The Power of the Middle Bit of a #P Function [1]

(Revised Version, November 1993)

Frederic Green[2] Clark University

Johannes Köbler[3 4] Universität Ulm

Kenneth W. Regan [5] SUNY/Buffalo

Thomas Schwentick [6] Universität Mainz

Jacobo Torán [7 3] U. Politecnica de Catalunya

[1]Preliminary versions of this work appeared in the *Proceedings of the Fourth Italian Conference on Theoretical Computer Science*, World Scientific, (1992), 317-329, and *Proceedings of the Seventh Annual Conference on Structure in Complexity Theory*, IEEE Computer Society Press, (1992), 111-117.

[2]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.

[3]Universität Ulm, Theoretische Informatik, Oberer Eselsberg, D-89069 Ulm, Germany.

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

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

[6]Universität Mainz, Institut für Informatik, D-55099 Mainz, Germany.

[7]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

This paper studies the class MP of languages which can be solved in polynomial time with the additional information of one bit from a #P function f. The middle bit of $f(x)$ is shown to be as powerful as any other bit, whereas the $O(\log n)$ bits at either end are apparently weaker. The polynomial hierarchy and the classes $\mathrm{Mod}_k\mathrm{P}$, $k \geq 2$, are shown to be low for MP. They are also low for a class we call AmpMP which is defined by abstracting the "amplification" methods of Toda (*SIAM J. Comput.* **20** (1991), 865–877). Consequences of these results for circuit complexity are obtained using the concept of a MidBit gate, which is defined to take binary inputs $x_1, \ldots, x_w$ and output the $\lfloor \log_2(w)/2 \rfloor^{\mathrm{th}}$ bit in the binary representation of the number $\sum_{i=1}^{w} x_i$. 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 celebrated results of Toda [Tod 91] have sparked renewed interest in the complexity classes #P [Va 79], PP (probabilistic polynomial time [Gi 77]), and $\oplus$P (parity polynomial time [PaZa 83, GoPa 86]). One relationship among these classes is that $\oplus$P comprises exactly those languages which are decided with the information in the rightmost bit of a #P function f, and PP, those decided with the information in the leftmost bit. (For the latter statement we arrange, as described later, that binary values $f(x)$ are padded with leading 0's out to a length which depends only on $|x|$.) Toda's two main theorems are that the polynomial-time hierarchy (PH) is contained in $\mathrm{BPP}^{\oplus\mathrm{P}}$, and that the evidently larger class $\mathrm{PP}^{\oplus\mathrm{P}}$ is contained in $\mathrm{P}^{\#\mathrm{P}}$. The observation which inspired the present work is that in Toda's proof of the second result, the full power of $\mathrm{P}^{\#\mathrm{P}}$ is not needed. Namely, the $\mathrm{P}^{\#\mathrm{P}}$ machine M in the proof makes only one query x to its oracle $f \in$ #P, and furthermore the machine's decision depends on only one bit of the answer $f(x)$ in binary notation. Hence it is natural to ask which other languages can be decided by looking at just one bit of a #P function. The class of languages with this property is defined by:

Definition 1.1 *A language L is in* MP *if there exists a* #P *function f and a polynomial-time computable function g such that for all x, $x \in L \Leftrightarrow (\lfloor f(x)/2^{g(x)} \rfloor \bmod 2) = 1$.*

Put another way, $x \in L$ iff there is a 1 at position $g(x)$ in the standard binary representation of $f(x)$. We call g a *bit-selection function*. The "M" stands for "middle bit," since we show, by judicious padding of values $f(x)$ to odd length, that g can be the function which indexes the middle bit of the binary representation of $f(x)$. In Section 3 we establish some basic properties of the class MP, including that MP is closed downward under polynomial-time many-one reductions and has complete problems.

One motivation for studying MP is that very little is known about the structure of classes in the neighborhood of $\mathrm{P}^{\#\mathrm{P}}$, even under relativizations. Indeed, it is not even known whether there exists an oracle set A for which $\mathrm{PP}^{\oplus\mathrm{P}^A} \neq \mathrm{PSPACE}^A$; hence no A for which $\mathrm{P}^{\#\mathrm{P}^A[1]} \neq \mathrm{PSPACE}^A$ is known either. One surprising property of $\mathrm{P}^{\#\mathrm{P}[1]}$ which follows from the proof of Toda's theorem is that a rich array of languages A are *low* for the class, meaning that $\mathrm{P}^{\#\mathrm{P}^A[1]} = \mathrm{P}^{\#\mathrm{P}[1]}$. The main result of this paper is that all of the languages that are known to be low for $\mathrm{P}^{\#\mathrm{P}[1]}$ are also low for the possibly smaller class MP. As shown in Section 4, the class $\mathrm{BPP}^{\oplus\mathrm{P}}$ from Toda's first theorem is low for MP. In proving these lowness results, we interpret the familiar probability amplification in Toda's first theorem as a means of inserting many 0's to the *right* of the important bit, and the novel "amplifying polynomials" in his second theorem as a means of inserting 0's to the *left* of that bit. We abstract these two properties to define the subclass AmpMP of MP languages whose representations in Definition 1.1 can be transformed to insert any desired number of 0's on both the left and the right of the selected bit. We show that any class

$\mathcal{C} \subseteq$ AmpMP which is closed under both $\leq^{\mathrm{P}}_{\mathrm{ctt}}$ and $\leq^{\mathrm{P}}_{\mathrm{dtt}}$ (that is, under polynomial-time conjunctive and disjunctive truth-table reducibilities, respectively) is low for both MP and for AmpMP itself. It follows that if MP = AmpMP, or even if $\mathrm{C_{=}P} \subseteq$ AmpMP, then the entire *counting hierarchy* [Wa 86] collapses to MP. We had hoped to be able to show lowness without any extra closure conditions on $\mathcal{C}$. However, very recently Köbler and Toda [KöTo 93] showed that if the class AmpMP defined in this paper is low for MP, then the counting hierarchy likewise collapses. We return to this question and give other open problems about MP in our concluding Section 7.

While it is immediate that $\oplus\mathrm{P} \subseteq$ MP, it is not obvious *à priori* that $\mathrm{Mod_3P} \subseteq$ MP, since in order to decide whether a number written in base 2 is congruent to 0 modulo 3, one needs the information of all of its bits. Nevertheless, in Section 5 we show that for all $k \geq 2$, all languages in $\mathrm{Mod}_k\mathrm{P}$ (see [CaHe 89b, He 90, BeGi 92]) are low for MP. Sections 5 and 6 together present our main application, which is an improvement on previous simulations of the circuit class ACC (originally defined by Barrington [Ba 89]).

Some background and explanation of how our work builds on prior techniques is in order. It is by now well known that relationships among Turing machine classes, both with and without relativization, correspond closely to circuit simulations or circuit-class separations. Toda proved his first theorem, $\mathrm{PH} \subseteq \mathrm{BP}\cdot\oplus\mathrm{P}$, using techniques introduced by Valiant and Vazirani [ValVaz 86]. The circuit analogue of this result was proved by Allender [Al 89], using polynomial methods introduced by Razborov [Raz 87] and Smolensky [Smo 87]. It states that any AC^0 predicate is computed with high probability by a family of circuits of *quasi-polynomial* size (i.e., size $2^{(\log n)^{O(1)}}$) which consist of a parity gate connected to AND gates which are *small* (i.e., have polylog fan-in). Allender and Hertrampf [AlHe 90] subsequently applied the Valiant-Vazirani technique to obtain a uniform version of Allender's result.

