

# Constant Width Planar Branching Programs Characterize $\text{ACC}^0$ in Quasipolynomial Size

Kristoffer Arnsfelt Hansen  
Department of Computer Science  
The University of Chicago\*  
arnsfelt@daimi.au.dk

## Abstract

We revisit the computational power of constant width polynomial size planar nondeterministic branching programs. We show that they are capable of computing any function computed by a  $\Pi_2 \circ \text{CC}^0 \circ \text{AC}^0$  circuit in polynomial size. In the quasipolynomial size setting we obtain a characterization of  $\text{ACC}^0$  by constant width planar nondeterministic branching programs.

## 1 Introduction

In this paper we revisit the computational power of constant polynomial size and quasipolynomial size planar branching programs. These are equivalent to constant width cylindrical branching programs [8], which were introduced and studied in [9].

The present work, as well as the previous work, belong to the study of the relationship between the computational power of width restricted circuits and depth restricted circuits. There is a strong relationship between the classes of functions computed by circuits under these two restrictions. Quantifying this relationship can provide great insight into both classes. Sometimes it is even possible to show an equivalence between two such classes. A good example of this is that the class of functions computed by quasipolynomial size circuits of polylogarithmic depth is equal to the class of functions computed by quasipolynomial size circuits of polylogarithmic width.

A surprising and beautiful connection in the lower level circuit classes was found by Barrington [4], namely that the class of functions computed by constant width circuits of polynomial size is exactly the class of functions computed by logarithmic depth AND/OR circuits of constant fanin

( $\text{NC}^1$  circuits). Thus there is a provable difference between the class of functions computed by unbounded fanin constant depth AND/OR circuits ( $\text{AC}^0$  circuits) and the class of functions computed by constant width circuits.

Barrington et al. [6] showed that by placed a *geometric* restriction on constant width circuits one obtains a characterization of  $\text{AC}^0$ . Namely, the class of functions computed by polynomial size *upward planar* constant width circuits is exactly  $\text{AC}^0$ . We say that a circuit is upward planar, if it as a digraph can be drawn in the plane with no arcs crossing, and such that all arcs are monotonically increasing in a common (upward, say) direction.

Recently a characterization of the class of constant depth circuits AND/OR circuits augmented with *modular counting* (MOD) gates ( $\text{ACC}^0$  circuits) was proved [8]. Namely, the class of functions computed by polynomial size *planar* constant width circuits is exactly  $\text{ACC}^0$ . Naturally, a circuit is said to be planar, if it as a digraph can be drawn in the plane with no arcs crossing. Allender et al. [2] extended this characterization of  $\text{ACC}^0$ , by showing that also constant width, polynomial size circuits of polylogarithmic *genus* only compute functions in  $\text{ACC}^0$ .

Thus in terms of circuits we have a complete characterization of the three important classes of circuits  $\text{AC}^0$ ,  $\text{ACC}^0$  and  $\text{NC}^1$  in terms of geometric restrictions of polynomial size constant width circuits.

In the present work we consider the (nondeterministic) branching program model. The characterization of  $\text{NC}^1$  by Barrington also holds for polynomial size constant width branching programs. Similarly, for  $\text{AC}^0$  we also have a characterization in terms of branching programs. Barrington et al. [5] proved that the class of functions computed by constant width upward planar nondeterministic branching programs is exactly  $\text{AC}^0$ .

Thus far we have no similar result for  $\text{ACC}^0$ . Here we come very close, and show that in the *quasipolynomial size* setting we can indeed obtain a characterization of  $\text{ACC}^0$  in terms of branching programs.

---

\*Supported by a Villum Kann Rasmussen postdoc fellowship. Currently at the University of Aarhus, supported by a postdoc fellowship from the Carlsberg Foundation.

**Theorem 1** *Constant width quasipolynomial size planar nondeterministic branching programs compute exactly quasipolynomial size  $\text{ACC}^0$ .*

One part of this theorem was obtained in [9]. More precisely the following upper bound was shown.

**Theorem 2 (Hansen, Miltersen and Vinay)** *Every function computed by a constant width polynomial size planar nondeterministic branching program is in  $\text{ACC}^0$ .*

By a simple translation, this result also holds in the quasipolynomial size setting. For the other part of Theorem 1 we use the method of approximation due to Razborov [10] and Smolensky [11] to reduce the simulation of  $\text{ACC}^0$  circuits to a specific type of circuits. These are included in the simulation by our next result.

