Chicago Journal of Theoretical Computer Science

The MIT Press

Volume 1999, Article 2

Complements of Multivalued Functions

ISSN 1073–0486. MIT Press Journals, Five Cambridge Center, Cambridge, MA 02142-1493 USA; (617)253-2889; *journals-orders@mit.edu*, *journals-info@mit.edu*. Published one article at a time in LaTeX source form on the Internet. Pagination varies from copy to copy. For more information and other articles see:

- *http://mitpress.mit.edu/CJTCS/*
- *http://www.cs.uchicago.edu/publications/cjtcs/*
- *ftp://mitpress.mit.edu/pub/CJTCS*
- *ftp://cs.uchicago.edu/pub/publications/cjtcs*

The *Chicago Journal of Theoretical Computer Science* is abstracted or indexed in *Research Alert,®SciSearch,®Current Contents®/Engineering Computing & Technology*, and *CompuMath Citation Index.®*

The *Chicago Journal of Theoretical Computer Science* is a peer-reviewed scholarly journal in theoretical computer science. The journal is committed to providing a forum for significant results on theoretical aspects of all topics in computer science.

Complements of Multivalued Functions

Stephen Fenner Frederic Green Steven Homer
Alan L. Selman Thomas Thierauf Heribert Vollmer

19 March, 1999

Abstract

We study the class coNPMV of complements of NPMV functions. Though defined symmetrically to NPMV, this class exhibits very different properties. We clarify the complexity of coNPMV by showing that it is essentially the same as that of $\mathrm{NPMV}^{\mathrm{NP}}$. Complete functions for coNPMV are exhibited and central complexity-theoretic properties of this class are studied. We show that computing maximum satisfying assignments can be done in coNPMV, which leads us to a comparison of NPMV and coNPMV with Krentel's classes MaxP and MinP. The difference hierarchy for NPMV is related to the query hierarchy for coNPMV. Finally, we examine a functional analogue of Chang and Kadin's relationship between a collapse of the Boolean hierarchy over NP and a collapse of the polynomial-time hierarchy.

1 Introduction

1-1 Consider the complexity class NPMV of partial multivalued functions that are computed nondeterministically in polynomial time. As this class captures the complexity of computing witnesses of sets in NP, by studying this class, and more generally, by studying relations among complexity classes of partial multivalued functions, we directly contribute to understanding the complexity of computing witnesses. It is well known that a partial multivalued function f belongs to NPMV if and only if it is polynomial length-bounded and $\mathrm{graph}(f) = \{\, \langle x, y\rangle : y$ is a value of $f(x)\,\}$ belongs to NP.

1-2 Now consider the class coNPMV. We will give a formal definition in the preliminaries section below. It will follow from the definition that a partial multivalued function f belongs to coNPMV if and only if it is polynomial length-bounded and $\mathrm{graph}(f)$ belongs to coNP. Given this symmetry, graphs of functions in NPMV are in NP while graphs of functions in coNPMV are in coNP, and given what we know about NP and coNP, one might expect that coNPMV has essentially the same complexity as NPMV. Indeed, it is easy to see that coNPMV = NPMV if and only if NP = coNP. However, the point of this paper is to show that in many ways coNPMV is a more powerful class than is NPMV. One can derive more information from computing the complement of a function in NPMV than from computing the function. For one example of this phenomenon, we prove here that coNPMV is not included in $\mathrm{FP}^{\mathrm{NPMV}}$ unless the polynomial hierarchy collapses. (This is an extension of a result of Fenner et al [FHOS97].) Thus, a coNPMV oracle provides more information than an NPMV oracle. This is surprising since function oracles, just as set oracles, provide knowledge about both their domains and their co-domains.

1-3 We will define many-one reductions between multivalued functions. This will be a straightforward adaptation of the many-one metric reducibility of Krentel [Kre88]. In Section 3, we will consider many-one complete functions for coNPMV.

1-4 Consider the partial multivalued function sat, defined so that y is a value of $sat(\varphi)$ if and only if y is a satisfying assignment of Boolean formula φ. The function *sat* is complete for NPMV. Nevertheless, in Section 4 we will see that sat and similar functions belong to coNPMV. Even the seemingly more powerful $\mathrm{FP}^{\mathrm{NP}}$-complete function *maxsat*, that gives the maximum satisfying assignment of a formula, is contained in coNPMV. However, we will see that *neither* NPMV nor $\mathrm{FP}^{\mathrm{NP}}$ are contained in coNPMV, and hence coNPMV is not closed under metric many-one reductions, unless the polynomial-time hierarchy collapses. Clearly, these function classes have strange closure properties, which we describe below.

1-5 As an upper bound on the complexity of coNPMV, we show that, for any $k \geq 2$,

$$\mathrm{coNPMV} \subseteq \mathrm{NPMV}(2) \subseteq \mathrm{NPMV}(k) \subseteq$$

$$\mathrm{NPMV}(k+1) \subseteq \mathrm{NPMV}(\mathrm{P}oly) \subseteq \mathrm{NPMV}^{\mathrm{NP}},$$

where $\mathrm{NPMV}(k)$ is the k-th level of the difference hierarchy for NPMV as defined by Fenner et al. [FHOS97].

1-6 On the other hand, even though there is an infinite hierarchy of complexity classes between coNPMV and $\mathrm{NPMV}^{\mathrm{NP}}$ (the difference hierarchy over NPMV does not collapse unless the polynomial-time hierarchy collapses [FHOS97]), our results suggest that the complexity of coNPMV is essentially the same as the complexity of $\mathrm{NPMV}^{\mathrm{NP}}$: We prove in Section 5 that $\mathrm{NPMV}^{\mathrm{NP}} = \pi_2^1 \circ \mathrm{coNPMV}$ (where π_2^1 is the projection function that maps a pair of strings to its first component). It follows that $\mathrm{NPMV}^{\mathrm{NP}}$ is the closure of coNPMV under metric many-one reductions.