Yao [Yao 90] then used these techniques to obtain the first non-trivial upper bound for ACC. By combining the Valiant-Vazirani method with improved versions of Toda's "amplifying polynomials," he showed that every language in ACC is recognized by a family of depth-2 probabilistic circuits of quasipolynomial size with a symmetric gate at the root and small AND gates at the leaves. Then Beigel and Tarui [BeTa 91] simplified Yao's proof and strengthened the result in two respects: (1) the circuits given by Yao can be made deterministic without increasing their size, and (2) the simulation applies not only to ACC circuits, but also to circuits consisting of one symmetric gate (at the root) connected to ACC subcircuits. This is the circuit analogue of the result that PH and all the $\mathrm{Mod}_k\mathrm{P}$ classes are low for $\mathrm{P}^{\#\mathrm{P}[1]}$, and shows that this lowness relationship holds relative to all oracles.

In the results of both [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 seems very hard to work directly with depth-2 circuits of the type given in [Yao 90] or [BeTa 91] in proving lower bounds, since all that can be said about the gates at the root is that they belong to an infinite subfamily of the symmetric functions. Our improvement is that such circuits can be restricted to have a symmetric gate of type MidBit at the root and still have the power of ACC. A MidBit gate over w inputs $x_1, \ldots, x_w$ is a gate which outputs the value of the $\lfloor \log_2(w)/2 \rfloor^{\text{th}}$ bit in the binary representation of the number $\sum_{i=1}^{w} x_i$. Let the term MidBit^+ refer to families of depth-2 deterministic circuits of quasipolynomial size with a MidBit gate at the root and small AND-gates at the leaves (see Definition 6.2). Our result is that ACC circuits can be simulated by MidBit^+ circuits. Furthermore, as follows from our lowness results, even a circuit consisting of a Midbit of ACC circuits can be simulated by MidBit^+ circuits. Much of the above can be proved by applying the ideas and techniques of [BeTa 91]. Our main technical contribution regarding the ACC problem is a means of converting representations involving counting modulo some prime p into "Midbit" representations in binary notation. 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 (Lemma 5.1). By choosing an appropriate Toda polynomial, the bit can be "isolated" from the rest of the string, and this leads to our Theorem 5.2, and in circuit form, Theorem 6.3.

Yao [Yao 90] conjectured that there are languages in TC_0 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, nominally 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 $|n| = \lceil \log_2(n+1) \rceil$. For a set A, $|A|$ denotes its cardinality. The characteristic function of a set A is denoted by χ_A. The set $\{0, 1, 2, \ldots\}$ of non-negative integers is denoted by $\mathcal{N}$. We fix some standard means of injectively encoding sequences $(x_1, \ldots x_k)$ of strings, where $k > 0$ is arbitrary, by single strings $\langle x_1, \ldots, x_k \rangle$, such that $|\langle x_1, \ldots, x_k \rangle|$ depends only on $\sum_{i=1}^{k} |x_i|$, and both the encoding and decoding are computable in polynomial time. Where intent is clear we write $f(x_1, \ldots, x_k)$ or $\chi_A(x, y)$ in place of $f(\langle x_1, \ldots, x_k \rangle)$ or $\chi_A(\langle x, y \rangle)$.

We write FP for the class of deterministic polynomial-time computable functions. To every polynomial-time bounded nondeterministic Turing machine (NTM) we associate the *counting function* $\#acc_N(x)$, defined to be the number of accepting computations which N has on input x. The class of such functions is denoted by #P. A language L belongs to the class PP if there is an NTM N with polynomial time bound p such that for all x, $x \in L$ iff $\#acc_N(x) > 2^{p(|x|)-1}$. For any natural number $k \geq 2$, L belongs to the class $\mathrm{Mod}_k\mathrm{P}$ if there exist N and p such that for all x, $x \in L$ iff $\#acc_N(x) \not\equiv 0 \pmod{k}$. With $k = 2$ we have $x \in L$ iff $\#acc_M(x)$ is odd, and this class is called $\oplus$P ("parity-P").

It is well-known that #P is closed not only under addition and multiplication but also under summation of exponentially many #P functions and multiplication of polynomially many #P functions. That is, if $f(\cdot,\cdot)$ is in #P and p is a polynomial, then the functions $\sum_{y,|y|\leq p(|x|)} f(x,y)$ and $\prod_{m\leq p(|x|)} f(x,0^m)$ are also in #P.

An oracle machine M may have a function f instead of a language as its oracle. M is *nonadaptive* if for every computation path leading up to some query z, the set of possible next queries does not depend on the answer $f(z)$. Otherwise M is *adaptive*. The class of languages (respectively, functions) computed by such machines M with an oracle from some class $\mathcal{C}$ is denoted by $\mathrm{P}^{\mathcal{C}}_{tt}$ (respectively, $\mathrm{FP}^{\mathcal{C}}_{tt}$). When M is deterministic and nonadaptive, and accepts iff at least one of its queries is answered "yes" (respectively, iff all of its queries are answered "yes"), M is said to compute a polynomial-time *disjunctive* (resp. *conjunctive*) truth-table reduction, and the corresponding reducibility relations are denoted by $\leq^{\mathrm{P}}_{\mathrm{dtt}}$ and $\leq^{\mathrm{P}}_{\mathrm{ctt}}$. Finally, given $k > 0$, a relativizable language class $\mathcal{C}$, and a class $\mathcal{D}$ of either languages or functions, $\mathcal{C}^{\mathcal{D}[k]}$ is the class of all languages in $\mathcal{C}^{\mathcal{D}}$ witnessed by a machine of type $\mathcal{C}$ which asks at most k queries on every computation path; these queries may be adaptive. The class $\mathcal{D}$ is *low* for $\mathcal{C}$ if the class $\mathcal{D}$, when used as an oracle for $\mathcal{C}$, does not increase the power of $\mathcal{C}$, that is, $\mathcal{C}^{\mathcal{D}} = \mathcal{C}$. Further information on these concepts may be found in [BaDiGa 87, Schö 86].

3 Bits of #P Functions and Counting Classes

If a given pair of functions $f \in \#\mathrm{P}$ and $g \in \mathrm{FP}$ satisfy Definition 1.1 for some language L, then we say $L \in \mathrm{MP}$ *via* f *and* g. For visual convenience we also use a second representation which involves a *bounding polynomial* for f, meaning a polynomial p such that for all x, $f(x) < 2^{p(|x|)}$. Then $f(x)$ is written as a binary string of length exactly $p(|x|)$, using leading 0's if necessary, with the least significant bit numbered 0 and written rightmost, and the most significant bit numbered $p(|x|) - 1$. As noted in Section 1, PP consists of the languages decided by the leftmost bit under this representation, and $\oplus$P, those decided by the rightmost bit. The name MP comes from our first result, which shows that all languages $L \in \mathrm{MP}$ have MP-representations by which membership of x in L is decided by the middle bit.