**Theorem 3** *Any function computed by a constant depth  $\Pi_2 \circ \text{CC}^0 \circ \text{AC}^0$  circuit of polynomial size is also computed by a constant width planar nondeterministic branching program of polynomial size.*

By a  $\Pi_2 \circ \text{CC}^0 \circ \text{AC}^0$  we mean a circuit with an AND gate at the output, OR gates at the next level, then an arbitrary constant number of levels consisting of MOD gates, and finally an arbitrary number of levels consisting of AND and OR gates.

Previously only a simulation of a single layer of MOD gates by planar branching programs was known [9].

**Theorem 4 (Hansen, Miltersen and Vinay)** *Any function computed by a constant depth  $\Pi_2 \circ \text{MOD} \circ \text{AC}^0$  circuit of polynomial size is also computed by a constant width planar nondeterministic branching program of polynomial size.*

In fact here, the MOD can be so-called *generalized* MOD gates. When we as in the present work consider an arbitrary number of levels of MOD gates this presents no additional generality.

A key ingredient to being able to simulate more than one layer of MOD gates is the modulus-amplifying polynomials introduced by Toda [14].

As noted previously, the class of constant width planar branching programs is equivalent to the class of constant width *cylindrical* branching programs. We say that a constant width branching program is cylindrical if it can be embedded on a cylinder surface with no arcs crossing in such a way that all arcs are monotonically increasing in the direction of the axis of the cylinder. While we have stated our results above in the presumably more familiar notion of planarity, we will in the remainder of the paper work with the notion of cylindricality, since this is more convenient to work with.

Similarly to constant width cylindrical branching programs one can consider constant width cylindrical circuits [9], and Theorem 2 and Theorem 4 holds also with “circuit” substituted for “nondeterministic branching program”. This played an important role in the characterization of  $\text{ACC}^0$  in terms of constant width planar circuits.

Since cylindrical circuits can simulate cylindrical branching programs, Theorem 3 and Theorem 1 also holds with “cylindrical circuit” substituted for “planar nondeterministic branching program”.

## 1.1 Organization of Paper

In Section 2 we define the notions of branching programs and circuits we consider in this paper. In Section 3 we will simulate a special case of the circuits of Theorem 3. The proof is concluded in Section 4 by showing that we can assume the special case without loss of generality. Finally in Section 5 we give the remaining proofs needed for our main result, Theorem 1.

## 2 Preliminaries

### 2.1 Constant Depth Circuits

A  $\text{MOD}_m$  takes  $n$  boolean inputs  $x_1, \dots, x_n$  and outputs 1 if  $\sum_{i=1}^n x_i \not\equiv 0 \pmod{m}$  and 0 otherwise.

We let  $\text{MOD}_m$  denote the family of  $\text{MOD}_m$  gates. Similarly let  $\text{AND}$  and  $\text{OR}$  denote the family of unbounded fanin AND and OR gates.

We will indicate restrictions on the fanin of AND and OR gates by a subscript.

If  $G$  is a family of boolean gates and  $\mathcal{C}$  is a family of circuits we let  $G \circ \mathcal{C}$  denote the class of polynomial size circuit families consisting of a  $G$  gate taking circuits from  $\mathcal{C}$  as inputs.

$\text{AC}^0$  is the class of functions computed by constant depth circuits built from unbounded fanin AND and OR gates.

$\text{CC}^0$  is the class of functions computed by constant depth circuits built entirely from  $\text{MOD}_m$  gates for constants  $m$ .

$\text{ACC}^0$  is the class of functions computed by constant depth circuits built from unbounded fanin AND, OR and  $\text{MOD}_m$  gates for constants  $m$ .

### 2.2 Cylindrical Branching Programs

A digraph  $D = (V, A)$  is called *layered* if there is a partition  $V = V_0 \cup V_1 \cup \dots \cup V_h$  such that all arcs of  $A$  goes from layer  $V_i$  to the next layer  $V_{i+1}$  for some  $i$ . We call  $h$  the *depth* of  $D$ ,  $|V_i|$  the width of layer  $i$  and  $k = \max |V_i|$  the width of  $D$ .

