1992

The power of the middle bit

Frederic Green
Clark University, fgreen@clarku.edu

Johannes Kobler
Clark University

Jacobo Toran
Clark University

The power of the middle bit

Frederic Green
Johannes Kobler
Jacobo Torán

Report LSI–91–50

The Power of the Middle Bit

Frederic Green[*] Clark University

Johannes Köbler[†‡] Universität Ulm

Jacobo Torán [§‡] U. Politecnica de Catalunya

Extended Abstract

Abstract

We study the class of languages that can be recognized in polynomial time with the additional information of one bit from a #P function. In particular we show that every MOD_k class and every class contained in PH are low for this class. We translate these results to the area of circuit complexity using MidBit (middle bit) gates. A MidBit gate over w inputs $x_1, \ldots, x_w$ is a gate which outputs the value of the $\lfloor \log(w)/2 \rfloor^{\text{th}}$ bit in the binary representation of the number $\sum_{i=1}^{w} x_i$. We show that every language in ACC can be computed by a family of depth-2 deterministic circuits of size $2^{(\log n)^c}$ with a MidBit gate at the root and AND-gates of fan-in $(\log n)^c$ at the leaves. This result improves the known upper bounds for the class ACC.

1 Introduction

The complexity classes PP (probabilistic polynomial time [Gi 77]) and $\oplus$P (parity P, [PaZa 83, GoPa 86]) have received much attention since the well known result by Toda [Tod 89] proving that the polynomial time hierarchy (PH) is Turing reducible to PP. These classes are closely related to the class of counting functions #P [Va 79] that count the number of accepting paths on nondeterministic Turing machines. Observe that sets in PP and $\oplus$P can be respectively decided with the information of the leftmost and rightmost bit of a #P function. Toda's proof combines two

[*] Clark University, Dept. of Math/CS, Worcester, Massachusetts 01610. Research partially supported by a grant from the Dirección General de Investigación Científica y Técnica (DGICYT), Spanish Ministry of Education, while the author was visiting the U. Politècnica de Catalunya, Barcelona.

[†] Universität Ulm, Theoretische Informatik, Oberer Eselsberg, D-7900 Ulm, Germany.

[‡] This research was supported by the DAAD (Acciones Integradas 1991, 313-AI-e-es/zk).

[§] U. Politecnica de Catalunya, Departamento L.S.I., Pau Gargallo 5, E-08028 Barcelona, Spain. Research partially supported by ESPRIT-II Basic Research Actions Program of the EC under Contract No. 3075 (project ALCOM).

important results; on one side he shows that PH is randomly reducible to $\oplus$P, and in a second part he proves that $\mathrm{PP}^{\oplus\mathrm{P}}$ is included in $\mathrm{P}^{\#\mathrm{P}}$. A careful observation of the proof of the last result shows that for this inclusion the whole power of $\mathrm{P}^{\#\mathrm{P}}$ is not needed. To decide an input x, a function $f \in \#\mathrm{P}$ has to be queried just once, and more interestingly, just one bit of information of f is needed, as in the case of PP or $\oplus$P. It is natural to ask what other problems can be computed by looking at just one bit of a #P function. This question has been independently considered by Schwentick [Scw 91] and Regan [Re 91] who define the class MidBitP (middle bit P) of languages that can be computed with the information of just one bit from a #P function.[1]

Definition 1.1 [Re 91, Scw 91] *A language L is in* MidBitP *if there exists a function f in* #P *and a function g in* FP *such that for all x, x is in L iff there is a 1 at position $g(x)$ in the binary representation of $f(x)$.*

At first sight it looks like the definition depends heavily on the base in which the representation of the #P function f is taken. It is clear that $\oplus$P is in MidBitP since the parity of a #P function in base 2 can be obtained by looking at just one bit, but how about other modular classes? For example, in order to decide whether a number written in base 2 is congruent to 0 modulo 3, one needs the information of each one of its bits. By constructing suitable #P functions we prove however that for each k the class $\mathrm{MOD}_k\mathrm{P}$ [BeGiHe 90] is included in MidBitP. Moreover we show that for every k, $\mathrm{MOD}_k\mathrm{P}$ is low for MidBitP. Intuitively this means that a language in $\mathrm{MOD}_k\mathrm{P}$ does not give any additional help when used as an oracle in a MidBitP computation and therefore this is stronger than a containment result. We also show that the classes BPP and PH are low for MidBitP.