Proposition 3.1 *Let $L \in$ 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. Furthermore, the index of the middle bit is given by a polynomial in $|x|$ alone.*

Proof. Take $f_0 \in$ #P and $g \in$ FP such that $L \in$ MP via f_0 and g, and let p be a bounding polynomial for f_0. Let p be a bounding polynomial for f. Then we may also suppose without loss of generality that for all x, $g(x) \leq p(|x|)$. Then the function f defined by $f(x) = 2^{2p(|x|)} + 2^{p(|x|)-g(x)} f_0(x)$ belongs to #P. Since for all x, $f_0(x) < 2^{p(|x|)}$, we have $f(x) < 2^{2p(|x|)+1}$, and so $|f(x)| = 2p(|x|)+1$. (Thus no leading 0's are needed.) The bit of $f_0(x)$ originally selected by $g(x)$ is now in position $p(|x|)$, which is the middle bit of $f(x)$. □

We note that in this and later proofs, in place of asking for one bit of the constructed #P function f evaluated on the input x, one can ask for the same bit of the #P-complete function #SAT evaluated at some other argument $y = h(x)$ where h is a polynomial-time computable polynomially honest function. The next result implies that, modulo the belief that PP and $\oplus$P are proper subclasses of $\mathrm{P}^{\#\mathrm{P}[1]}$, the bits at distance $O(\log n)$ (where $n = |x|$) from either the left or right end of #SAT are easier than the middle bit (this follows easily from known results). On the other hand, it is possible to push the decision bit of any MP language quite close to either end of the binary string representing $f(x)$, in comparison with the length of $f(x)$. That is, for any $L \in$MP, for any $\epsilon > 0$, there is a #P function f such that the deciding bit for $x \in L$ is as close as $|f(x)|^\epsilon$ to either end of the binary representation of $f(x)$. In this sense, a wide range of bits around the middle are equally as powerful as the middle bit.

Proposition 3.2

(a) *Let L be in MP via a function $f \in$ #P and a bit selection function $g \in$ FP. If $g(x) = O(\log(|x|))$, then $L \in \oplus$P, while if $|f(x)| - g(x) = O(\log(|x|))$, then $L \in$ PP.*

(b) *Let $L \in$ MP. Let $0 < \epsilon < 1$, and let g be a polynomial-time computable function which takes a string x and a number m as arguments, such that always $m^\epsilon < g(x,m) < m - m^\epsilon$. Then there exists a #P function f' with bounding polynomial p' such that upon taking $g'(x) = g(x, p'(|x|))$ as bit-selection function, $L \in$ MP via f' and g'.*

Proof. (a) The statement for $\oplus$P is immediate from [BeGi 92]. The statement for PP follows quickly from the result $\mathrm{P}^{\mathrm{PP}[O(\log n)]} = \mathrm{PP}$ in [BeReSp 91]: Let c be a constant such that $|f(x)| - g(x) \leq c\log_2(|x|)$ for all $x \in \Sigma^*$. Then $f(x) < 2^{g(x)+c\log_2(|x|)}$, and the bits at the positions $g(x) + c\log_2(|x|) - 1, \ldots, g(x)$ in the binary representation of $f(x)$ can be computed in polynomial time by binary search asking $c\log_2(|x|)$ many queries to the PP oracle set $\{\langle x, i\rangle \mid f(x) \geq i\}$.

(b) Let f and the middle-bit selector p be as in Proposition 3.1, and let $p'(n) = (p(n)+2)^{\lceil 1/\epsilon \rceil}$. With g' as given in the statement of this proposition, define $f'(x) = 2^{p'(n)-1} + 2^{g'(x)-p(n)} f(x)$, where $n = |x|$. Then bit number $p(n)$ of $f(x)$ is the same as bit number $g'(x)$ of $f'(x)$. Because g' is in FP and $g'(x) > p'(n)^\epsilon \geq p(n)$, f' is in #P. Hence f' and g' form an MP representation for L. (Note that p' also serves as a bounding polynomial for f'.) □

With $m = p'(n)$, the selected bit may be as low as m^ϵ or as high as $m - m^\epsilon$, depending only on g. We do not have any quantitative results on the difficulty of the bits in positions between $O(\log n)$ and m^ϵ. Next we collect some basic structural properties of MP.

Proposition 3.3

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

(b) MP *has complete sets under* $\leq_{\mathrm{m}}^{\mathrm{P}}$.

(c) MP *is closed downward under* $\leq_{\mathrm{m}}^{\mathrm{P}}$, *and is closed under complements and join.*

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

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

Proof. (a) The inclusion $\mathrm{PP}^{\oplus \mathrm{P}} \subseteq \mathrm{MP}$ follows from inspection of Toda's proof [Tod 91] that $\mathrm{PP}^{\oplus \mathrm{P}} \subseteq \mathrm{P}^{\#\mathrm{P}}$, as discussed in Section 1. The inclusion $\mathrm{MP} \subseteq \mathrm{P}^{\#\mathrm{P}[1]}$ is obvious.

(b) 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 $x\}$ is easily seen to be complete for MP under $\leq_{\mathrm{m}}^{\mathrm{P}}$. Complete languages based on counting satisfying assignments to Boolean formulas can also be defined.

(c) Let B be in MP via some $f_B \in \#\mathrm{P}$ and $g \in \mathrm{FP}$, and suppose that $A \leq_{\mathrm{m}}^{\mathrm{P}} B$ via some FP function h. Then the function $f_A = f_B \circ h$ is in #P, and the function $g \circ h$ is in FP. For all $x \in \Sigma^*$, $x \in A$ if and only if there is a 1 at position $g(h(x))$ in the binary representation of $f(h(x))$, so $A \in \mathrm{MP}$.

For complements, consider the function $f'(x) = f(x) + 2^{g(x)}$, which belongs to #P and flips the bit at position $g(x)$ in the binary representation of $f(x)$. Given $A, B \in \mathrm{MP}$ with #P functions f_A and f_B respectively, the join $0A \cup 1B$ is represented by the #P function f defined by $f(0x) = f_A(x)$, $f(1x) = f_B(x)$.

(d) This holds for any class which contains P and has the closure properties in (c).

(e) Likewise, since MP contains P and is closed under $\leq_{\mathrm{m}}^{\mathrm{P}}$, it follows (see [KöScWa 87]) that $\bigcup_{k \geq 1} \mathrm{P}^{\mathrm{MP}[k]}$ coincides with the Boolean closure of MP, which equals MP iff MP is closed under intersection. □

Finally we compare the polynomial-time Turing and truth-table closures of MP.

Proposition 3.4

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

(b) $\mathrm{FP}^{\mathrm{MP}}_{tt} = \mathrm{FP}^{\#\mathrm{P}}_{tt} = \mathrm{FP}^{\#\mathrm{P}[1]}$.

(c) $\mathrm{P}^{\mathrm{MP}}_{tt} = \mathrm{P}^{\#\mathrm{P}[1]}$.

Proof. Part (a) is obvious, and (c) follows immediately from (b). The first equality in (b) follows from the fact that the value of a #P function can be computed in polynomial time by asking nonadaptive queries to an MP oracle. The second equality, namely that a round of nonadaptive queries to a #P oracle function can be simulated by one query to a #P oracle function, is shown in [CaHe 89a]. □

Fortnow and Reingold [FoRe 91] proved that PP is closed downward under polynomial-time truth-table reductions ($\leq^{\mathrm{P}}_{\mathrm{tt}}$). Since their result relativizes, the class $\mathrm{PP}^{\oplus\mathrm{P}}$ is also closed downward under $\leq^{\mathrm{P}}_{\mathrm{tt}}$. Hence if $\mathrm{MP} = \mathrm{PP}^{\oplus\mathrm{P}}$, then both classes equal $\mathrm{P}^{\#\mathrm{P}[1]}$. The open problems of whether $\mathrm{MP} = \mathrm{PP}^{\oplus\mathrm{P}}$ or whether MP is closed under intersection are discussed at the end of the paper.

