On Closure Properties of #P in the Context of PF $\circ$ #P *

Mitsunori Ogihara†
Dept. Computer Science
Univ. of Rochester
(ogiwara@cs.rochester.edu)

Thomas Thierauf
Fakultät für Informatik
Universtät Ulm
(thierauf@informatik.uni-ulm.de)

Seinosuke Toda
Dept. Computer Science
Univ. Electro-Communications
(toda@cs.uec.ac.jp)

Osamu Watanabe
Dept. Computer Science
Tokyo Institute of Technology
(watanabe@cs.titech.ac.jp)

Abstract

For any operator τ on integer-valued functions, we say that #P *is closed under* τ *in context* PF$\circ$#P if, for every $f \in \#\mathrm{P}$, $\tau[f]$ belongs to PF $\circ$ #P. For several operators τ, it is shown that the closure properties of #P under τ in the above sense is closely related to the relationships between $\mathrm{P}^{\#\mathrm{P}[1]}$ and higher classes such as $\mathrm{PH}^{\mathrm{PP}}$ and $\mathrm{PP}^{\mathrm{PP}}$.

1. Introduction

Counting is one of the key notions in computation. Recently, various counting problems have received considerable attention (see, e.g., [Sch90]) and, in order to model them, there have been introduced and extensively studied complexity classes called *counting classes*, typified by function classes #P [Val79], spanP [KST89], and GapP [FFK94], and language classes PP [Gil77], $\oplus$P [PZ83], $\mathrm{C}_=\mathrm{P}$ [Sim75, Wag86a], and the counting hierarchy CH [Tor91, Wag86b]. Unfortunately, many of the questions regarding counting classes, even the ones about inclusion relation, are left open. Confronted with such difficulties in resolving problems absolutely, researchers have

*A part of the work was done while the first author was visiting Department of Computer Science, State University of New York at Buffalo, and the second, third, and fourth authors were visiting Department of Computer Science, University of Rochester. This research is supported in part by JSPS/NSF International Collaboration Grant JSPS-ENGR-207/NSF-INT-9116781. The first author is supported in part by NSF Grant grant CCR-9002292. The second author is supported in part by DFG Postdoctorial Scholarship Th 472/1-1 and NSF grant CCR-8957604.

†Previously known as Mitsunori Ogiwara.

devised tools to obtain relative answers that promote a better understanding of the original questions. (*Cf.* Even though the P=?NP question is open, through various research, now we have ample knowledge about how NP would be different from P if they were different.) The purpose of this paper is to introduce a structural concept that helps us to deepen our understanding on relationships between counting classes.

The central counting class is #P, the class of functions that count the number of solutions to NP decision problems. The class #P is known to contain many natural functions, such as the permanent of integer matrices, which is one of the first nontrivial functions proven to be in #P and, in fact, proven to be #P-complete [Val79]. With the increase in the number of interesting examples, the properties of #P, especially, the *closure properties* of #P, has become a central research topic. Intuitively, we say that #P is closed under an operation τ if the functions constructed by applying τ to #P functions always belong to #P. For instance, for any #P functions $f(x)$ and $g(x)$, the functions $f(x) + g(x)$ and $f(x)g(x)$ also belong to #P, Here we say that #P is closed under addition and multiplication and that both addition and multiplication are closure properties of #P. Closure properties of #P have played important roles, both explicitly and implicitly, in the study of counting classes of languages [BHW91, BRS91, CH90, FR91], and many closure properties possessed by #P have been found (See [OH93]). Nevertheless, the class does not seem to possess closure properties under some primitive operations, such as modified subtraction.[1]

Ogiwara and Hemachandra [OH93] have established the theory for closure properties of function classes. They have clarified why #P seems to lack such primitive closure properties. They showed that #P is closed under modified subtraction if and only if the counting hierarchy collapses to UP, which is the smallest counting class. Informally put, we cannot hope that modified subtraction of #P functions is done by #P unless all the decision problems in the counting hierarchy, including those belonging to the polynomial-time hierarchy, are solved by NP machines that have at most one accepting path per input.

Although it is not likely that #P functions can compute modified subtraction of #P-functions, we notice that subtraction is *almost* computed by #P functions. Let $f(x)$ and $g(x)$ be two #P functions and $p(n)$ be a polynomial such that $\max\{f(x), g(x)\} < 2^{p(|x|)}$ for all x. Then it is easy to design a #P function $h(x)$ such that for all x, $h(x) = 2^{p(|x|)} + (f(x) - g(x))$. Clearly, the first bit of $h(x)$ represents the sign of $f(x) - g(x)$ and the last $p(|x|)$ bits of $h(x)$ represent $f(x) - g(x)$. So, we can easily retrieve $f(x) - g(x)$ from $h(x)$. Here we may say that *the function $h(x)$ realizes the subtraction of $f(x)$ and $g(x)$*, as the actual value of the subtraction is encoded in the binary representation of $h(x)$, and we might as well say that #P is *closed under subtraction in some weaker sense*, as we only have to do some simple post-computation on the

[1] The *modified subtraction* of m from n, denoted by $n \ominus m$, is $\max\{n - m, 0\}$. Since #P functions are always nonnegative, #P is provably not closed under the usual subtraction.

outcome of a #P function. This observation is generalized to the following definition of closure properties of #P in context $\mathrm{PF} \circ \#\mathrm{P}$.

Definition 1.1. For any operator (or, functor) τ, let $\tau[\#\mathrm{P}]$ denote the class of functions obtained by applying τ to some function in #P, and let $\mathrm{PF} \circ \#\mathrm{P} = \{h \circ f \ : \ h \in \mathrm{PF}, f \in \#\mathrm{P}\}$, where $h \circ f$ denotes the ordinary composition of the two functions and PF denotes the class of all polynomial-time computable functions.

We say that #P *is closed under* τ *in context* $\mathrm{PF} \circ \#\mathrm{P}$ if $\tau[\#\mathrm{P}] \subseteq \mathrm{PF} \circ \#\mathrm{P}$.

In other words, τ is a closure property of #P in context $\mathrm{PF} \circ \#\mathrm{P}$ if the function generated by applying τ to #P can be computed by #P with supplementary polynomial-time post-computation. We have chosen $\mathrm{PF} \circ \#\mathrm{P}$ from the point of view that we should keep our context as close as possible to #P. But, in fact, the above definition can be easily extended to an arbitrary context. However, as far as it concerns our results, our proof techniques can be applied to any larger context to show results similar to the ones we will prove.

By allowing polynomial-time post-computation and extending the context from #P to $\mathrm{PF} \circ \#\mathrm{P}$, we have cured the weakness of #P, i.e., the lack of closure properties under some primitive operations. Indeed, it is easy to see that, in context of $\mathrm{PF} \circ \#\mathrm{P}$, the class is closed not only under modified subtraction but also under many 'hard' closure properties [OH93]. This leads us to question "What is the limit of the closure properties of #P in context $\mathrm{PF} \circ \#\mathrm{P}$?" In order to answer this question, we seek to find closure properties that are provably possessed by #P (lower bounds) as well as those that do not seem to be possessed by #P (upper bounds). We believe that clarifying the limit will shed lights on the computational power of $\mathrm{PF} \circ \#\mathrm{P}$ and, in turn, on the structure of #P.

