On Information From #P Functions

Lide Li[*]
University of Chicago

Mitsunori Ogihara[†]
University of Rochester

Kenneth W. Regan[‡]
State University of NY at Buffalo

May 1994

Abstract

This paper studies complexity classes defined by restrictions on access to information about the values of functions in Valiant's class #P. These classes are at the frontier of what is currently known about complexity classes even under relativizations. We show relations among classes defined by access to finitely many or $O(\log n)$ bits of a #P function value. This work continues recent research on bounded query classes and truth-table reductions.

Topics. Computational complexity, complexity classes.
AMS classifications. 68Q15, 68Q10.

1 Introduction

This paper is motivated by important open questions about the structure of counting classes in complexity theory. The class #P consists of the counting functions associated to languages in NP, typified by $f_{SAT}(x)$ = the number of satisfying assignments of the Boolean formula encoded by the string x. Valiant [Val79] proved that both f_{SAT} and the permanent function are complete for #P. In particular, every language in NP can be solved in polynomial time with one query to f_{SAT}; hence in symbols, NP $\subseteq \mathrm{P}^{\#\mathrm{P}[1]}$. Toda [Tod91] answered long-standing questions of which other languages belong to $\mathrm{P}^{\#\mathrm{P}[1]}$ by proving first, that the polynomial hierarchy randomly reduces with bounded error to parity polynomial time (in symbols, PH $\subseteq$ BP[$\oplus$P]), and second, that even the class PP[$\oplus$P] defined with unbounded error is contained in $\mathrm{P}^{\#\mathrm{P}[1]}$. Several researchers [RS92, GKR$^+$92] noted that Toda's second theorem only requires knowing one bit (indeed, the middle bit) of the queried value $f(x)$, and studied the class MP of languages defined for one bit. Current knowledge about major counting classes between P and polynomial space (PSPACE) is summarized by:

$$\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PH} \subseteq \mathrm{BP}[\oplus\mathrm{P}] \subseteq \mathrm{PP}[\oplus\mathrm{P}] \subseteq \mathrm{MP} \subseteq \mathrm{P}^{\#\mathrm{P}[1]} \subseteq \mathrm{P}^{\#\mathrm{P}} \subseteq \mathrm{PSPACE}.$$

[*]Supported in part by the NSF under grant CCR-9253582. E-mail: lilide@cs.uchicago.edu
[†]Supported in part by the NSF under grant CCR-9002292 and the JSPS under grant NSF-INT-9116781/JSPS-ENG-207. E-mail: ogihara@cs.rochester.edu
[‡]Supported in part by the NSF under grant CCR- 9011248. E-mail: regan@cs.buffalo.edu

It is not known whether $P \neq PSPACE$, so none of these classes are known to differ. In the face of such gaps in our knowledge, complexity theorists often next study how classes behave under *relativizations* to certain *oracle* sets. Indeed, there are oracle sets A relative to which the first five classes are all different. However, it is not even known whether there exists an oracle set A that makes $\mathrm{PP}[\oplus\mathrm{P}^A] \neq \mathrm{PSPACE}^A$. Thus the classes $\mathrm{PP}[\oplus\mathrm{P}]$, MP, and $\mathrm{P}^{\#\mathrm{P}[1]}$ represent the frontier beyond which we really know very little about complexity classes, and this is important ground for study. One other fact known is that if $\mathrm{MP} = \mathrm{PP}[\oplus\mathrm{P}]$, then both classes equal $\mathrm{P}^{\#\mathrm{P}[1]}$ as well [GKR+92]. This further draws attention to the region between $\mathrm{PP}[\oplus\mathrm{P}]$ and $\mathrm{P}^{\#\mathrm{P}[1]}$, the subject of this paper.

The issue can be understood in terms of the amount of information in the value $f(x)$ of a #P function f that is required to solve a given problem. In $\mathrm{P}^{\#\mathrm{P}[1]}$, one has access to all of the information; i.e., polynomially many bits. MP is exactly the class of languages solvable with one bit of information in $f(x)$. In probing the difference, we focus on the power of $O(\log n)$ bits of information, in two forms:

(a) Knowing $O(\log n)$ bits of the value $f(x)$ itself.

