition 2. Thus the overall circuit is a CC^0 circuit of polynomial size and can be constructed in DPLOGTIME. (The random walk and hash function evaluation is thus not performed by the circuit itself but rather by the procedure constructing the circuit. The hash function evaluation could easily be done by the circuit but we do not know how to evaluate the random walk by a CC^0 circuit. This leads to DPLOGTIME-uniformity.) ■

This provides the following strengthening of the main theorem.

Theorem 7. *Both uniformly and non-uniformly we have*

$$\mathrm{ACC}^0 \subseteq \mathrm{rand}(\log n){-}\mathrm{CC}^0\ .$$

We will use later the fact that Corollary 5 allows us to convert any ACC^0 circuit into a $\mathrm{rand}(\log n){-}\mathrm{CC}^0$ circuit which computes the same function with probability of error bounded by $1/n^k$, for any fixed $k \ge 1$.

Ajtai and Ben-Or [4] show that non-uniformly $\mathrm{rand}{-}\mathrm{AC}^0$ is equal to AC^0; the same technique applies to ACC^0.

Proposition 8 (Ajtai and Ben-Or). *Non-uniformly we have*

$$\mathrm{ACC}^0 = \mathrm{rand} - \mathrm{ACC}^0\ .$$

However, we do not know how to derandomize $\mathrm{rand} - \mathrm{CC}^0$ even non-uniformly as we do not know how to compute AND in CC^0 (Catch 22). However, since non-uniformly $\mathrm{rand} - \mathrm{CC}^0 \subseteq \mathrm{rand} - \mathrm{ACC}^0 \subseteq \mathrm{ACC}^0$ we can non-uniformly reduce the number of random bits in any CC^0 circuit to obtain:

Corollary 9. *Non-uniformly we have*

$$\mathrm{ACC}^0 = \mathrm{rand}{-}\mathrm{ACC}^0 = \mathrm{rand}{-}\mathrm{CC}^0 = \mathrm{rand}(\log n){-}\mathrm{CC}^0.$$

III. Derandomization

Our $\mathrm{rand} - \mathrm{CC}^0$ circuits are allowed to have error $\le 1/3$. It is clear that one can uniformly decrease the error probability to $1/2^{\log^k n}$, for any fixed $k \ge 1$, by taking $O(\log^k n)$ independent copies of the circuit and outputting the majority output. Since it suffices to compute *approximate* majority of $O(\log^k n)$ output bits we can use uniform AC^0 circuits for computing approximate majority of $O(\log^k n)$ bits as provided by Ajtai [2]. These circuits are built out of $\mathrm{AND}_{\log^{O(1)} n}$ and $\mathrm{OR}_{\log^{O(1)} n}$ gates so they can be converted to CC^0 circuits by Proposition 2 to obtain uniform polynomial size $\mathrm{rand} - \mathrm{CC}^0$ circuits computing the original function with error probability at most $1/2^{\log^k n}$. Using the technique of Ajtai and Ben-Or [4] one can then non-uniformly derandomize $\mathrm{rand} - \mathrm{CC}^0$ circuits as follows.

Theorem 10. *If f_n is computable by $\mathrm{rand} - \mathrm{CC}^0$ circuits with error $\leq 1/3n$ then f_n is computable by $\mathrm{AND}_n \circ \mathrm{OR}_n \circ \mathrm{CC}^0$ non-uniform circuits.*

Proof: To obtain an $\mathrm{AND} \circ \mathrm{OR} \circ \mathrm{CC}^0$ circuit computing f correctly on inputs of length n we proceed as follows. Take OR of n independent copies of the $\mathrm{rand} - \mathrm{CC}^0$ circuit for f. This yields an $\mathrm{OR} \circ \mathrm{rand} - \mathrm{CC}^0$ circuit that computes the function f on one-inputs, i.e., inputs in $f^{-1}(1)$, correctly with probability $\geq 1 - \frac{1}{(3n)^n}$. At the same time the $\mathrm{OR} \circ \mathrm{rand} - \mathrm{CC}^0$ circuit computes f correctly on zero-inputs, i.e., inputs in $f^{-1}(0)$, with probability at least $1 - \frac{n}{3n} \geq \frac{2}{3}$. Taking AND of n independent copies of the $\mathrm{OR} \circ \mathrm{rand} - \mathrm{CC}^0$ gives circuit that is correct on one-inputs with probability at least $1 - \frac{n}{(3n)^n}$ and on zero-inputs with probability at least $1 - 3^{-n}$. As on every input of length n, the error of the resulting $\mathrm{AND} \circ \mathrm{OR} \circ \mathrm{rand} - \mathrm{CC}^0$ circuit is smaller than 2^{-n}, there is a certain choice of the random bits which yields an $\mathrm{AND} \circ \mathrm{OR} \circ \mathrm{CC}^0$ circuit that computes f correctly on all inputs of length n. ■