1-7 In Section 6, we show that if the difference hierarchy for NPMV collapses, then the NPMV oracle hierarchy collapses. This is the functional analogue of the well-known result by Chang and Kadin relating a collapse of the Boolean hierarchy over NP to a collapse of the polynomial-time hierarchy.

1-8 Finally, we remark that the phenomenon that universal quantification seems to lead to larger function classes was previously observed by Toda. We show in Section 7 how this observation follows from our results.

2 Preliminaries

2-1 We fix Σ to be the finite alphabet $\{0, 1\}$. Let $<$ denote the standard lexicographic order on Σ^*. For $n \geq 0$ we define $\Sigma^n = \{x \in \Sigma^* \mid |x| = n\}$. By $\langle \cdot, \cdot \rangle$ we denote a standard pairing function on $\Sigma^* \times \Sigma^*$.

2-2 We use the standard complexity classes P and NP for (nondeterministic) polynomial time, Σ_k^p and $\Delta_k^p = \mathrm{P}^{\Sigma_{k-1}^p}$ for the levels of the polynomial-time hierarchy, and $\mathrm{NP}(k)$ for the levels of the Boolean hierarchy, for $k \geq 1$.

2-3 Let f be a relation on $\Sigma^* \times \Sigma^*$. We will call f a *(partial) multivalued function* from Σ^* to Σ^*. By $f(x) \mapsto y$ we denote that $(x, y) \in f$ and say that f *maps* x *to* y. By $\text{set-}f(x)$ we denote the set of outcomes of f on x, $\text{set-}f(x) = \{\, y : f(x) \mapsto y \,\}$. The *graph of* f is $\mathrm{graph}(f) = \{\, \langle x, y \rangle : f(x) \mapsto y \,\}$. The *domain of* f, $\mathrm{dom}(f)$, is the set of x where $\text{set-}f(x)$ is nonempty. We will say that f is undefined at x if $x \notin \mathrm{dom}(f)$. The domain of a class $\mathcal{F}$ of functions is $\mathrm{dom}(\mathcal{F}) = \{\mathrm{dom}(f) \mid f \in \mathcal{F}\}$.

2-4

2-5 Given partial multivalued functions f and g, define g to be a *refinement of* f if $\mathrm{dom}(g) = \mathrm{dom}(f)$ and $\mathrm{graph}(g) \subseteq \mathrm{graph}(f)$. Let $\mathcal{F}$ and $\mathcal{G}$ be classes of partial multivalued functions. Purely as a convention, if f is a partial multivalued function, we define $f \in_c \mathcal{G}$ if $\mathcal{G}$ contains a refinement of f, and we define $\mathcal{F} \subseteq_c \mathcal{G}$ if, for every $f \in \mathcal{F}$, $f \in_c \mathcal{G}$. This notation is consistent with our intuition that $\mathcal{F} \subseteq_c \mathcal{G}$ should entail that the complexity of computing values of functions in $\mathcal{F}$ is not greater than the complexity of computing values of functions in $\mathcal{G}$.

2-6 A transducer T is a nondeterministic Turing machine with a read-only input tape, a write-only output tape, read-write work tapes, and accepting states in the usual manner. T computes a value y on an input string x if there is an accepting computation of T on x for which y is the final content of T's output tape. (In this case, we will write $T(x) \mapsto y$.) Such transducers compute partial, multivalued functions. (As transducers do not typically accept all input strings, when we write "function," "partial function" is always intended. If a function f is total, it will always be explicitly noted.)

2-7 The following classes of partial functions were first defined by Book, Long, and Selman [BLS84].

2-8

- NPMV is the set of all partial, multivalued functions computed by nondeterministic polynomial time-bounded transducers;
- NPSV is the set of all $f \in \mathrm{NPMV}$ that are single-valued;
- FP is the set of all partial functions computed by deterministic polynomial time-bounded transducers.

2-9 A function f belongs to NPMV if and only if it is polynomially length-bounded and $\mathrm{graph}(f)$ belongs to NP. In this paper we will adopt the convention, different from other papers on the subject, that all outputs of a function $f \in$ NPMV on input x are of the same length, namely, $p(|x|)$, where p is some polynomial. This convention is merely for convenience and can easily be removed in all our results by using a padding argument.

2-10 The domain of every function in NPMV belongs to NP. An example is *sat*, which maps Boolean formulas to their satisfying assignments.

2-11 Fenner et al. [FHOS97] define the *difference hierarchy over* NPMV as follows. Let $\mathcal{F}$ be a class of partial multivalued functions. A partial multivalued function f is in $\mathrm{co}\,\mathcal{F}$ if there exist $g \in \mathcal{F}$ and a polynomial p such that for every x,

$$\text{set-}f(x) \;=\; \Sigma^{p(|x|)} - \text{set-}g(x).$$

Let $\mathcal{F}$ and $\mathcal{G}$ be two classes of partial multivalued functions. A partial multivalued function h is in $\mathcal{F} \wedge \mathcal{G}$, respectively $\mathcal{F} \vee \mathcal{G}$, if there exist partial multivalued functions $f \in \mathcal{F}$ and $g \in \mathcal{G}$ such that for every x,

$$\text{set-}h(x) = \text{set-}f(x) \cap \text{set-}g(x), \text{ respectively}$$
$$\text{set-}h(x) = \text{set-}f(x) \cup \text{set-}g(x).$$

Let $\mathcal{F} - \mathcal{G}$ denote $\mathcal{F} \wedge \mathrm{co}\,\mathcal{G}$. Then, $\mathrm{NPMV}(k)$ is the class of partial multivalued functions defined in the following way:

$$\text{NPMV}(1) = \text{NPMV},$$

and, for $k \geq 2$,

$$\text{NPMV}(k) = \text{NPMV} - \text{NPMV}(k-1).$$

Fenner et al. prove that, for every $k \geq 1$, $f \in \text{NPMV}(k)$ if and only if f is polynomially length-bounded and $\text{graph}(f) \in \text{NP}(k)$.