4 The Class AmpMP

In this section we introduce a subclass AmpMP of MP that will be very useful in obtaining lowness results for a variety of complexity classes including $\oplus$P, BPP, PH, and $\mathrm{Mod}_k\mathrm{P}$, $k \geq 2$. Toda's proof, which as mentioned in Proposition 3.3(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, for every polynomial r, there is a #P function f such that for all x, not only does the middle bit of $f(x)$ equal 1 iff $x \in L$, but also the $r(|x|)$ bits to the left of this bit are always 0. We call this property "amplification on the left of the decision bit." As noted below, familiar probability-amplification methods for languages L in BPP yield #P functions which allow 0's to be inserted to the *right* of the bit $\chi_L(x)$. When the construction implicit in the whole of Toda's paper is carried out on a language L in the polynomial hierarchy (more precisely, to L in $\mathrm{BPP}^{\oplus\mathrm{P}}$), any desired number m of 0's (bounded by a polynomial in $|x|$) can be inserted on *both* the left and the right of the decision bit. This motivated us to study the class of all languages with this amplification property, and to call the class AmpMP.

Definition 4.1 *A language L is in* AmpMP *if there are functions $f \in \#\mathrm{P}$ and $u \in \mathrm{FP}$ such that for all $x \in \Sigma^*$ and $m > 0$ there exist nonnegative integers a and b with $b < 2^{u(x,0^m)}$ such that*

$$f(x,0^m) = a2^{u(x,0^m)+2m+1} + \chi_L(x)2^{u(x,0^m)+m} + b.$$

Put another way, L is in AmpMP if there are functions $f \in \#\mathrm{P}$ and $u, v \in \mathrm{FP}$ such that for every $x \in \Sigma^*$ and $m > 0$, the binary representation of $f(x, 0^m)$ is of the form

$$a_{v(x,0^m)-1} \dots a_0 \underbrace{0 \dots 0}_{m \text{ times}} \chi_L(x) \underbrace{0 \dots 0}_{m \text{ times}} b_{u(x,0^m)-1} \dots b_0. \tag{1}$$

We say the functions f, u, v form an AmpMP representation of L. Here and later, it is understood that $a_{v(x,0^m)-1} \dots a_0$ and $b_{u(x,0^m)-1} \dots b_0$ are strings which may depend on x and m; only their lengths and not their actual values matter.

Proposition 4.2 BPP *and* $\oplus$P *are subclasses of* AmpMP.

Proof. Let $L \in \mathrm{BPP}$. The standard method for reducing the error probability to $2^{-(m+1)}$ (see e.g. [Sch 83]) takes the majority result of $O(m)$ trials. Making m a second argument yields a #P function $f(x, 0^m)$ and a polynomial $p(n, m)$ with the following property: Whenever $x \in L$, $f(x)$ in binary begins with 1^{m+1} (or equals $2^{p(n,m)}$), and when $x \notin L$, $f(x)$ begins with 0^{m+1}. Then the function $f'(x, 0^m) = f(x, 0^m) + 2^{p(n,m)-(m+1)}$ belongs to #P and has the decision bit $\chi_L(x)$ in position $p(n, m)$, followed by 0^m to its right. Since $\chi_L(x)$ is the leading bit, this satisfies Definition 4.1 with $a = 0$.

Given $L \in \oplus\mathrm{P}$, Toda's amplifying polynomials yield a #P function f such that for all x and m, if $x \in L$ then $f(x, 0^m) \equiv 0 \pmod{2^m}$, and if $x \notin L$, then $f(x, 0^m) \equiv -1 \pmod{2^m}$ (see [Tod 91]). Defining $f'(x, 0^m) = 2^m(f(x, 0^{m+1}) + 1)$ then places 0^m on the right as well as the left of the bit $\chi_L(x)$. $\square$

It follows by bit-shifting methods similar to those of Proposition 3.1 that every AmpMP language L has a representation as in (1) above, in which u and v are polynomials in m and $|x|$. In fact, one can arrange that $v = u$, so that the decision bit $\chi_L(x)$ is in the middle. However, we prefer to keep v and u separate.

Lemma 4.3

(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, and let $u, v \in \mathrm{FP}$ and $f \in \#\mathrm{P}$ provide the AmpMP representation as in (1). Here and below we write v and u as short for $v(x, 0^m)$ and $u(x, 0^m)$. Consider the function $f'(x, 0^m)$ whose value in binary is

$$a_{v-1} \dots a_0 \underbrace{1 \dots 1}_{m \text{ times}} \chi_L(x) \underbrace{1 \dots 1}_{m \text{ times}} b_{u-1} \dots b_0.$$

Since this equals $f(x, 0^m)$ plus some powers of 2 determined by m and u alone, f' is in #P. Let p' be a polynomial which bounds the running time of some nondeterministic Turing machine N whose counting function $\#acc_N$ is the same as f'. Then, by interchanging accept and reject states in N, the function $f''(x, 0^m) = 2^{p'(|\langle x,0^m\rangle|)+1} - 1 - f'(x, 0^m)$ is also in #P. In $f''(x, 0^m)$, the complement of the bit $\chi_L(x)$ is flanked by m 0's.

(b) Let L_1, L_2 be two sets in AmpMP, with respective representations f_1, u_1, v_1 and f_2, u_2, v_2. Then $f_1(x, 0^m)$ has the form

$$a_{v_1-1} \dots a_0 \underbrace{0 \dots 0}_{m \text{ times}} \chi_{L_1}(x) \underbrace{0 \dots 0}_{m \text{ times}} b_{u_1-1} \dots b_0.$$

Let $h(x, 0^m) = v_1(x, 0^m) + 2m + 1 + u_1(x, 0^m)$, which bounds the length of $f_1(x, 0^m)$. Let v'_2 abbreviate $v_2(x, 0^{h(x,0^m)})$, and let u'_2 abbreviate $u_2(x, 0^{h(x,0^m)})$. Then $f_2(x, 0^{h(x,0^m)})$ has the form

$$a'_{v'_2-1} \dots a'_0 \quad \underbrace{0 \dots 0}_{h(x,0^m) \text{ times}} \quad \chi_{L_2}(x) \quad \underbrace{0 \dots 0}_{h(x,0^m) \text{ times}} \quad b'_{u'_2-1} \dots b'_0,$$

for some strings a' and b'. The function $f_3(x, 0^m) = f_1(x, 0^m) \cdot f_2(x, 0^{h(x,0^m)})$ also belongs to #P. Because of all the 0's around the bit $\chi_{L_2}(x)$, the value $\chi_{L_2}(x) \cdot f_1(x, 0^m)$ appears as a substring of $f_3(x, 0^m)$, and in particular, the decision bit $\chi_{L_1}(x) \cdot \chi_{L_2}(x)$ for $L_1 \cap L_2$ appears at position $u_1 + m + u'_2 + h(x, 0^m)$.