(b) Knowing the number $wt(f(x))$ of 1's in the binary representation of $f(x)$, which is an $O(\log n)$-bit number.

It is known that computing $wt(f(x))$ in polynomial time, or even narrowing it down to one of $O(|f(x)|^{1/2})$-many values, is as hard as #P itself when multiple calls are allowed [Ram92, RS93]. This paper is the first to study the class CBP of languages decidable with one such call. The main results giving relationships among (a), (b), and the foregoing classes are in Sections 3 and 4. One curiosity is that CBP is the first class we know with Boolean closure properties whose only known constructions increase the lengths of representations more than linearly. We show the significance of this for circuit reductions studied recently by Gottlob [Got93] and Ogihara [Ogi94] for classes between NP and P^{NP}, and it is interesting that the answers for classes between MP and P^{MP} currently seem to be quite different.

2 Preliminaries

A binary relation R on strings is a P-*predicate* if $R(x, y)$ is polynomial-time decidable and there exists a polynomial p such that whenever $R(x, y)$ holds, $|y| = p(|x|)$. Here $|x|$ denotes the length of a string x, while $\|A\|$ stands for the cardinality of a set A. Then a function $f : \Sigma^* \to \mathbf{N}$ belongs to #P [Val79] if there is a P-predicate R such that for all $x \in \Sigma^*$, $f(x)$ equals $\|\{ x : R(x, y) \}\|$. Values of #P functions are written in standard binary notation, with the least significant bit denoted as bit 0, and with $\Sigma = \{ 0, 1 \}$ can be regarded as strings in Σ^*. Bit i of string x is denoted by x_i. We remark for later use that the class #P is unchanged if one restricts attention to R such that $R(x, y)$ never holds when $y \in 0^*$. It follows that for any $f \in$ #P there is a polynomial p such that the functions $2^{p(|x|)} - f(x)$ and $2^{p(|x|)} - 1 - f(x)$ also belong to #P; then p (along with any polynomial greater than p) is called a *bounding polynomial* for f. Note that $2^{p(|x|)} - 1 - f(x)$ complements every bit in the binary representation of $f(x)$ after the first. It is also well known that #P is closed under addition and multiplication. An idea we use frequently is that given two #P functions f_1, f_2 and a bounding polynomial p for f_1, the function $2^{p(|z|)} f_2(z) + f_1(z)$ also belongs to #P. This idea can be extended for polynomially many #P functions, and allows their values to be combined into one.

In some proofs we refer to polynomial-time nondeterministic Turing machines N instead of P-predicates, and denote by $acc_N(x)$ the number of accepting computations N has on input x. Functions of the form acc_N are an equivalent characterization of #P. Many much-studied language classes can be defined in terms of #P and the class PF of polynomial-time computable functions. For instance, a language L belongs to:

(a) NP if for some $f \in$ #P, $L = \{ x : f(x) > 0 \}$.

(b) $\oplus$P if for some $f \in$ #P, $L = \{ x : f(x) \text{ is odd} \}$.

(c) UP if the characteristic function χ_L belongs to #P.

(d) PP if for some $f \in$ #P and $g \in$ PF, $L = \{ x : f(x) > g(x) \}$.

(e) MP if for some $f \in$ #P and $g \in$ PF, $L = \{ x : \text{ bit } g(x) \text{ of } f(x) \text{ is a '1'} \}$.

Access to information in a function f is formalized by the model of an *oracle Turing machine* M, which at points during its computation may write down a *query string* z and receive in one step the value $f(z)$. A language oracle A is the same as having its characteristic function χ_A as oracle. In this paper we study cases where M does not receive $f(z)$, but instead receives some partial information about $f(z)$.

3 Counting-Bit Functions and the Class CBP

Given a string $x \in \{ 0, 1 \}^*$, denote by $wt(x)$ the *Hamming weight* of x, which equals the number of 1's in x.

Definition 3.1. (a) CBF denotes the class of functions $g(x) = wt(f(x))$ for $f \in$ #P. We call g a "counting-bit function" for #P.