2-12 In particular, we are interested in the class coNPMV. It follows that a function f belongs to coNPMV if and only if it is polynomially length-bounded and $\text{graph}(f)$ belongs to coNP. Observe that the classes NPMV and coNPMV satisfy the nice symmetry that graphs of functions in the former class are in NP and those in the latter class are in coNP.

2-13 Just as the definition of the Boolean hierarchy over NP leads to the class $\text{NP}(\text{P}oly)$ (see [Wag90]), we now introduce the class $\text{NPMV}(\text{P}oly)$. It can be shown that a function h belongs to $\text{NPMV}(k)$ if and only if there is a 2-ary function $f \in \text{NPMV}$ such that

$$\begin{aligned} \text{set-}h(x) = \text{set-}f(x,k) - \big(&\text{set-}f(x,k-1) \\ &-\big(\text{set-}f(x,k-2) \\ &-\big(\cdots - \text{set-}f(x,1)\cdots\big)\big)\big). \end{aligned}$$

We say that $h \in \text{NPMV}(\text{P}oly)$ if and only if there is a function $f \in \text{NPMV}$ and a polynomial p such that

$$\begin{aligned} \text{set-}h(x) = \text{set-}f(x,p(|x|)) - \big(&\text{set-}f(x,p(|x|)-1) \\ &-\big(\text{set-}f(x,p(|x|)-2) \\ &-\big(\cdots - \text{set-}f(x,1)\ldots\big)\big)\big). \end{aligned}$$

The above-mentioned result by Fenner et al. can be extended to show that $f \in \text{NPMV}(\text{P}oly)$ if and only if f is polynomially length-bounded and $\text{graph}(f) \in \text{NP}(\text{P}oly)$.

2-14 The primary new contribution of Fenner et al. is the development of hierarchies of classes of functions that access classes of partial functions as oracles. This development is based on the following description of oracle Turing machines with oracles that compute partial functions. Assume first that the oracle is a single valued partial function g. Let $\perp$ be a symbol not belonging to the finite alphabet Σ. In order for a machine M to access a partial function oracle, M has a write-only input oracle tape, a separate read-only output oracle tape, and a special oracle call state q. To query g on a string x, M enters state q with x on the oracle input tape in the usual fashion. The oracle then returns the value $g(x)$ on the oracle output tape if the value exists, and writes $\perp$ on the tape otherwise. (It is possible that M may read only a portion of the oracle's output if the oracle's output is too long to read with the resources of M.) We shall assume, without loss of generality, that M never makes the same oracle query more than once on any possible computation path.

2-15 If g is a single-valued partial function and M is a deterministic oracle transducer as just described, then we let $M[g]$ denote the single-valued partial function computed by M with oracle g.

Definition 1 *[FHOS97] Let f and g be multivalued partial functions.* f is Turing reducible to g in polynomial time, $f \leq^{\mathrm{P}}_{\mathrm{T}} g$, *if for some deterministic polynomial-time oracle transducer M, for every single-valued refinement g' of g, $M[g']$ is a single-valued refinement of f.*

2-16 Fenner et al. prove that $\leq^{\mathrm{P}}_{\mathrm{T}}$ is a reflexive and transitive relation over the class of all partial multivalued functions.

2-17 Let $\mathcal{F}$ be a class of partial multivalued functions. $\mathrm{FP}^{\mathcal{F}}$ denotes the class of partial multivalued functions f that are $\leq^{\mathrm{P}}_{\mathrm{T}}$-reducible to some $g \in \mathcal{F}$. $\mathrm{FP}^{\mathcal{F}[k]}$ (respectively, $\mathrm{FP}^{\mathcal{F}[\log]}$) denotes the class of partial multivalued functions f that are $\leq^{\mathrm{P}}_{\mathrm{T}}$-reducible to some $g \in \mathcal{F}$ via a machine that, on input x, makes k adaptive queries (respectively, $\mathcal{O}(\log|x|)$ adaptive queries) to its oracle.

2-18 This definition template defines classes of multivalued partial functions such as $\mathrm{FP}^{\mathrm{NPMV}}$ and can easily be extended to define $\mathrm{NPMV}^{\mathrm{NPMV}}$. If $\mathcal{K}$ is a class of sets, then $\mathrm{FP}^{\mathcal{K}}$ is defined as usual, except that we allow it to compute partial functions (at the discretion of the oracle machine).

2-19 We will use the following generalization of the many-one metric reducibility of Krentel [Kre88] in order to discuss complete functions for classes of multivalued functions.

Definition 2-1 Definition 2 *Given partial multivalued functions $f, g : \Sigma^* \mapsto \Sigma^*$, we say f* is metric many-one reducible to g, *or symbolically, $f \leq^{\mathrm{P}}_{m} g$, if there are functions $t_1, t_2 \in$* FP *such that the multivalued partial function h defined by*

$$h(x) = t_2(x, (g \circ t_1)(x))$$

is a refinement of f, where $\text{set-}h(x)$ *is defined as*

$$\{\, t_2(x, y) : g(t_1(x)) \mapsto y \,\}.$$

Definition 2-2 *If, in addition, we have* set-$h(x) =$ set-$f(x)$ *for all* x*, we call it a* strong metric many-one reduction*, denoted by* $f \leq_{sm}^{\mathrm{P}} g$*.*

2-20 The motivation underlying this definition is that, given a value of $g(x)$, one can compute in polynomial time a value of $f(x)$. In the case of a strong reduction, one gets all values of $f(x)$ when varying over all values of $g(x)$. Obviously, $f \leq_{m}^{\mathrm{P}} g$ implies $f \leq_{\mathrm{T}}^{\mathrm{P}} g$.