A similar proof establishes that $\mathrm{rand} - \mathrm{CC}^0 \subseteq \mathrm{OR} \circ \mathrm{AND} \circ \mathrm{CC}^0$. We can thus replace arbitrarily complex ACC^0 circuit by MOD gates, n OR gates of fan-in n and a single AND gate of fan-in n. In fact, one can make the number of OR gates and the fan-in of the AND and OR gates slightly sub-linear. This seems to suggest that MOD gates have sufficient power to compute AND and OR.

These results are similar in spirit to several known results about depth reduction of constant depth circuits. The difference is that those results eliminate the AND and OR gates by means of the depth reduction, while our results utilize the power of modular counting to do the elimination. A drawback of the depth reduction results is that they incur an inherent quasipolynomial increase in the size of the circuit, which we are able to avoid in our results.

Allender and Hertrampf prove that for a prime p, quasipolynomial size $\mathrm{CC}^0[p]$ is equal to quasipolynomial size $\mathrm{AND} \circ \mathrm{OR} \circ \mathrm{MOD}_p \circ \mathrm{AND}_{\log^{O(1)} n}$ circuits as well as $\mathrm{OR} \circ \mathrm{AND} \circ \mathrm{MOD}_p \circ \mathrm{AND}_{\log^{O(1)} n}$ circuits [6]. Beigel and Tarui show that quasipolynomial size ACC^0 can be computed by quasipolynomial size circuits consisting of a gate computing a symmetric function fed by AND gates of fan-in $\log^{O(1)} n$ [18]. Tarui shows that quasipolynomial size AC^0 can be computed by quasipolynomial size depth 3 circuits consisting of an OR gate at the output, fed by n MAJ gates that are fed by AND gates of fan-in $\log^{O(1)} n$ [43] (the fan-in of the top gate can in fact be slightly sub-linear).

Next we consider uniform derandomization. In [27] a theorem of Allender and Hertrampf [6] was used to convert a uniform probabilistic circuit to a deterministic uniform circuit.

Proposition 11 (Allender and Hertrampf)**.** *Let $\{C_n\}$ be a uniform family of probabilistic circuits taking $r(n)$ random bits and computing a family of Boolean functions $\{f_n\}$ with error probability less than $1/r(n)$. Then there is a uniform family of deterministic circuits computing $\{f_n\}$ that consists of circuits with the top gates being:*

$$\mathrm{OR}_{2^{r(n)^2}} \circ \mathrm{AND}_{2^{r(n)}} \circ \mathrm{OR}_{O(r(n))} \ ,$$

that take as their inputs outputs of copies of C_n with random bits hardwired.

This theorem is not suitable for us as the resulting circuit is of quasi-polynomial size. So we need a more efficient version of this conversion. Viola [47] considered a similar question of efficiently converting probabilistic circuits into deterministic circuits, albeit in a somewhat different language. His objective was to minimize the running time needed to calculate the connectivity language of the circuit. This objective also lead to somewhat quasi-polynomial size circuits. We use techniques similar to [6] and [47] based on Lautemann's proof of $\mathrm{BPP} \in \Sigma^2$ [34] to establish the following claim.

Theorem 12. *Let $\{C_n\}$ be a uniform family of probabilistic circuits taking $r(n) \geq 1$ random bits and computing a family of Boolean functions $\{f_n\}$ with error probability less than $1/(21r(n))$. Then there is a uniform family of deterministic circuits computing $\{f_n\}$ that consists of circuits with the top gates being:*

$$\mathrm{OR}_{2^{O(r(n))}} \circ \mathrm{AND}_{2^{r(n)}} \circ \mathrm{OR}_{O(r(n))} \ ,$$

that take as their inputs outputs of copies of C_n with random bits hardwired.

