The Power of the Middle Bit of a #P Function

(Revised Version)

Frederic Green[1]
Clark University

Johannes Köbler[2 3]
Universität Ulm

Keneth W. Regan [4]
SUNY/Buffalo

Thomas Schwentick [5]
University of Mainz

Jacobo Torán [6 3]
U. Politecnica de Catalunya

May 1994

[1]Clark University, Dept. of Math/CS, Worcester, MA 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.

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

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

[4]SUNY/Buffalo, Computer Science Department, Buffalo, NY 14260. Supported by NSF Grant CCR-9011248.

[5]Universität Mainz, Im Zuckergarten 17, D-6500 Mainz 42, Germany.

[6]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).

Abstract

We introduce the class MP of languages L which can be solved in polynomial time with the additional information of one bit from a #P function f. We prove that the polynomial hierarchy and the classes $\mathrm{Mod}_k\mathrm{P}$, $k \geq 2$, are low for this class. We show that the middle bit of $f(x)$ is as powerful as any other bit, and that a wide range of bits around the middle have the same power. By contrast, the $O(\log n)$ many least significant bits are equivalent to $\oplus$P [BeGiHe 90], and by a simple corollary to the result that PP is closed under intersection [BeReSp 91], the $O(\log n)$ many most significant bits are equivalent to PP; hence these bits are probably weaker. We study also the subclass AmpMP of languages whose MP representations can be "amplified," showing that $\mathrm{BPP}^{\oplus\mathrm{P}} \subseteq \mathrm{AmpMP}$, and that important subclasses of AmpMP are low for MP.

We translate some of 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 $\lfloor \log(w)/2 \rfloor^{\mathrm{th}}$ 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 $2^{(\log n)^{O(1)}}$ with a MidBit gate at the root and AND-gates of fan-in $(\log n)^{O(1)}$ 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 polynomial time [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 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.

We consider the complexity class MP of languages that can be decided with the help of any one selected bit. This class is a natural generalization of both PP and $\oplus$P, and seems easier than the much-studied class $\mathrm{P}^{\#\mathrm{P}}$. We suppose that the values of a #P function $f(x)$ are encoded as binary numbers, possibly with leading zeroes, where the length of the encoding is computable in polynomial time and with the least significant bits indexed first (the index of the least significant bit being 0; note also that the most significant bit is zero only if the length of the encoding is greater than $\log f(x)$).

Definition 1.1 *A language L is in* MP *if there exists a function f in* #P *and a function g in* FP *(called a bit selection function) such that for all x, x is in L if and only if there is a 1 at position $g(x)$ in the binary representation of $f(x)$.*

That is, for all x, $x \in L \Leftrightarrow \lfloor f(x)/2^{g(x)} \rfloor \bmod 2 = 1$. The "M" stands for "middle bit", since we show that without loss of generality g can be the function which indexes the middle bit of the binary representation of $f(x)$.

We investigate in Section 3 the basic properties of MP. It is known that the rightmost $O(\log n)$ bits of a #P function still give the power of $\oplus$P [BeGiHe 90]; it is a simple consequence of [BeReSp 91] that the leftmost $O(\log n)$ bits do likewise for PP. MP is closed downward under polynomial time many-one reducibility and has complete problems. The problem of whether MP is closed under intersection leads to a question of independent mathematical interest about the size of integer-valued polynomials which satisfy certain congruence equations. We discuss this question at the end of the section.

In Section 4 we consider subclasses of MP that correspond to special kinds of #P functions, having many zeroes around the deciding bit. We show that these classes are low for MP. Although those #P functions have a very special form, important classes like the polynomial hierarchy (PH) and $\oplus$P in fact have such a representation. The class of all languages that fulfill this "amplification condition" will be called AmpMP. We give closure properties of AmpMP and show that any subclass of AmpMP which is closed under conjunctive and disjunctive reducibilities is low not only for MP but also for AmpMP. Thus many important subclasses of this class, including BPP and PH, are low for AmpMP and for MP. Furthermore, if AmpMP = MP, or even if $\mathrm{C_{=}P} \subseteq$ AmpMP, then the *counting hierarchy* [Wa 86] collapses to MP.

Definition 1.1 makes sense even when $f(x)$ is written in base k, $k \geq 3$, rather than base 2, and it is natural to ask whether the class defined remains the same. On one hand, the classes $\mathrm{Mod}_k\mathrm{P}$ analogous to $\oplus$P for the least significant bit are all believed to be different. On the other hand, the "most significant bit = 1" definition of PP yields the same class in any base, since PP is closed under Boolean operations [BeReSp 91]. We had hoped to show that if MP is closed under intersection then its definition is independent of the base, but are unable to do so in this paper. While it is immediate that $\oplus\mathrm{P} \subseteq \mathrm{MP}$ it is not so obvious whether $\mathrm{Mod}_3\mathrm{P} \subseteq \mathrm{MP}$ since 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 in Section 5 that for each k the class $\mathrm{Mod}_k\mathrm{P}$ [BeGiHe 90] is in fact included in MP. Furthermore, we show that for every k, $\mathrm{Mod}_k\mathrm{P}$ is in AmpMP and since $\mathrm{Mod}_k\mathrm{P}$ is closed under polynomial-time Turing reductions, it too is low for MP. We describe the proof techniques for this result in greater detail below, in the context of circuits.

In Section 6 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., $\mathrm{TC}_0 \not\subseteq$ACC, but no proof of this fact has been obtained.

The intimate connection between results about Turing machine classes and results about circuit classes is by now well-known. Exponential lower bounds for circuit classes imply relativized separations of complexity classes. The contrapositive of this implication has useful consequences for circuit complexity: that is, containment of one complexity class in another relative to all oracles implies a quasi-polynomial size circuit simulation result. The other direction also works in many cases, that is a quasi-polynomial size circuit simulation can be used to prove a relativizable containment. Hence there are many available techniques that can be applied in either domain. Our proofs, both for Turing machine classes and circuit classes, rest on a number of techniques which have, over several years, led to some very important results regarding the class ACC. It is worth-while to briefly review these developments.

Toda proved the first part of his theorem, PH$\subseteq$BP$\cdot\oplus$P, using techniques introduced by Valiant and Vazirani [ValVaz 86]. Using polynomial methods introduced by Razborov and Smolensky, Allender [Al 89] proved the circuit analog of this result: any $\mathrm{AC}^{(0)}$ predicate is computed with high probability by a quasi-polynomial size circuit consisting of a parity gate connected to small AND's. Allender and Hertrampf [AlHe 90] subsequently found that the technique of Valiant and Vazirani could also be applied to circuits to obtain a uniform version of Allender's result.