Consider the following two notions of majority computing operators, which we call the weak majority and the (strong) majority, respectively.[2] For any function $f : \Sigma^* \to \mathbf{N}$ and any string x in Σ^*,

$$\mathrm{maj_w}[f](x) = \begin{cases} y, & \text{if more than half of } f(\langle 1, x\rangle), \ldots, f(\langle 2^{|x|}, x\rangle) \text{ are equal to } y, \\ \text{some value}, & \text{otherwise;} \end{cases}$$

$$\mathrm{maj}[f](x) = \begin{cases} y, & \text{if more than half of } f(\langle 1, x\rangle), \ldots, f(\langle 2^{|x|}, x\rangle) \text{ are equal to } y, \\ ?, & \text{otherwise,} \end{cases}$$

where "?" $\in \Sigma^*$ is a special symbol not representing an element in $\mathbf{N}$.

Both the weak majority $\mathrm{maj_w}[f]$ and the (strong) majority $\mathrm{maj}[f]$ take the same value y if y gains a majority in the values of f. But, when there is no majority, they behave differently;

[2] As we shall see in the next section, formally, we will consider classes of operators instead of one fixed operator.

$\mathrm{maj}_{\mathrm{w}}[f]$ takes the value "?" to inform that there is no majority while $\mathrm{maj}_{\mathrm{w}}[f]$ may take an arbitrary value. The difference seems crucial, for, as we shall see in Section 3, the following results hold:

(1) #P is closed under $\mathrm{maj}_{\mathrm{w}}$ in context PF∘#P,[3]

(2) #P is closed under maj in context PF∘#P if and only if $\mathrm{P}^{\#\mathrm{P}[1]} = \mathrm{PP}^{\mathrm{PP}}$ (or, equivalently, CH collapses to $\mathrm{P}^{\#\mathrm{P}[1]}$).

Thus, we conclude that the limit of the closure properties of #P in context PF ∘ #P is between the weak majority and the strong majority, and that the crucial factor that (possibly) separates $\mathrm{P}^{\#\mathrm{P}[1]}$ and $\mathrm{PP}^{\mathrm{PP}}$ is that only one question to #P does not help to detect whether the majority exists among exponentially many values of a #P function.

We also seek to find results similar to (2) above, i.e., the results characterizing collapses of counting classes in terms of the closure properties of #P. We think such characterizations will be useful (in some cases) for analyzing relationships among counting classes. In Section 4, we provide such results with respect to closure properties median, plurality, and maximum.

2. Preliminaries

In this paper, we follow standard definitions and notations in computational complexity theory (see, e.g., [BDG88, BDG91]).

Throughout this paper, we fix our alphabet to $\Sigma = \{0, 1\}$; by a *string* we mean an element of Σ^*, and by a *language* we mean a subset of Σ^*. Natural numbers are encoded in Σ^* in an ordinary way, and let $\mathbf{N}$ denote the set of (encoded) natural numbers. For any string x, let $|x|$ denote the length of x, and for any set X, let $\|X\|$ denote the cardinality of X. For any language L, let $L^{\leq n}$ be the set $\{\, x \in L \,:\, |x| \leq n \,\}$. The standard lexicographic ordering of Σ^* is used; that is, for strings $x, y \in \Sigma^*$, x is *lexicographically smaller* than y (denoted by $x < y$) if either (i) $|x| < |y|$, or (ii) $|x| = |y|$ and there exists $z, u, v \in \Sigma^*$ such that $x = z0u$ and $y = z1v$. We consider a standard one-to-one pairing function from $\Sigma^* \times \Sigma^*$ to Σ^* that is computable and invertible in polynomial time. For inputs x and y, we denote the output of the pairing function by $x\#y$; this notation is extended to denote every n tuple. Furthermore, we assume that for all (x, y) and (x', y') such that $|x| = |x'|$ and $|y| = |y'|$, we have $|x\#y| = |x'\#y'|$.

Throughout this paper we assume that functions are *total.*

For our computation model, we consider standard Turing machines. A machine is either deterministic or nondeterministic, and a deterministic machine is either an *acceptor* or a *transducer*, while a nondeterministic Turing machine is always an acceptor. We also consider a *query machine*, i.e., a machine that can ask queries to a given oracle. In this paper, an oracle is either a set or a function; for each oracle type, we adopt the standard query mechanism for our

[3]We show that for an appropriate choice of the values when there is no majority, the weak majority of #P functions can be done in context PF ∘ #P.

query machines. We assume that the nondeterministic branching degree at each guessing state is always two. For a nondeterministic machine M and any string x, let $acc_M(x)$ (resp., $rej_M(x)$, $total_M(x)$) denote the number of accepting paths (resp., the number of rejecting paths, the total number of paths) of M on input x.

In what follows, we define the complexity classes used in this paper. Below, we denote by $\mathcal{C}$ any class of either languages or functions, and we define those classes relative to $\mathcal{C}$. Nonrelativized classes are defined as special cases in which the empty oracle is used.

(1) $\mathrm{P}^{\mathcal{C}}$ is the class of languages L for which there exist some polynomial time-bounded deterministic query acceptor M and some oracle X in $\mathcal{C}$ such that for all $x \in \Sigma^*$, $x \in L$ if and only if M^X accepts x.

(2) $\mathrm{NP}^{\mathcal{C}}$ is the class of languages L for which there exist some polynomial time-bounded nondeterministic query acceptor M and some oracle X in $\mathcal{C}$ such that for all $x \in \Sigma^*$, $x \in L$ if and only if $acc_{M^X}(x) > 0$.

(3) $\mathrm{PP}^{\mathcal{C}}$ is the class of languages L for which there exist some polynomial time-bounded nondeterministic acceptor M and some oracle X in $\mathcal{C}$ such that for all $x \in \Sigma^*$, $x \in L$ if and only if $acc_{M^X}(x) > total_{M^X}(x)/2$.

(4) $\mathrm{C_{=}P}^{\mathcal{C}}$ is the class of languages L for which there exist some polynomial time-bounded nondeterministic acceptor M, some integer-valued function f in PF, and some oracle X in $\mathcal{C}$ such that for all $x \in \Sigma^*$, $x \in L$ if and only if $acc_{M^X}(x) = f(x)$.

(5) $\mathrm{PF}^{\mathcal{C}}$ is the class of functions that are computable by some polynomial time-bounded query transducer with some oracle in $\mathcal{C}$.

(6) $\#\mathrm{P}^{\mathcal{C}}$ is the class of total functions $f : \Sigma^* \to \mathbf{N}$ for which there exist some polynomial time-bounded nondeterministic query acceptor M and some oracle X in $\mathcal{C}$ such that for all $x \in \Sigma^*$, $f(x) = acc_{M^X}(x)$.

By restricting the way of asking queries, we can define various subclasses of the above classes. Here we define those that are used in our discussion.

(7) $\mathrm{P}^{\mathcal{C}[1]}$ (resp., $\mathrm{PF}^{\mathcal{C}[1]}$) is the class of languages accepted (resp., computed) by some polynomial-time deterministic query machine relative to some oracle in $\mathcal{C}$, where the query machine asks at most one query per input. (Such query machines are called *one-query machines.*)

The polynomial-time hierarchy and the counting hierarchy are defined as follows.