2-22 The classes that we have been considering relate in interesting ways to studies of the complexity of optimization problems. In order to capture the complexity of optimization problems, Krentel [Kre88] defined the complexity classes MaxP and MinP as the functions computable by taking the maximum, respectively minimum, over sets of feasible solutions of problems in NP. Further, Krentel extended these classes to hierarchies of classes of optimization functions [Kre92]. Krentel defined these functions using the notion of a *metric Turing machine*, which we now review. Consider nondeterministic polynomial-time Turing machines that print an output value on every path. We associate with every inner node of the computation tree either the function $\min$ or the function $\max$ (for the classes MinP and MaxP, all nodes are associated with the same function). Thus, metric Turing machines define (total) functions from input words to integers via the usual bottom-up evaluation of the machine's computation tree. Since all the function classes considered in this paper are partial, we extend the metric Turing machine just defined by allowing the machine to output a special symbol $\perp$ that denotes that the computation on the corresponding path ends with an undefined result. We extend the $\min$ and $\max$ functions in the obvious way: define $\max(x, \perp) = \max(\perp, x) = x$ and $\min(x, \perp) = \min(\perp, x) = x$, for all x (including $\perp$ itself). Vollmer and Wagner [VW93, VW95] gave a detailed structural examination of Krentel's hierarchy. Here, we just define the class MaxP using an operator-characterization from [VW95]. MinP is defined analogously.

2-23

$$h \in \mathrm{MaxP} \iff \exists f,\ g \in \mathrm{FP} :\ h(x) = \max_{0 \leq y \leq g(x)} f(x, y).$$

3 Functions Complete for coNPMV

3-1 NPMV is precisely the class of functions that compute witnesses for NP sets in the following sense: For any set $L \in \mathrm{NP}$ there exist a set $A \in \mathrm{P}$ and a polynomial p such that for all x, we have

$$x \in L \iff \exists y \in \Sigma^{p(|x|)} :\ (x, y) \in A.$$

Any y such that $(x, y) \in A$ is called a *witness for* x (with respect to A). Clearly, there is a function $f_A \in$ NPMV such that set-$f(x)$ is exactly the set of witnesses for x. On the other hand, any NPMV function f defines a set in NP, namely $\mathrm{dom}(f)$. As a consequence of this discussion, we see that $\mathrm{dom}(\text{NPMV}) = \text{NP}$.

3-2 Next, we extend the notion of a witness to Σ_2^p. For any Σ_2^p set L there exist a set $B \in$ coNP and a polynomial p such that for all x, we have $x \in L \iff \exists y \in \Sigma^{p(|x|)} : (x, y) \in B$. A y such that $(x, y) \in B$ is called a *witness for* x (with respect to B). What function class captures the computation of witnesses for Σ_2^p sets? Since $\Sigma_2^p = \text{NP}^{\text{NP}}$, certainly witnesses can be computed in NPMV^{NP}. However, we will see below that the seemingly weaker class coNPMV already suffices to do so.

3-3 Let us consider set L again. We may safely assume that, for all $(x, y) \in B$, we have $y \in \Sigma^{p(|x|)}$. Since $B \in$ coNP, it is then the graph of a coNPMV function f, so that set-$f(x)$ is exactly the set of witnesses for x. Hence, coNPMV can compute witnesses for sets in Σ_2^p. Conversely, for any coNPMV function f, we have $\mathrm{dom}(f) \in \Sigma_2^p$. This is because, for any x, $x \in \mathrm{dom}(f) \iff \exists y \in \Sigma^{p(|x|)} : y \in$ set-$f(x)$. Thus, coNPMV is *precisely* the class of functions that computes witnesses for Σ_2^p sets. As a consequence, we have the following proposition.

Proposition 1 $\mathrm{dom}(\text{coNPMV}) = \Sigma_2^p$.

3-4 Witnesses of Σ_2^p-complete sets can give rise to complete functions for coNPMV. Consider, for example, the satisfiability problem QBF_2 for Boolean formulas with two quantifiers. Let φ be a Boolean formula in the variables $\mathrm{x} = (x_1, \ldots, x_k)$ and $\mathrm{y} = (y_1, \ldots, y_l)$. Then we define

$$\varphi(\mathrm{x}, \mathrm{y}) \in \text{QBF}_2 \iff \exists \mathrm{x}\ \forall \mathrm{y} : \varphi(\mathrm{x}, \mathrm{y}) = 1.$$

Let F_2 be the multivalued function that computes witnesses, i.e., partial assignments $\mathrm{x} = (x_1, \ldots, x_k)$, for QBF_2 formulas φ as above.

Theorem 1 F_2 *is* $\leq_{sm}^{\text{P}}$*-complete for* coNPMV.

Proof of Theorem 1 We have argued already that $F_2 \in$ coNPMV. Let f be any coNPMV function. There is an NP transducer M and a polynomial p such that for all x, we have set-$f(x) = \Sigma^{p(|x|)} -$ set-$M(x)$. We show how to compute a $y \in$ set-$f(x)$ from $F_2(\varphi_x)$, for an appropriately constructed formula φ_x.

3-5 Define a machine M' on input x as follows. First, M' guesses a $y \in \Sigma^{p(|x|)}$. Then, M' simulates M on input x. If M outputs y on the simulated path, then M' rejects. Otherwise, M' accepts.