In section 3 we give an application of the previous results improving the known upper bound for the circuit class ACC. This class was defined by Barrington [Ba 89] as the class of languages accepted by bounded depth polynomial-size circuits with AND, OR, NOT and a finite set of MOD_k gates. Clearly ACC contains AC_0 and is contained in TC_0. Since the PARITY function cannot be computed in AC_0 the first inclusion is proper; Barrington [Ba 89] conjectured that the second inclusion is also proper i.e., $\mathrm{TC}_0 \not\subset$ACC, but no proof of this fact has been obtained.

Using Toda's result [Tod 89] and building on some work on AC_0 by Allender and Hertrampf [Al 89], [AlHe 90], Yao [Yao 90] proved the first non-trivial upper bound for ACC. He showed that every language in ACC is recognized by a family of depth-2 probabilistic circuits of size $2^{(\log n)^c}$ with a symmetric gate at the root and AND-gates of fan-in $(\log n)^c$ at the leaves. Recently Beigel and Tarui [BeTa 91] have improved this result showing that the circuits given by Yao can be made deterministic without increasing their size. However in both cases the symmetric gate at the root depends on the type of the modular gates used in the ACC circuit. It is therefore very hard

[1] Independently, Regan [Re 91] and Schwentick [Scw 91] have also observed that Toda's proof implies that $\mathrm{PP}^{\oplus\mathrm{P}}$ and $\mathrm{PP}^{\mathrm{PH}}$ are contained in MidBitP.

to prove that a certain function cannot be computed by depth-2 circuits of the type given in [Yao 90] or [BeTa 91] since all that can be said about the gates in the root is that they belong to an infinite subfamily of the symmetric functions. We improve the above upper bounds showing that the mentioned circuits can be restricted to have a symmetric gate of type MidBit at the root. A MidBit gate over w inputs $x_1, \ldots, x_w$ is a gate which outputs the value of the $\lfloor \log(w)/2 \rfloor^{\text{th}}$ bit in the binary representation of the number $\sum_{i=1}^{w} x_i$. We prove that ACC can be computed by a family of depth-2 deterministic circuits of size $2^{(\log n)^c}$ with a MidBit gate at the root and AND-gates of fan-in $(\log n)^c$ at the leaves. We believe that there are TC_0 languages which cannot be computed by circuits of this kind, and that the study of these circuits can therefore provide a way to show that TC_0 is not contained in ACC.

2 Lowness of Mod Classes for the Class MidBitP

The concept of lowness in the context of computational complexity theory was first introduced by Schöning [Sch 83] and was first studied in counting classes by Torán [Tor 88]. A class $\mathcal{A}$ is *low* for a relativizable complexity class $\mathcal{C}$ if the sets in $\mathcal{A}$, when used as an oracle for $\mathcal{C}$, do not help, i.e., $\mathcal{C}^{\mathcal{A}} = \mathcal{C}$. In this section we prove that for any k, Mod_kP is low for MidBitP.

Toda has shown that for every function f in $\#\text{P}^{\oplus\text{P}}$ and every polynomial p there is a function $g \in \#\text{P}$ such that $f(x)$ and $g(x)$ agree in the last $p(|x|)$ bits.

Theorem 2.1 [Tod 89] *For all functions f in $\#\text{P}^{\oplus\text{P}}$ and for every polynomial t there exists a function h in $\#\text{P}$ such that*

$$h(x) \equiv f(x) \pmod{2^{t(|x|)}}.$$

By this result, $\oplus$P is low for $\oplus$P (which is proved by Papadimitriou and Zachos [PaZa 83] using a different technique) and MidBitP.

Corollary 2.2 $\oplus$P *is low for* MidBitP.