A *nondeterministic branching program* is an acyclic digraph where all arcs are labeled by either a literal, i.e. a variable or a negated variable, or a boolean constant, and an initial and a terminal node. An input is accepted if and only if there is a path from the initial node to the terminal node in the digraph that results from substituting constants for the literals according to the input and then deleting arcs labeled by 0.

We only consider branching programs in layered form, i.e. we assume that when viewed as a digraph they are layered. This translates the definition of size and width from layered digraphs to branching programs.

To make the informal definition of cylindricity defined earlier a little more precise, we say that a layered digraph is cylindrical if every layer can be placed on a cross section of the cylinder, and all arcs between layers only travel between two adjacent cross sections with no intersections. For a more precise combinatorial definition we refer to [9].

### 3 Improved Simulation of Constant Depth Circuits by Cylindrical Branching Programs

Our starting point is the simulation of a single  $\text{MOD}_m$  gate as done in [9]. The construction there is essentially as the example illustrated in Figure 1 (with an added twist allowing simulation of a layer of OR gates on top). The general construction is as follows. The branching program will have  $n + 3$  layers. The first and last layer containing a single node and the middle layers containing  $m$  nodes. The node in the first layer has an arc to node 1 in the second layer. Every node in the next-to-last layer except node 1 has an arc to the node in the last layer. The nodes in the middle layers represent the sum of a prefix of the input modulo  $m$  in the obvious way.

For simulating a  $\text{MOD}_m$  gate in general, all the arcs from the next-to-last layer to the last layer are necessary, since in the case that the  $\text{MOD}_m$  gate evaluates to 1, we only know that the sum of the inputs modulo  $m$  is nonzero.

The key to simulating an arbitrary number of  $\text{MOD}$  gates is to only consider circuits on a very special form. In the remainder of this section we will give the simulation proving the following.

**Proposition 5** *Any function computed by a constant depth  $\Pi_2 \circ \text{MOD}_{m_1} \circ \dots \circ \text{MOD}_{m_h} \circ \text{AC}^0$  circuit of polynomial size is also computed by a constant width nondeterministic cylindrical branching program of polynomial size, provided the following two condition holds.*

- $m_1 < \dots < m_h$ .
- *The sum of the inputs of every  $\text{MOD}_{m_i}$  gate always evaluate to 0 or 1 modulo  $m_i$ .*

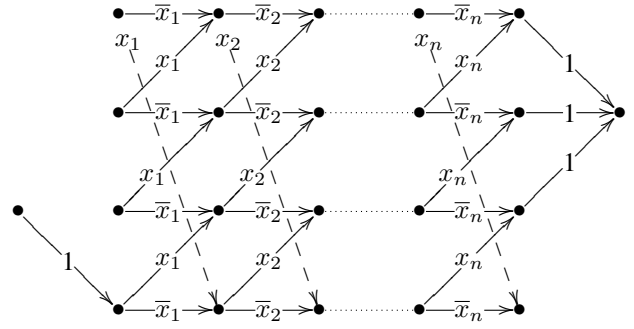


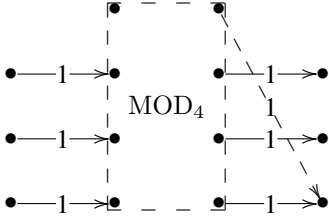
Figure 1. A cylindrical branching program computing  $\text{MOD}_4$ . The dashed arcs indicate a cylindrical embedding.

Suppose we want to simulate a  $\text{MOD}_{m_1} \circ \dots \circ \text{MOD}_{m_h}$  circuit satisfying the conditions of Proposition 5, by a cylindrical branching program.

Consider the branching program fragment obtained by removing the first and last layer of the branching program described for computing  $\text{MOD}_m$ . We can view it as computing a relation on  $[m]$ . The relation computed describes exactly the paths from a node in the first layer to a node in the last layer in the digraph obtained by substituting constants for the literals according to a given input and deleting arcs labeled with 0. Under our assumption, exactly two possible such relations can be computed, namely the identity relation and the relation sending  $i$  to  $i + 1$  (where we count modulo  $m$ ).