3-6 We have to define the reduction functions t_1 and t_2 as required in Definition 2. Function t_1 is the Cook-Levin reduction[1] applied to x with M' as the underlying machine. This will give a Boolean formula φ_x that, intuitively, describes the work of M' on input x. The variables of φ_x can be partitioned into two parts, for example,

- $y_1, \ldots, y_k$, that are used to describe that M' guesses a $y \in \Sigma^{p(|x|)}$, and
- $z_1, \ldots, z_l$, that are used to describe the subsequent simulation of M.

Furthermore, from any setting of the variables $y_1, \ldots, y_k$ of φ_x, we can reconstruct in polynomial time the $y \in \Sigma^{p(|x|)}$ guessed by M'. This is done by function t_2.

3-7 Let us fix a setting of the variables $y_1, \ldots, y_k$ and let $y \in \Sigma^{p(|x|)}$ be the corresponding string guessed by M'. Then we have

$$\begin{aligned} & \forall z_1, \ldots, z_l : \; \varphi_x(y_1, \ldots, y_k, z_1, \ldots, z_l) = 1 \\ \iff \; & M' \text{ accepts on all paths following } y \\ \iff \; & y \notin \text{set-}M(x) \\ \iff \; & f(x) \mapsto y, \end{aligned}$$

and hence, $\text{set-}t_2(x, F_2 \circ t_1(x)) = \text{set-}f(x)$, where $t_1(x) = \varphi_x$.

Proof of Theorem 1 □

3-8 A crucial point in the above proof is that the Cook-Levin reduction maintains witnesses. That is, from a given assignment for the constructed formula φ_x, one can recover a corresponding path of the nondeterministic machine. Thus, any Σ_2^p-complete set sharing this property with QBF_2 defines a coNPMV-complete function in an analogous way.

3-9 As an example, consider the following set L_f. For any NPMV function f and any even-valued polynomial p such that f maps strings of length n to strings of length $p(n)$, $x \in L_f$ if and only if

$$\exists y \in \Sigma^{p(|x|)/2} \; \forall z \in \Sigma^{p(|x|)/2} : \; f(x) \not\mapsto yz.$$

In other words, string y is not a prefix of an output of $f(x)$.

3-10 Clearly, for every $f \in \mathrm{NPMV}$, we have that L_f is in Σ_2. Thus, in particular, taking $f = \mathit{sat}$, L_{sat} is Σ_2^p-complete and has the above-mentioned property. We

[1] This is the well-known reduction that transforms the computation of an NP machine to a Boolean formula that is satisfiable iff the machine accepts; see, e.g., [Coo71].

conclude that the corresponding witness function, *not-pre-sat*, is complete for coNPMV, where (by definition) *not-pre-sat*$(\varphi) \mapsto y \iff y$ is a truth assignment of the first half of φ's variables that is not a prefix of a satisfying assignment of φ.

Theorem 2 *not-pre-sat is* $\leq_{sm}^{\mathrm{P}}$*-complete for the class* coNPMV.

3-11 *Not-pre-sat* is a trivial transformation of F_2, so Theorem 2 can be seen directly via a straightforward metric reduction from F_2.

4 Properties of coNPMV

4-1 NPMV is closed under $\leq_{sm}^{\mathrm{P}}$-reductions, but not under $\leq_{m}^{\mathrm{P}}$-reductions; in fact, it is possible to have $g \in$ NPMV and $f \leq_{m}^{\mathrm{P}} g$ but $\mathrm{graph}(f)$ be noncomputable. (For example, define f to map x to two values, the first of which is either 0 or 1 and solves the halting problem on x and the second of which is the constant 10. Then clearly $\mathrm{graph}(f)$ is not computable, but the constant function 10 is a refinement of f in NPMV.) However, NPMV *is* closed under this reduction in a weaker sense, defined below.

Definition 3 *A class* $\mathcal{F}$ *is* c-closed *under reducibility* $\leq_r$*, if* $g \in \mathcal{F}$ *and* $f \leq_r g$ *implies* $f \in_c \mathcal{F}$.

4-2 It is immediate from this definition that NPMV is c-closed under $\leq_{m}^{\mathrm{P}}$-reductions. One might suspect that this same fact holds for coNPMV. However, it is quite unlikely that coNPMV is c-closed under this reducibility: otherwise, since $sat \in$ coNPMV and sat is complete for NPMV, we would get that NPMV $\subseteq_c$ coNPMV. But this seems to be very unlikely, as the following extension of a result of Fenner et al. [FHOS97] shows.

Theorem 3 NPMV $\subseteq$ coNPMV $\iff$ NPMV $\subseteq_c$ coNPMV $\iff$ NP $=$ coNP.

Proof of Theorem 3 We cycle through the implications. The first implication is trivial. For the second, let $L \in$ NP. Define function

$$\chi_L(x) \;=\; \begin{cases} 1 & \text{if } x \in L \\ \bot & \text{otherwise}\,. \end{cases}$$

Then we have $\chi_L \in$ NPMV, and hence, by assumption, $\chi_L \in$ coNPMV. Therefore, $\mathrm{graph}(\chi_L) \in$ coNP, which implies that $L \in$ coNP since $x \in L \iff (x, 1) \in \mathrm{graph}(\chi_L)$.

4-3 Now suppose that NP = coNP and let $f \in$ NPMV. Then $\mathrm{graph}(f) \in$ NP and, therefore, in coNP by assumption. Thus $f \in$ coNPMV.

Proof of Theorem 3 □

Corollary 1 coNPMV *is c-closed under* $\leq_m^{\mathrm{P}}$*-reducibility if and only if* NP $=$ coNP.