It will follow from the next theorem that for every function $f \in \#\text{P}^{\text{BPP}}$ and every polynomial p there is a function $g \in \#\text{P}$ such that $f(x)$ and $g(x)$ agree in the first $p(|x|)$ bits where the first bit of a binary number is the most significant bit which is 1.

Theorem 2.3 *For every function $f \in \#\text{P}^{\text{BPP}}$ there exist a polynomial t and a function $g \in \#\text{P}$ such that*

$$f(x) = \lfloor g(x)/2^{t(|x|)} \rfloor.$$

Proof. Let f be in $\#\text{P}^{\text{BPP}}$. Since BPP is closed under Turing reductions, there exists a language L in BPP and a polynomial q such that

$$f(x) = \sum_{y \in \Sigma^{q(|x|)}} \chi_L(\langle x, y \rangle).$$

Furthermore, by the probability amplification lemma for BPP, there exists a function $h \in \#\mathrm{P}$ and a polynomial t such that

$$\begin{aligned}\langle x,y\rangle \in L &\Rightarrow h(\langle x,y\rangle) \geq 2^{t(|x|)} - 2^{t(|x|)-q(|x|)-2},\\ \langle x,y\rangle \notin L &\Rightarrow h(\langle x,y\rangle) \leq 2^{t(|x|)-q(|x|)-2},\end{aligned}$$

and therefore h fulfills the following inequalities:

$$\chi_L(\langle x,y\rangle)2^{t(|x|)} - 2^{t(|x|)-q(|x|)-2} \leq h(\langle x,y\rangle) \leq \chi_L(\langle x,y\rangle)2^{t(|x|)} + 2^{t(|x|)-q(|x|)-2}.$$

Since $\#\mathrm{P}$ is closed under addition, $h'(\langle x,y\rangle) = h(\langle x,y\rangle) + 2^{t(|x|)-q(|x|)-2}$ is also a $\#\mathrm{P}$ function fulfilling the inequalities

$$\chi_L(\langle x,y\rangle)2^{t(|x|)} \leq h'(\langle x,y\rangle) \leq \chi_L(\langle x,y\rangle)2^{t(|x|)} + 2^{t(|x|)-q(|x|)-1}.$$

Now we can define the $\#\mathrm{P}$ function

$$g(x) = \sum_{y\in\Sigma^{q(|x|)}} h'(\langle x,y\rangle),$$

and since

$$2^{q(n)}2^{t(|x|)-q(|x|)-1} < 2^{t(n)},$$

it follows that

$$\lfloor g(x)/2^{t(|x|)}\rfloor = f(x).$$

□

By this result, BPP is low for BPP [Za 82], PP, [KöScToTo 89], and MidBitP.

Corollary 2.4 BPP *is low for* MidBitP.

Since all the proofs in this section relativize we immediately obtain the lowness of $\mathrm{BPP}^{\oplus\mathrm{P}}$ and since PH is contained in $\mathrm{BPP}^{\oplus\mathrm{P}}$ [Tod 89], also of PH for MidBitP.

Corollary 2.5 $\mathrm{BPP}^{\oplus\mathrm{P}}$ *and therefore* PH *are low for* MidBitP.

Using relativized versions of Theorem 2.3 and Theorem 2.1, we can easily prove the following theorem which states that for every function f in $\#\mathrm{P}^{\mathrm{BPP}^{\oplus\mathrm{P}}}$ a $\#\mathrm{P}$ function h can be constructed such that the binary representation of f is a substring of the binary representation of h. Note that Theorem 2.6 additionally allows us to "isolate" the binary representation of f inside h, i.e. for each polynomial p we can construct h such that there are at least $p(|x|)$ $0's$ in the binary representation of $h(x)$ to the left and to the right of the binary representation of $f(x)$. A similar result was used in [TodWa 91] to show that every function in the counting version of PH is metric reducible to a $\#\mathrm{P}$ function.