(b) A language L belongs to CBP ("counting bits-P") if there exist $g \in$ CBF and a polynomial-time oracle machine M such that M is allowed to make one query z and receive the value $g(z)$, and M accepts L with this oracle.

Since f_{SAT} is complete for #P under reductions that preserve values (see [Sim75, Val79]), one can fix f to be f_{SAT} in this definition. Alternatively, one can let f depend on M and arrange that the query z equals the input x itself. First we show some relationships between CBF and #P itself.

Proposition 3.1 *Every* #P *function that is bounded by a polynomial is in* CBF.

Proof. Let $f \in \#\mathrm{P}$, and let q be a polynomial such that $f(x) \leq q(|x|)$ for all x. Let the NTM N_1 witness f. We construct an NTM N_2 as follows. On input x, N_2 nondeterministically guesses an integer i with $1 \leq i \leq q(|x|)$, then nondeterministically guesses i computation paths of N_1 in lexicographically increasing order, and simulates N_1 on these paths. N_2 accepts iff all of them are accepting paths of N_1. Suppose $f(x) = k \leq q(|x|)$, i.e. N_1 has k accepting paths on input x. For a fixed i, $1 \leq i \leq k$, the number of accepting paths of N_2 by guessing i is $\binom{k}{i}$. Using the binomial formula, we have $acc_{N_2}(x) = \binom{k}{1} + \binom{k}{2} + \cdots + \binom{k}{k} = 2^k - 1$. Thus $wt(acc_{N_2}(x)) = bit(2^k - 1) = k = f(x)$ for all x. Therefore $f \in \mathrm{CBF}$. □

By definition every function in CBF is polynomially bounded, but we show that the converse to Proposition 3.1 is highly unlikely to hold.

Theorem 3.2 CBF $\subseteq \#\mathrm{P}$ *if and only if* PP = UP

Proof. First suppose CBF $\subseteq \#\mathrm{P}$, and let $L \in \mathrm{PP}$. Then by [Gil77] there exists a #P function with bounding polynomial p such that for all x, $x \in L \iff f(x) \geq 2^{p(|x|)-1}$. (This is the special case of the previous definition with $g(x) = 2^{p(|x|)-1}$, which is half the maximum possible value of $f(x)$.) By remarks at the beginning of Section 2, the function $f'(x) = 2^{p(|x|)} - 1 - f(x)$ also belongs to #P. Then $wt(f(x)) + wt(f'(x)) = p(|x|)$. Let $g(x) = 2^{p(|x|)}(f'(x) + 2^{p(|x|)-1}) + f(x)$. Clearly $g \in \#\mathrm{P}$, and for all x,

$$x \in L \implies wt(g(x)) = p(|x|) + 1 \tag{1}$$

$$x \notin L \implies wt(g(x)) = p(|x|) \tag{2}$$

Now by hypothesis, $wt(g) \in \#\mathrm{P}$, so there exists a P-predicate R such that for all x, $wt(g(x)) = \|\{\, y : R(x,y) \,\}\|$. Now let N be an NTM that on input x, nondeterministically guesses $q = p(|x|) + 1$ candidates $y_1, \ldots, y_q$ in lexicographically-increasing order, and accepts iff $R(x, y_i)$ holds for all i, $1 \leq i \leq q$. Then $x \in L$ iff $N(x)$ has a unique accepting computation, and otherwise $N(x)$ has no accepting computations, so $L \in \mathrm{UP}$.

Conversely, suppose PP = UP. By Theorem 3.2(4) of [OH91], it follows that $\mathrm{P}^{\#\mathrm{P}} = \mathrm{UP}$. Let $f \in \#\mathrm{P}$, and define $A = \{\, (x,m) : wt(f(x)) = m \,\}$. Clearly $A \in \mathrm{P}^{\#\mathrm{P}}$, thus $A \in \mathrm{UP}$. Hence there is a P-predicate R such that for all x and m, $(x,m) \in A \implies \|\{\, y : R(x,m,y) \,\}\| = 1$, and $(x,m) \notin A \implies \|\{\, y : R(x,m,y) \,\}\| = 0$. Now define $R'(x,m,y,j)$ to hold iff $R(x,m,y)$ holds and $1 \leq j \leq m$. Then for all x, $\|\{\, (m,y,j) : R(x,m,y,j) \,\}\| = wt(f(x))$. Hence $wt(f(x)) \in \#\mathrm{P}$, so CBF $\subseteq \#\mathrm{P}$. □

Now we show some technical properties of the class CBP.