Proof: We want to uniformly construct $\mathrm{OR}_{2^{O(r(n))}} \circ \mathrm{AND}_{2^{r(n)}} \circ \mathrm{OR}_{O(r(n))}$-type circuit computing f_n. For a fixed input $w \in \{0,1\}^n$, let $S_w \subseteq \{0,1\}^{r(n)}$ be the set of random strings for which C_n outputs 1 on input w. If $f(w) = 1$ then $|S_w| \geq 2^{r(n)}(1 - 1/21r(n))$ and $|S_w| \leq 2^{r(n)}/21r(n)$ otherwise. We will build a circuit that will distinguish these two cases. We use the method of Lautemann [34]. Lautemann shows that any set $A \subseteq \{0,1\}^m$ has the following two properties:

1) For any integer $\ell > 1$, if $|A| < 2^m/\ell$ then $\forall x_1, \ldots, x_\ell \in \{0,1\}^m$, $\bigcup_{i=1}^{\ell}(A \oplus x_i) \subsetneq \{0,1\}^m$.
2) If $|A| \geq 2^{m-1}$ then $\exists x_1, \ldots, x_m \in \{0,1\}^m$ such that $\bigcup_{i=1}^{m}(A \oplus x_i) = \{0,1\}^m$.

Here, $A \oplus x_i = \{y \oplus x_i;\ y \in A\}$, where the XOR is bitwise. The proof is a simple counting argument and we return to it later. This directly allows one to build $\textsc{Or}_{2^{O(r(n)^2)}} \circ \textsc{And}_{2^{r(n)}} \circ \textsc{Or}_{O(r(n))}$-type circuit computing f_n. Namely, the following circuit computes f_n:

$$\bigvee_{x_1,\ldots,x_{r(n)} \in \{0,1\}^{r(n)}} \bigwedge_{y \in \{0,1\}^{r(n)}} \bigvee_{i=1}^{r(n)} C_n(y \oplus x_i) \ ,$$

where $C_n(y \oplus x_i)$ is the circuit C_n with random bits hardwired to $y \oplus x_i$. It is straightforward to verify using Lautemann's properties that this circuit computes f_n on each input $w \in \{0,1\}^n$, as the set S_w is of size either larger than $2^{r(n)}/2$ or smaller than $2^{r(n)}/21r(n)$ depending on the value $f(w)$.

The size of the top-most OR is too large, however as there are $2^{r(n)^2}$ choices for $x_1, \ldots, x_{r(n)}$. Using the standard method of random walks on expanders we reduce the number of necessary bits to $2^{O(r(n))}$. Margulis and Gaber and Galil [23], [35] describe a sequence of simple 8-regular graphs G_n, with G_n having n^2 vertices, that are known to satisfy the following property [3], [8].

Proposition 13 ([30], Theorems 3.6 and 8.2). *Let $n, \ell > 1$ be integers and $A \subseteq V(G_n)$. If $|A| \geq \frac{19}{20}|V(G_n)|$ then the probability that a simple random walk of length ℓ on G_n starting from a vertex chosen uniformly at random from $V(G_n)$ does not visit some vertex in A is at most $\left(\frac{19}{20}\right)^{\ell}$.*

Pick G_n with the number of vertices equal to 2^m, where m is an integer. For $z \in \{0,1\}^{m+3\ell}$ let $\mathrm{rw}(z)_0, \mathrm{rw}(z)_1, \ldots, \mathrm{rw}(z)_\ell$ be the sequence of vertices visited by a walk of length ℓ on G_n determined by z; the first m bits of z determine the starting vertex $\mathrm{rw}(z)_0$ of the walk, and each consecutive three bits of z determine the neighbor of the current vertex that will be the next vertex. We claim that any set $A \subseteq \{0,1\}^m$ has the following two properties:

1) For any integer $\ell > 1$, if $|A| < 2^m/(\ell+1)$ then $\forall z \in \{0,1\}^{m+3\ell}$, $\bigcup_{i=0}^{\ell}(A \oplus \mathrm{rw}(z)_i) \subsetneq \{0,1\}^m$.
2) If $|A| \geq \frac{19}{20}2^m$ then $\exists z \in \{0,1\}^{61m}$ such that $\bigcup_{i=0}^{20m}(A \oplus \mathrm{rw}(z)_i) = \{0,1\}^m$.