Yao [Yao 90] then showed that these techniques could be used to obtain the first non-trivial upper bound for ACC. Applying the techniques of Valiant and Vazirani, as well as the polynomials Toda constructed to prove that $\mathrm{PP}^{\oplus\mathrm{P}} \subseteq \mathrm{P}^{\mathrm{PP}}$, Yao showed that every language in ACC is recognized by a family of depth-2 probabilistic circuits of size $2^{(\log n)^{O(1)}}$ with a symmetric gate at the root and AND-gates of fan-in $(\log n)^{O(1)}$ at the leaves. Recently Beigel and Tarui [BeTa 91] have simplified Yao's proof and improved the result, showing that the circuits given by Yao can be made deterministic without increasing their size. (In fact, even circuits with a symmetric gate at the root and ACC subcircuits can be simulated by a depth 2 circuit consisting of a symmetric gate over small AND's. As Beigel and Tarui indicate, this is the circuit analog of the result that PH and all the Mod_mP classes are low for $\mathrm{P}^{\#\mathrm{P}[1]}$, relative to all oracles.)

In both results ([Yao 90] and [BeTa 91]), the symmetric gate at the root depends on the number of inputs and the types of modular gates used in the ACC circuit. It is therefore very hard 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 at 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 $\sum_{i=1}^{w} x_i$. Thus we prove that any family of ACC circuits can be computed by a family of depth-2 deterministic circuits of size $2^{(\log n)^{O(1)}}$ with a MidBit gate at the root and AND-gates of fan-in $(\log n)^{O(1)}$ at the leaves (we refer to these as MidBit^+ circuits; see Definition 6.2). Furthermore, as is evident from our lowness results, even a circuit consisting of a Midbit of ACC circuits can be simulated by MidBit^+ circuits. Most of the techniques of [BeTa 91] can be applied to prove this. The main technical contribution of this paper regarding the ACC problem is a technique to prove that a MidBit of Mod_p gates (for p prime) can be simulated by a MidBit^+ circuit. This same technique is used to prove our lowness results. By multiplying by a carefully chosen number which is not too large, the rightmost "bit" of a number written in base p can be represented as a single bit in the middle of a binary string. This key result is implicit in Lemma 5.1. By choosing an appropriate Toda polynomial, the bit can be "isolated" from the rest of the string, giving the lowness result (Theorem 5.2 and, in circuit form, Theorem 6.3).

Yao [Yao 90] conjectured that there are TC_0 languages which cannot be computed by probabilistic circuits consisting of a symmetric gate over small AND's, and Beigel and Tarui [BeTa 91] make a similar, probably weaker conjecture for deterministic circuits of this kind. Likewise, we believe that there are TC_0 languages that cannot be computed by MidBit^+ circuits. The study of these circuits may therefore provide a way to show that TC_0 is not contained in ACC.

2 Preliminaries and Notation

All languages considered here are over the alphabet $\Sigma = \{0, 1\}$. The length of a string $x \in \Sigma^*$ is denoted by $|x|$. If n is a natural number, $|n|$ denotes the length of its binary encoding, namely $|n| = \lceil \log_2(n + 1) \rceil$. The notation $\langle \cdot, \cdot \rangle : \Sigma^* \times \Sigma^* \to \Sigma^*$ denotes a pairing function that is computable in polynomial time and has inverses also computable in polynomial time. For a set A, $|A|$ denotes its cardinality. The characteristic function of a set A is denoted by χ_A.

We assume that the reader is familiar with (nondeterministic, polynomial time bounded, oracle) Turing machines and complexity classes (see [BaDiGa 87, Schö 86]). FP is the class of functions computable by a deterministic polynomial time bounded Turing transducer.

The class of functions computable by a deterministic polynomial time bounded oracle Turing transducer asking parallel queries to a (set or function) oracle in $\mathcal{C}$ is denoted by $\mathrm{FP}^{\mathcal{C}}_{tt}$.

An NP machine is a nondeterministic polynomial time bounded Turing machine M that on every computation path either accepts or rejects. The number of all accepting computation paths of M on input x is denoted by $\#acc_M(x)$. A set L is said to be in the class PP if there exists an NP machine M whose running time is bounded by a polynomial p, such that for any x, $x \in L$ iff $\#acc_M(x) > 2^{p(|x|)-1}$. A set L is in $\oplus$P if there exists an NP machince M such that for any x, $x \in L$ iff $\#acc_M(x)$ is odd. For any natural number $k > 2$ the class $\mathrm{Mod}_k\mathrm{P}$ is similarly defined except that $x \in L$ iff $\#acc_M(x) \not\equiv 0 \pmod{k}$.

For a relativizable language class $\mathcal{C}$, $\mathcal{C}^{\mathcal{B}[k]}$ is the class of all sets in $\mathcal{C}^{\mathcal{B}}$ witnessed by a machine of type $\mathcal{C}$ asking at most k adaptive queries on every computation path.

Let $\leq_\alpha$ be any reducibility. The reduction class $\{A \mid \exists B \in \mathcal{C} : A \leq_\alpha B\}$ of all sets $\leq_\alpha$-reducible to some set in $\mathcal{C}$ is denoted by $R_\alpha(\mathcal{C})$.

3 Counting Classes and Bits of #P Functions

As indicated above, $(\mathrm{PP} \cup \oplus\mathrm{P}) \subseteq \mathrm{MP}$. In fact,

Proposition 3.1

(a) $\mathrm{PP}^{\oplus\mathrm{P}} \subseteq \mathrm{MP} \subseteq \mathrm{P}^{\#\mathrm{P}[1]}$.

(b) MP *is closed under complementation.*

(c) MP *has complete sets under* $\leq^{\mathrm{p}}_{\mathrm{m}}$ *and is closed under* $\leq^{\mathrm{p}}_{\mathrm{m}}$.

Proof. (a) The inclusion $\mathrm{PP}^{\oplus\mathrm{P}} \subseteq \mathrm{MP}$ follows from inspection of Toda's proof [Tod 89] that $\mathrm{PP}^{\oplus\mathrm{P}} \subseteq \mathrm{P}^{\#\mathrm{P}}$. The inclusion $\mathrm{MP} \subseteq \mathrm{P}^{\#\mathrm{P}[1]}$ is obvious.

(b) Let f be a #P function and let $g \in \mathrm{FP}$ be a bit selection function witnessing $L \in \mathrm{MP}$. Consider the #P function $h(x) = f(x) + 2^{g(x)}$. Since there is a 1 at position $g(x)$ in the binary representation of $f(x)$ if and only if there is a 0 at position $g(x)$ in the binary representation of $h(x)$, it follows that $\overline{L} \in \mathrm{MP}$.