(8) $\mathrm{PH}^{\mathcal{C}}$ is the class $\mathrm{NP}^{\mathcal{C}} \cup \mathrm{NP}^{\mathrm{NP}^{\mathcal{C}}} \cup \mathrm{NP}^{\mathrm{NP}^{\mathrm{NP}^{\mathcal{C}}}} \cup \ldots$, where classes $\mathrm{NP}^{\mathrm{NP}^{\mathcal{C}}}$, $\mathrm{NP}^{\mathrm{NP}^{\mathrm{NP}^{\mathcal{C}}}}, \ldots$ are defined inductively. $\mathrm{PFH}^{\mathcal{C}}$ is the class of functions that are computable in polynomial time relative to any language in $\mathrm{PH}^{\mathcal{C}}$.

(9) $\mathrm{CH}^{\mathcal{C}}$ is the class $\mathrm{PP}^{\mathcal{C}} \cup \mathrm{PP}^{\mathrm{PP}^{\mathcal{C}}} \cup \mathrm{PP}^{\mathrm{PP}^{\mathrm{PP}^{\mathcal{C}}}} \cup \ldots$, where classes $\mathrm{PP}^{\mathrm{PP}^{\mathcal{C}}}$, $\mathrm{PP}^{\mathrm{PP}^{\mathrm{PP}^{\mathcal{C}}}}, \ldots$ are

defined inductively.

We will mainly deal with the following language classes: $\mathrm{P}^{\#\mathrm{P}[1]}$, $\mathrm{P}^{\#\mathrm{P}}$, PH, $\mathrm{PH}^{\mathrm{PP}}$, $\mathrm{PP}^{\mathrm{PP}}$, and CH. We know that $\mathrm{PP}^{\mathrm{PH}} \subseteq \mathrm{P}^{\#\mathrm{P}[1]}$ [Tod91] and $\mathrm{P}^{\#\mathrm{P}[1]} \subseteq \mathrm{P}^{\#\mathrm{P}} \subseteq \mathrm{PH}^{\mathrm{PP}} \subseteq \mathrm{P}^{\mathrm{PP}^{\mathrm{PP}}} \subseteq \mathrm{CH}$. None of these inclusions are known to be proper. Below, we list several elementary facts on these classes, which are used in proving our results. They are either obvious or easy to prove.

Proposition 2.1.
(1) $\mathrm{PF} \circ \#\mathrm{P} = \mathrm{PF}^{\#\mathrm{P}[1]}$.
(2) $\mathrm{PF}^{\mathrm{CH}} = \mathrm{CH}$.
(3) $\mathrm{PP}^{\mathrm{PP}} \subseteq \mathrm{P}^{\#\mathrm{P}[1]}$ if and only if $\mathrm{CH} = \mathrm{P}^{\#\mathrm{P}[1]}$ if and only if $\mathrm{PF}^{\mathrm{CH}} \subseteq \mathrm{PF}^{\#\mathrm{P}[1]}$.
(4) $\mathrm{NP}^{\mathrm{PP}} \subseteq \mathrm{P}^{\#\mathrm{P}[1]}$ if and only if $\mathrm{CH} = \mathrm{P}^{\#\mathrm{P}[1]}$ if and only if $\mathrm{PF}^{\mathrm{CH}} \subseteq \mathrm{PF}^{\#\mathrm{P}[1]}$.

We will further use the following technical result on $\mathrm{C}_=\mathrm{P}$ due to Simon [Sim75] and the results on $\mathrm{PP}^{\mathrm{PP}}$ and $\mathrm{NP}^{\mathrm{PP}}$ that are slight modification of the results due to Torán [Tor91].

Lemma 2.2.
(1) [Sim75] Let $A \in \mathrm{C}_=\mathrm{P}$. Then there exist a polynomial q and a polynomial-time nondeterministic machine M such that for all x, the following conditions are satisfied:
  (i) $total_M(x) = 2^{q(|x|)}$, (ii) $acc_M(x) \leq 2^{q(|x|)-1}$, and
  (iii) $x \in A$ if and only if $acc_M(x) = rej_M(x) = 2^{q(|x|)-1}$.
(2) [Tor91] A set L is in $\mathrm{PP}^{\mathrm{PP}}$ if and only if there exist a polynomial p and a set $A \in \mathrm{C}_=\mathrm{P}$ such that for every x,

$$\begin{array}{lcll} x \in L & \Longleftrightarrow & \|\{\, w \in \Sigma^{p(|x|)} : x\#w \in A \,\}\| \geq 2^{p(|x|)-1} + 1, & \text{and} \\ x \notin L & \Longleftrightarrow & \|\{\, w \in \Sigma^{p(|x|)} : x\#w \in A \,\}\| \leq 2^{p(|x|)-1} - 1. & \end{array}$$

(3) [Tor91] A set L is in $\mathrm{NP}^{\mathrm{PP}}$ if and only if there exist a polynomial p and a set $A \in \mathrm{C}_=\mathrm{P}$ such that for every x, we have $x \in L \iff \|\{\, w \in \Sigma^{p(|x|)} : x\#w \in A \,\}\| \geq 1$.

The operators we study as closure properties are based on the following functions on $\mathbf{N}^*$, where $\mathbf{N}^*$ is the set of tuples of $\mathbf{N}$. Let $(x_1, ..., x_m)$ be any element in $\mathbf{N}^*$ and let ϕ be some fixed function from $\mathbf{N}^*$ to $\mathbf{N}$.

$$\mathrm{maj}(x_1, \ldots, x_m) = \begin{cases} y, & \text{if more than half of } x_1, \ldots, x_m \text{ are equal to } y, \\ ?, & \text{otherwise (where ? is some symbol not in } \mathbf{N}), \end{cases}$$

$$\mathrm{maj}_{\mathrm{w}}^{\phi}(x_1, \ldots, x_m) = \begin{cases} y, & \text{if more than half of } x_1, \ldots, x_m \text{ are equal to } y, \\ \phi(x_1, \ldots, x_m), & \text{otherwise,} \end{cases}$$

$$\mathrm{mid}(x_1, \ldots, x_m) = \text{the } \lfloor (m+1)/2 \rfloor\text{-th smallest value}$$

$$
\begin{aligned}
& && \text{in the ordering } x_{i_1} \le \cdots \le x_{i_m} \text{ of } x_1, \ldots, x_m, \\
\mathrm{plu}(x_1, \ldots, x_m) &= && \text{the } \textit{set} \text{ of the most commonly occurring number(s)} \\
& && \text{amongst } x_1, \ldots, x_m, \\
\mathrm{plu}^*(x_1, \ldots, x_m) &= && \text{the smallest value in } \mathrm{plu}(x_1, \ldots, x_m), \\
\max(x_1, \ldots, x_m) &= && \text{the largest number in } \{x_1, \ldots, x_m\},
\end{aligned}
$$

Let us say a few words about "mid". When the number m of elements is odd, then the median, i.e., the middle element, is unambiguous since it is the $(m+1)/2$-th smallest element. However, when m is even, there are two candidates for the median, namely the $\lfloor (m+1)/2 \rfloor$-th and the $\lceil (m+1)/2 \rceil$-th smallest element, which are called the *left* and *right median*, respectively. We defined "mid" as a function taking the left median. As shown in [OH93], sometimes one has to be careful about which median function is chosen. However, our results concerned with the median operator hold for the right median operator as well.