We can convert such a branching program fragment, into another branching program fragment computing this relation for a given  $m' < m$ . This is done by adding a new layer before the first layer and adding a new layer after the last layer. From the first layer we have  $m'$  arcs from the first  $m'$  nodes to the corresponding nodes in the old first layer. Similarly, from the old last layer we will have  $m'$  arcs from the first  $m'$  nodes to the corresponding nodes in the new last layer, but additionally we will have an arc from node  $m' + 1$  in the old last layer to node 1 in the new old layer. An example of this construction is shown in Figure 2.

We can then construct a cylindrical branching program for a  $\text{MOD}_{m_1} \circ \dots \circ \text{MOD}_{m_h}$  circuit satisfying the conditions of Proposition 5, as follows. By induction we can assume that we have constructed branching program fragments computing the relation for  $\text{MOD}_{m_2}$  corresponding to the outputs of every  $\text{MOD}_{m_2} \circ \dots \circ \text{MOD}_{m_h}$  subcircuit. We then convert all these into computing the relation for  $\text{MOD}_{m_1}$  as described above. Then by simply concatenating these fragments we obtain a branching program fragment computing the relation for  $\text{MOD}_{m_1}$  that corresponds

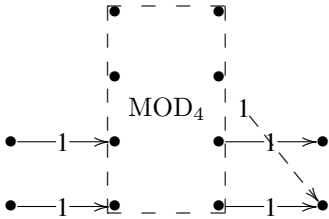


**Figure 2. Converting a fragment for  $\text{MOD}_4$  into a fragment for  $\text{MOD}_3$**

to the output of the entire circuit, thereby completing the construction.

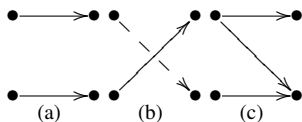
The rest of the simulation is essentially as in [9]. For completeness we give the full construction of the parts that are slightly changed.

We will now show that we can combine the fragments constructed above to compute the OR function of such  $\text{MOD}_{m_1} \circ \dots \circ \text{MOD}_{m_h}$  circuits. The starting point is first to transform these fragments to compute the relation for  $\text{MOD}_2$ , described above, as illustrated in Figure 3 (alternatively, we could simply assume without loss of generality that  $m_1 = 2$ ).



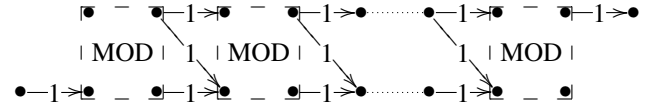
**Figure 3. Preparing a fragment for  $\text{MOD}_4$  to compute OR.**

Consider now the relations over  $\{1, 2\}$  as shown in Figure 4. The branching program fragment we have constructed computes the relation (a) if the output is 0 and computes the relation (b) if the output is 1. The construction for simulating an OR gate at the output is now done by interleaving branching program fragments computing the relation (c) between the fragments computing the relations corresponding to the inputs to the OR gate.



**Figure 4. Some elements of  $M_2$ .**

This construction is a way of “short circuiting” the branching program in the case that one of the  $\text{MOD}_{m_1}$  gates that are inputs to the OR gate in fact evaluates to 1. Finally we add layers at both ends, connecting the bottom node in the previous first layer and to the top node in the previous layer. The entire construction is shown in Figure 5.



**Figure 5. A cylindrical branching program computing  $\text{OR} \circ \text{MOD} \circ \dots \circ \text{MOD}$ .**

This establishes that we can simulate  $\text{OR} \circ \text{MOD}_{m_1} \circ \dots \circ \text{MOD}_{m_h}$  circuits satisfying the conditions of Proposition 5 by cylindrical branching programs.

For the remaining parts we refer to the following results. First as we need the fact that we can compute  $\text{AC}^0$  by upward planar branching programs, being one part of the following theorem due to Vinay [15] and Barrington et al. [5].

**Theorem 6** *A language is in  $\text{AC}^0$  if and only if it is accepted by a polynomial size, constant width upward planar branching program.*

And for making using these upward planar branching programs we make use of the following substitution lemma from [9].