(c) The language $U_{\mathrm{MP}} = \{\langle N, x, 0^k, 0^m\rangle \mid N$ is a nondeterministic TM and there is a 1 at position k in the binary representation of the number of all accepting paths of length $\leq m$ of N on input $x\}$ can easily be seen to be complete for MP under $\leq^{\mathrm{p}}_{\mathrm{m}}$. Now let B be in MP via some #P function f and bit selection function $g \in \mathrm{FP}$, and suppose that $A \leq^{\mathrm{p}}_{\mathrm{m}} B$ via some FP function h. Then $f \circ h$ is in #P and $g \circ h$ is in FP, and it holds for all $x \in \Sigma^*$ that $x \in A$ if and only if there is a 1 at position $g(h(x))$ in the binary representation of $f(h(x))$, i.e. $A \in \mathrm{MP}$. □

Proposition 3.2 *Let L be in MP via a function $f \in \#\mathrm{P}$ and a bit selection function $g \in \mathrm{FP}$.*

(a) [BeGiHe 90] *If $g(x) = O(\log(|x|))$, then $L \in \oplus\mathrm{P}$.*

(b) [BeReSp 91] *If $|f(x)| - g(x) = O(\log(|x|))$, then $L \in \mathrm{PP}$.*

Proof. (b) Let c be a constant such that $|f(x)| - g(x) \leq c\log(|x|)$ for all $x \in \Sigma^*$. Then $f(x) < 2^{g(x)+c\log(|x|)}$, and the bits at the positions $g(x) + c\log(|x|) - 1, \ldots, g(x)$ in the binary representation of $f(x)$ can be computed in polynomial time by binary search asking $c\log(|x|)$ many queries to the PP oracle set $\{\langle x, i\rangle \mid f(x) \geq i\}$. This shows that $L \in \mathrm{P}^{\mathrm{PP}[O(\log n)]}$, which equals PP [BeReSp 91]. □

The previous theorem shows that the bits at either end are weak. However it is easy to see that the bit in the middle is strong. Furthermore, a wide range of bits around the middle are also strong including bits whose distance from either end of the string is as small as any polynomial fraction of the length of the string.

Proposition 3.3 *(a) Let $L \in \mathrm{MP}$. Then there is a #P function f such that for all x, $|f(x)|$ is odd, and $x \in L$ iff the middle bit of $f(x)$ is a 1.*

(b) Let $L \in \mathrm{MP}$, and let the polynomial q and the constants $\epsilon, \delta > 0$ be fixed. Let g be any FP function. Let p be any polynomial such that for all inputs x, with $n = |x|$, $p(n)\delta/q(n) < g(x) < (1 - \epsilon/q(n))p(n)$. Then $L \in \mathrm{MP}$ via a #P function f where $f(x) \leq 2^{p(|x|)}$ and using g as a bit-selection function.

Proof. (a) Let the NTM M, polynomial p, and bit-selection function $g \in \mathrm{FP}$ be such that $L \in \mathrm{MP}$ via f and g, and for all x, $\#acc_M(x) < 2^{p(|x|)}$. Then let M' be an NTM which on any input x first calculates $d := p(|x|) - g(x)$, makes d-many dummy nondeterministic moves, and then simulates $M(x)$. This multiplies the number of accepting computations of $M(x)$ by 2^d, and thus moves bit $g(x)$ of $\#acc_M(x)$ to position $p(|x|)$. Now build an NTM M'' such that for all x, $\#acc_{M''}(x) = \#acc_{M'}(x) + 2^{2p(|x|)}$, and let $f(x) := \#acc_{M''}(x)$. Then f has the desired property. Part (b) is proved by similar "bit-shifting" methods. □

Define a *bit-query machine* to be a machine which takes a function f as oracle and makes queries of the form (y, i), receiving bit i of $f(y)$ from the oracle. Since $\mathrm{MP} \subseteq \mathrm{P}^{\mathrm{PP}}$, the leftmost bit can be used as an oracle for the middle bit. That is, for every $L \in \mathrm{MP}$ there is a polynomial-time bit-query machine M, a #P function f, and a polynomial p which bounds the lengths of values of f, such that M makes queries of the form $(y, p(|y|))$ and accepts L with oracle f. On the other hand, the rightmost bit probably cannot be used as an oracle for the middle bit, since $\mathrm{P}^{\oplus\mathrm{P}} = \oplus\mathrm{P}$ and MP is unlikely to be contained in $\oplus\mathrm{P}$. The next result extends the range to the right of bits which can be used as oracles for the middle bit.

Proposition 3.4 *Let $L \in \mathrm{MP}$, let $\delta > 0$ and $k \geq 1$ be fixed, and let $g \in \mathrm{FP}$ be such that for all y, $g(y) > \delta|y|^{1/k}$. Then there is a polynomial-time machine M with oracle function $f \in \#\mathrm{P}$ such that on any input x, $M(x)$ makes one bit query $(y, g(y))$, and accepts iff the bit returned is a 1.*

Proof. The string y has the form $0^m 1x$, where $m := \lceil |x|^k/\delta \rceil - |x| - 1$. The remaining details are similar to those of Proposition 3.3 and are left to the reader. □

The proof gives a many-one reduction; we do not know whether a Turing reduction could use lower-order bits. Nor do we know more about the power of bits $g(y)$ for functions g which are $\omega(\log |y|)$ and $o(|y|^\epsilon)$ for all $\epsilon > 0$.

Since a bit-query machine with a #P function as oracle can always be simulated by a standard oracle TM with an MP language as oracle, and vice-versa, we revert to the standard formalism in assessing the power of the number of bit-queries to a #P function:

Theorem 3.5

(a) $\mathrm{P}^{\mathrm{MP}[1]} = \mathrm{MP}$.

(b) MP *is closed under intersection if and only if* $\mathrm{MP} = \bigcup_{k\geq 1} \mathrm{P}^{\mathrm{MP}[k]}$.

Proof. (a) It is easy to see that MP is closed under join. By Proposition 3.1, MP is also closed under $\leq_{\mathrm{m}}^{\mathrm{p}}$ and under complementation, and thus it follows that MP is closed under $\leq_{1tt}^{p}$, i.e. $\mathrm{P}^{\mathrm{MP}[1]} = \mathrm{MP}$.

(b) Since MP contains P and is closed under polynomial time many-one reductions, it follows that the bounded truth-table closure of MP (which is easily seen to be $\bigcup_{k\geq 1} \mathrm{P}^{\mathrm{MP}[k]}$) coincides with the Boolean closure of MP [KöScWa 87]. Now, since MP is closed under complementation, if MP is also closed under intersection then the Boolean closure of MP equals MP. □

Proposition 3.6

(a) $\mathrm{P}^{\mathrm{MP}} = \mathrm{P}^{\mathrm{PP}} = \mathrm{P}^{\#\mathrm{P}}$.

(b) $\mathrm{FP}_{\|}^{\mathrm{MP}} = \mathrm{FP}_{\|}^{\#\mathrm{P}}$

(c) The closure of MP *under polynomial time tt-reductions equals* $\mathrm{P}^{\#\mathrm{P}[1]}$.

Proof. (a) is obvious.

(b) follows from the fact that the value of a #P function can be computed in polynomial time by asking parallel queries to an MP oracle.