(c) Since by (a) and (b) AmpMP is closed under Boolean operations, it suffices to show that AmpMP is closed under many-one reductions. Suppose $A \leq_{\mathrm{m}}^{\mathrm{p}} L$ where $L \in$ AmpMP. Let f, u, and v provide the representation as in (1) for L, and let t be the polynomial-time computable function used in the reduction. Define the function $f'(x, 0^m) = f(t(x), 0^m)$. The value $f'(x, 0^m)$ has the form

$$a_{v(t(x),0^m)-1} \dots a_0 \underbrace{0 \dots 0}_{m \text{ times}} \chi_L(t(x)) \underbrace{0 \dots 0}_{m \text{ times}} b_{u(t(x),0^m)-1} \dots b_0.$$

Then $\chi_L(t(x)) = \chi_A(x)$, and the functions $u'(x, 0^m) = u(t(x), 0^m)$ and $v'(x, 0^m) = v(t(x), 0^m)$ complete an AmpMP-representation for A. $\square$

The technical key to our lowness results is a lemma which shows that the value of a function which is in #P with a bounded number of queries to AmpMP can be obtained as a substring of the value of a function in #P, with no queries. By (c) above, for any $k > 0$, $\mathrm{P}^{\mathrm{AmpMP}[k]} = \mathrm{AmpMP}$. Hence $\#\mathrm{P}^{\mathrm{AmpMP}[k]} = \#\mathrm{AmpMP}$, where #AmpMP equals the class of functions f such that for some language A in AmpMP and polynomial q, and all $x \in \Sigma^*$,

$$f(x) = \sum_{y \in \{0,1\}^{q(|x|)}} \chi_A(x, y). \tag{2}$$

Lemma 4.4 *For every function $f \in$ #AmpMP there is a function $f' \in$ #P such that for all $x \in \Sigma^*$ and $m \geq 0$, the binary representation of $f'(x, 0^m)$ has the form*

$$a_{v-1} \dots a_0 \underbrace{0 \dots 0}_{m \text{ times}} f(x) \underbrace{0 \dots 0}_{m \text{ times}} b_{u-1} \dots b_0.$$

As usual we suppose that $f(x)$ is written as a string of length exactly $p(|x|)$, where p is some bounding polynomial. Then the above states that $\lfloor f'(x, 0^m)/2^{u+m} \rfloor$ is congruent to $f(x)$ modulo $2^{p(|x|)+m}$.

Proof. Given f, take $A \in$ AmpMP and the polynomial q from (2). By the remarks following Proposition 4.2, there are polynomials u, v and a #P function f_A such that for all pairs $\langle x, y \rangle \in \Sigma^*$ and $k > 0$, the binary representation of $f_A(\langle x, y \rangle, 0^k)$ has the form

$$a_{v(n',k)-1} \dots a_0 \underbrace{0 \dots 0}_{k \text{ times}} \chi_A(x, y) \underbrace{0 \dots 0}_{k \text{ times}} b_{u(n',k)-1} \dots b_0, \tag{3}$$

where $n' = |\langle x, y \rangle|$. By our particular choice of tupling function in Section 2, and since $|y| = q(|x|)$, $|\langle x, y \rangle|$ depends only on $|x|$. Thus the position of the bit $\chi_A(x, y)$ is the same for all y. Now given $x \in \Sigma^n$ and $m \geq 0$, take $k = m + q(n) + 1$, and define

$$f'(x, 0^m) = \sum_{y \in \{0,1\}^{q(n)}} f_A(\langle x, y \rangle, 0^k).$$

Then $f' \in$ #P. The binary representation of $f'(x, 0^m)$ is obtained by summing the representations in (3), and by Eq. (2) and the choice of k, $f(x)$ appears as a substring of length exactly $q(n) + 1$ flanked by m 0's on both sides. (Also note u and v are polynomials in n and m.) $\square$

Corollary 4.5

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

(b) $\bigcup_{k>1} \mathrm{AmpMP}^{\mathrm{AmpMP}[k]} = \mathrm{AmpMP}$.

Proof. Given $L \in \mathrm{MP}^{\mathrm{AmpMP}[k]}$, L has a representing function $f \in \#\mathrm{P}^{\mathrm{AmpMP}[k]}$. Then the function f' in Lemma 4.4 provides an MP-representation for L—here m is immaterial and can be fixed to 0. For (b), if f is an AmpMP-representation, then so is f'. $\square$

The limitation of this corollary is that it only applies to bounded truth-table reductions to AmpMP. To apply it for full Turing reductions, we seek technical conditions on subclasses $\mathcal{C}$ of AmpMP under which any number of queries to $\mathcal{C}$ made by a nondeterministic machine can be replaced by two queries.

Theorem 4.6 *Let $\mathcal{C}$ be a subclass of* AmpMP *which is closed downward under both* $\leq^{\mathrm{P}}_{\mathrm{ctt}}$ *and* $\leq^{\mathrm{P}}_{\mathrm{dtt}}$. *Then $\mathcal{C}$ is low both for* MP *and for* AmpMP.

Proof. Let $L \in \mathrm{MP}^A$ where $A \in \mathcal{C}$. Let $B = \{0\langle x_1, \ldots, x_k\rangle :$ each x_i belongs to $A\}$, and let $C = \{1\langle x_1, \ldots, x_k\rangle :$ some x_i belongs to $A\}$. By the closure properties of $\mathcal{C}$, B and C belong to $\mathcal{C}$, and by the closure properties of AmpMP, the language $D = B \cup C$ belongs to AmpMP. Let N be a nondeterministic oracle TM such that the counting function $\#acc_{N^A}(x)$ gives an MP representation for L. By a standard trick, replace N by a nondeterministic OTM N' which on any input x guesses a computation path of $N(x)$ and also guesses the oracle answers along the path. Let $y_1, \ldots, y_k$ denote the query strings whose answers were guessed as "yes" along this path, and $z_1, \ldots, z_l$, those guessed "no." Then this path by N' accepts iff $0\langle y_1, \ldots, y_k\rangle \in B$, $1\langle z_1, \ldots, z_l\rangle \notin C$, and the path guessed by N accepts. N' need make only two queries to its AmpMP oracle D, and since this trick does not change the number of accepting computations, Corollary 4.5 implies that $L \in \mathrm{MP}$. The case $\mathrm{AmpMP}^A = \mathrm{AmpMP}$ is similar. □

Since BPP and $\oplus$P are closed under polynomial-time Turing reductions ($\leq^{\mathrm{P}}_{\mathrm{T}}$) [Ko 82, PaZa 83], it follows from Proposition 4.2 and Theorem 4.6 that they are low for both MP and AmpMP. We can quickly show the same lowness property for the class $\mathrm{BPP}^{\oplus\mathrm{P}}$ shown in [Tod 91] to contain the polynomial hierarchy (PH).

Proposition 4.7 $\mathrm{BPP}^{\oplus\mathrm{P}}$ *and* PH *are low for* MP *and for* AmpMP.

Proof. Proposition 4.2 relativizes to show that for any oracle set A, $\mathrm{BPP}^A \subseteq \mathrm{AmpMP}^A$. Hence $\mathrm{BPP}^{\oplus\mathrm{P}} \subseteq \mathrm{AmpMP}^{\oplus\mathrm{P}}$. Since $\oplus$P is low for AmpMP, it follows that $\mathrm{BPP}^{\oplus\mathrm{P}} \subseteq$ AmpMP. Since $\mathrm{BPP}^{\oplus\mathrm{P}}$ is closed downward under $\leq^{\mathrm{P}}_{\mathrm{T}}$, it too is low for AmpMP. Lowness for MP is similar, and the conclusions for PH follow since $\mathrm{PH} \subseteq \mathrm{BPP}^{\oplus\mathrm{P}}$. □

By careful examination of the proof in [Tod 91], we find that the lowness of $\mathrm{BPP}^{\oplus\mathrm{P}}$ for MP is a consequence of the more general result that any $\#\mathrm{P}^{\mathrm{BPP}^{\oplus\mathrm{P}}}$ function reduces to the "middle bits" of some #P function (see [ToWa 92] for related results about $\#\mathrm{P}^{\mathrm{PH}}$). Furthermore, the bits can be isolated any distance m from the left part of the string, independent of the input length $|x|$.