4-4 We observe that the proof of Theorem 3 shows also that NPSV $\subseteq$ coNPMV $\Longleftrightarrow$ NP $=$ coNP, even though it is fairly easy to see that NPSV$_t$, the class of all total NPSV functions, *is* contained in coNPMV. We also note that Theorem 3 extends to higher levels of the difference hierarchies over NPMV and NP, that is, NPMV$(k) \subseteq$ coNPMV$(k) \Longleftrightarrow$ NPMV$(k) \subseteq_c$ coNPMV$(k) \Longleftrightarrow$ NP$(k) =$ coNP(k). By a result of Kadin [Kad88], a collapse of the Boolean hierarchy implies a collapse of the polynomial-time hierarchy. Hence, there is likely to be a whole hierarchy between coNPMV and NPMV$^{\mathrm{NP}}$.

Theorem 4 *For all* $k \geq 2$*, we have*

$$\text{coNPMV} \subseteq \text{NPMV}(2) \subseteq \text{NPMV}(k) \subseteq \text{NPMV}(k+1) \subseteq \text{NPMV}(\mathrm{P}oly) \subseteq \text{NPMV}^{\mathrm{NP}}.$$

Furthermore, all of the inclusions are strict unless the polynomial-time hierarchy collapses.

Proof of Theorem 4 It remains to show the last inclusion. Let $f \in$ NPMV$(\mathrm{P}oly)$. Then the graph of f is in NP$(\mathrm{P}oly)$, which is known to be equal to P$^{\mathrm{NP}[\log]}$ [Wag90]. Obviously, f can be computed by an NPMV algorithm with access to a P$^{\mathrm{NP}[\log]}$ oracle: simply guess an output of f and, querying its graph, check that the guess is correct. Thus, NPMV$(\mathrm{P}oly) \subseteq$ NPMV$^{\mathrm{P}^{\mathrm{NP}[\log]}} \subseteq$ NPMV$^{\mathrm{NP}}$.

Proof of Theorem 4 □

4-5 Under the likely assumption that $\mathrm{NP} \neq$ coNP, we see, by Theorem 3, that the class NPMV is not included in coNPMV, even though the function sat, which is complete for NPMV, belongs to coNPMV. This phenomenon happens again for *maxsat*, the function that maps a Boolean formula to its lexicographically largest satisfying assignment. Fenner et al. [FHOS97] show that *maxsat* $\in$ NPMV(2). In fact, it is even in coNPMV. However, we will show that the corresponding classes, namely MaxP or FP$^{\mathrm{NP}}$, are included in coNPMV if and only if $\mathrm{NP} =$ coNP.

4-6

Theorem 5 *maxsat* $\in$ coNPMV.

Proof of Theorem 5 Consider an NPMV machine M that, on input of a formula φ, guesses an assignment y for φ. If y does not satisfy φ, then M accepts and outputs y. Otherwise, if y does satisfy φ, M guesses another assignment $y' > y$. If y' also satisfies φ, M outputs y; otherwise, M rejects (and outputs nothing).

4-7 M outputs every assignment except the maximum satisfying one (if there is one). Hence, *maxsat* $\in$ coNPMV.

Proof of Theorem 5 □

4-8 Krentel [Kre92] showed that $\mathrm{FP}^{\mathrm{NP}} = \mathrm{FP}^{\mathrm{MaxP}[1]}$. Since $\mathrm{FP}^{\mathrm{NPMV}} = \mathrm{FP}^{\mathrm{NP}}$ [FHOS97] and *maxsat* is complete for MaxP, we have that $\mathrm{FP}^{\mathrm{NPMV}} \subseteq \mathrm{FP}^{\mathrm{coNPMV}[1]}$. That is, polynomially many queries of a FP function to NPMV can be replaced by one query to coNPMV. Hence, as we have mentioned, coNPMV seems to be a more powerful class than NPMV. We will give more evidence for this in the next section.

4-9

Corollary 2 $\mathrm{MaxP} \subseteq$ coNPMV $\Longleftrightarrow$ $\mathrm{MinP} \subseteq$ coNPMV $\Longleftrightarrow$ NP $=$ coNP.

Proof of Corollary 2 If $\mathrm{MaxP} \subseteq$ coNPMV, then NPMV $\subseteq_c \mathrm{MaxP} \subseteq$ coNPMV, and, therefore, NPMV $\subseteq_c$ coNPMV. But by Theorem 3, this implies NP = coNP. Conversely, if NP = coNP, then $\mathrm{NPMV}^{\mathrm{NP}} = \mathrm{NPMV}^{\mathrm{NP}\cap\mathrm{coNP}} = \mathrm{NPMV}$. This implies $\mathrm{MaxP} \subseteq$ NPMV and, since the hypothesis also implies NPMV = coNPMV, that $\mathrm{MaxP} \subseteq$ coNPMV.

Proof of Corollary 2 □

4-10 We conclude this section with an observation regarding the relationship between MaxP and NPMV. First, note that trivially NPSV $\subseteq \mathrm{MaxP} \cap \mathrm{MinP}$ since *the* output of an NPSV function is both the minimum and the maximum. Similarly, NPMV $\subseteq_c \mathrm{MaxP} \cap \mathrm{MinP}$. The more interesting question is whether these inclusions are strict. This is quite likely.

Theorem 6 $\mathrm{MaxP} \subseteq$ NPMV $\Longleftrightarrow$ $\mathrm{MinP} \subseteq$ NPMV $\Longleftrightarrow$ NP $=$ coNP.

Proof of Theorem 6 If NP $=$ coNP, then $\mathrm{FP}^{\mathrm{NP}} \subseteq$ NPMV [Sel94]; thus, especially $\mathrm{MaxP} \cup \mathrm{MinP} \subseteq$ NPMV.