(c) Since $\mathrm{P}^{\#\mathrm{P}[1]}$ is closed under polynomial time tt-reductions [CaHe 89], it follows that $R_{tt}(\mathrm{MP}) \subseteq R_{tt}(\mathrm{P}^{\#\mathrm{P}[1]}) = \mathrm{P}^{\#\mathrm{P}[1]}$. The converse inclusion follows from (b). □

There are two unresolved structural properties of the class MP which seem both important and amenable to attack. The first is the problem of whether MP is closed under intersection. The direct attempt to solve this by writing and solving equations leads to the following purely numerical question, which we have circulated among mathematicians. (Say x is *top modulo* 2^k if $x \bmod 2^k \geq 2^{k-1}$.)

> In terms of k, what is the minimum degree of an integer-valued polynomial $p(x, y)$ such that for some polynomial t and for all x, y it is true that $p(x, y)$ is top modulo $2^{t(k)} \iff$ both x and y are top modulo 2^k?

The simplest polynomial we know which satisfies this congruence relation is $p(x, y) := \binom{2^{k-1}}{x}\binom{2^{k-1}}{y}2^{k-1}$. Smaller ones have been found by A. Odlyzko and M. Coster [personal communication, 1991], but they still have exponential degree and coefficient size. If such p can be found with degree polynomial in k, then p can be written as a polynomial-sized sum of small binomial coefficients in x and y, which can then be used in building polynomial-time NTMs. Then it would follow, after "lining-up" decision bits, that MP is closed under intersection. A similar congruence relation modulo 2^k with the same open problem is $p(x, y) = 0 \iff (x = 0 \wedge y = 0)$.

The second open problem concerns whether the inclusions in Proposition 3.1(a) are proper. It is not even known whether there is an oracle separating $\mathrm{PP}^{\oplus\mathrm{P}}$ from $\mathrm{P}^{\#\mathrm{P}[1]}$ or even from PSPACE. Since $\mathrm{PP}^{\oplus\mathrm{P}}$ is closed under polynomial-time truth-table reductions, as follows by relativizing the proof for PP in [FoRe 91], $\mathrm{MP} = \mathrm{PP}^{\oplus\mathrm{P}}$ implies that both classes are equal to $\mathrm{P}^{\#\mathrm{P}[1]}$.

4 The Class AmpMP

Toda's proof, which as mentioned in Prop. 3.1(a) yields $\mathrm{PP}^{\oplus\mathrm{P}} \subseteq \mathrm{MP}$, actually shows that languages L in $\mathrm{PP}^{\oplus\mathrm{P}}$ have MP-representations of a special kind. Namely, there is a #P function f such that for any input x and number m, not only does the middle bit of $f(x, 0^m)$ equal '1' iff $x \in L$, but also the m bits to the left of this bit are always '0'. We call this property "amplification on the left of the decision bit." Technically, $(x, 0^m)$ stands for the string $x10^m$, and the point is that m can be made as large as desired. In this section we study the stronger notion of "amplification on both sides of the decision bit," which leads to the class AmpMP formalized as follows:

Definition 4.1 *A language L is in* AmpMP *if there are a polynomial p and a* #P *function f such that for every $x \in \Sigma^*$ and $m > 0$, $f(x, 0^m)$ is of the form*

$$f(x, 0^m) = a(x, 0^m)2^{p(n)+2m+1} + \chi_L(x)2^{p(n)+m} + b(x, 0^m)$$

where $n = |x| + m$ and $b(x, 0^m) < 2^{p(n)}$.

In other words, L is in AmpMP if there are polynomials p, r and a #P function f such that for every $x \in \Sigma^*$ and $m > 0$, the binary representation of $f(x, 0^m)$ is of the form

$$a_{r(n)} \cdots a_0 \underbrace{0 \ldots 0}_{m \text{ times}} \chi_L(x) \underbrace{0 \ldots 0}_{m \text{ times}} b_{p(n)-1} \cdots b_0$$

where $b_0, \ldots, b_{p(n)-1}, a_0, \ldots, a_{r(n)} \in \{0, 1\}$. The next lemma shows that the class AmpMP is very robust, and closed under Boolean operations.

Lemma 4.2

(a) AmpMP *is closed under complementation,*

(b) AmpMP *is closed under intersection,*

(c) AmpMP *is closed under bounded truth table reductions.*

Proof. (a) Let L be in AmpMP. We have to show that there are a polynomial p and a #P function h fulfilling the condition in the definition of AmpMP for the complement of L. Since L is in AmpMP there are a polynomial p and a #P function f such that for every $x \in \Sigma^*$ and $m > 0$, the binary representation of $f(x, 0^m)$ is of the form

$$a_{r(n)} \cdots a_0 \underbrace{0 \ldots 0}_{m \text{ times}} \chi_L(x) \underbrace{0 \ldots 0}_{m \text{ times}} b_{p(n)-1} \cdots b_0$$

where $n = |x| + m$. Consider the following function $f'(x, 0^m)$ whose value in binary is

$$a_{r(n)} \cdots a_0 \underbrace{1 \ldots 1}_{m \text{ times}} \chi_L(x) \underbrace{1 \ldots 1}_{m \text{ times}} b_{p(n)-1} \cdots b_0$$

Clearly, $f' \in$ #P, and there are a polynomial t and an NP machine M having on input $(x, 0^m)$ exactly $2^{t(n)}$ different computation paths such that $\#acc_M(x, 0^m) = f'(x, 0^m) + 1$. Now the desired #P function h can be obtained by inverting f' bitwise, i.e. $h(x, 0^m) = 2^{t(n)} - 1 - f'(x, 0^m) = \#acc_{\overline{M}}(x, 0^m)$, where $\overline{M}$ is obtained from M by interchanging accepting and rejecting states.

(b) Let A, B be two sets in AmpMP. We have to show that there are a polynomial p and a #P function h fulfilling the condition in the definition of AmpMP for the set $A \cap B$. Since A, B are in AmpMP, there are polynomials p_A, p_B, t and #P functions h_A, h_B such that $t(n) \geq p_A(n) + 2m$ (letting $n = |x| + m$) and

$$h_A(x, 0^m) = a(x, 0^m) 2^{p_A(n)+2m+1} + \chi_A(x) 2^{p_A(n)+m} + b(x, 0^m) < 2^{t(n)}$$

where $b(x, 0^m) < 2^{p_A(n)}$, and

$$h_B(x, 0^m) = a'(x, 0^m) 2^{p_B(n)+2m+1} + \chi_B(x) 2^{p_B(n)+m} + b'(x, 0^m)$$

where $b'(x, 0^m) < 2^{p_B(n)}$. Now define $h(x, 0^m) = h_A(x, 0^m) \cdot h_B(x, 0^{t(n)})$, then

$$\begin{aligned}
& h(x, 0^m) \\
&\quad = [a'(x, 0^{t(n)}) 2^{p_B(n)+2t(n)+1} + \chi_B(x) 2^{p_B(n)+t(n)} + b'(x, 0^{t(n)})] \cdot h_A(x, 0^m) \\
&\quad = a''(x, 0^m) 2^{p_A(n)+p_B(n)+t(n)+2m+1} + \chi_{A \cap B}(x) 2^{p_A(n)+p_B(n)+t(n)+m} + b''(x, 0^m)
\end{aligned}$$

where $b''(x, 0^m) = \chi_B(x) 2^{p_B(n)+t(n)} b(x, 0^m) + b'(x, 0^{t(n)}) h_A(x, 0^m) < 2^{p_A(n)+p_B(n)+t(n)}$, and $a''(x, 0^m) = a'(x, 0^{t(n)}) 2^{t(n)-p_A(n)-2m} h_A(x, 0^m) + \chi_B(x) a(x, 0^m)$.