Theorem 4.8 *For every function f in $\#\mathrm{P}^{\mathrm{BPP}^{\oplus\mathrm{P}}}$ there exist a function $h \in \#\mathrm{P}$ and a polynomial p such that for all x and m,*

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

Proof. Let f be in $\#\mathrm{P}^{\mathrm{BPP}^{\oplus\mathrm{P}}}$. Since $\mathrm{P}^{\mathrm{BPP}^{\oplus\mathrm{P}}} = \mathrm{BPP}^{\oplus\mathrm{P}}$, there exist a language $A \in \mathrm{BPP}^{\oplus\mathrm{P}}$ and a polynomial q such that for all $x \in \Sigma^*$,

$$f(x) = \sum_{y \in \{0,1\}^{q(|x|)}} \chi_A(x, y).$$

By Proposition 4.2 for BPP relativized to $\oplus$P, we obtain a function $f' \in \#\mathrm{P}^{\oplus\mathrm{P}}$ and a polynomial u such that for all x and y, the binary representation of $f'(\langle x, y\rangle, 0^m)$ has the form

$$\chi_A(x, y)\,\underbrace{0\ldots0}_{m\ \text{times}}\,b_{u(|\langle x,y\rangle|)-1}\cdots b_0.$$

For all x, with $n = |x|$, define

$$g(x) = \sum_{y \in \{0,1\}^{q(n)}} f'(\langle x, y\rangle, 0^{q(n)}).$$

Because of the choice $m = q(|x|)$, the sum of the $b_{u-1}\ldots b_0$ terms does not spill any carries into the sum of the $\chi_A(x, y)$ terms. Since $|\langle x, y\rangle|$ depends only on n for $y \in \Sigma^{q(n)}$, there is a polynomial $p(n)$ such that for all x, $g(x)$ has the form

$$g(x) = f(x)b'_{p(|x|)-1}\cdots b'_0.$$

Now g also belongs to $\#\mathrm{P}^{\oplus\mathrm{P}}$. Since $\mathrm{P}^{\oplus\mathrm{P}} = \oplus\mathrm{P}$, there exists a language $B \in \oplus\mathrm{P}$ and a polynomial r such that for all x,

$$g(x) = \sum_{z \in \{0,1\}^{r(|x|)}} \chi_B(x, z).$$

Next, by using Toda's amplifying polynomials as in the proof of Proposition 4.2, we obtain a function $g' \in \#\mathrm{P}$ such that for all $x, z \in \Sigma^*$ and $m \in \mathcal{N}$,

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

Finally define

$$h(x, 0^m) = \sum_{z \in \{0,1\}^{r(|x|)}} g'(x, z, 0^m).$$

Then $h \in \#\mathrm{P}$, and for all x and m,

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

The conclusion follows on noting that $\lfloor g(x)/2^{p(|x|)} \rfloor = f(x)$. $\square$

We return to this in connection with open problems about AmpMP in Section 7. Before presenting our new idea which gives analogous lowness results for the classes $\mathrm{Mod}_k\mathrm{P}$, $k \geq 3$, we observe one more consequence of the results in this section.

Proposition 4.9 *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, GuNaWe 90, Gr 93, BeChOg 93]) it follows from Theorem 4.6 that $\mathrm{C_{=}P}$ would be low for MP. However, from the result of [Tor 88] that $\mathrm{PP^{PP}} = \mathrm{PP^{C_{=}P}}$, this would give $\mathrm{PP^{PP}} \subseteq \mathrm{MP^{C_{=}P}} = \mathrm{MP}$, implying that the entire counting hierarchy collapses to MP. $\square$

5 Lowness of Mod Classes for the Class MP

In this section we show that for any k, $\mathrm{Mod}_k\mathrm{P}$ 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* $k > 1$*, let* $f \in \#\mathrm{P}$ *with bounding polynomial* s*, and let FP functions* q *and* r *be given such that for all* z*,* $2^{q(z)}$ *and* $k^{r(z)}$ *are at most* $2^{s(|z|)}$*. For all* z *write* $a(z) = \lfloor f(z)/k^{r(z)} \rfloor$ *and* $b(z) = f(z) \bmod k^{r(z)}$*. Suppose that for all* z*,* $b(z) \leq k^{r(z)}/2^{q(z)+1}$*. Then there exist a #P function* h*, an FP function* u*, and a non-negative integer-valued function* a' *such that for all* z*,*

$$h(z) = a'(z)a(z)2^{u(z)+q(z)} + b(z)2^{u(z)} + c(z), \quad \textit{where} \quad c(z) < 2^{u(z)}. \tag{4}$$

Proof. We have that for all z, $f(z) = a(z)k^{r(z)} + b(z)$, where $b(z) \leq k^{r(z)}/2^{q(z)+1}$. We first claim that we can find $g \in \#\mathrm{P}$, a polynomial u, and a function $b' : \Sigma^* \longrightarrow \mathcal{N}$ such that for all z,

$$g(z) = a(z)2^{u(z)} + b'(z), \quad \text{where} \quad b'(z) < 2^{u(z)-q(z)}.$$

For all $z \in \Sigma^*$, let $u(z) = q(z) + s(z) + 1$ and $g(z) = f(z)\lceil 2^{u(z)}/k^{r(z)} \rceil$. The quantity $\lceil 2^{u(z)}/k^{r(z)} \rceil$ is polynomial-time computable, because the functions $u(z)$ and $r(z)$ are bounded by a polynomial in $|z|$. Hence g is in #P.

Clearly $g(z) \geq a(z)2^{u(z)}$. Then

$$\begin{aligned}
g(z) - a(z)2^{u(z)} &\leq f(z)\left(1 + \frac{2^{u(z)}}{k^{r(z)}}\right) - a(z)2^{u(z)} \\
&= f(z) + (a(z)k^{r(z)} + b(z))\frac{2^{u(z)}}{k^{r(z)}} - a(z)2^{u(z)} \\
&= f(z) + b(z)\frac{2^{u(z)}}{k^{r(z)}} \\
&< 2^{s(z)} + 2^{u(z)-q(z)-1} = 2^{u(z)-q(z)}.
\end{aligned}$$

The last line follows by $f(z) < 2^{s(z)}$, $b(z) \leq k^{r(z)}/2^{q(z)+1}$, and the definition of u. This proves the claim. Now define

$$h(z) = f(z)2^{u(z)} + g(z)i(z),$$

where $i(z)$ is the unique integer which satisfies $0 \leq i(z) < 2^{q(z)}$ and $i(z) \equiv -k^{r(z)} \pmod{2^{q(z)}}$. Then it follows that

$$\begin{aligned} h(z) &= a(z)k^{r(z)}2^{u(z)} + b(z)2^{u(z)} + a(z)2^{u(z)}i(z) + b'(z)i(z) \\ &= a(z)2^{u(z)}[k^{r(z)} + i(z)] + b(z)2^{u(z)} + b'(z)i(z), \end{aligned}$$

where $b'(z)i(z) < 2^{u(z)}$ and $k^{r(z)} + i(z) \equiv 0 \pmod{2^{q(z)}}$. □

Theorem 5.2 *For every prime k,* $\mathrm{Mod}_k\mathrm{P} \subseteq \mathrm{AmpMP}$.