4-11 Now suppose $\mathrm{MaxP} \subseteq$ NPMV (the case for MinP is analogous). Let $L \in$ coNP. Define

$$f(x) = \begin{cases} 0 & \text{if } x \in L \\ 1 & \text{otherwise.} \end{cases}$$

Then $f \in \mathrm{MaxP}$ and hence, by assumption, is in NPMV. Since $x \in L$ if and only if $f(x) = 0$, we have $L \in$ NP.

Proof of Theorem 6 □

4-12 The last two results relativize; analogous results hold for higher levels of the NPMV hierarchy and Krentel's min/max hierarchy [FHOS97, VW95]. For the relativized version of Theorem 6 one has to use techniques from Krentel [Kre92] and Vollmer and Wagner [VW95].

5 A Characterization of coNPMV

5-1 As we have already seen in the preceding section, coNPMV seems to be a more powerful class than NPMV. This is somewhat surprising in light of the aforementioned symmetry in the definitions of coNPMV and NPMV by their graphs.

5-2 The following theorem shows that coNPMV is in fact very close to $\mathrm{NPMV}^{\mathrm{NP}}$. This is surprising as well, as we have already seen in Theorem 4 that there is a hierarchy of function classes between coNPMV and $\mathrm{NPMV}^{\mathrm{NP}}$.

5-3 If f is a multivalued function and g is a single-valued function, then $g \circ f$ is defined by $\mathrm{graph}(g \circ f) = \{ \langle x, g(y) \rangle : f(x) \mapsto y \}$. Let π_2^1 denote the projection function that maps a pair of strings to its first component. By $\pi_2^1 \circ$ coNPMV we denote $\{ \pi_2^1 \circ f : f \in \text{coNPMV} \}$.

Theorem 7 $\mathrm{NPMV}^{\mathrm{NP}} = \pi_2^1 \circ$ coNPMV.

Proof of Theorem 7 The right-to-left containment follows from Theorem 4 and the fact that the projection of any $\mathrm{NPMV}^{\mathrm{NP}}$ function is still in $\mathrm{NPMV}^{\mathrm{NP}}$; hence, $\pi_2^1 \circ \text{coNPMV} \subseteq \mathrm{NPMV}^{\mathrm{NP}}$.

5-4 For the other direction, let $f \in \mathrm{NPMV}^{\mathrm{NP}}$. By a standard argument, $\mathrm{graph}(f) \in \Sigma_2^p$, and thus there is a polynomial q and a predicate $R \in$ coNP such that for any x and $y \in \Sigma^{q(|x|)}$

$$f(x) \mapsto y \Longleftrightarrow \exists z \in \Sigma^{q(|x|)} : R(x, y, z).$$

Define f' such that for any x and any $y, z \in \Sigma^{q(|x|)}$

$$f'(x) \mapsto \langle y, z \rangle \iff R(x, y, z).$$

5-5 So R witnesses that $f' \in$ coNPMV. But $f(x) = \pi_2^1 \circ f'(x)$, which shows that $f \in \pi_2^1 \circ$ coNPMV.

Proof of Theorem 7 □

5-6 The reason why it is likely that coNPMV is a proper subclass of NPMV^{NP} is not because outputs of coNPMV functions give too little information, but rather that they give too much. We can compute an arbitrary NPMV^{NP} function simply by throwing away part of the output of a coNPMV function. This is what the projection operator accomplishes, and it is most likely necessary.

5-7 If we apply Theorem 7, many properties of NPMV^{NP} now carry over to coNPMV. In Section 3 we have shown functions F_2 and *not-pre-sat* complete for coNPMV. Since the projection function is in FP, we get that those functions are complete for NPMV^{NP} as well.

Corollary 3 NPMV^{NP} *is the c-closure of* coNPMV *under* $\leq_m^{\text{P}}$*-reducibility and the closure of* coNPMV *under* $\leq_{sm}^{\text{P}}$*-reducibility.*

5-8 In particular, we get the following corollary.

Corollary 4 $\text{FP}^{\text{coNPMV}[k]} = \text{FP}^{\text{NPMV}^{\text{NP}}[k]}$ *for all* $k \geq 1$, *and* $\text{FP}^{\text{coNPMV}} = \text{FP}^{\text{NPMV}^{\text{NP}}} = \text{FP}^{\text{NP}^{\text{NP}}}$.

5-9 Observe, by contrast, that $\text{FP}^{\text{NPMV}} = \text{FP}^{\text{NP}} = \text{FP}^{\text{MinP}} = \text{FP}^{\text{MaxP}}$, so coNPMV and NPMV define different Δ-levels of the functional polynomial hierarchy.

5-10 Fenner et al. [FHOS97] have shown that $\text{NPMV}(2) \subseteq \text{FP}^{\text{NPMV}} \iff \Sigma_2^p = \Delta_2^p$. Note that, in contrast for the corresponding language classes, we have $\text{NP}(k) \subseteq \text{P}^{\text{NP}}$ for all k. We can now improve the result of Fenner et al.

Corollary 5 $\text{coNPMV} \subseteq \text{FP}^{\text{NPMV}} \iff \Sigma_2^p = \Delta_2^p$.

Proof of Corollary 5 If $\Sigma_2^p = \Delta_2^p$, then

$$\begin{aligned}\text{coNPMV} \subseteq \text{FP}^{\text{coNPMV}} &= \text{FP}^{\text{NPMV}^{\text{NP}}} = \text{FP}^{\Sigma_2^p} \\ &= \text{FP}^{\Delta_2^p} = \text{FP}^{\text{NP}} = \text{FP}^{\text{NPMV}},\end{aligned}$$

where the last equality is Theorem 1 in [FHOS97] and the second equally follows from the relativized version of the same theorem. Conversely, if coNPMV $\subseteq$ $\mathrm{FP}^{\mathrm{NPMV}}$, then $\mathrm{dom}(\mathrm{coNPMV}) \subseteq \mathrm{P}^{\mathrm{NP}} = \Delta_2^p$, so that $\Sigma_2^p \subseteq \Delta_2^p$.