The first property holds trivially and the second property holds for the following reason. Let $|A| \geq \frac{19}{20}2^m$. For $u, v \in \{0,1\}^m$, u is not in $A \oplus v$ iff $v \notin A \oplus u$. Hence, for $u \in \{0,1\}^m$ and $z \in \{0,1\}^{61m}$, u is not in $\bigcup_{i=0}^{20m}(A \oplus \mathrm{rw}(z)_i)$ iff $\mathrm{rw}(z)_0, \mathrm{rw}(z)_1, \ldots, \mathrm{rw}(z)_{20m} \notin A \oplus u$. For a fixed $u \in \{0,1\}^{r(n)}$ and random $z \in \{0,1\}^{61m}$, the probability of $\mathrm{rw}(z)_0, \mathrm{rw}(z)_1, \ldots, \mathrm{rw}(z)_{20m} \notin A \oplus u$ is at most $\left(\frac{19}{20}\right)^{20m} < 2^{-m}$ by the previous proposition. By union bound there must by some $z \in \{0,1\}^{61m}$, such that for every $u \in \{0,1\}^m$, u is in $\bigcup_{i=0}^{20m}(A \oplus \mathrm{rw}(z)_i)$. So the properties hold. Chose $m = r(n)$ and $\ell = 20m$ (we assume without loss of generality that $r(n)$ is even.) Clearly, the following circuit computes f_n:

$$\bigvee_{z \in \{0,1\}^{61r(n)}} \bigwedge_{y \in \{0,1\}^{r(n)}} \bigvee_{i=0}^{20r(n)} C_n(y \oplus \mathrm{rw}(z)_i) \ .$$

Since the graphs G_n are very simple to describe and the i-th neighbor of any vertex can be computed in polynomial time in the length of the description of the vertex, the connectivity language of this circuit can be computed in DPLOGTIME. ■

This together with our Theorem 7 and Proposition 2 yields the following corollary.

Corollary 14. *Both uniformly and non-uniformly we have*

$$\mathrm{ACC}^0 = \textsc{And} \circ \textsc{Or} \circ \mathrm{CC}^0 = \textsc{Or} \circ \textsc{And} \circ \mathrm{CC}^0 \ .$$

IV. Constant width planar branching programs

In this section we will use our results to improve upon a recent characterization of ACC^0 circuits by constant width *planar* nondeterministic branching programs [27]. All results in this section hold in the non-uniform as well as logspace-uniform setting. The characterization obtained was the following.

Theorem 15 (Hansen). *Constant width quasipolynomial size planar nondeterministic branching programs compute exactly quasipolynomial* ACC^0.

By a constant width quasipolynomial size planar nondeterministic branching program is simply meant a nondeterministic branching program in layered form where every layer contains a constant number of nodes, having the property that as a digraph it can be drawn in the plane with no arcs crossing. For precise definitions we refer the reader to [27], [28].

The proof of the above characterization involved a number of steps where all except one could be done in polynomial size. One direction of the characterization was obtained by Hansen, Miltersen and Vinay [28].

Proposition 16 (Hansen, Miltersen, Vinay). *Every function computed by constant width polynomial size planar nondeterministic branching programs is in* ACC^0.

For the other direction, the following part of the characterization was done in the polynomial size setting [27].

Proposition 17 (Hansen). *Any function computed by constant depth* $\textsc{And} \circ \textsc{Or} \circ \mathrm{CC}^0$ *circuits of polynomial size is also computed by a constant width planar nondeterministic branching programs of polynomial size.*

The final part of Theorem 15 was proved by a quasipolynomial version of our Corollary 14. The proof of this used ideas similar to parts of the proofs of our results here. The main cause for the inefficiency leading only to a result in the quasipolynomial setting was the use of probabilistic polynomials instead of our use here of more complicated CC^0 circuits.

With the improved result as stated by Corollary 14 we finally obtain the following characterization of ACC^0.

Theorem 18. *A function is computable by a constant width polynomial size planar non-deterministic branching programs if and only if it is in* ACC^0.

We thus answer the open question of [27] affirmatively.

Acknowledgment

We thank Eric Allender for useful comments and pointers to previous literature. The work of the first author is supported by a postdoc fellowship from the Carlsberg Foundation. The work of the second author is partially supported by grant GA ČR 201/07/P276, project No. 1M0021620808 of MŠMT ČR, Institutional Research Plan No. AV0Z10190503 and grant IAA100190902 of GA AV ČR. Part of this work was done while the second author was visiting Aarhus University.