**Lemma 7** *If  $f(x_1, \dots, x_n)$  is computed by a cylindrical branching program of size  $s_1$  and width  $w_1$  and  $g_1, \dots, g_n$  and  $\bar{g}_1, \dots, \bar{g}_n$  are computed by upward planar branching programs, each of size  $s_2$  and width  $w_2$  then  $f(g_1, \dots, g_n)$  is computed by a cylindrical branching program of size  $O(s_1 s_2)$  and width  $O(w_1 w_2)$ .*

Thus by combining Theorem 6 with Lemma 7 and the above constructions we can simulate  $\text{OR} \circ \text{MOD}_{m_1} \circ \dots \circ \text{MOD}_{m_h} \circ \text{AC}^0$  circuits satisfying the conditions of Proposition 5 by cylindrical branching programs, and using the fact that branching programs are closed under AND simply by concatenation we obtain the full construction for the proof of Proposition 5

## 4 Transformation of $\text{CC}^0$ Circuits

In this section we show that without loss of generality  $\Pi_2 \circ \text{CC}^0 \circ \text{AC}^0$  can be assumed to satisfy the requirements of Proposition 5, and thereby complete the proof of Theorem 3.

The key ingredient is the modulus-amplifying polynomials introduced by Toda [14].

We say that a single-variate integer polynomial  $A$  is degree- $d$  modulus amplifying if for all integers  $n$  and  $m$  we have the following property.

$$\begin{aligned} n \equiv 0 \pmod{m} &\Rightarrow A(n) \equiv 0 \pmod{m^d} \\ n \equiv 1 \pmod{m} &\Rightarrow A(n) \equiv 1 \pmod{m^d} \end{aligned}$$

Optimal constructions of these polynomials was provided by Beigel and Tarui [7]. Previous use of these polynomials in circuit complexity have used these polynomials where the degree  $d$  is polylogarithmic. What distinguishes our use here is that we only make use of the polynomials for constant  $d$ .

**Theorem 8 (Toda, Beigel and Tarui)** *For every  $d \geq 1$ , there is a unique degree  $2d - 1$  polynomial that is degree- $d$  modulus amplifying.*

The polynomial constructed by Beigel and Tarui can in fact be explicitly described by the following formula.

$$A_d(n) = (-1)^{d+1}(n-1)^d \left( \sum_{j=0}^{d-1} \binom{d+j-1}{j} n^j \right) + 1 .$$

We will first by standard means obtain the second condition of Proposition 5, at the expense of introducing constant fanin AND gates in the circuit. We will then use modulus amplifying polynomials to enforce the first condition and finally bring the AND gates to the bottom. By this we will obtain the following result.

**Proposition 9** *Any function computed by a constant depth  $\text{CC}^0$  circuit of polynomial size is also computed by a constant depth  $\text{MOD}_{m_1} \circ \dots \circ \text{MOD}_{m_h} \circ \text{AND}_{O(1)}$  circuit of polynomial size satisfying the condition of Proposition 5. Furthermore  $m_h$  is a constant depending on the depth and the moduli  $m$  of the  $\text{CC}^0$  circuit.*

We will need the following standard simulation of a  $\text{MOD}_{p^e}$  gate by a  $\text{MOD}_p \circ \text{AND}_{O(1)}$  circuit (see e.g [7] for a proof).

**Lemma 10** *Let  $q = p^e$  for a prime  $p$ . Then the  $\text{MOD}_q$  function is computed by a  $\text{MOD}_p \circ \text{AND}_{q-1}$  circuit. Furthermore the circuit satisfies that the sum of the inputs to the  $\text{MOD}_p$  gate is always 0 or 1 modulo  $p$ .*

To bring the AND gates to the bottom of the circuit we will use the following.

**Lemma 11** *Let  $C$  be a  $\text{AND}_s \circ (\text{MOD}_m)_r \circ \text{AND}_t$  circuit, satisfying that the sum of the inputs to every  $\text{MOD}_m$  gate is always 0 or 1 modulo  $m$ . Then there is a  $(\text{MOD}_m)_{r^s} \circ \text{AND}_{st}$  circuit computing the same function as  $C$  satisfying that the sum of the inputs to the  $\text{MOD}_m$  gate is always 0 or 1 modulo  $m$ .*