(c) We first show that AmpMP is closed under many-one reductions. Let B be in AmpMP, and suppose that $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ via some FP function g. Since B is in AmpMP there are a polynomial p and a #P function f such that for every $x \in \Sigma^*$ and $m > 0$, the binary representation of $f(x, 0^m)$ is of the form

$$f(x, 0^m) = a(x, 0^m)2^{p(n)+2m+1} + \chi_B(x)2^{p(n)+m} + b(x, 0^m)$$

where $n = |x| + m$ and $b(x, 0^m) < 2^{p(n)}$. Consider the function $f'(x, 0^m) = f(g(x), 0^m)$ which is of the form

$$f'(x, 0^m) = a(g(x), 0^m)2^{p(n)+2m+1} + \chi_A(x)2^{p(n)+m} + b(g(x), 0^m)$$

where $n = |g(x)| + m$ and $b(g(x), 0^m) < 2^{p(n)}$. Let q be a polynomial such that $q(|x|+m) \geq p(|g(x)| + m)$. Then the #P function $f''(x, 0^m) = f'(x, 0^m) \cdot 2^{q(|x|+m)-p(|g(x)|+m)}$ and the polynomial q form an AmpMP representation for A.

Since AmpMP contains P and is closed under many-one reductions it follows that the bounded truth-table closure of AmpMP coincides with the Boolean closure of AmpMP [KöScWa 87]. This completes the proof because by (a) and (b) AmpMP is closed under Boolean operations. $\square$

It is an open problem whether AmpMP is closed under conjunctive reductions. As we will see this problem is related to the lowness properties of the class. 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., $\mathcal{C}^{\mathcal{A}} = \mathcal{C}$.

We would like to prove that AmpMP is low for the class MP. This would happen if AmpMP were closed under conjunctive polynomial time reducibility. We can prove however a series of lowness results which are based on the following theorem.

Theorem 4.3 *Let k be a constant. For every function $f \in \#\mathrm{P}^{\mathrm{AmpMP}[k]}$ there are a function $g \in \#\mathrm{P}$ and a polynomial p such that for every $x \in \Sigma^*$ and $m > 0$,*

$$f(x) \equiv \lfloor g(x, 0^m)/2^{p(|x|+m)} \rfloor \pmod{2^m}.$$

Proof. Since AmpMP is closed under bounded truth table reductions there is a language $A \in$ AmpMP and a polynomial q such that

$$f(x) = \sum_{y \in \Sigma^{q(|x|)}} \chi_A(x, y)$$

Because of $A \in$ AmpMP there are a polynomial r and a #P function h such that for every $x \in \Sigma^*$ and $m > 0$, $h(x, y, 0^m)$ is of the form

$$h(x, y, 0^m) = a(x, y, 0^m)2^{r(n)+2m+1} + \chi_A(x, y)2^{r(n)+m} + b(x, y, 0^m)$$

where $n = |x| + m$ and $b(x, y, 0^m) < 2^{r(n)}$. Now the proof of the theorem is completed by choosing the polynomial $p(n) \geq r(n) + q(|x|) + m$ and defining $g(x, 0^m) = \sum_{y \in \Sigma^{q(|x|)}} h(x, y, 0^{m+q(|x|)}) 2^{p(n)-r(n)-q(|x|)-m}$. $\square$

Note that Theorem 4.3 even allows us to isolate the binary representation of $f(x)$ inside the binary representation of some #P function $h(x, 0^m)$ by m 0's to the left and to the right, i.e., $h(x, 0^m)$ is of the form

$$a_{r(n)} \cdots a_0 \underbrace{0 \ldots 0}_{m \text{ times}} \mathit{bin}(f(x)) \underbrace{0 \ldots 0}_{m \text{ times}} b_{p(n)-1} \cdots b_0$$

where $n = |x| + m$, p, r are polynomials, $b_0, \ldots, b_{p(n)-1}, a_0, \ldots, a_{r(n)} \in \{0, 1\}$, and $\mathit{bin}(f(x)) \in \Sigma^{t(|x|)}$ is the binary representation of $f(x)$ (possibly with leading 0's) for some polynomial t. To see this, first define $f'(x, 0^m) = f(x) \cdot 2^m$ and apply Theorem 4.3 to get a function $h(x, 0^m)$ such that $f'(x, 0^m) \equiv \lfloor h(x, 0^m)/2^{p(|x|+m)} \rfloor \pmod{2^{t(|x|)+2m}}$.

Now we are ready to state our first "lowness" result.

Corollary 4.4

1. $\bigcup_{k>1} \mathrm{MP}^{\mathrm{AmpMP}[k]} = \mathrm{MP}$
2. $\bigcup_{k>1} \mathrm{AmpMP}^{\mathrm{AmpMP}[k]} = \mathrm{AmpMP}$

Proof. By Theorem 4.3 it follows that for every function $f \in \#\mathrm{P}^{\mathrm{AmpMP}[k]}$ there exist a function $g \in$ #P and a polynomial t such that the binary representation of $f(x)$ is reproduced inside the binary representation of $g(x, 0^{t(|x|)})$. The rest is clear. $\square$

Corollary 4.5 *Let C be a subclass of* AmpMP. *If C is closed under conjunctive and disjunctive reducibilities then C is low for* MP *and for* AmpMP.

Proof. This is a direct consequence of Corollary 4.4 since if C is closed under conjunctive and disjunctive reducibilities then it is easy to see that $\mathrm{MP}^C = \mathrm{MP}^{C[2]}$ and $\mathrm{AmpMP}^C = \mathrm{AmpMP}^{C[2]}$. $\square$

Corollary 4.6 *If* $\mathrm{C_{=}P} \subseteq \mathrm{AmpMP}$ *then* $\mathrm{CH} = \mathrm{MP}$.

Proof. Assume that $\mathrm{C_{=}P} \subseteq \mathrm{AmpMP}$. Since the class $\mathrm{C_{=}P}$ is closed under disjunctive and conjunctive reductions ([Tor 88],[Gr 91],[BeChOg 91]) it follows by Corollary 4.5 that $\mathrm{C_{=}P}$ is low for MP. Now the collapse of the counting hierarchy follows easily using the equality $\mathrm{MP}^{\mathrm{MP}} = \mathrm{MP}^{\mathrm{C_{=}P}}$ [Tor 88]. $\square$

We show now that several important complexity classes are included in AmpMP. It will follow from the next result proved by Toda [Tod 89] that $\oplus$P is contained in AmpMP (even in the subclass of AmpMP where $p(n) = 0$).