An *operator* is defined as a functor mapping one function to another. We define now the operator classes that we are interested in. Let f be a function on Σ^*, and let ϕ be some function from $\mathbf{N}^*$ to $\mathbf{N}$. Below, e denotes a polynomial-time computable function of Σ^* to $\mathbf{N}$ (in binary).

$$
\begin{aligned}
\textbf{poly-pre}[f] &= \{\, f \circ h \,:\, h \in \mathrm{PF} \,\}, \\
\textbf{poly-post}[f] &= \{\, h \circ f \,:\, h \in \mathrm{PF} \,\}, \\
\textbf{poly}[f] &= \textbf{poly-pre}[f] \cup \textbf{poly-post}[f], \\
\textbf{maj}[f] &= \{\, g \,:\, g(x) = \mathrm{maj}(\, f(1,x), \ldots, f(e(x),x)\,) \text{ for some } e \in \mathrm{PF} \,\}, \\
\textbf{maj}_{\mathrm{w}}^{\phi}[f] &= \{\, g \,:\, g(x) = \mathrm{maj}_{\mathrm{w}}^{\phi}(\, f(1,x), \ldots, f(e(x),x)\,) \text{ for some } e \in \mathrm{PF} \,\}, \\
\textbf{mid}[f] &= \{\, g \,:\, g(x) = \mathrm{mid}(\, f(1,x), \ldots, f(e(x),x)\,) \text{ for some } e \in \mathrm{PF} \,\}, \\
\textbf{plu}[f] &= \{\, g \,:\, g(x) = \mathrm{plu}^*(\, f(1,x), \ldots, f(e(x),x)\,) \text{ for some } e \in \mathrm{PF} \,\}, \text{ and} \\
\textbf{max}[f] &= \{\, g \,:\, g(x) = \max(\, f(1,x), \ldots, f(e(x),x)\,) \text{ for some } e \in \mathrm{PF} \,\}.
\end{aligned}
$$

For any class of functions $\mathcal{F}$ and any operator class τ, we define $\tau[\mathcal{F}]$ to be the class $\bigcup\{\, \tau[f] : f \in \mathcal{F} \,\}$.

It is clear that #P is closed under **poly-pre**. Thus, for discussing closure properties of #P, our choice of a pairing function $\langle \cdot, \cdot \rangle$ is not essential. On the other hand, #P is not known to be closed under **poly-post**.

We can now reformulate our questions concerning closure properties of #P in context PF $\circ$ #P as follows. Let τ be any of the operator classes defined above. Then we ask whether $\tau[\#\mathrm{P}] \subseteq \textbf{poly}[\#\mathrm{P}]$. We will show that this in fact holds for $\tau = \textbf{maj}_{\mathrm{w}}^{\phi}$ for an appropriate choice of ϕ, and this does not hold for $\tau = \textbf{maj}$, **mid**, **plu**, or **max** unless some implausible collapse occurs.

3. On the Majority Operators

We show in this section that #P is closed under the weak majority operator in context PF $\circ$ #P, but not closed under the majority operator in context PF$\circ$#P, unless the counting hierarchy collapses. In the proof of our first theorem, we need the following result of Toda [Tod91].

Lemma 3.1. [Tod91] Let $T' \in \#\mathrm{P}$, q be a polynomial, and $m \geq 2$ be a natural number. Then there is a function $T \in \#\mathrm{P}$ such that for all $x \in \Sigma^*$ of length n,

$$\begin{aligned} T'(x) \equiv 0 \pmod{m} &\implies T(x) \equiv 0 \pmod{m^{q(n)}}, \text{ and} \\ T'(x) \equiv -1 \pmod{m} &\implies T(x) \equiv -1 \pmod{m^{q(n)}}. \end{aligned}$$

Theorem 3.2. #P is closed under $\mathbf{maj}_{\mathrm{w}}^{\phi}$ in context PF $\circ$ #P, for some function $\phi : \mathbf{N}^* \to \mathbf{N}$.[4]

Proof. Let $f \in \#\mathrm{P}$, let e be a polynomial-time computable function, and let $g(x) = \mathrm{maj}(f(1,x), \ldots, f(e(x),x))$. Our goal is to design a polynomial time-bounded deterministic transducer M_0 that, for each input x, asks one query to some function $f_0 \in \#\mathrm{P}$ and outputs $g(x)$, if the majority exists. Noting that $\mathrm{PF} \circ \#\mathrm{P} = \mathrm{PF}^{\#\mathrm{P}[1]}$, this clearly proves the theorem. As we do not have to worry about detecting the nonexistence of the majority, we may define the function $\phi : \mathbf{N}^* \to \mathbf{N}$ as the output of M_0.

Let $x \in \Sigma^n$ and p be a polynomial such that for all $i \leq e(x)$, $f(i,x) < 2^{p(n)}$ and $e(x) < 2^{p(n)}$. Let m_i denote the i-th prime number. By the Prime Number Theorem, $m_i \leq 2i^2$, for every $i \geq 1$. Hence, primes $m_1, \ldots, m_{p(n)}$ are computable within polynomial time in n. Also, note that $f(i,x) < m_1 \cdots m_{p(n)}$, for all $i \leq e(x)$.

We define a function u' as follows. For all strings x and integers i,j,k such that $1 \leq i \leq e(x)$, $1 \leq j \leq p(n)$, and $0 \leq k < m_j$,

$$u'(i,x,j,k) = (f(i,x) + (m_j - k))^{m_j - 1}.$$

Clearly, u' is in #P. By the Fermat's little Theorem, for all integers i,j,k such that $1 \leq i \leq e(x)$, $1 \leq j \leq p(n)$, and $0 \leq k < m_j$, we have

- $f(i,x) \equiv k \pmod{m_j} \implies u'(i,x,j,k) \equiv 0 \pmod{m_j}$, and
- $f(i,x) \not\equiv k \pmod{m_j} \implies u'(i,x,j,k) \equiv 1 \pmod{m_j}$.

Apply Lemma 3.1 to $T'(i,x,j,k) = u'(i,x,j,k) + (m_j - 1)$ and $q = p$. Then we get $T \in \#\mathrm{P}$ satisfying the conditions mentioned in the lemma. Define $u = T + 1$. Then we have:

- $u'(i,x,j,k) \equiv 0 \pmod{m_j} \implies u(i,x,j,k) \equiv 0 \pmod{m_j^{p(n)}}$, and

[4] Preceding to this work, essentially the same result was proven in a quite different form in a manuscript by Beigel, Tarui, and Toda. But, in the conference version [BTT92], the result is stated in a weaker form. Though the result will appear in their journal version, since its style is quite different and does not fit in our paper, we include the full proof of the theorem.

- $u'(i,x,j,k) \equiv 1 \pmod{m_j} \Longrightarrow u(i,x,j,k) \equiv 1 \pmod{m_j^{p(n)}}$.

Define a function v by

$$v(x,j,k) = \sum_{i \le e(x)} u(i,x,j,k).$$

Clearly, v is in #P. Furthermore, for all strings x of length n and all integers j,k such that $1 \le j \le p(n)$ and $0 \le k < m_j$, we have

$$v(x,j,k) \bmod m_j^{p(n)} = ||\{\, i \le e(x) \,:\, f(i,x) \not\equiv k \pmod{m_j} \,\}||,$$

and therefore

$$e(x) - (v(x,j,k) \bmod m_j^{p(n)}) \;=\; ||\{\, i \le e(x) \,:\, f(i,x) \equiv k \pmod{m_j} \,\}||.$$

Now, suppose $g(x) \neq ?$, i.e., the majority exists. Then, for each prime m_j, there exists a unique $k_j < m_j$ such that $g(x) \equiv k_j \pmod{m_j}$. Therefore, more than $e(x)/2$ of the i's satisfy $f(i,x) \equiv k_j \pmod{m_j}$. Conversely, for all $k < m_j$ that are different from k_j, there are less than $e(x)/2$ of the i's such that $f(i,x) \equiv k \pmod{m_j}$. Thus, we observe that for every j and k with $1 \le j \le p(n)$ and $0 \le k < m_j$,

$$g(x) \equiv k \pmod{m_j} \iff e(x) - (v(x,j,k) \bmod m_j^{p(n)}) > e(x)/2.$$

By the last observation, when we get the values $v(x,j,k)$ for all j and k with $1 \le j \le p(n)$ and $0 \le k < m_j$, we can compute the unique $k_j < m_j$ such that $g(x) \equiv k_j \pmod{m_j}$. Then, using the Chinese Remainder Theorem, we can compute $g(x)$ from the m_j's and k_j's within polynomial time in n.