Proof. Let A be a set in $\mathrm{Mod}_k\mathrm{P}$ and let r be the FP function $r(x, 0^m) = 2m + 2$, so that $k^{r(x,0^m)} \geq 2^{2m+2}$. Since k is prime, we can adapt results from Toda [Tod 91] and Beigel and Gill [BeGi 92] to obtain a function $f_A \in \#\mathrm{P}$ such that for all x and m,

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

Now let $f(x, 0^m) = f_A(x, 0^{r(x,0^m)}) \cdot 2^m$. Then $f \in \#\mathrm{P}$, and there is a function $a_0 : \Sigma^* \to \mathcal{N}$ such that for all x and m,

$$f(x, 0^m) = a_0(x, 0^m)2^m k^{r(x,0^m)} + \chi_A(x)2^m,$$

where $\chi_A(x)2^m \leq k^{r(x,0^m)}/2^{m+2}$. With reference to the statement of Lemma 5.1 and the quantities $a(z)$ and $b(z)$ in the proof, taking $z = \langle x, 0^m \rangle$, we have $a(z) = a_0(x, 0^m)2^m$, $b(z) = \chi_A(x)2^m$, and $q(z) = m + 1$. Then Lemma 5.1 yields $h \in \#\mathrm{P}$, a polynomial u, and $a', c : \Sigma^* \to \mathcal{N}$ such that for all x and m,

$$h(x, 0^m) = a'(x, 0^m)a_0(x, 0^m)2^m \cdot 2^{u(x,0^m)+m+1} + \chi_A(x)2^{m+u(x,0^m)} + c(x, 0^m),$$

where $c(x, 0^m) < 2^{u(x,0^m)}$. In binary representation, this places m 0's on both the left and the right of the bit $\chi_A(x)$. □

Corollary 5.3 *For any $k \geq 2$,* $\mathrm{Mod}_k\mathrm{P}$ *is low for* MP *and for* AmpMP.

Proof. First suppose k is prime. Then $\mathrm{Mod}_k\mathrm{P}$ is closed under $\leq_{\mathrm{T}}^{\mathrm{P}}$ [BeGi 92]. Hence by Theorems 4.6 and 5.1, $\mathrm{Mod}_k\mathrm{P}$ is low for MP and for AmpMP. Now suppose k is composite;

then one can write $k = p^e k'$ where p is prime, $e \geq 1$, and $\gcd(p, k') = 1$. Then by the representation theorem of Hertrampf [He 90],

$$\mathrm{Mod}_k\mathrm{P} \subseteq \mathrm{Mod}_p\mathrm{P}^{\mathrm{Mod}_{k'}\mathrm{P}}.$$

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

The next statement follows quickly from the above by a proof similar to that of Theorem 4.8.

Corollary 5.4 *For any $k \geq 2$ and every function f in $\#\mathrm{P}^{\mathrm{Mod}_k\mathrm{P}}$ there exist a function $g \in \#\mathrm{P}$ and a polynomial p such that for all x,*

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

As a side remark, let ModPH be the closure of P under the operations $\mathcal{C} \mapsto \mathrm{NP}^{\mathcal{C}}$ and $\mathcal{C} \mapsto \mathrm{Mod}_k\mathrm{P}^{\mathcal{C}}$ for $k \geq 2$. This is intuitively an extension of the polynomial hierarchy by the $\mathrm{Mod}_k\mathrm{P}$ classes, and can be regarded as the polynomial-time analogue of the circuit class ACC.

Corollary 5.5 ModPH *is low for* MP *and for* AmpMP.

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 $\lfloor \log_2(w)/2 \rfloor^{\mathrm{th}}$ 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 $w \leq 2^{r(\log n)}$. 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. This is 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 and Gill [BeGi 92], 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 of a type first constructed by Toda [Tod 91] and successively improved by [Yao 90, BeTa 91]. The *modulus-amplifying* polynomials Q_d have the property that for every $k \geq 1$ and $X \geq 0$,

$$\begin{aligned} X \equiv 0 \pmod{k} \quad &\Rightarrow \quad Q_d(X) \equiv 0 \pmod{k^d}, \\ X \equiv 1 \pmod{k} \quad &\Rightarrow \quad Q_d(X) \equiv 1 \pmod{k^d}. \end{aligned}$$

(The modulus-amplifying polynomials constructed by Beigel and Tarui [BeTa 91] have degree $2d - 1$.) Now it follows that

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

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 $i(n) \equiv -k^{p'} \pmod{2^t}$ and q is a polynomial such that $q \geq s + t + 2$, following the proof of Lemma 5.1. Analogously 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

$$h_n(x) \quad \equiv \quad 2^q b_n(x) + c_n(x) \pmod{2^{q+t}} \tag{5}$$

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 $< 2^{q+t}$. 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 the congruence (5) 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]. A family $\{C_n\}$ of circuits belongs to pure-ACC if there is a fixed m such that for all n, every gate in C_n is a Mod_m gate. This theorem as well as its proof is the circuit analogue of Corollary 5.3.

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 $\mathrm{Mod}_{k_1}, \mathrm{Mod}_{k_2}, ..., \mathrm{Mod}_{k_l}$ 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 91], Allender and Hertrampf [AlHe 90], and the lowness methods of Section 4. It says that circuits consisting of a MidBit gate over ACC subcircuits can be simulated by MidBit$^+$ circuits. The proof is similar to those of Theorems 1 and 2 in [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 $\lfloor \log_2(s)/2 \rfloor^{\text{th}}$ 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 polylog-many random bits, using the techniques of [ValVaz 86] as applied by [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 $t(\log n)$ random 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 $\lfloor \log_2(s)/2 \rfloor + t(\log n)$ of S'. □

7 Conclusion and Open Problems

The class MP is important not only for its role as the implicit upper bound in Toda's proofs [Tod 91], but also for its place at the frontier of counting classes whose relationships are not well understood. We have established several properties of the class which make it a natural and attractive object of study, and used results about it to improve the known upper bounds on the circuit class ACC. The first open problem is whether one can construct an oracle relative to which the inclusions in Proposition 3.3(a) are proper. It is not even known whether there exists an oracle A such that $\mathrm{PP}^{\oplus\mathrm{P}^A}$ is different from PSPACE^A.

A second problem which seems amenable to attack is whether MP is equal to $\mathrm{PP}^{\oplus\mathrm{P}}$. If so, then by Proposition 3.4, both classes are equal to $\mathrm{P}^{\#\mathrm{P}[1]}$. Interestingly, we can entertain intuitive arguments both for and against $\mathrm{MP} = \mathrm{PP}^{\oplus\mathrm{P}}$. Let L be in MP via the #P function f and midbit-selecting polynomial p. On the "for" side, one can seek a probabilistic hashing construction whose object is to divide the number of witnesses by $2^{-p(n)}$, and try to prove a slight correlation between bit $p(n)$ of $f(x)$ being 1 and the reduced number of witnesses being odd. On the negative side, every language $L \in \mathrm{PP}^{\oplus\mathrm{P}}$ has an MP representation which allows 0's to be inserted to the left of the bit $\chi_L(x)$, and it would be noteworthy if every MP language had this property. A sub-problem is whether MP is closed under intersection. The direct attempt to solve this by writing polynomial equations, after the fashion of the proof that PP is closed under intersection [BeReSp 91], 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 all x and y, $p(x, y)$ 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 (with $t(k) = k$) is $p(x, y) = \binom{x}{2^{k-1}}\binom{y}{2^{k-1}}2^{k-1}$. A. Odlyzko and M. Coster [personal communication, 1991] have found solutions with degree and coefficient size that are smaller, but still $2^{\Theta(k)}$. 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 by "lining-up" decision bits as in the proof of Proposition 3.1, it would follow 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 remaining discussion is motivated by the important general problem of comparing the power of computing mod 2 versus computing mod k for $k > 2$. First, we ask whether the class MP remains the same when values $f(x)$ are written in some other prime or composite base, where the acceptance condition may be defined either as the selected bit being a 1, or as its being nonzero. If $\mathrm{MP} = \mathrm{PP}^{\oplus\mathrm{P}}$ then the answer is immediately yes, but unconditionally we have not been able to extend the methods of Section 5 to show this. In view of the strong belief that the classes $\mathrm{Mod}_k\mathrm{P}$ are different for all different values of k, it would not seem surprising if the answer were no.