Theorem 2.6 *For all functions f in $\#\mathrm{P}^{\mathrm{BPP}^{\oplus\mathrm{P}}}$ there exists a polynomial q such that for every polynomial t there exists a function h in $\#\mathrm{P}$ such that*

$$\lfloor h(x)/2^{q(|x|)} \rfloor \equiv f(x) \pmod{2^{t(|x|)}}.$$

Proof of Theorem 2.6. Let f be in $\#\mathrm{P}^{\mathrm{BPP}^{\oplus\mathrm{P}}}$. Since Theorem 2.3 relativizes, there exist a polynomial q and a function $g \in \#\mathrm{P}^{\oplus\mathrm{P}}$ such that

$$f(x) = \lfloor g(x)/2^{q(|x|)} \rfloor.$$

By Theorem 2.1, there exists a function $h \in \#\mathrm{P}$ such that

$$h(x) \equiv g(x) \pmod{2^{q(|x|)+t(|x|)}},$$

and therefore

$$\lfloor h(x)/2^{q(|x|)} \rfloor \equiv f(x) \pmod{2^{t(|x|)}}.$$

□

Next we prove the main result of this section, namely that every $\#\mathrm{P}^{\mathrm{Mod}_k\mathrm{P}}$ function can be isolated inside some $\#\mathrm{P}$ function.

Theorem 2.7 *Let k be prime. For all functions b in $\#\mathrm{P}^{\mathrm{Mod}_k\mathrm{P}}$ and for every polynomial t there exist a polynomial q and a function h in $\#\mathrm{P}$ such that*

$$\lfloor h(x)/2^{q(|x|)} \rfloor \equiv b(x) \pmod{2^{t(|x|)}}.$$

Because the proof of Theorem 2.7 relativizes, we can state the following corollaries.

Corollary 2.8 *For any k, $\mathrm{Mod}_k\mathrm{P}$ is low for $\mathrm{MidBitP}$.*

Proof of Corollary 2.8. By the representation theorem of Hertrampf [He 90], it follows that if $k = p^{e_1}q$ for a prime number p and $\gcd(p,q) = 1$, then

$$\mathrm{Mod}_k\mathrm{P} \subseteq \mathrm{Mod}_p\mathrm{P}^{\mathrm{Mod}_q\mathrm{P}}.$$

The result follows iterating this argument for all the prime factors of k and using the relativized version of the previous theorem. □

The result stated in theorem 2.7 works also for every complexity class in ModPH, a generalization of the polynomial time hierarchy that includes also ModP classes. ModPH can be considered as the polynomial time analogue to the circuit class ACC.

Definition 2.9 *ModPH is the smallest family of languages containing the class P and satisfying that for any class K in ModPH the classes NP^K, co-NP^K and Mod_nP^K (for any positive integer n) also belong to ModPH.*

Corollary 2.10 *For all functions f in $\#\mathrm{P}^{\mathrm{ModPH}}$ and for every polynomial t there exist a polynomial q and a function h in $\#\mathrm{P}$ such that*

$$\lfloor h(x)/2^{q(|x|)} \rfloor \equiv f(x) \pmod{2^{t(|x|)}}.$$

Proof of Theorem 2.7. Let b be in $\#\mathrm{P}^{\mathrm{Mod}_k\mathrm{P}}$. Since k is prime, $\mathrm{Mod}_k\mathrm{P}$ is closed under Turing reductions [BeGiHe 90]. Thus there exists a language L in $\mathrm{Mod}_k\mathrm{P}$ and a polynomial r such that

$$b(x) = \sum_{y \in \Sigma^{r(|x|)}} \chi_L(\langle x, y \rangle).$$

Let p be a polynomial such that $k^{p(n)} > 2^{r(n)+t(n)+2}$. Adapting results from Toda [Tod 89] and Beigel, Gill and Hertrampf [BeGiHe 90] we can assume that there is a function c in $\#\mathrm{P}$ s.t.

$$c(\langle x, y \rangle) \equiv \chi_L(\langle x, y \rangle) \pmod{k^{p(|x|)}}.$$