By using standard methods, we can construct a function f_0 in #P such that all the values $v(x,j,k)$ for all j and k with $1 \le j \le p(n)$ and $0 \le k < m_j$, are computable from $f_0(x)$ within polynomial time in n. Hence, some polynomial time-bounded deterministic query transducer M_0, given any input x, can compute $g(x)$ by asking one query, namely x, to f_0. $\square$

Theorem 3.2 states that the majority of exponentially many values of a #P function can be computed by a #P function as long as the majority exists. Can we expect from the new function to receive information on the existence of the majority? The following theorem states that we cannot expect this unless the counting hierarchy collapses.

Theorem 3.3. #P is closed under **maj** in context PF $\circ$ #P if and only if $\mathrm{P}^{\#\mathrm{P}[1]} = \mathrm{PP}^{\mathrm{PP}}$.

Proof. Suppose that #P is closed under **maj** in context PF $\circ$ #P. We will show $\mathrm{PP}^{\mathrm{PP}} \subseteq \mathrm{P}^{\#\mathrm{P}[1]}$. Let L be any set in $\mathrm{PP}^{\mathrm{PP}}$. By Lemma 2.2 (2), there exist a set $A \in \mathrm{C_=P}$ and a polynomial p such that for all $x \in \Sigma^n$,

$$\begin{aligned} x \in L &\iff ||\{\, w \in \Sigma^{p(n)} \,:\, x\#w \in A \,\}|| \ge 2^{p(n)-1} + 1, \quad \text{and} \\ x \notin L &\iff ||\{\, w \in \Sigma^{p(n)} \,:\, x\#w \in A \,\}|| \le 2^{p(n)-1} - 1. \end{aligned}$$

Furthermore, by Lemma 2.2 (1), there exist a polynomial time-bounded nondeterministic machine M and a polynomial $q(\cdot,\cdot)$ such that for all $x \in \Sigma^n$ and $w \in \Sigma^{p(n)}$, it holds that $total_M(x\#w) = 2^{q(n,p(n))}$, $acc_M(x\#w) \leq 2^{q(n,p(n))-1}$, and $x\#w \in A$ if and only if $acc_M(x\#w) = rej_M(x\#w) = 2^{q(n,p(n))-1}$.

Define f and g as follows. For each $x \in \Sigma^*$ and each $i, 1 \leq i \leq 2^{p(|x|)}$, let $f(i,x) = acc_M(x\#w)$, where w is the i-th smallest string among those of length $p(|x|)$ (in the lexicographic ordering), and let

$$g(x) = \mathrm{maj}(\, f(1,x), \ldots, f(2^{p(|x|)}, x)\,).$$

Clearly, $f \in \#\mathrm{P}$. So, by our supposition that $\#\mathrm{P}$ is closed under **maj** in context $\mathrm{PF} \circ \#\mathrm{P}$, g is in $\mathrm{PF} \circ \#\mathrm{P} = \mathrm{PF}^{\#\mathrm{P}[1]}$.

We claim that for all $x \in \Sigma^n$, $x \in L$ if and only if $g(x) = 2^{q(n,p(n))-1}$. To see one direction, assume $x \in L$. Then more than half of the strings $w \in \Sigma^{p(n)}$ satisfy $x\#w \in A$, and therefore, more than half of the integers i with $1 \leq i \leq 2^{p(n)}$ satisfy $f(i,x) = 2^{q(n,p(n))-1}$. Thus, we have $g(x) = 2^{q(n,p(n))-1}$. To see the converse, assume $x \notin L$. Then less than half of the strings $w \in \Sigma^{p(n)}$ satisfy $x\#w \in A$. This implies that less than half of the integers i with $1 \leq i \leq 2^{p(n)}$ satisfy $f(i,x) = 2^{q(n,p(n))-1}$. Thus $2^{q(n,p(n))-1}$ is not the majority of $(f(1,x), \ldots, f(2^{p(n)}, x))$.

Hence, using the one-query machine for g, we can construct a machine that accepts L in polynomial time asking one query to a $\#\mathrm{P}$ function. We leave the details to the reader.

Next suppose that $\mathrm{P}^{\#\mathrm{P}[1]} = \mathrm{PP}^{\mathrm{PP}}$. By Proposition 2.1, we have $\mathrm{CH} = \mathrm{P}^{\#\mathrm{P}[1]}$. Let $f \in \#\mathrm{P}$ and $e \in \mathrm{PF}$. It suffices to show that $g(x) = \mathrm{maj}(f(1,x), \ldots, f(e(x),x))$ is in $\mathrm{PF}^{\mathrm{CH}}$, for since $\mathrm{CH} = \mathrm{P}^{\#\mathrm{P}[1]}$, we have $g \in \mathrm{PF}^{\#\mathrm{P}[1]} = \mathrm{PF} \circ \#\mathrm{P}$.

Define a set G by $G = \{\, x\#k : x \in \Sigma^*,\ k$ is a positive integer, and $g(x) = k \,\}$. Obviously, for all $x\#k$, we have $x\#k \in G$ if and only if $f(i,x) = k$ for more than $e(x)/2$ of the integers i with $1 \leq i \leq e(x)$. We conclude that G is in $\mathrm{PP}^{\mathrm{C_=P}}$. Furthermore, define a set H by $H = \{\, x\#j : x \in \Sigma^*,\ j$ is a positive integer, $g(x) \neq\ ?$, and the j-th bit of the binary representation of $g(x)$ is $1 \,\}$. It is easy to see that H is in $\mathrm{NP}^G \subseteq \mathrm{CH}$. This implies that g is in $\mathrm{PF}^{\mathrm{CH}}$, because g is in PF^H. □

The following corollary is immediate from the theorem.

Corollary 3.4. $\#\mathrm{P}$ is closed under **maj** in context $\mathrm{PF} \circ \#\mathrm{P}$ if and only if the counting hierarchy CH collapses to $\mathrm{P}^{\#\mathrm{P}[1]}$.

4. On the Median, Plurality, and Maximum Operators

In this section, we consider the closure properties of $\#\mathrm{P}$ under the median, plurality, and maximum operators. We will show that, as for (strong) majority, $\#\mathrm{P}$ is not closed under the median or plurality operators in context $\mathrm{PF} \circ \#\mathrm{P}$, unless the counting hierarchy collapses. For the maximum operators, we can argue along the same line, but we need a slightly stronger hypothesis.

We start by considering the median operators. In light of Toda's result [Tod90] that the **mid** operators applied to polynomial-time computable functions characterize $\mathrm{PF}^{\#\mathrm{P}} = \mathrm{PF}^{\mathrm{PP}}$, we can observe that the **mid** operators are strong enough to capture the computational power of PP-computations. Our result below is inspired with this observation.