Theorem 4.7 [Tod 89] *For every language $A \in \oplus\mathrm{P}$ there exists a #P function h such that $h(x, 0^m) \equiv \chi_A(x) \pmod{2^m}$.*

Corollary 4.8 *$\oplus$P is contained in AmpMP and therefore $\oplus$P is low for MP and AmpMP.*

Proof. Let A be in $\oplus$P. By Theorem 4.7 there is a function $h \in \#\mathrm{P}$ such that $h(x, 0^m) \equiv \chi_A(x) \pmod{2^{m+1}}$. Then also the function $f(x, 0^m) = h(x, 0^m) \cdot 2^m$ is in #P, witnessing $A \in \mathrm{AmpMP}$. The lowness follows by Corollary 4.5 since $\oplus$P is closed under Turing reductions [PaZa 83]. □

The next corollary states that for every function f in $\#\mathrm{P}^{\oplus\mathrm{P}}$ there is a function $g \in \#\mathrm{P}$ such that $f(x)$ and $g(x, 0^m)$ agree in the last m bits.

Corollary 4.9 *For every function f in $\#\mathrm{P}^{\oplus\mathrm{P}}$ there exists a function g in #P such that*

$$g(x, 0^m) \equiv f(x) \pmod{2^m}.$$

Proof. Let f be in $\#\mathrm{P}^{\oplus\mathrm{P}}$. Since $\oplus$P is closed under Turing reductions [PaZa 83] there exist a language A in $\oplus$P and a polynomial q such that

$$f(x) = \sum_{y \in \Sigma^{q(|x|)}} \chi_A(x, y)$$

By Theorem 4.7 there is a function $h \in \#\mathrm{P}$ such that $h(x, y, 0^m) \equiv \chi_A(x, y) \pmod{2^m}$. Now the corollary follows defining $g(x, 0^m) = \sum_{y \in \Sigma^{q(|x|)}} h(x, y, 0^m)$. □

As a consequence of the next theorem we will get the containment of BPP in AmpMP (even in the subclass of AmpMP where $a(x, 0^m) = 0$).

Theorem 4.10 *For every language $L \in \mathrm{BPP}$ there exist a polynomial t and a function $h \in \#\mathrm{P}$ such that*

$$\chi_L(x) \cdot 2^m = \lfloor h(x, 0^m)/2^{t(n)-m} \rfloor$$

where $n = |x| + m$.

Proof. Let L be in BPP. By the probability amplification lemma for BPP, there exists a function $h \in \#\mathrm{P}$ and a polynomial t such that

$$x \in L \;\Rightarrow\; h(x, 0^m) \geq 2^{t(n)} - 2^{t(n)-m-2},$$
$$x \notin L \;\Rightarrow\; h(x, 0^m) \leq 2^{t(n)-m-2},$$

and therefore h fulfills the following inequalities,

$$\chi_L(x, 0^m)2^{t(n)} - 2^{t(n)-m-2} \leq h(x, 0^m) \leq \chi_L(x, 0^m)2^{t(n)} + 2^{t(n)-m-2}$$

Because #P is closed under addition, the proof can be completed by defining $h(x, 0^m) = h(x, 0^m) + 2^{t(n)-m-2}$ since h fulfills the inequalities

$$\chi_L(x, 0^m)2^{t(n)} \leq h(x, 0^m) \leq \chi_L(x, 0^m)2^{t(n)} + 2^{t(n)-m-1}$$

□

Corollary 4.11 BPP *is contained in* AmpMP *and therefore is low for* MP *and* AmpMP*.*

Proof. The containment of BPP in AmpMP follows immediately from Theorem 4.10, and the lowness follows by Corollary 4.5 since BPP is closed under Turing reductions. □

The next corollary states that for every function $f \in \#\mathrm{P}^{\mathrm{BPP}}$ and every polynomial p there is a function $g \in \#\mathrm{P}$ such that $f(x)$ and $g(x)$ agree in the leftmost $p(|x|)$ many bits where the leftmost bit of a binary number is the most significant bit which is 1.

Corollary 4.12 *For every function $f \in \#\mathrm{P}^{\mathrm{BPP}}$ there exist a polynomial r and a function $g \in \#\mathrm{P}$ such that*

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

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

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

By Theorem 4.10 there is a #P function h and a polynomial t such that

$$\chi_L(x, y) \cdot 2^m = \lfloor h(x, y, 0^m)/2^{t(|x|+m)-m} \rfloor$$

Defining the #P function g as

$$g(x) = \sum_{y \in \Sigma^{q(|x|)}} h(x, y, 0^{q(|x|)})$$

it follows that

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

which completes the proof if we choose $r(n) = t(n + q(n))$. □

Corollary 4.13 $\mathrm{BPP}^{\oplus\mathrm{P}}$ *and the polynomial hierarchy* PH *are low for both* MP *and* AmpMP.

Proof. Since $\mathrm{BPP}^{\oplus\mathrm{P}}$ is closed under Turing reductions, it suffices by Corollary 4.5 to show that it is contained in AmpMP. This follows by relativizing Corollary 4.11 to $\oplus$P and observing $\mathrm{AmpMP}^{\oplus\mathrm{P}} \subseteq \mathrm{AmpMP}$ by Corollary 4.8. The lowness of PH follows since $\mathrm{PH} \subseteq \mathrm{BPP}^{\oplus\mathrm{P}}$. □

Using relativized versions of Theorems 4.12 and 4.9, we get the following theorem which is stronger than what we would get by Theorem 4.3 because the polynomial p only depends on $|x|$ and not on m.

Theorem 4.14 *For every function f in $\#\mathrm{P}^{\mathrm{BPP}^{\oplus\mathrm{P}}}$ there are a function $g \in \#\mathrm{P}$ and a polynomial p such that*

$$f(x) \equiv \lfloor g(x, 0^m)/2^{p(|x|)} \rfloor \pmod{2^m}.$$

Proof. Let f be in $\#\mathrm{P}^{\mathrm{BPP}^{\oplus\mathrm{P}}}$. Since Corollary 4.12 relativizes, there exists a polynomial p and a function $h \in \#\mathrm{P}^{\oplus\mathrm{P}}$ such that

$$f(x) = \lfloor h(x)/2^{p(|x|)} \rfloor$$

By Theorem 4.9, there exists a function $g \in \#\mathrm{P}$ such that

$$g(x, 0^m) \equiv h(x) \pmod{2^{m+p(|x|)}},$$

and therefore

$$\lfloor g(x, 0^m)/2^{p(|x|)} \rfloor \equiv f(x) \pmod{2^m}.$$

□

5 Lowness of Mod Classes for the Class MP

In this section we show that for any k, $\mathrm{Mod}_k\mathrm{P}$ is included in AmpMP, thereby proving that $\mathrm{Mod}_k\mathrm{P}$ is low for MP and AmpMP.