Consider the function

$$f(x) := \sum_{y \in \Sigma^{r(|x|)}} c(\langle x, y \rangle).$$

Then

$$f(x) = a(x) k^{p(|x|)} + b(x)$$

where $b(x) \leq 2^{r(|x|)} < \frac{k^{p(|x|)}}{2^{t(|x|)+2}}$. The proof of Theorem 2.7 is now completed by the following lemma. □

Lemma 2.11 *If $f \in \#\mathrm{P}$ is of the form $f(x) = a(x)k^{p(|x|)} + b(x)$, where*

$$b(x) < \frac{k^{p(|x|)}}{2^{t(|x|)+2}},$$

then there exist a function h in $\#\mathrm{P}$ and a polynomial q such that

$$h(x) = a'(x) 2^{q(|x|)+t(|x|)} + b(x) 2^{q(|x|)} + c(x),$$

where $c(x) < 2^{q(|x|)}$.

Proof of Lemma 2.11. Since f is in $\#\mathrm{P}$ there exists a polynomial s such that $f(x) < 2^{s(|x|)}$ for all x. We first prove the following claim.

Claim. There exist a polynomial q and a function g in $\#\mathrm{P}$ such that

$$g(x) = a(x) 2^{q(|x|)} + b'(x) \text{ and } b'(x) < 2^{q(|x|)-t(|x|)-1}.$$

Proof of Claim. Define

$$g(x) = f(x)\left\lceil \frac{2^{q(|x|)}}{k^{p(|x|)}} \right\rceil .$$

Then it follows that

$$\begin{aligned} a(x)2^{q(|x|)} \leq g(x) &= (a(x)k^{p(|x|)} + b(x))\left\lceil \frac{2^{q(|x|)}}{k^{p(|x|)}} \right\rceil \\ &< a(x)2^{q(|x|)} + b(x)\frac{2^{q(|x|)}}{k^{p(|x|)}} + a(x)k^{p(|x|)} + b(x). \\ &< a(x)2^{q(|x|)} + 2^{q(|x|)-t(|x|)-2} + a(x)k^{p(|x|)} + b(x). \\ &< a(x)2^{q(|x|)} + 2^{q(|x|)-t(|x|)-1}. \end{aligned}$$

The last inequality can be achieved by choosing $q > t + s + 2$. □

To complete the proof of Lemma 2.11 we define

$$h(x) = f(x)2^{q(|x|)} + g(x)i(|x|),$$

where

$$i(n) \equiv -k^{p(n)} \pmod{2^{t(n)}} \text{ and } i(n) < 2^{t(n)}.$$

Then it follows that

$$\begin{aligned} h(x) &= a(x)k^{p(|x|)}2^{q(|x|)} + b(x)2^{q(|x|)} + a(x)2^{q(|x|)}i(|x|) + b'(x)i(|x|) \\ &= 2^{q(|x|)}a(x)(k^{p(|x|)} + i(|x|)) + b(x)2^{q(|x|)} + b'(x)i(|x|), \end{aligned}$$

where

$$k^{p(n)} + i(n) \equiv 0 \pmod{2^{t(n)}}$$

and

$$b'(x)i(|x|) < 2^{q(|x|)-1}.$$

□

From corollary 2.10 we know that ModPH is the largest class of languages known to be low for MidBitP. It is an open question whether the class PP is also low for MidBitP. A positive answer to this problem would imply that the polynomal time counting hierarchy, [Wa 86], collapses to the class MidBitP.

3 A New Upper Bound for ACC

The methods of the preceding section relativize. It is thus not surprising that there are analogous circuit results. In this section we prove them directly.

Yao [Yao 90] showed that ACC circuits can be simulated by probabilistic depth-2 circuits consisting of some symmetric gate over subexponentially many AND gates of polylogarithmic fan-in. Beigel and Tarui [BeTa 91] subsequently improved his construction and showed that depth-2 *deterministic* circuits of a symmetric gate over small AND's also can simulate ACC. Note that in both cases it was proved that *some* symmetric function over small AND's is sufficient. This leaves open the possibility that the symmetric function needed will be different for different input sizes. However our main result in this section is that there is *one particular* symmetric function which, together with AND gates of small fan-in, can capture all of ACC: namely, the symmetric function which outputs the middle bit of the sum of the inputs.