Lemma 3.3

(a) For any function value $f(x)$ and place $k \geq 0$, if bit k of $f(x)$ is 0 then $wt(f(x) + 2^k) = wt(f(x)) + 1$, else $wt(f(x) + 2^k) \leq wt(f(x))$.

(b) *Given* $f_1, f_2 \in \#\mathrm{P}$ *and a polynomial* q, *we can find* $f_3 \in \#\mathrm{P}$ *such that for all* x, $wt(f_3(x)) = q(|x|)wt(f_1(x)) + wt(f_2(x))$.

Proof. Part (a) is immediate by properties of addition in binary notation, and (b) follows by the ability to string together $q(|x|)$-many copies of $f_1(x)$ followed by one of $f_2(x)$ as described in the first paragraph of Section 2. □

Proposition 3.4 *(a)* CBP *is closed under all Boolean operations.*

(b) $\mathrm{MP} \subseteq \mathrm{CBP} \subseteq \mathrm{P}^{\#\mathrm{P}[1]}$.

Proof. Part (a) follows because Lemma 3.3(b) intuitively lets one set up a "pairing function" on weights upon selecting q to be greater than some bounding polynomial for f_2. Namely, for any $k_1, k_2 \geq 0$, let $k_3 = q(|x|)k_1 + k_2$; then $wt(f_3(x)) = k_3 \iff wt(f_1(x)) = k_1 \wedge wt(f_2(x)) = k_2$. Hence if A_1 and A_2 belong to CBP, and S_1 and S_2 are the deterministic polynomial-time computable sets of accepting weight values as per the remark after Definition 3.1, then S_3 can be computed under this mapping for the desired Boolean combination of A_1 and A_2.

For part (b), that $\mathrm{CBP} \subseteq \mathrm{P}^{\#\mathrm{P}[1]}$ is obvious, and for $\mathrm{MP} \subseteq \mathrm{CBP}$, we need to show how to tell whether bit k of $f(x)$ equals 0 or 1 with *one* evaluation of $wt(f'(x))$ for some other #P function f'. The method is to take $f_1(x) = f(x)$ and $f_2(x) = f(x) + 2^k$ in this pairing construction. □

In the next section we give results that are technically stronger than (b), and compare CBP to other modes of restricted access to the values of #P functions.

4 Results on Counting Classes

Two much-studied restrictions on access by oracle machines M are to limit M to some number $q(n)$ of queries on any input x of length n, and to make M *nonadaptive*; i.e., required to write down all its queries $z_1, z_2, \ldots$ in advance before getting any information from the oracle. $\mathrm{P}^{\mathcal{C}[q(n)]}$ denotes the class of languages accepted in polynomial time by OTMs M with oracles in $\mathcal{C}$ under the former restriction, and $\mathrm{P}^{\mathcal{C}}_{\mathrm{tt}}$ under the latter. The subscript 'tt' is used because the nonadaptive M embodies a truth-table with one row for each possible vector of query answers and entries that tell whether the row accepts or rejects x.

An important special case with a language oracle $A \in \mathcal{C}$ is when the table can be encoded as a Boolean formula ϕ_x of size polynomial in $|x|$ over the yes/no query answers $\vec{a}$, such that $x \in L \iff \phi_x(\vec{a})$ is true. Then following [Wil87, Ogi94] we write $L \in \mathrm{NC}^1(\mathcal{C})$. (The standard uniformity condition on the reduction is that $M(x)$ be able to write down ϕ_x before making any queries.) When ϕ_x is a conjunct (respectively, disjunct) of polynomially many 'yes' answers, we have the notion of a 'ctt' (resp. 'dtt') reduction, and write $L \in \mathrm{P}^{\mathcal{C}}_{\mathrm{ctt}}$ (resp. $L \in \mathrm{P}^{\mathcal{C}}_{\mathrm{dtt}}$). If for some fixed $k > 0$, each ϕ_x produced by M is a depth-k formula of such polynomial-fan-in conjuncts and disjuncts, with negations allowed, then $L \in \mathrm{AC}^0(\mathcal{C})$.