The key to this result is the following lemma, which says that the "amplification" of a #P-function in k-adic representation can, in some sense, be saved in dyadic representation.

Lemma 5.1 *Let r, q be polynomials.*

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

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

then there exist a function h in #P and a polynomial p such that

$$h(x) = a'(x)2^{p(|x|)+q(|x|)} + b(x)2^{p(|x|)} + c(x),$$

where $c(x) < 2^{p(|x|)}$ and $a'(x)$ is a multiple of $a(x)$.

Proof. Since f is in #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 p and a function g in #P such that

$$g(x) = a(x)2^{p(|x|)} + b'(x) \text{ and } b'(x) < 2^{p(|x|)-q(|x|)-1}.$$

Proof of Claim. Define

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

Then it follows that

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

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

To complete the proof of Lemma 5.1 we define

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

where

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

Then it follows that

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

where

$$k^{r(n)} + i(n) \equiv 0 \pmod{2^{q(n)}}$$

and

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

$\square$

Theorem 5.2 *For every prime k,* $\mathrm{Mod}_k\mathrm{P}$ *is included in* AmpMP

Proof. Let A be a set in $\mathrm{Mod}_k\mathrm{P}$ and let r be a polynomial such that $k^{r(m)} > 2^{2m+3}$. Adapting results from Toda [Tod 89] and Beigel, Gill and Hertrampf [BeGiHe 90] we can assume that there is a function c in #P such that

$$c(x, 0^m) \equiv \chi_A(x) \pmod{k^m}$$

Now let $f(x, 0^m) = c(x, 0^{r(m)}) \cdot 2^{m+1}$. Then we have

$$f(x, 0^m) = a(x, 0^m)2^{m+1}k^{r(m)} + \chi_A(x)2^{m+1}$$

where $\chi_A(x)2^{m+1} < k^{r(m)}/2^{m+2}$, so we can apply Lemma 5.1 to obtain a polynomial p and a function h in #P such that

$$h(x, 0^m) = a'(x, 0^m)2^{m+1+p(n)} + \chi_A(x)2^{m+1+p(n)} + c(x, 0^m)$$

where $c(x, 0^m) < 2^{p(n)}$. Remembering that $a'(x, 0^m)$ is a multiple of 2^{m+1} we get an AmpMP characterization for A. $\square$

Because Theorem 5.2 relativizes, we can state the following corollary.

Corollary 5.3 *For any k,* $\mathrm{Mod}_k\mathrm{P}$ *is low for* MP *and* AmpMP*.*

Proof. First observe that for prime k, $\mathrm{Mod}_k\mathrm{P}$ is closed under Turing reductions [BeGiHe 90] and therefore is low for MP and AmpMP by Corollary 4.5 and Theorem 5.2. In the case that k is composite it follows by the representation theorem of Hertrampf [He 90] that if $k = p^{e_1}q$ for a prime number p and $\gcd(p, q) = 1$, then

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

Since the above lowness proof for the prime case relativizes the lowness of $\mathrm{Mod}_k\mathrm{P}$ follows iterating this argument for all the prime factors of k. $\square$

Corollary 5.3 together with Theorem 4.3 immediately imply the main result of this section.

Corollary 5.4 *For any k and every function f in $\#\mathrm{P}^{\mathrm{Mod}_k\mathrm{P}}$ there are a function $g \in \#\mathrm{P}$ and a polynomial p such that*

$$f(x) \equiv \lfloor g(x, 0^m)/2^{p(|x|+m)} \rfloor \pmod{2^m}.$$

The result stated in Corollary 5.4 works also for the class ModPH, a generalization of the polynomial time hierarchy that contains also ModP classes. ModPH can be considered as the polynomial time analogue to the circuit class ACC.

Definition 5.5 *ModPH is the smallest family of languages containing the class P and satisfying that for any set A in ModPH the classes NP^A, co-NP^A and Mod_kP^A (for any positive integer k) also are contained in ModPH.*

Corollary 5.6 ModPH *is contained in* AmpMP *and therefore is low for* MP *and* AmpMP.

Corollary 5.7 *For all functions f in $\#\mathrm{P}^{\mathrm{ModPH}}$ there exist a polynomial t and a function h in $\#\mathrm{P}$ such that*

$$\lfloor h(x, 0^m)/2^{t(|x|+m)} \rfloor \equiv f(x) \pmod{2^m}.$$

6 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.

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 6.1 *A MidBit gate over w inputs $x_1, ..., x_w$ is a gate which outputs the value of the $\lfloor log(w)/2 \rfloor^{\mathrm{th}}$ bit in the binary representation of the number $\sum_{i=1}^{w} x_i$.*

A Mod_k gate over w inputs $x_1, ..., x_w$ is defined to output 1 if $\sum_{i=1}^{w} x_i \not\equiv 0 \pmod{k}$ and 0 otherwise.

In our simulations circuits consisting of a particular gate over small AND gates arise frequently, so we introduce the following notation.

Definition 6.2 *Let G be a Boolean gate. A family of circuits $\{C_n\}$ is called a family of G^+ circuits if there is a polynomial p such that for each n, C_n consists of a gate of type G at the root whose inputs are at most $2^{p(log(n))}$ AND gates each of size at most $p(log(n))$. A family of Boolean functions $\{f_n\}$ is computable by a family of G^+ circuits $\{C_n\}$ if for each n, $f_n(x_1, ..., x_n) = C_n(x_1, ..., x_n)$.*

Note that we will always speak of families of MidBit$^+$ or Mod$_k^+$ circuits. Even when we refer to a MidBit$^+$ or Mod$_k^+$ circuit individually, it should be understood that what is meant is a member of a particular family of such circuits.

The following theorem gives the circuit analogue of Corollary 5.4. We find that for any family of functions which can be expressed as sums of Mod$_k^+$ circuits, there is a family of low-degree polynomials whose middle bits agree with the bits of the original functions.