Definition 3.1 *A MidBit gate over w inputs $x_1, ..., x_w$ is a gate which outputs the value of the $\lfloor log(w)/2 \rfloor^{th}$ bit in the binary representation of the number $\sum_{i=1}^{w} x_i$. A MidBit-of-AND circuit of order r is a circuit consisting of a MidBit gate over AND-gates where each AND-gate has fan-in at most r. A family of functions $\{f_n\}$ is computable by a family of MidBit$^+$ circuits if there is a polynomial p such that for each n there is a MidBit-of-AND circuit of order $p(log(n))$ and size $2^{p(log(n))}$ which computes $f_n(x_1, ..., x_n)$.*

A *Mod$_k$* gate over w inputs $x_1, ..., x_w$ is defined to output 1 if $\sum_{i=1}^{w} x_i \neq 0 \pmod{k}$ and 0 otherwise. We similarly define Mod$_k$-of-AND circuits and families of Mod$_k^+$ circuits. Note that we will always speak of families of MidBit$^+$ or Mod$^+$ circuits. Even when we refer to a MidBit$^+$ or Mod$^+$ circuit individually, it should be understood that what is meant is a member of a particular family of such circuits.

The following theorems give the circuit analogue of the lowness result MidBitP$^{\text{Mod}_k\text{P}}$ = MidBitP.

Theorem 3.2 *Let k be prime and let $\{C_n\}$ be a family of circuits such that there exists a polynomial r where for each n, C_n consists of a MidBit gate over at most $2^{r(log(n))}$ Mod$_k$-gates. Then $\{C_n\}$ is computable by a family of MidBit$^+$-circuits.*

Proof. Similar to the proof of Theorem 2.7. Let t be any polynomial. We have that C_n outputs 1 if and only if the $\lfloor log(w)/2 \rfloor^{th}$ bit in the binary representation of $b(x)$ equals 1, where

$$b(x) = \sum_{i=1}^{w} \text{Mod}_k(x_{i_1}, ..., x_{i_{s_i}}),$$

$s_i \leq n$ (i.e., the Mod gates can have any number of the n inputs) and $w \leq 2^{r(log(n))}$. Using techniques of Toda [Tod 89] and Beigel, Gill and Hertrampf [BeGiHe 90], there is a polynomial Q_d of degree d which has the property that if $X \neq 0 \pmod{k}$ then $Q_d(X) = 1 \pmod{k^d}$, and if $X = 0 \pmod{k}$ then $Q_d(X) = 0 \pmod{k^d}$. Thus

$$b(x) = \sum_{i=1}^{w} [Q_d(\sum_{l=1}^{s_i} x_{j_l}) \bmod k^d].$$

Suppose we choose $d = p(log(n))$ where p is a polynomial such that $k^p > \max\{2^{r+t+2}, 2^{t+2}\}$. (To simplify notation, in the remainder of the proof, where it is not confusing we use p, q, r and t for $p(log(n)), q(log(n)), r(log(n))$ and $t(log(n))$, respectively.) Then $b(x) \leq 2^r < k^p$. Now the outer sum in the equation above for b is less than k^p, so the "mod" can be moved outside:

$$b(x) = [\sum_{i=1}^{w} Q_p(\sum_{l=1}^{s_i} x_{j_l})] \pmod{k^p}.$$

We write

$$f(x) = \sum_{i=1}^{w} Q_p(\sum_{l=1}^{s_i} x_{j_l}).$$

Then

$$f(x) = ak^p + b.$$

Note that f is a polynomial in the variables $x_1, ..., x_n$ of degree $p(log(n))$. If we multiply f by $\lceil 2^q/k^p \rceil$ we still have such a polynomial. Following the proof of Lemma 2.11, we find that $\lceil 2^q/k^p \rceil f(x) = a2^q + b'$ where $b' < 2^{q-t-1}$. Setting $i = -k^p \pmod{2^t}$ as in the proof of Lemma 2.11,

$$2^q f(x) + i \lceil 2^q/k^p \rceil f(x) = 2^q b + ib' \pmod{2^{q+t}} \quad (*).$$