Some results, the first and last easy, which also illustrate the notation are:

(a) $\mathrm{P}^{\mathrm{PP}} = \mathrm{P}^{\#\mathrm{P}}$ [Sim75, Val79].

(b) $\mathrm{P}^{\mathrm{PP}[O(\log n)]} = \mathrm{PP}$ [BRS91].

(c) $\mathrm{P}^{\mathrm{PP}}_{\mathrm{tt}} = \mathrm{PP}$ [FR91].

(d) $\mathrm{P}^{\mathrm{MP}}_{\mathrm{tt}} = \mathrm{P}^{\#\mathrm{P}[1]}$ [GKR+92].

It is well known that the class $\mathrm{P}^{\#\mathrm{P}[1]}$ is unchanged if one fixes the oracle f to be the #P-complete function f_{SAT} described in section 1, or if one fixes $y_x = x$ (but then the oracle function f depends on M). The last result reinforces our designation that MP is the class of languages that depend on one bit of a #P function. The results on PP give another motivation to ask about the power of log-many bits, in studying the classes $\mathrm{P}^{\mathrm{MP}[O(\log n)]}$ and $\mathrm{P}^{\mathrm{MP}[O(\log n)]}_{\mathrm{tt}}$. First we show that several possible ways of formalizing the power of $O(\log n)$ bits from #P functions define the same class.

Lemma 4.1 *For any language A, the following are equivalent:*

(a) $A \in \mathrm{P}^{\mathrm{MP}[O(\log n)]}$.

(b) A is accepted in polynomial time by a machine M that on inputs x of length n is allowed to make $O(\log n)$ queries of the form (z, f, k), where f is (an encoding of) a #P function, receiving as answer bit k of the value $f(z)$.

(c) There is a fixed $f \in$ #P and a polynomial-time machine M' that accepts A while making $O(\log n)$ queries of the form k, each receiving bit k of $f(x)$.

Furthermore, the restrictions of (b) and (c) where M behaves nonadaptively are equivalent to $A \in \mathrm{P}^{\mathrm{MP}[O(\log n)]}_{\mathrm{tt}}$.

Proof. The implications (a) $\Longrightarrow$ (b) and (c) $\Longrightarrow$ (a) are clear enough. The key to (b) $\Longrightarrow$ (c) is that since the "query tree" of M on input x is binary-branching and has height $O(\log n)$, its size $q(n)$ is polynomial, and M' can trace it out to identify all possible #P functions $f_1, \ldots f_{q(n)}$ with bounding polynomials $p_1, \ldots, p_{q(n)}$ and arguments $z_1, \ldots z_{q(n)}$ that M may invoke, before making any queries of its own. These actions depend only on x, and via the idea at the end of the first paragraph in Section 2, there is a #P function f such that $f(x)$ strings together all the values $f_i(z_i)$. Since M' knows the bounding polynomials, it can translate the k components of queries made by M into corresponding indices into the one value $f(x)$. The case where M is nonadaptive is simpler, involving only one branch of the tree. $\square$

J. Köbler [personal communication, 1991] considered the class of languages L such that there exist functions $f \in$ #P and $g \in$ PF giving for all x, $x \in L$ iff $f(x)$

"masks" $g(x)$. The masking relation on two strings $x, y \in \{0,1\}^*$ is defined by: x masks y if x majorizes y bitwise; i.e. for all k such that bit k of y equals 1, then bit k of x also equals 1. Call this class "MaskP." Köbler noted that MP $\subseteq$ MaskP and that MaskP is closed under intersection, which follow from (a) below, and asked whether MaskP is closed under complements.

Proposition 4.2 *(a)* $\mathrm{MaskP} = \mathrm{P}^{\mathrm{MP}}_{\mathrm{ctt}}$, *and also co-*$\mathrm{MaskP} = \mathrm{P}^{\mathrm{MP}}_{\mathrm{dtt}}$.