Theorem 6.3 *Let k be prime and let $\{b_n\}$ be a family of functions such that there exists a polynomial r where for each n, b_n is of the form*

$$b_n(x_1, ..., x_n) = \sum_{i=1}^{w} c_i(x_1, ..., x_n),$$

where each c_i is a Mod$_k^+$ circuit and $w \leq r(log(n))$. Then for any polynomial t there are polynomials p and q and a family of polynomials $\{h_n\}$ of degree $p(log(n))$ such that for each n,

$$b_n(x_1, ..., x_n) \equiv \lfloor h_n(x_1, ..., x_n)/2^{q(log(n))} \rfloor \pmod{2^{t(log(n))}}$$

Proof. Similar to the proof of Theorem 5.2. To simplify notation, unless explicitly stated, p, p', q, r, s, and t denote $p(log(n))$, $p'(log(n))$,$q(log(n))$, $r(log(n))$, $s(log(n))$, and $t(log(n))$, respectively. Also denote any function g of $x_1, ..., x_n$ as $g(x)$. We have that each Mod$_k^+$ circuit c_i outputs 1 if and only if a certain sum σ_i of AND-gates is nonzero mod k. (From an observation of Beigel, Gill and Hertrampf [BeGiHe 90], without loss of generality σ_i is always 0 or 1 (mod k), by Fermat's little theorem.) Note that we can think of each σ_i as a polynomial in $\{x_1, ..., x_n\}$ of polylog degree. We make use of polynomials Q_d originally written down by Toda [Tod 89], and improved by Beigel and Tarui [BeTa 91]. The polynomial Q_d is of degree $2d - 1$ and has the property that if $X \not\equiv 0 \pmod{k}$ then $Q_d(X) \equiv 1 \pmod{k^d}$, and if $X \equiv 0 \pmod{k}$ then $Q_d(X) \equiv 0 \pmod{k^d}$. Thus

$$b_n(x) = \sum_{i=1}^{w} \left[Q_d(\sigma_i) \bmod k^d\right]$$

We choose $d = p'(log(n))$ where p' is a polynomial such that $k^{p'} > 2^{r+t+2}$. Then $b_n(x) \leq 2^r < k^{p'}$. Now the outer sum in the equation above for b_n is less than $k^{p'}$, so the "mod" can be moved outside:

$$b_n(x) \equiv \left[\sum_{i=1}^{w} Q_p(\sigma_i)\right] \pmod{k^{p'}}$$

We write

$$f_n(x) = \sum_{i=1}^{w} Q_{p'}(\sigma_i)$$

Then

$$f_n(x) = a_n(x)k^{p'} + b_n(x)$$

for some $a_n(x)$. Note that for some polynomial s, $f_n(x) < 2^s$. Also note that since σ_i is a polynomial of polylog degree, there is some polynomial p such that f_n is a polynomial of degree $p(log(n))$ in the variables $x_1, ..., x_n$. Define the degree $p(log(n))$ polynomial h_n as follows:

$$h_n(x) = i(n) \left\lceil 2^q / k^{p'} \right\rceil f_n(x) + 2^q f_n(x),$$

where $i(n) \equiv -k^{p'} \pmod{2^t}$, following the proof of Lemma 5.1 . Thus we find that $\lceil 2^q/k^{p'} \rceil f_n(x) = a_n(x) 2^q + b'_n(x)$ where $b'_n(x) < 2^{q-t-1}$. Hence

$$h_n(x) \equiv 2^q b_n(x) + i(n) b'_n(x) \pmod{2^{q+t}}$$

where $i(n)b'_n(x) < 2^{q-1}$. This completes the proof. □

Corollary 6.4 *Let k be prime and $\{C_n\}$ be a family of circuits where for each n, C_n consists of a MidBit gate over $2^{polylog}$ Mod_k^+ circuits. Then $\{C_n\}$ is computable by a family of MidBit$^+$ circuits.*

Proof. Each C_n is the MidBit of a sum b_n of Mod_k^+ circuits. Using the previous theorem and adopting the notations of the proof, we can find a family of polylog-degree polynomials $\{h_n\}$ obeying

$$h_n(x) \equiv 2^q b_n(x) + c_n(x) \pmod{2^{q+t}} \quad (*)$$

for some $c_n(x) < 2^{q-1}$. Choose $t > r$. We can express $h_n \pmod{2^{q+t}}$ as a sum of non-negative terms with coefficients $< 2^{q+t-1}$. This can further be rewritten as a sum $h'_n(x)$ of AND gates by replacing terms with coefficients > 1 by a sum of identical terms with unit coefficients. Reducing the right hand side of eq. (*) mod 2^{q+t}, we obtain $2^q(b_n(x) \bmod 2^t) + c_n(x)$. Now the output bit of C_n is in position $\lfloor r/2 \rfloor$ of $b_n(x)$ and is therefore in position $q + \lfloor r/2 \rfloor$ of $h'_n(x)$. We can multiply the sum by repeated addition so that this is precisely the middle bit. □

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 6.5 *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. They also showed that a polylog-size AND of Mod_k gates (for k prime) can be switched with the Mod_k's to produce a Mod_k^+ circuit. Using these facts, Corollary 6.4 and an inductive argument as in the proof of Lemma 6 in [BeTa 91], each layer of Mod_{k_i} gates can be "absorbed" in the MidBit gate, and the resulting polylog fan-in 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 MP. 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 Theorems 1 and 2 of [BeTa 91].

Theorem 6.6 *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, OR, and NOT 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, NOT, 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 $\Pr(c_i' \neq c_i) \leq 2^{-q(log(n))-2}$. 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'. □

Acknowledgements

We wish to thank V. Arvind, Jim Royer and Richard Beigel for helpful discussions.

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.

[AHOW 91] E. Allender, L. Hemachandra, M. Ogiwara, and O. Watanabe, Relating equivalence and reducibility to sparse sets. In *Proceeding 6th Structure in Complexity Theory Conference*, 1991, 220-237.

[BaDiGa 87] J.L. Balcázar, J. Díaz, J. Gabarró, *Structural Complexity I.* Springer, 1987.

[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.

[BeChOg 91] R. Beigel, R. Chang, and M. Ogiwara, A relationship between difference hierarchies and relativized polynomial hierarchies. To appear in *Mathematical Systems Theory.*

[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.

[BeReSp 91] R. Beigel, N. Reingold and D. Spielman, PP is closed under intersection. In *Proceedings of the 23rd ACM Symposium on the Theory of Computation* 1991, 1-11.

[BeTa 91] R. Beigel, J. Tarui, On ACC. In *Proceedings of the 32nd Symposium on Foundations of Computer Science* 1991, 783-792.

[CaHe 89] J. Cai, L. Hemachandra, Enumerative Counting is Hard. In *Information and Computation 92(1)* (1989), 34-44.

[FoRe 91] L. Fortnow and N. Reingold, PP is closed under truth-table reductions. *Proceedings of the 6th Annual Conference on Structure in Complexity Theory* 1991, 13-15.

[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.

[Gr 91] F. Green, On the power of deterministic reductions to $\mathrm{C}_=\mathrm{P}$. To appear in *Mathematical Systems Theory.*

[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.

[KöScWa 87] J. Köbler, U. Schöning, K. W. Wagner. The difference and truth-table hierarchies of NP. In *Theoretical Informatics and Applications*, 21(4):419-435, 1987.

[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.

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

[Schö 86] U. Schöning. *Complexity and Structure.* Springer-Verlag *Lecture Notes in Computer Science* 211, 1986.

[Tod 89] S. Toda, On the computational power of PP and $\oplus$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.