Now if we reduce the left hand side of the above equation mod 2^{q+t}, we obtain a polynomial of degree $p(log(n))$ with all positive coefficients (none larger than 2^{q+t-1}). Replace any term with coefficient > 1 by a sum of identical terms with unit coefficients, and substitute multiplication with logical AND. The result is a sum σ of $\leq 2^{polylog}$ AND gates each of polylog fan-in. Reducing the right hand side of eq. (*), we obtain $2^q(b \bmod t) + ib'$, where $ib' \leq 2^{q-1}$. Thus the output bit of C_n is the bit in position $q + \lfloor log(w)/2 \rfloor$ in the binary expansion of σ. We can add constant inputs or multiply the sum of AND's so that this is precisely the middle bit. □

It is not hard to see that if the inputs to the MOD gates in the previous theorem are AND gates of polylog fan-in, that the resulting depth-3 circuits can still be simulated by MidBit$^+$ circuits. The reason is that we are composing the polylog degree polynomial Q_p with another polylog degree polynomial.

Corollary 3.3 *Let k be prime and let $\{C_n\}$ be a family of depth-3 circuits such that there exists a polynomial s where for each n, C_n consists of a MidBit gate over at most $2^{s(log(n))}$ Mod_k^+ circuits. Then $\{C_n\}$ is computable by a family of MidBit$^+$-circuits.*

We now turn our attention to MidBit gates at the root and *pure ACC* subcircuits [Yao 90] (families of constant-depth polynomial size circuits which consist only of Mod_m gates for some natural number m).

Theorem 3.4 *Let $\{C_n\}$ be a family of depth-d circuits consisting of a MidBit gate at the root and Mod_m gates at remaining levels. Then $\{C_n\}$ is computable by a family of MidBit$^+$-circuits.*

Proof. Beigel and Tarui [BeTa 91] have shown that a Mod_m gate can be simulated by a "stratified" circuit of $\text{Mod}_{k_1}, \text{Mod}_{k_2}, ..., \text{Mod}_{k_l}$ gates where $k_1, k_2, ..., k_l$ are the prime divisors of m, on levels $1, 2, ..., l$, respectively, and polylog fan-in AND gates on the lowest level. Using the previous theorem and an inductive argument as in the proof of Theorem 6 in [BeTa 91], each layer of Mod_{k_i} gates can be "absorbed" in the MidBit gate, and the resulting polylog AND gates "pushed" down to the leaves. The resulting circuit is a MidBit$^+$ circuit. □

The following main theorem uses a combination of the above results, techniques of Valiant and Vazirani [ValVaz 86], Toda [Tod 89], Allender [Al 89], and Allender and Hertrampf [AlHe 90], and the technique by which we showed that BPP is low for MidBitP. It says that circuits consisting of a MidBit gate over ACC subcircuits can be simulated by MidBit$^+$ circuits. The proof is similiar to those given in Corollaries 8 and 9 of [BeTa 91].

Theorem 3.5 *Let $\{C_n\}$ be a family of depth-d circuits of size $2^{polylog(n)}$ consisting of a MidBit gate at the root and Mod_m, AND, and OR gates at remaining levels. Then $\{C_n\}$ is computable by a family of MidBit$^+$-circuits.*

Proof. Let $C_n = 1$ iff the $\lfloor log(s)/2 \rfloor^{\text{th}}$ bit of S is 1, where $S = \sum_{i=1}^{s} c_i$, with each subcircuit c_i consisting of AND, OR, and Mod_m gates, and without loss of generality, $s = 2^{q(log(n))}$ where q is a polynomial. The AND and OR gates in each c_i can be replaced by probabilistic Mod_m^+ circuits with polylogarithmically many random bits, using the techniques of [ValVaz 86], [Al 89], and [AlHe 90]. By pushing the AND-gates to the leaves, as in the preceding theorem, c_i can be simulated by a probabilistic circuit c'_i comprised of Mod_m gates and AND gates of polylog fan-in at the lowest level, so that and $\Pr(c'_i \neq c_i) \leq 2^{-q(log(n))-2}$. It is possible to simulate c_i with such a c'_i using $t(log(n))$ bits where t is a polynomial such that $t > q + 2$. Let c''_i denote the sum of c'_i over all possible settings of the random bits of c'_i, and let $S' := \sum_{i=1}^{s}(c''_i + 2^{t(log(n))-q(log(n))-2})$. One can show that $S' = 2^{t(log(n))}S + r$ where $r < 2^{t(log(n))}$. The output of the desired MidBit$^+$ circuit is the bit in position $\lfloor log(s)/2 \rfloor + t(log(n))$ of S'. □