Theorem 4.1. #P is closed under **mid** in context $\mathrm{PF} \circ \#\mathrm{P}$ if and only if $\mathrm{P}^{\#\mathrm{P}[1]} = \mathrm{PP}^{\mathrm{PP}}$.

Proof. Suppose that #P is closed under **mid** in context $\mathrm{PF} \circ \#\mathrm{P}$. We will show $\mathrm{PP}^{\mathrm{PP}} \subseteq \mathrm{P}^{\#\mathrm{P}[1]}$. Let $L \in \mathrm{PP}^{\mathrm{PP}}$. By Lemma 2.2 (2), there exist a set $A \in \mathrm{C_{=}P}$ and a polynomial p such that for all $x \in \Sigma^n$,

$$\begin{aligned} x \in L &\iff ||\{\, w \in \Sigma^{p(n)} \,:\, x\#w \in A \,\}|| \geq 2^{p(n)-1} + 1, \quad \text{and} \\ x \notin L &\iff ||\{\, w \in \Sigma^{p(n)} \,:\, x\#w \in A \,\}|| \leq 2^{p(n)-1} - 1. \end{aligned}$$

Furthermore, by Lemma 2.2 (1), there is a polynomial time-bounded nondeterministic machine M and a polynomial $q(\cdot,\cdot)$ such that for all $x \in \Sigma^n$ and $w \in \Sigma^{p(n)}$, we have $total_M(x\#w) = 2^{q(n,p(n))}$, $acc_M(x\#w) \leq 2^{q(n,p(n))-1}$, and $x\#w \in A$ if and only if $acc_M(x\#w) = rej_M(x\#w) = 2^{q(n,p(n))-1}$.

Define f and g as follows. For each $x \in \Sigma^*$ and each $i, 1 \leq i \leq 2^{p(|x|)}$,

- $f(i,x) = acc_M(x\#w)$, where w is the i-th smallest string among those of length $p(|x|)$, and
- $g(x) = \mathrm{mid}(\, f(1,x), \ldots, f(2^{p(|x|)}, x)\,)$.

We claim that for every $x \in \Sigma^n$, $x \in L$ if and only if $g(x) = 2^{q(n,p(n))-1}$. Suppose $x \in L$. Then, for more than half of $w \in \Sigma^{p(n)}$, $x\#w \in A$. Thus, for more than half of integers i with $1 \leq i \leq 2^{p(n)}$, $f(i,x) = 2^{q(n,p(n))-1}$. Moreover, there is no integer i with $1 \leq i \leq 2^{p(n)}$ such that $f(i,x) > 2^{q(n,p(n))-1}$. Therefore, we have $g(x) = 2^{q(n,p(n))-1}$. On the other hand, suppose $x \notin L$. Then, for all strings w of length $p(n)$, $acc_M(x\#w) < 2^{q(n,p(n))-1}$. So, $2^{q(n,p(n))-1}$ never appears in $(f(1,x), \ldots, f(2^{p(n)}, x))$ and hence, it cannot be $g(x)$. Thus, the claim holds. Since g is in $\mathrm{PF} \circ \#\mathrm{P} = \mathrm{PF}^{\#\mathrm{P}[1]}$ by our assumption, we can conclude that L is in $\mathrm{P}^{\#\mathrm{P}[1]}$.

Conversely, suppose $\mathrm{PP}^{\mathrm{PP}} = \mathrm{P}^{\#\mathrm{P}[1]}$. Let $g(x) = \mathrm{mid}(\, f(1,x), \ldots, f(e(x), x)\,)$, where f is a function in #P and e is a function in PF. We will show that g is in $\mathrm{PF}^{\mathrm{CH}}$.

For all $x \in \Sigma^n$ and all positive integers k, $g(x) = k$ if and only if the following conditions are satisfied:

(1) $f(i,x) = k$ for some i with $1 \leq i \leq e(x)$,

(2) $||\{\, i \leq e(x) \,:\, f(i,x) < k \,\}|| < e(x)/2$, and

(3) $||\{\, i \leq e(x) \,:\, f(i,x) > k \,\}|| \leq e(x)/2$.

Define $G = \{x\#k \;:\; g(x) = k\}$. From the above conditions, we have $G \in \mathrm{PP}^{\mathrm{PP}}$. Define $H = \{x\#j \;:\; \text{the } j\text{-th bit of the binary representation of } g(x) \text{ is } 1\,\}$. Clearly, H is in NP^G and hence in CH. Since $g \in \mathrm{PF}^H$, we conclude that g is in $\mathrm{PF}^{\mathrm{CH}}$, which is, by our assumption combined with Proposition 2.1, $\mathrm{PF}^{\#\mathrm{P}[1]}$. □

Corollary 4.2. #P is closed under **mid** in context PF∘#P if and only if the counting hierarchy CH collapses to $\mathrm{P}^{\#\mathrm{P}[1]}$.

Next, we consider the plurality operators. Since there is a certain similarity between plurality and majority, one might expect that one can somehow simulate the majority operators by the plurality operators. The proof of the following result is based on this intuition.

Theorem 4.3. #P is closed under **plu** in context PF ∘ #P if and only if $\mathrm{P}^{\#\mathrm{P}[1]} = \mathrm{PP}^{\mathrm{PP}}$.

Proof. Assume that #P is closed under **plu** in context PF ∘ #P. We will show that $\mathrm{PP}^{\mathrm{PP}} \subseteq \mathrm{P}^{\#\mathrm{P}[1]}$. Let $L \in \mathrm{PP}^{\mathrm{PP}}$. By Lemma 2.2 (2), there exist a set $A \in \mathrm{C_=P}$ and a polynomial p such that for all $x \in \Sigma^n$,

$$\begin{array}{lcll} x \in L & \Longleftrightarrow & \|\{\, w \in \Sigma^{p(n)} : x\#w \in A\,\}\| \geq 2^{p(n)-1}+1, & \text{and} \\ x \notin L & \Longleftrightarrow & \|\{\, w \in \Sigma^{p(n)} : x\#w \in A\,\}\| \leq 2^{p(n)-1}-1. & \end{array}$$

Furthermore, by Lemma 2.2 (1), there is a polynomial time-bounded nondeterministic machine M and a polynomial $q(\cdot,\cdot)$ such that for all $x \in \Sigma^n$ and $w \in \Sigma^{p(n)}$, we have $total_M(x\#w) = 2^{q(n,p(n))}$, $acc_M(x\#w) \leq 2^{q(n,p(n))-1}$, and $x\#w \in A$ if and only if $acc_M(x\#w) = rej_M(x\#w) = 2^{q(n,p(n))-1}$.

We define N to be a nondeterministic machine that, given an input of the form $x\#wb$ with $|w| = p(|x|)$ and $b \in \{0,1\}$, operates as follows:

(1) If $b = 0$, then N simulates M on input $x\#w$.

(2) If $b = 1$ and the last bit of w is 0, then N nondeterministically guesses u of length $q(|x|, p(|x|))$ and halts in an accepting state.

(3) If $b = 1$ and the last bit of w is 1, then N nondeterministically guesses u of length $q(|x|, p(|x|))$ and halts in a rejecting state.

For any $x \in \Sigma^n$, the following facts on N follow immediately:

(a) For exactly one forth of strings v of length $p(n)+1$, $acc_N(x\#v) = 0$.

(b) For exactly one fourth of strings v of length $p(n)+1$, $acc_N(x\#v) = 2^{q(n,p(n))}$.