References

[1] M. Ajtai, "Σ^1_1-formulae on finite structures," *Annals of Pure and Applied Logic*, vol. 24, no. 1, pp. 1–48, 1983.

[2] ——, "Approximate counting with uniform constant depth circuits," in *Advances in Computational Complexity Theory*, ser. DIMACS Series in Disc. Math. and Theoret. Comp. Sci., J.-Y. Cai, Ed., 1993, pp. 1–20.

[3] M. Ajtai, J. Komlós, and E. Szemerédi, "Deterministic simulation in LOGSPACE," in *Proceedings of the 19th annual ACM symposium on Theory of computing*. ACM, 1987, pp. 132–140.

[4] M. Ajtai and M. Ben-Or, "A theorem on probabilistic constant depth computations," in *Proceedings of the 16th Annual ACM Symposium on Theory of computing*. ACM, 1984, pp. 471–474.

[5] E. Allender and M. Koucký, "Amplifying lower bounds by means of self-reducibility," in *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity*. IEEE Computer Society Press, 2008, pp. 31–40.

[6] E. Allender and U. Hertrampf, "Depth reduction for circuits of unbounded fan-in," *Information and Computation*, vol. 112, no. 2, pp. 217–238, 1994.

[7] E. Allender, J. Jiao, M. Mahajan, and V Vinay, "Non-commutative arithmetic circuits: Depth reduction and size lower bounds," *Theoretical Computer Science*, vol. 209, no. 1–2, pp. 47–86, 1998.

[8] N. Alon, U. Feige, A. Wigderson, and D. Zuckerman, "Derandomized graph products," *Computational Complexity*, vol. 5, no. 1, pp. 60–75, 1995.

[9] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[10] D. A. M. Barrington, "Some problems involving Razborov-Smolensky polynomials," in *Boolean Function Complexity*, ser. London Mathematical Society Lecture Note Series, M. S. Paterson, Ed. Cambridge University Press, 1992, vol. 169, pp. 109–128.

[11] D. A. M. Barrington, R. Beigel, and S. Rudich, "Representing boolean functions as polynomials modulo composite numbers," *Computational Complexity*, vol. 4, pp. 367–382, 1994.

[12] D. A. M. Barrington, N. Immerman, and H. Straubing, "On uniformity within NC^1," *Journal of Computer and System Sciences*, vol. 41, no. 3, pp. 274–306, 1990.

[13] D. A. M. Barrington, H. Straubing, and D. Thérien, "Non-uniform automata over groups," *Information and Computation*, vol. 89, no. 2, pp. 109–132, 1990.

[14] D. A. M. Barrington, "Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1," *Journal of Computer and System Sciences*, vol. 38, no. 1, pp. 150–164, 1989.

[15] D. A. M. Barrington, C.-J. Lu, P. B. Miltersen, and S. Skyum, "Searching constant width mazes captures the AC^0 hierarchy," in *Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science*, ser. Lecture Notes in Computer Science, vol. 1373. Springer, 1998, pp. 73–83.

[16] ——, "On monotone planar circuits," in *Proceedings of the 14th Annual IEEE Conference on Computational Complexity*. IEEE Computer Society Press, 1999, pp. 24–31.

[17] D. A. M. Barrington and D. Thérien, "Finite monoids and the fine structure of NC^1," *Journal of the ACM*, vol. 35, no. 4, pp. 941–952, 1988.

[18] R. Beigel and J. Tarui, "On ACC," *Computational Complexity*, vol. 4, no. 4, pp. 350–366, 1994.

[19] J. L. Carter and M. N. Wegman, "Universal classes of hash functions," *Journal of Computer and System Sciences*, vol. 18, no. 2, pp. 143–154, 1979.

[20] S. Chari, P. Rohatgi, and A. Srinivasan, "Randomness-optimal unique element isolation with applications to perfect matching and related problems," in *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*. ACM, 1993, pp. 458–467.

[21] K. Efremenko, "3-query locally decodable codes of subexponential length," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (to appear)*, 2009.

[22] M. Furst, J. B. Saxe, and M. Sipser, "Parity, circuits, and the polynomial-time hierarchy," *Mathematical Systems Theory*, vol. 17, no. 1, pp. 13–27, 1984.

[23] O. Gabber and Z. Galil, "Explicit constructions of linear-sized superconcentrators," *Journal of Computer and System Sciences*, vol. 22, no. 3, pp. 407–420, 1981.