References

[Al 89] E. Allender, A note on the power of threshold circuits. In *Proceedings of the 30th Symposium on Foundations of Computer Science* 1989, 580-584.

[AlHe 90] E. Allender, U. Hertrampf, On the power of uniform families of constant depth threshold circuits. In *Proceedings 15th Symposium on Mathematical Foundations Computer Science, Lecture Notes in Computer Science 452* (1990), 158-164.

[Ba 89] D. Barrington, Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. In *J. Comput. Syst. Sci., 38* (1989), 150-164.

[BeGiHe 90] R. Beigel, J. Gill, U. Hertrampf, Counting classes: Thresholds, parity, mods, and fewness. In *Proceedings 7th Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science 415* (1990), 49-57.

[BeTa 91] R. Beigel, J. Tarui, On ACC. In *Proceedings of the 32nd Symposium on Foundations of Computer Science* 1991.

[Gi 77] J. Gill, Computational complexity of probabilistic Turing machines. In *SIAM Journal on Computing 6* (1977), 675-695.

[GoPa 86] L. Goldschlager, I. Parberry, On the construction of parallel computers from various bases of boolean functions. In *Theoretical Computer Science 21* (1986), 43-58.

[He 90] U. Hertrampf, Relations among MOD-classes. In *Theoretical Computer Science 74* (1990), 325-328.

[KöScToTo 89] J. Köbler, U. Schöning, J. Torán and S. Toda, Turing Machines with few accepting computations and low sets for PP. In *Proceedings of the 4th Structure in Complexity Theory Conference* 1989, 208-216.

[PaZa 83] C. Papadimitriou, S. Zachos, Two remarks on the power of counting. In *6th GI Conference on Theoretical Computer Science, Lecture Notes in Computer Science 145* (1983) 269-276.

[Re 91] K. Regan, Private communication (1991).

[Sch 83] U. Schöning, A low and a high hierarchy within NP, Journal of Computer and System Sciences 27 (1983) 14-28.

[Scw 91] T. Schwentick, The complexity of computing the middle bit of a #P function. Technical report University of Mainz (1991).

[Tod 89] S. Toda, On the computational power of PP and ⊕P. In *Proceedings of the 30th Symposium on Foundations of Computer Science* 1989, 514-519.

[TodWa 91] S. Toda and O. Watanabe, Polynomial time 1-Turing reducibility from #PH to #P. To appear in *Theoretical Computer Science.*

[Tor 88] J. TORÁN, An Oracle Characterization of the Counting Hierarchy, *Proceedings of the 3rd Annual Conference on Structure in Complexity Theory* 1988, 213-223.

[Va 79] L.G. VALIANT, The complexity of computing the permanent. In *Theoretical Computer Science 8* (1979), 189-201.

[ValVaz 86] L. VALIANT AND V. VAZIRANI, NP is as easy as detecting unique solutions. In *Theoretical Computer Science 47* (1986) 85-93.

[Wa 86] K. WAGNER, The complexity of combinatorial problems with succint input representation. In *Acta Informatica 23* (1986) 325-356.

[Yao 90] A. YAO, On ACC and threshold circuits. In *Proceedings of the 31st Symposium on Foundations of Computer Science* 1990, 619-627.

[Za 82] S. ZACHOS, Robustness of probabilistic computational complexity classes under definitional perturbations. In *Information and Control 54* (1982), 143-154.