(b) $\mathrm{P}_{\mathrm{ctt}}(\mathrm{MP}) \subseteq \mathrm{CBP}$ *and* $\mathrm{P}_{\mathrm{dtt}}(\mathrm{MP}) \subseteq \mathrm{CBP}$.

(c) For all fixed $m \geq 1$, $\mathrm{P}^{\mathrm{CBP}[m]} = \mathrm{CBP}$.

Proof. That $\mathrm{MaskP} \subseteq \mathrm{P}^{\mathrm{MP}}_{\mathrm{ctt}}$ is clear. For the reverse inclusion, a ctt-reduction to a language in MP can be converted to a conjunct of conditions of the form "bit k of $f(x)$ is 1" and "bit k' of $f(x)$ is 0." Let p be a bounding polynomial for $f(x)$; then $f'(x) = 2^{p(|x|)} - 1 - f(x)$ is also in #P, and so is $f''(x) = 2^{p(|x|)} \cdot f'(x) + f(x)$, which essentially concatenates $f(x)$ and the bit-complement of $f(x)$. Then the conjunct can be converted into a mask for 1's. The second part of (a) follows similarly.

For (b), given $L \in \mathrm{P}_{\mathrm{ctt}}(\mathrm{MP})$ we can find $f \in$ #P and a polynomial-time non-oracle machine that given any input x constructs a set S_x such that $x \in L$ iff for every $k \in S_x$, bit k of $f(x)$ is 0. Then the function $f'(x) = f(x) + \sum_{k \in S_x} 2^k$ belongs to #P, and by iterating the argument in Lemma 3.3(a), $wt(f'(x)) = k + wt(f(x))$ iff $x \in L$. The case for $L \in \mathrm{P}_{\mathrm{dtt}}(\mathrm{MP})$ follows because both MP and CBP are closed under complements.

Finally, (c) is proved by extending the pairing idea of Proposition 3.4(a) to m-tuples. The length of the representation may expand from $p(n)$ to $p(n)^m$, but since m is a fixed constant, this is still polynomially bounded. □

This raises several open questions. The first is whether $O(\log n)$ queries to CBP or to MP are equivalent to one. With CBP, the difficulty is that the pairing idea of Proposition 3.4(a) changes a #P function f with bounding polynomial p into one with bounding polynomial p^2. Applying it $O(\log n)$ times creates a representation of length $p(n)^{O(\log n)}$, which is no longer polynomially bounded. With MP, the difficulty is that MP is not known to be closed under union or intersection at all; this problem is described in detail at the end of [GKR+92]. In all previous cases in the literature where a complexity class $\mathcal{C}$ has been shown to be closed under intersection or other Boolean operations, even the very difficult theorem for PP from [BRS91] cited above, the theorem provides a *construction* that only expands the lengths of the representations *linearly*. That is, let R_A and R_B be representations for two languages A, B in $\mathcal{C}$; for instance, they can be the codes of machines that compute the P-predicates and auxiliary functions for the classes defined in Section 2. Then there is a polynomial-time computable function F that given any two strings R_A and R_B outputs a representation R_C for $C = A \cap B$, and in addition $|R_C| \leq c(|R_A| + |R_B|)$, where c is fixed. We spell out the requirements in full detail for MP and CBP:

Definition 4.1. MP is *polynomial-time constructively closed under intersection* if there is a polynomial-time computable function F and $c > 0$ such that given the codes of any two P-predicates R_1, R_2 and bit-selection functions g_1, g_2, $F(R_1, g_1, R_2, g_2)$ outputs the codes of a P-predicate R_3 and a bit-selection function g_3 such that

(a) for all x, bit $g_3(x)$ of $\|\{ y : R_3(x,y) \}\|$ equals '1' iff bit $g_1(x)$ of $\|\{ y : R_1(x,y) \}\|$ equals '1' AND bit $g_2(x)$ of $\|\{ y : R_2(x,y) \}\|$ equals '1,' and

(b) $|R_3| + |g_3| \leq c(|R_1| + |g_1| + |R_2| + |g_2|)$.