**Proof** Let  $P_1, \dots, P_s$  be polynomials of degree  $t$  with  $r$  terms corresponding to each of the subcircuits with a  $\text{MOD}_m$  gate at the output. Define  $P(x) = \prod_{i=1}^s P_i(x)$ . Since each  $P_i$  is always 0 or 1 modulo  $m$ , we have  $P(x) \equiv C(x) \pmod{m}$ . Note also that  $P$  has degree  $st$  and  $r^s$  terms. Thus the desired circuit is given by  $P$ .  $\square$

**Proof (Proposition 9)** Let  $C$  be a  $\text{CC}^0$  circuit of depth  $h$  consisting of  $\text{MOD}_m$  gates and let  $m = p_1^{e_1} \dots p_k^{e_k}$  be the prime factorization of  $m$ . Since  $a \equiv 0 \pmod{m}$  if and only if  $a \equiv 0 \pmod{p_i^{e_i}}$  for all  $i$ , we can compute  $\text{MOD}_m$  by an OR of  $\text{MOD}_{p_i^{e_i}}$  gates. This OR gate can be replaced by a  $\text{MOD}_p$  gate for  $p > k$ . By the above lemma we can convert all  $\text{MOD}_{p_i^{e_i}}$  gates into  $\text{MOD}_{p_i}$  gates at the expense of introducing  $\text{AND}_m$  gates.

By introducing dummy nodes we can thus get an equivalent polynomial size circuit on the form

$$\text{MOD}_{p'_1} \circ \text{AND}_{m_1} \circ \dots \circ \text{MOD}_{p'_{h'}} \circ \text{AND}_m$$

having the property that the sum of the inputs to every  $\text{MOD}_{m_i}$  gate is always 0 or 1 modulo  $m_i$ .

Define the numbers  $m_1, \dots, m_{h'}$  and  $d_1, \dots, d_{h'}$  inductively as follows. Let  $m_1 = p'_1$  and  $d_1 = 1$ . Now assuming  $m_{i-1}$  and  $d_{i-1}$  has been defined, pick  $d_i$  minimal such that  $p_i^{d_i} > m_{i-1}$  and define  $m_i = p_i^{d_i}$ . Note that  $m_{h'}$  is a constant only depending on  $h$  and  $m$ .

Next we use the modulus amplifying polynomials for all  $\text{MOD}$  gates, converting a  $\text{MOD}_{p'_i}$  gate into a  $\text{MOD}_{m_i}$  gate. Consider a  $\text{MOD}_{p'_i} \circ \text{AND}_m$  subcircuit  $C'$  and let  $P$  be the corresponding polynomial of degree  $m$  over  $\mathbf{Z}_{p'_i}$ , in the inputs  $y_i$  of the subcircuit. Let  $Q(y) = A_{d_i}(P(y))$ . Then we have

$$\begin{aligned} C'(y) = 0 &\Rightarrow Q(y) \equiv 0 \pmod{m_i} \\ C'(y) = 1 &\Rightarrow Q(y) \equiv 1 \pmod{m_i} \end{aligned}$$

We now interpret  $Q$  as a  $\text{MOD}_{m_i} \circ \text{AND}_{m d_i}$  circuit and substitute it for  $C'$ .

Having done this for every  $\text{MOD}_{p'_i}$  gate we have obtained an equivalent polynomial size circuit of the form

$$\text{MOD}_{m_1} \circ \text{AND}_{m d_1} \circ \dots \circ \text{MOD}_{m_{h'}} \circ \text{AND}_{m d_{h'}}$$

of alternating AND and MOD gates, where  $h' = O(h)$  and  $p'_i < m$  for all  $i$ , having the property that the sum of the inputs to every  $\text{MOD}_{p'_i}$  gate is always 0 or 1 modulo  $p'_i$ .

We now finally proceed from the top to the bottom, moving the AND gates to the bottom of the circuit by use of Lemma 11 thereby obtaining an equivalent polynomial size circuit of the form

$$\text{MOD}_{m_1} \circ \text{MOD}_{m_2} \circ \dots \circ \text{MOD}_{m_{h'}} \circ \text{AND}_s$$

where  $s = \prod_{i=1}^{h'} m d_i$  and  $m'_{h'}$  are constants depending only on  $h'$  and  $m$ , satisfying the condition of Proposition 5 thereby completing the proof.  $\square$

## 5 Transformation of $\text{ACC}^0$ Circuits

In this section we provide a proof of the fact that  $\text{ACC}^0$  circuits of quasipolynomial size can be computed by probabilistic  $\text{CC}^0$  circuits of quasipolynomial size. These can then be converted into deterministic  $\Pi_2 \circ \text{CC}^0$  circuits of quasipolynomial size computing the same function.