(c) If $x \in L$, then for more than one fourth of strings v of length $p(n)+1$, $acc_N(x\#v) = 2^{q(n,p(n))-1}$.

(d) If $x \notin L$, then for less than one fourth of strings v of length $p(n)+1$, $acc_N(x\#v) = 2^{q(n,p(n))-1}$.

Now define functions f and g as follows. For each $x \in \Sigma^n$ and $i, 1 \leq i \leq 2^{p(n)+1}$, $f(i,x) = acc_N(x\#v)$, where v is the i-th smallest string among those of length $p(n)+1$, and for each x,

$$g(x) \ = \ \mathrm{plu}^*(\ f(1,x),\ldots,f(2^{p(|x|)+1},x)\ ).$$

We claim that for all $x \in \Sigma^*$ of length n, $x \in L$ if and only if $g(x) = 2^{q(n,p(n))-1}$. To see one direction, assume $x \in L$. Then, by condition (c) above, more than one forth of the integers i with $1 \leq i \leq 2^{p(n)+1}$ satisfy $f(i,x) = 2^{q(n,p(n))-1}$. Moreover, by conditions (a), (b), and (d), for all positive integers k other than $2^{q(n,p(n))-1}$, there are less than one forth of the integers i with $1 \leq i \leq 2^{p(n)+1}$ such that $f(i,x) = k$. Therefore, $2^{q(n,p(n))-1}$ is the most commonly occurring number in $(f(1,x),\ldots,f(2^{p(n)+1},x))$; that is, $g(x) = 2^{q(n,p(n))-1}$. To see the converse, assume $x \notin L$. Then, by condition (a) above, one forth of the integers i with $1 \leq i \leq 2^{p(n)+1}$ satisfy $f(i,x) = 0$. On the other hand, by condition (d), less than one forth satisfy $f(i,x) = 2^{q(n,p(n))-1}$. Thus $2^{q(n,p(n))-1}$ is not a most commonly occurring number in $(f(1,x),\ldots,f(2^{p(n)+1},x))$; that is, $g(x) \neq 2^{q(n,p(n))-1}$.

Since g is in $\mathrm{PF}^{\#\mathrm{P}[1]}$ by our assumption on the closure property of #P under the plurality operators, we can conclude that L is in $\mathrm{P}^{\#\mathrm{P}[1]}$.

To show the converse implication, assume that $\mathrm{PP}^{\mathrm{PP}} = \mathrm{P}^{\#\mathrm{P}[1]}$. Let $g(x) = \mathrm{plu}^*(\ f(1,x),\ldots,f(e(x),x)\ )$, where f is a function in #P and e is a function in PF. We show that g is in $\mathrm{PF}^{\mathrm{CH}}$.

We first define a set G by $G = \{x\#k \ : \ g(x) = k\}$. The following characterization of G is immediate from the definition of the plu* operator: for all $x \in \Sigma^*$ and all integers k, we have $x\#k \in G$ if and only if

(i) $\forall k' \ [\ ||\{\, i \leq e(x) \ : \ f(i,x) = k' \,\}|| \ \leq \ ||\{\, i \leq e(x) \ : \ f(i,x) = k \,\}||\ ]$, and

(ii) $\forall k' < k \ [\ ||\{\, i \leq e(x) \ : \ f(i,x) = k' \,\}|| \ < \ ||\{\, i \leq e(x) \ : \ f(i,x) = k \,\}||\ ]$.

This implies that G is in co-$\mathrm{NP}^{\mathrm{PP}^{\mathrm{PP}}} \subseteq \mathrm{CH}$. Next, we define a set H by $H = \{x\#j \ : \ \text{the } j\text{-th bit of } g(x) \text{ is one}\ \}$. It obvious that H is in NP^G. Hence H is also in CH. Since g is in PF^H, we can conclude that g is in $\mathrm{PF}^{\mathrm{CH}}$. Combining Proposition 2.1 with our assumption, this implies that g is in $\mathrm{PF}^{\#\mathrm{P}[1]}$. $\square$

Corollary 4.4. #P is closed under **plu** in context $\mathrm{PF} \circ \#\mathrm{P}$ if and only if the counting hierarchy CH collapses to $\mathrm{P}^{\#\mathrm{P}[1]}$.

Finally, we consider the maximum operators. Here, we have a slightly different result, which indicates in turn that the maximum operators are weaker than the other operators considered so far. Krentel [Kre88] showed that the maximum operators applied to polynomial-time computable functions characterize $\mathrm{PF}^{\mathrm{NP}}$. By this result, we can observe that the maximum operators are strong enough to capture the computational power of NP-computations. The following result is inspired with this observation.

Theorem 4.5. #P is closed under **max** in context PF $\circ$ #P if and only if $\mathrm{P}^{\#\mathrm{P}[1]} = \mathrm{NP}^{\mathrm{PP}}$.

Proof. Assume that #P is closed under **max** in context PF $\circ$ #P. We will show that every language in $\mathrm{NP}^{\mathrm{PP}}$ belongs to $\mathrm{P}^{\#\mathrm{P}[1]}$. Let L be in $\mathrm{NP}^{\mathrm{PP}}$. By Lemma 2.2 (3), there exist a set $A \in \mathrm{C}_{=}\mathrm{P}$ and a polynomial p such that for all $x \in \Sigma^*$,

$$x \in L \iff ||\{\, w \in \Sigma^{p(|x|)} \ : \ x\#w \in A \,\}|| \geq 1.$$

Moreover, by Lemma 2.2 (1), there exist a polynomial time-bounded nondeterministic machine M and a polynomial $q(\cdot,\cdot)$ such that for all $x \in \Sigma^n$ and all $w \in \Sigma^{p(n)}$, we have $total_M(x\#w) = 2^{q(n,p(n))}$, $acc_M(x\#w) \leq 2^{q(n,p(n))-1}$, and $x \in A$ if and only if $acc_M(x\#w) = rej_M(x\#w) = 2^{q(n,p(n))-1}$.

Define functions f and g as follows. For each $x \in \Sigma^n$ and $i, 1 \leq i \leq 2^{p(n)}$, $f(i,x) = acc_N(x\#w)$, where w is the i-th smallest string in $\Sigma^{p(n)}$, and for each x,

$$g(x) \ = \ \max(\, f(1,x), \ldots, f(2^{p(n)}, x)\,).$$

We claim that for all $x \in \Sigma^*$ of length n, $x \in L$ if and only if $g(x) = 2^{q(n,p(n))-1}$. To see one direction, assume $x \in L$. Then there exists $w \in \Sigma^{p(n)}$ such that $x\#w \in A$; that is, there exists an integer $i, 1 \leq i \leq 2^{p(n)}$, such that $f(i,x) = 2^{q(n,p(n))-1}$. Since $f(i,x) \leq 2^{q(n,p(n))-1}$ for all integers i, we see that $g(x) = 2^{q(n,p(n))-1}$. To see the converse, assume $x \notin L$. Then there exists no string w of length $p(n)$ such that $x\#w \in A$; that is, for all integers i with $1 \leq i \leq 2^{p(n)}$, we have $f(i,x) < 2^{q(n,p(n))-1}$. Thus we get that $g(x) < 2^{q(n,p(n))-1}$. Thus the claim holds. Since g is in PF $\circ$ #P $= \mathrm{PF}^{\#\mathrm{P}[1]}$ by our assumption, the above observation implies $L \in \mathrm{P}^{\#\mathrm{P}[1]}$.