The corresponding condition for CBP is defined similar, with the code M_1 of the polynomial-time OTM in Definition 3.1 in place of g_1, similarly for M_2 and M_3, and using the acceptance conditions on M_1, M_2, M_3 from Definition 3.1 in place of (a).

For short, we say that MP and/or CBP would be *p-closed* under intersection. Since both these classes are closed under complementation via constructions of the required efficient kind, p-closure under intersection does likewise for all Boolean operations. For MP to be closed under intersection says that knowing two bits of a #P function value is no more powerful than knowing just one, and part (a) of the next result says that if this knowledge can be transformed constructively, then one bit is as powerful as knowing $O(\log n)$ bits.

Theorem 4.3 *(a) If* MP *is p-closed under intersection, then* $\mathrm{NC}^1(\mathrm{MP}) = \mathrm{MP}$*, which implies in particular that* $\mathrm{P}^{\mathrm{MP}[O(\log n)]} = \mathrm{MP}$ *and* $\mathrm{MaskP} = \mathrm{MP}$*.*

(b) If CBP *is p-closed under intersection, then* $\mathrm{NC}^1(\mathrm{CBP}) = \mathrm{CBP}$*, and in particular,* $\mathrm{P}^{\mathrm{CBP}[O(\log n)]} = \mathrm{CBP}$*.*

Proof. Let ϕ be the formula over queries to MP constructed by the machine that carries out the NC^1 reduction for the given language $L \in \mathrm{P}^{\mathrm{MP}[O(\log n)]}$. By known techniques (see [HS65, Bus87, MP92]), ϕ can be "rebalanced" into an equivalent formula ϕ' of $O(\log n)$ depth. Since ϕ' has $O(\log n)$ depth, the increase in size of the representation is bounded by a polynomial times $c^{O(\log n)}n$, which is polynomially bounded. The treatment of CBP is similar. (See also remarks below.) □

Since MP and CBP both have complete sets under polynomial-time many-one reductions (the constructions are standard and left to the reader), we can use a specific analogue of Definition 4.1 for these classes. Let U be some complete set for MP, and define $A = \{ (x,y) : x \in U \}$ and $B = \{ (x,y) : y \in U \}$. Now suppose that $A \cup B$ and $A \cap B$ both belong to MP, so in particular, they reduce back to U. The specific analogue of the definition is that there should be a function u reducing $A \cup B$ to U and a function v reducing $A \cap B$ to U, such that u and v increase the length of their argument by at most a factor of some fixed constant $c > 0$. Now suppose we have an $\mathrm{NC}^1(\mathrm{MP})$ machine M that accepts its input x iff the formula

$$\phi_x = (((a_1 \wedge a_2) \vee (a_3 \wedge a_4)) \wedge ((a_5 \wedge a_6) \vee (a_7 \wedge a_8)))$$

holds, where each a_i is the answer to a query (x, k_i) made by M. (In general we may suppose that negations are pushed to the leaves in ϕ, and by the idea of Proposition 4.2(a), eliminated altogether.) Finally let $D = \{ (x, k) : \text{ bit } k \text{ of } f(x) \text{ equals } 1 \}$ be the MP language used as oracle by M, and let g reduce D to U. Write g_i as short for $g(x, k_i)$, $1 \leq i \leq 8$. Then the instance of U equivalent to whether M accepts x is

$$z_x = v(u(v(g_1, g_2), v(g_3, g_4))), u(v(g_5, g_6), v(g_7, g_8))).$$

This illustrates that the nesting depth of the reductions from $A \cap B$ and $A \cup B$ to U is what governs the complexity of computing z_x from x. In the general case of a depth-d formula, under our hypothesis on u and v, the length of z_x is bounded by $2^d c^d n$, which is polynomially bounded if $d = O(\log n)$. The *time* to compute z_x is majorized by the worst-case time for the outermost u or v, and the g's do not nest, so the time issue is less important. This reinforces the import of Definition 4.1(b) that the linear length expansion of the representations is what matters most.