**Theorem 12** *Let  $k$  be any integer. Then any function computed by an  $\text{ACC}^0$  circuit on  $n$  inputs of quasipolynomial size is also computed by a probabilistic  $\text{CC}^0$  circuit of quasipolynomial size on  $n$  inputs with error less than  $\frac{1}{n^k}$ .*

We will use the method of approximation of OR (and AND) gates due to Razborov [10] and Smolensky [11] (cf. [3]).

**Theorem 13 (Razborov, Smolensky)** *For any prime  $p$  and any epsilon there is a probabilistic  $\text{MOD}_p \circ \text{AND}_{O(\log(\frac{1}{\epsilon}))}$  circuit that computes the OR function with error less than  $\epsilon$ . Furthermore the circuit satisfies that the sum of its inputs is always 0 or 1 modulo  $p$ .*

This will allow us to reduce the fanin of the AND and OR gates sufficiently (to polylogarithmic fanin) to be able to move them to the bottom of the circuit without increasing the size of the circuit beyond quasipolynomial. To get the even cleaner statement described above we need the following observation implicit in a result by Straubing and Thérien [12].

**Proposition 14 (Straubing and Thérien)** *Let  $p$  be a prime not dividing  $m$ . Then any function computed by a  $\text{MOD}_p \circ \text{AND}_d \circ \text{MOD}_m$  circuit of size  $S$  is also computed by a  $\text{MOD}_p \circ \text{MOD}_m$  circuit of size  $S^d$ .*

We will instantiate this result with  $S$  being quasipolynomial and  $d$  polylogarithmic, thus implying that  $S^d$  is quasipolynomial.

**Proof (Theorem 12)** Let  $C$  be any quasipolynomial size  $\text{ACC}^0$  circuit of depth  $h$  consisting of AND, OR and  $\text{MOD}_m$  gates. As in the proof of Proposition 9 we can assume that all MOD gates are on the form  $\text{MOD}_{p_i}$  for some prime divisor  $p_i$  of  $m$ . Next by Theorem 13 we can replace all AND and OR gates by probabilistic  $\text{MOD}_p \circ \text{AND}_{O(\log^{k'} n)}$  circuit for an arbitrary prime  $p$  and a suitable large  $k'$ . By a union bound the resulting circuit will then compute the correct output with error less than  $\frac{1}{n^k}$ . By possible introduction of dummy gates we can now assume that the circuit is on the form

$$\text{MOD}_{p'_1} \circ \text{AND}_{O(\log^{k'} n)} \circ \text{MOD}_{p'_2} \circ \dots \circ \text{AND}_{O(\log^{k'} n)} \circ \text{MOD}_{p'_{h'}}$$

where  $h' = O(h)$ , and the sum of the inputs to every  $\text{MOD}_{p'_i}$  gate is always 0 or 1 modulo  $p'_i$ . We can then simply proceed from the top to the bottom, moving the AND gates to the bottom of the circuit (by multiplying linear polynomials corresponding to the inputs of an AND gate) obtaining a

$$\text{MOD}_{p'_1} \circ \text{MOD}_{p'_2} \circ \dots \circ \text{MOD}_{p'_{h'}} \circ \text{AND}_{O(\log n)^{k' h'}}$$

circuit of quasipolynomial size. We can then introduce a new layer of  $\text{MOD}_m$  gates at the bottom of the circuit to be absorb the AND gates using Proposition 14, thereby completing the proof.  $\square$

We can convert the probabilistic circuit of Theorem 12 into a deterministic circuit using the technique of Ajtai and Ben-Or [1].

**Theorem 15** *Any function computed by an  $\text{ACC}^0$  circuit on of quasipolynomial size is also computed by a  $\Pi_2 \circ \text{CC}^0$  circuit of quasipolynomial size.*

**Proof** Let  $f$  be a Boolean function computed by an  $\text{ACC}^0$  circuit and let  $C'$  be the probabilistic  $\text{CC}^0$  circuit given by Theorem 12 computing  $f$  with error less than  $\frac{1}{2m^2}$ .