To show the converse implication, assume $\mathrm{NP}^{\mathrm{PP}} = \mathrm{P}^{\#\mathrm{P}[1]}$. Let $g(x) = \max(\, f(1,x), \ldots, f(e(x), x)\,)$, where f is a function in #P and e is a function in PF. We show that g is in $\mathrm{PFH}^{\mathrm{PP}}$. Then we can conclude, by Proposition 2.1, that g is in $\mathrm{PF}^{\#\mathrm{P}[1]}$.

Define $G = \{x\#k \ : \ g(x) = k\}$. It is obvious that for all $x \in \Sigma^*$ and all integers k, we have $x\#k \in G$ if and only if (i) there exists some $i \leq e(x)$ such that $f(i,x) = k$, and (ii) for all $j \leq e(x)$, $f(j,x) \leq k$. From (i) and (ii) we get that G is in $\mathrm{PH}^{\mathrm{PP}}$. Furthermore, define $H = \{x\#j \ : \ \text{the } j\text{-th bit of the binary representation of } g(x) \text{ is } 1\,\}$. It is obvious that H is in $\mathrm{NP}^{G} \subseteq \mathrm{PH}^{\mathrm{PP}}$. Since g is in PF^{H}, we conclude that g is in $\mathrm{PFH}^{\mathrm{PP}}$. $\square$

Corollary 4.6. #P is closed under **max** in context PF $\circ$ #P if and only if $\mathrm{PH}^{\mathrm{PP}} = \mathrm{P}^{\#\mathrm{P}[1]}$.

5. Concluding Remarks

We have studied closure properties of #P in context PF $\circ$ #P. As we have mentioned in Section 1, we are not restricting the context to the one we have chosen. Indeed, one can think of any complexity class with access to #P as context. Regarding the operators in this paper, however, our proof techniques can be carried over to larger classes. For example, we can show that #P is closed under **maj** in context $\mathrm{PF}^{\#\mathrm{P}}$ if and only if $\mathrm{P}^{\#\mathrm{P}} = \mathrm{PP}^{\mathrm{PP}}$.

But, for smaller classes, the situation seems to be different. In Section 1, we have mentioned that the modified subtraction of #P functions can be retrieved from another #P function. As a matter of fact, the post-computation can even be done by small circuits of constant depth, i.e., by AC^0-circuits. So, we may say that #P is closed under modified subtraction in context $\mathrm{AC}^0 \circ \#\mathrm{P}$. But, this argument does not seem to hold for several other 'hard' closure properties in [OH93]. Consider, for example, $\lfloor f(x)/g(x) \rfloor$ for $f \in \#\mathrm{P}$ and nonzero $g \in \#\mathrm{P}$. It is easy to design a #P function, say $h(x) = f(x)2^{p(|x|)} + g(x)$ for some suitably large polynomial p, from which logarithmically depth-bounded circuits can compute the division (see [BCH86]). But, combining the result of Furst, Saxe and Sipser [FSS84] with the easily provable fact that the parity function is AC^0-reducible to integer division, it is seen that no AC^0 circuit can compute the division from h above. Thus, studying the closure properties of #P in context $\mathrm{AC}^0 \circ \#\mathrm{P}$ would give us other insight on the nature of #P-computations and hence of the counting hierarchy. Particularly, as a first trial along this line, it is interesting to ask whether there is a #P function from which the division can be computed by AC^0 circuits.

It would be meaningful to continue the investigation along the line described in this paper. In particular, it would be interesting to find more nontrivial closure properties of #P with respect to some reasonable contexts. Especially, exhibiting an operator, like majority, that, with a slight change in the definition, will drastically change its behavior as closure properties, will shed light on the properties of its related complexity classes, and may give some hint on how to actually separate those classes.

Acknowledgement

The material presented in this paper was initiated and studied while the authors were visiting Lane Hemaspaandra at the University of Rochester. The authors would like to thank him for leading them to this subject and his hospitality. Without him, the present paper would not have existed. They also would like to thank Jun Tarui for helpful comments related to Theorem 3.2.

References

[BDG88] J. Balcázar, J. Díaz, and J. Gabarró, *Structural Complexity I*, EATCS Monographs on Theoretical Computer Science, Springer-Verlag (1988).

[BDG91] J. Balcázar, J. Díaz, and J. Gabarró, *Structural Complexity II*, EATCS Monographs on Theoretical Computer Science, Springer-Verlag (1991).

[BCH86] P. Beame, S. Cook and H. Hoover, Log depth circuits for division and related problems, *SIAM J. Comput.* **15** (1986), 994–1003.

[BHW91] R. Beigel, L. Hemachandra, and G. Wechsung, Probabilistic polynomial time is closed under parity reductions, *Inform. Proc. Lett.* **37** (1991), 91–94.

[BRS91] R. Beigel, N. Reingold, and D. Spielman, PP is closed under intersection, In *Proc. 23rd Symp. on Theory of Comput.*, ACM Press (1991), pp. 1–9.

[BTT92] R. Beigel, J. Tarui, and S. Toda, On probabilistic ACC circuits with an exact-threshold output gate, In *Proc. 3rd Int. Symp. on Algorithms and Computation*, Lecture Notes in Computer Science #650 (1992), pp. 420–429.

[CH90] J. Cai and L. Hemachandra, On the power of parity polynomial time, *Math. Systems Theory* **23** (1990), 95–106.

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

[FFK94] S. Fenner, L. Fortnow, and S. Kurtz, Gap-definable counting classes, *J. Comput. System Sci.* **48** (1994), 116–148.

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

[FSS84] M. Furst, J. Saxe and M. Sipser, Party, circuits and the polynomial-time hierarchy, *Math. Systems Theory* **17** (1984), 13–27.

[KST89] J. Köbler, U. Schöning, and J. Torán, On counting and approximation, *Acta Inform.* **26** (1989), 363–379.

[Kre88] M. Krentel, The complexity of optimization Problems, *J. Comput. System Sci.* **36** (1988), 490–509.

[OH93] M. Ogiwara and L. Hemachandra, A complexity theory for feasible closure properties, *J. Comput. System Sci.* **46** (1993), 295–325.

[PZ83] C. Papadimitriou and S. Zachos, Two remarks on the power of counting, In *Proc. 6th GI Conf. on Theoret. Comput. Sci.*, Lecture Notes in Computer Science #145 (1983), pp. 269–276.

[Sch90] U. Schöning, The power of counting, In *Complexity Theory Retrospective* (A. Selman, ed.), Springer-Verlag (1990), 204–223.

[Sim75] J. Simon, *On some central problems in computational complexity,* PhD thesis, Cornell University, Ithaca, N.Y., January 1975. Available as Cornell Department of Computer Science Technical Report TR75-224.

[Tod90] S. Toda, Simple characterization of P(#P) and complete problems, *J. Comput. System Sci.* **49** (1994), 1–17.

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

[Tor91] J. Toran, An oracle characterization of the counting hierarchy, *J. Assoc. Comput. Mach.* **38** (1991), 753–774.

[Val79] L. Valiant, The complexity of enumeration and reliability problems, *SIAM J. Comput.* **8** (1979), 410–421.

[Wag86a] K. Wagner, Some observations on the connection between counting and recursion, *Theoret. Comput. Sci.* **47** (1986), 131–147.

[Wag86b] K. Wagner, The complexity of combinatorial problems with succinct input representations, *Acta Inform.* **23** (1986), 325–356.