A second open question is whether $\mathrm{P}^{\mathrm{MP}[O(\log n)]} \subseteq \mathrm{CBP}$. This would follow if uniform OR-OF-AND circuits over queries to MP (where the OR gate has polynomial fan-in and the AND gates have log fan-in) can be simulated in CBP. Note that our result $\mathrm{P}^{\mathrm{MP}}_{\mathrm{ctt}} \subseteq \mathrm{CBP}$ shows that circuits with a single unbounded fan-in AND or OR can be thus simulated, but we have not found a way to take the reasoning up a level. A third problem is whether $\mathrm{NC}^1(\mathrm{MP}) = \mathrm{AC}^0(\mathrm{MP})$. This holds for NP in place of MP [Got93, Ogi94], but seems to require a property even stronger than MP being closed under union.

5 Conclusion

This paper has mapped out the structure of classes between MP and $\mathrm{P}^{\#\mathrm{P}[1]}$. These two classes represent the power of knowing one bit and of knowing all the bits of a #P function value, respectively, and we found some relationships among different kinds of partial knowledge in between. Although some of our new definitions are very technical, our results showed their significance to natural, long-studied problems in complexity theory. We have also found several challenging questions related to other work in progress, and look toward further research on counting classes.

References

[BRS91] R. Beigel, N. Reingold, and D. Spielman. The perceptron strikes back. In *Proc. 6th Annual IEEE Conference on Structure in Complexity Theory*, pages 286–291, 1991.

[Bus87] S. Buss. The boolean formula value problem is in ALOGTIME. In *Proc. 19th Annual ACM Symposium on the Theory of Computing*, pages 123–131, 1987.

[FR91] L. Fortnow and N. Reingold. PP is closed under truth-table reductions. In *Proc. 6th Annual IEEE Conference on Structure in Complexity Theory*, pages 13–15, 1991.

[Gil77] J. Gill. Computational complexity of probabilistic Turing machines. *SIAM J. Comput.*, 6:675–695, 1977.

[GKR$^+$92] F. Green, J. Köbler, K. Regan, T. Schwentick, and J. Torán. The power of the middle bit of a #P function. Technical Report UBCS–TR 92–27, Computer Science Dept., University at Buffalo, 1992. Accepted to *J. Comp. Sys. Sci.*

[Got93] G. Gottlob. NP trees and Carnap's modal logic. In *Proc. 34th Annual IEEE Symposium on Foundations of Computer Science*, pages 42–51, 1993.

[HS65] J. Hartmanis and R. Stearns. On the computational complexity of algorithms. *Transactions of the AMS*, 117:285–306, 1965.

[MP92] D. Muller and F. Preparata. Parallel restructuring and evaluation of expressions. *J. Comp. Sys. Sci.*, 44:43–62, 1992.

[Ogi94] M. Ogihara. $\mathrm{NC}^k(\mathrm{NP}) = \mathrm{AC}^{k-1}(\mathrm{NP})$. In *Proc. 11th Annual Symposium on Theoretical Aspects of Computer Science*, pages 313–324, 1994.

[OH91] M. Ogiwara and L. Hemachandra. A complexity theory for feasible closure properties. In *Proc. 6th Annual IEEE Conference on Structure in Complexity Theory*, pages 16–29, 1991.

[Ram92] J. Ramachandran. Set-bit enumeration is hard. In *Proc. 7th Annual IEEE Conference on Structure in Complexity Theory*, pages 58–70, 1992.

[RS92] K. Regan and T. Schwentick. On the power of one bit of a #p function. In *Proc. 4th Annual Italian Conference on Theoretical Computer Science*, pages 317–329. World Scientific, Singapore, 1992.

[RS93] J. Ramachandran and T. Schwentick. A note on set-bit enumeration, 1993. Technical Report, University of Mainz, 11/26/93.

[Sim75] J. Simon. *On some central problems in computational complexity.* PhD thesis, Cornell University, 1975.

[Tod91] S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20:865–877, 1991.

[Val79] L. Valiant. The complexity of computing the permanent. *Theor. Comp. Sci.*, 8:189–201, 1979.

[Wil87] C. Wilson. Relativized NC. *Math. Sys. Thy.*, 20:13–29, 1987.