[24] V. Grolmusz, "Superpolynomial size set-systems with restricted intersections mod 6 and explicit Ramsey graphs," *Combinatorica*, vol. 20, no. 1, pp. 71–86, 2000.

[25] K. A. Hansen, "Constant width planar computation characterizes ACC^0," *Theory of Computing Systems*, vol. 39, no. 1, pp. 79–92, 2006.

[26] ——, "On modular counting with polynomials," in *Proceedings of the 21st Annual IEEE Conference on Computational Complexity*. IEEE Computer Society Press, 2006, pp. 202–212.

[27] ——, "Constant width planar branching programs characterize ACC^0 in quasipolynomial size," in *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity*. IEEE Computer Society Press, 2008, pp. 92–99.

[28] K. A. Hansen, P. B. Miltersen, and V Vinay, "Circuits on cylinders," *Computational Complexity*, vol. 15, no. 1, pp. 62–81, 2006.

[29] J. Håstad, *Computational limitations of small-depth circuits*. MIT Press, 1987.

[30] S. Hoory, N. Linial, and A. Wigderson, "Expander graphs and their applications," *Bulletin of the American Mathematical Society*, vol. 43, pp. 439–561, 2006.

[31] R. Impagliazzo and D. Zuckerman, "How to recycle random bits," in *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, 1989, pp. 248–253.

[32] M. Koucký, "Circuit complexity of regular languages," *Theory of Computing Systems*, to appear.

[33] M. Koucký, P. Pudlák, and D. Thérien, "Bounded-depth circuits: separating wires from gates," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*. ACM, 2005, pp. 257–265.

[34] C. Lautemann, "BPP and the polynomial hierarchy," *Information Processing Letters*, vol. 17, no. 4, pp. 215–217, 1983.

[35] G. Margulis, "Explicit constructions of expanders," *Problemy Peredači Informacii*, vol. 9, no. 4, pp. 71–80, 1973.

[36] N. Nisan, "Pseudorandom generators for space-bounded computation," *Combinatorica*, vol. 12, no. 4, pp. 449–461, 1992.

[37] A. A. Razborov, "Lower bounds for the size of circuits of bounded depth with basis $(\wedge, \oplus)$," *Mathematical Notes of the Academy of Science of the USSR*, vol. 41, no. 4, pp. 333–338, 1987.

[38] W. L. Ruzzo, "On uniform circuit complexity," *Journal of Computer and System Sciences*, vol. 22, no. 3, pp. 365–383, 1981.

[39] R. Smolensky, "Algebraic methods in the theory of lower bounds for Boolean circuit complexity," in *Proceedings of the 19th annual ACM Symposium on Theory of Computing*. ACM, 1987, pp. 77–82.

[40] ——, "On interpolation by analytic functions with special properties and some weak lower bounds on the size of circuits with symmetric gates," in *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, 1990, pp. 628–631.

[41] H. Straubing, *Finite Automata, Formal Logic, and Circuit Complexity*. Birkhäuser, 1994.

[42] H. Straubing and D. Thérien, "A note on MOD_p - MOD_m circuits," *Theory of Computing Systems*, vol. 39, no. 5, pp. 699–706, 2006.

[43] J. Tarui, "Probabilistic polynomials, AC^0 functions and the polynomial-time hierarchy," *Theoretical Computer Science*, vol. 113, no. 1, pp. 167–183, 1993.

[44] D. Thérien, "Circuits constructed with MOD_q gates cannot compute "and" in sublinear size," *Computational Complexity*, vol. 4, pp. 383–388, 1994.

[45] V Vinay, "Hierarchies of circuit classes that are closed under complement," in *Proceedings of the 11th Annual IEEE Conference on Computational Complexity*. IEEE Computer Society Press, 1996, pp. 108–117.

[46] L. G. Valiant and V. V. Vazirani, "NP is as easy as detecting unique solutions," in *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*. ACM, 1985, pp. 458–463.

[47] E. Viola, "On approximate majority and probabilistic time," in *Proceedings of the 22nd Annual IEEE Conference on Computational Complexity*. IEEE Computer Society Press, 2007, pp. 155–168.

[48] A. C.-C. Yao, "Separating the polynomial-time hierarchy by oracles," in *Proceedings of the 26th Annual IEEE Symposium on Foundations of Computer Science*, 1985, pp. 1–10.