Take  $n^2$  independent copies of  $C'$  and take the OR of these. This decreases the error on the positive side to less than  $\frac{1}{(2n^2)^{n^2}}$  and keeps the error on the negative side less than  $\frac{n^2}{2n^2} = \frac{1}{2}$ . Now take  $n$  independent copies of this circuit and take the AND of these. This decreases the error on the negative side to less than  $\frac{1}{2^n}$  and keeps the error on the positive side less than  $\frac{n}{(2n^2)^{n^2}} < \frac{1}{2^n}$ . Thus there is a fixed setting of the random bits that always computes the correct result.  $\square$

Combining this theorem with Proposition 5 concludes the proof of Theorem 1.

## 6 Concluding Remarks

We have shown that constant width cylindrical nondeterministic branching programs are (presumably) significantly more powerful than previously suggested [9]. Here we can only say presumably, since we still have no separation between the class of circuits  $\Pi_2 \circ \text{MOD} \circ \text{AC}^0$  and  $\text{ACC}^0$ .

We have obtained a characterization of  $\text{ACC}^0$  in the quasipolynomial size setting by cylindrical nondeterministic branching programs and circuits. Naturally, the main open question is now whether a characterization can be obtained in the polynomial size setting.

In [9], the question of the relationship between the class of constant width cylindrical nondeterministic branching programs and constant width cylindrical circuits was raised. We now know that they have equivalent power in the quasipolynomial size setting.

Another question is whether the class of functions computed by constant width cylindrical nondeterministic branching programs is closed under complementation. We now know that the answer is yes in the quasipolynomial size setting.

## References

- [1] M. Ajtai and M. Ben-Or. A theorem on probabilistic constant depth computations. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 471–474. ACM, 1984.
- [2] E. Allender, S. Datta, and S. Roy. Topology inside  $\text{NC}^1$ . In *20th Annual IEEE Conference on Computational Complexity*, pages 298–307. IEEE Computer Society Press, 2005.
- [3] E. Allender and U. Hertrampf. Depth reduction for circuits of unbounded fan-in. *Information and Computation*, 112(2):217–238, 1994.
- [4] D. A. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in  $\text{NC}^1$ . *J. Comput. System Sci.*, 38(1):150–164, 1989.
- [5] D. A. M. Barrington, C.-J. Lu, P. B. Miltersen, and S. Skyum. Searching constant width mazes captures the  $\text{AC}^0$  hierarchy. In *Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science*, volume 1373 of *Lecture Notes in Computer Science*, pages 73–83. Springer, 1998.
- [6] D. A. M. Barrington, C.-J. Lu, P. B. Miltersen, and S. Skyum. On monotone planar circuits. In *14th Annual IEEE Conference on Computational Complexity*, pages 24–31. IEEE Computer Society Press, 1999.
- [7] R. Beigel and J. Tarui. On  $\text{ACC}$ . *Computational Complexity*, 4(4):350–366, 1994.
- [8] K. A. Hansen. Constant width planar computation characterizes  $\text{ACC}^0$ . *Theory of Computing Systems*, 39(1):79–92, 2006.
- [9] K. A. Hansen, P. B. Miltersen, and V. Vinay. Circuits on cylinders. *Computational Complexity*, 15(1):62–81, 2006.
- [10] 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*, 41(4):333–338, 1987.
- [11] R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings of the 19th ACM Symposium on Theory of Computing*, pages 77–82, 1987.
- [12] H. Straubing and D. Thérien. A note on  $\text{MOD}_p$  -  $\text{MOD}_m$  circuits. *Theory of Computing Systems*, 39(5):699–706, 2006.
- [13] G. Tardos and D. A. M. Barrington. A lower bound on the mod 6 degree of the OR function. *Computational Complexity*, 7(2):99–108, 1998.
- [14] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991.
- [15] V. Vinay. Hierarchies of circuit classes that are closed under complement. In *11th Annual IEEE Conference on Computational Complexity*, pages 108–117. IEEE Computer Society Press, 1996.