Second, it follows from the proof of Theorem 4.8 (which is essentially Toda's proof) that languages in $\mathrm{BPP}^{\oplus\mathrm{P}}$ enjoy a property which is somewhat stronger than our definition of AmpMP in Section 4.

Proposition 7.1 *For every language $L \in \mathrm{BPP}^{\oplus\mathrm{P}}$ there are functions $f \in \#\mathrm{P}$ and $u, v \in$ FP such that for all $x \in \Sigma^*$ and $m_1, m_2 \in \mathcal{N}$, $f(x, 0^{m_1}, 0^{m_2})$ has the form*

$$a_{v(x,m_1,m_2)-1} \cdots a_0 \underbrace{0 \ldots 0}_{m_1 \text{ times}} \chi_L(x) \underbrace{0 \ldots 0}_{m_2 \text{ times}} b_{u(x,m_2)-1} \cdots b_0. \tag{6}$$

The point is that $u(x, m_2)$ is independent of m_1. Intuitively speaking, this says that languages $L \in \mathrm{BPP}^{\oplus\mathrm{P}}$ have AmpMP representations in which one can first amplify on the right of the bit $\chi_L(x)$, fix the length of the "garbage term" b, and then amplify on the left to insert as many 0's as desired.

However, we were unable to obtain this stronger amplification property given $L \in \mathrm{Mod}_k\mathrm{P}$, $k \geq 3$ (and k prime). In Theorem 5.2, the trick was to multiply f_A by 2^m to get f, and this makes it hard to separate m into m_1 and m_2. Moreover, the polynomial u which bounds the length of the "garbage term" c depends on a bounding polynomial for f, which in turn depends on the number of 0's to be inserted on the left anyway.

The interest in whether these results can be improved was heightened by recent work of one of the present authors and Toda [KöTo 93]. They define a language L to belong to the class ModP if there are functions $f \in \#\mathrm{P}$ and $g \in \mathrm{FP}$ such that for all x, $g(x) = 0^p$ where p is prime, and $x \in L \iff f(x) \not\equiv 0 \pmod{p}$. Then they prove that ModP $\subseteq$ AmpMP and that $\mathrm{P}_{tt}^{\mathrm{ModP}} = \mathrm{P}^{\#\mathrm{P}[1]}$. Hence if either AmpMP or ModP is low for MP, then the counting hierarchy collapses to MP. This is fairly strong evidence that AmpMP itself is not low for MP, and that Theorem 4.6 cannot be improved much further. However, intuitively speaking, the proof in [KöTo 93] that a given language L in ModP belongs to AmpMP first amplifies on the *left* of the bit $\chi_L(x)$ ("Claim 1" in [KöTo 93]), and *then* on the right ("Claim 2").

We considered the stronger amplification property (6) in early work on this paper, but rejected it because the simpler Definition 4.1 expedited our main results. With (6) we were able to weaken the condition on the class $\mathcal{C}$ in Theorem 4.6 from closure under $\leq_{\mathrm{ctt}}^{\mathrm{P}}$ and $\leq_{\mathrm{dtt}}^{\mathrm{P}}$ to closure under $\leq_{\mathrm{m}}^{\mathrm{P}}$. This still does not achieve our desire for a natural structural condition for a language to be low for MP (or for $\mathrm{P}^{\#\mathrm{P}}$). We leave as our final open problem the question of whether the stronger amplification property does capture lowness for MP, or to the contrary, whether the arguments of [KöTo 93] can be applied to this case as well. This last problem may seem very arcane, but the results of [KöTo 93] show that a slight technical distinction can make a large difference in the power of a class, and we suspect that at least some of the keys to unlocking the secrets of counting classes may be concealed in problems like this one.

Acknowledgments We wish to thank V. Arvind, Jim Royer and Richard Beigel for helpful discussions. We also thank the anonymous referees for perceptive comments on earlier drafts of this paper and for suggestions in improving the exposition.

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.

[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 *Journal of Computer and System Sciences* **38**, (1989), 150-164.

[BeChOg 93] R. BEIGEL, R. CHANG, AND M. OGIWARA, A relationship between difference hierarchies and relativized polynomial hierarchies. In *Mathematical Systems Theory* **26**, (1993) 293-310.

[BeGi 92] R. BEIGEL AND J. GILL. Counting classes: thresholds, parity, mods, and fewness. In *Theoretical Computer Science* **103**, (1992), 3-23.

[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. A journal version is to appear in *Journal of Computer and System Sciences.*

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

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

[CaHe 89b] J. Cai, L.A. Hemachandra, On the power of parity. In *Proceedings 6th Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science 349*, (1989), 229-240.

[FoRe 91] L. Fortnow and N. Reingold, PP is closed under truth-table reductions. In *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 93] F. Green, On the power of deterministic reductions to $C_=P$. In *Mathematical Systems Theory* **26**, (1993), 215-233.

[GuNaWe 90] T. Gundermann, N. Nasser, and G. Wechsung, A survey on counting classes. In *Proceedings of the 5th Annual Conference on Structure in Complexity Theory*, (1990), 140-153.

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

[Ko 82] K. Ko, Some observations on the probabilistic algorithms and NP-hard problems. In *Information Processing Letters* **14**, (1982), 39-43.

[KöScToTo 92] J. Köbler, U. Schöning, J. Torán, and S. Toda, Turing Machines with few accepting computations and low sets for PP. In *Journal of Computer and System Sciences* **44**, (1992), 272-286.

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

[KöTo 93] J. Köbler and S. Toda, On the power of generalized MOD-classes. In *Proceedings of the 8th Structure in Complexity Theory Conference*, (1993), 147-155.

[LLS 75] R. Ladner, N. Lynch, and A. Selman, A comparison of polynomial time reducibilities. In *Theoretical Computer Science* **1**, (1975), 103-123.

[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.

[Raz 87] A. RAZBOROV, Lower bounds for the size of circuits of bounded depth with basis $\{\wedge, \oplus\}$. In *Math. Notes Acad. Sci. USSR* **41(4)**, (1987), 333-338.

[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).

[Smo 87] R. SMOLENSKY, Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings of the 19th ACM Symposium on the Theory of Computation*, (1987), 77-82.

[Tod 91] S. TODA. PP is as hard as the polynomial-time hierarchy. In *SIAM Journal on Computing* **20**, (1991) 865-877.

[ToWa 92] S. TODA AND O. WATANABE, Polynomial time 1-Turing reducibility from #PH to #P. In *Theoretical Computer Science* **100**, (1992), 205-221.

[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 succinct 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.