Proof of Corollary 5 □

Corollary 6 *For any $k \geq 1$, we have*

$$\mathrm{NPMV}^{\mathrm{NP}} \subseteq \mathrm{FP}^{\mathrm{coNPMV}[1]} \subseteq \mathrm{FP}^{\mathrm{coNPMV}[k]} \subseteq \mathrm{FP}^{\mathrm{coNPMV}[k+1]} \subseteq \mathrm{FP}^{\mathrm{coNPMV}} = \mathrm{FP}^{\mathrm{NP}^{\mathrm{NP}}}.$$

Furthermore, all inclusions are strict unless the polynomial-time hierarchy collapses.

Proof of Corollary 6 It remains to show the strictness of the inclusions. Suppose $\mathrm{FP}^{\mathrm{coNPMV}[1]} \subseteq \mathrm{NPMV}^{\mathrm{NP}}$. This is equivalent to $\mathrm{FP}^{\mathrm{NPMV}^{\mathrm{NP}}[1]} \subseteq \mathrm{NPMV}^{\mathrm{NP}}$ by Corollary 4, which implies $\mathrm{P}^{\Sigma_2^p[1]} \subseteq \Sigma_2^p$. But then $\Pi_2^p = \Sigma_2^p = \mathrm{PH}$. For the other inclusions, suppose $\mathrm{FP}^{\mathrm{coNPMV}[k]} = \mathrm{FP}^{\mathrm{coNPMV}[k+1]}$. Then $\mathrm{FP}^{\mathrm{NPMV}^{\mathrm{NP}}[k]} = \mathrm{FP}^{\mathrm{NPMV}^{\mathrm{NP}}[k+1]}$. By a theorem of Fenner et al. [FHOS97], this implies that $\mathrm{FP}^{\Sigma_2^p[k]} = \mathrm{FP}^{\Sigma_2^p[k+1]}$, which, by a relativization of Kadin's theorem [Kad88], implies that the polynomial hierarchy collapses.

Proof of Corollary 6 □

5-11 Thus we see, combining Theorem 4 and Corollary 6, that all classes of the difference hierarchy over NPMV are included in the query hierarchy over coNPMV, in fact, already in its first level. There are (under reasonable assumptions) no inclusions in the opposite direction. Concerning the relationship between the query hierarchy over NPMV and the difference hierarchy over NPMV, we know from Fenner et al. [FHOS97] that all classes of the first hierarchy are included in certain classes of the second hierarchy. Any inclusion in the opposite direction implies coNPMV $\subseteq$ $\mathrm{FP}^{\mathrm{NPMV}}$, which again implies a collapse of the polynomial hierarchy, by Corollary 5.

6 Relationships between the Functional Difference and Polynomial-Time Hierarchies

6-1 Chang and Kadin [CK96] showed that if the Boolean hierarchy over NP collapses to the *k-th* level, then the polynomial hierarchy collapses to the *k-th* level of the

Boolean hierarchy over $\mathrm{NP}^{\mathrm{NP}}$: if $\mathrm{NP}(k+1) = \mathrm{NP}(k)$, then $\mathrm{PH} = \mathrm{NP}^{\mathrm{NP}}(k)$. It is a simple consequence of known results that a similar connection exists for the corresponding functional hierarchies, namely $\mathrm{NPMV}(k)$ and $\Sigma\mathrm{MV}_k = \mathrm{NPMV}^{\Sigma^p_{k-1}}$.

Theorem 8 *For any* $k \geq 1$, *if* $\mathrm{NPMV}(k+1) = \mathrm{NPMV}(k)$, *then* $\Sigma\mathrm{MV}_3 = \mathrm{NPMV}^{\mathrm{NP}}(k)$.

Proof of Theorem 8 $\mathrm{NPMV}(k+1) = \mathrm{NPMV}(k)$ is equivalent to $\mathrm{NP}(k+1) = \mathrm{NP}(k)$ [FHOS97], which implies $\Sigma^p_3 = \mathrm{NP}^{\mathrm{NP}}(k)$ [CK96] (relativized). By considering the graphs of functions [FHOS97], we immediately get that $\Sigma\mathrm{MV}_3 = \mathrm{NPMV}^{\mathrm{NP}}(k)$.

Proof of Theorem 8 □

6-5 Since $\mathrm{NP}^{\mathrm{NP}}(k) \subseteq \mathrm{P}^{\mathrm{NP}^{\mathrm{NP}}[k]}$, a consequence of Chang and Kadin's theorem is that, if $\mathrm{NP}(k+1) = \mathrm{NP}(k)$, then $\Sigma^p_3 = \mathrm{P}^{\mathrm{NP}^{\mathrm{NP}}[k]}$ (indeed, they prove this directly in their paper before treating the stronger result). The functional analogue of such a collapse would be $\Sigma\mathrm{MV}_3 = \mathrm{FP}^{\mathrm{NPMV}^{\mathrm{NP}}[k]}$ or, equivalently, $\Sigma\mathrm{MV}_3 = \mathrm{FP}^{\mathrm{coNPMV}[k]}$. We cannot expect this as a direct consequence of Theorem 8, since the difference and query hierarchies are not intertwined in this context. Nevertheless, such an analogous result does hold. To see this, we have to modify the proof of the Chang and Kadin theorem.

Theorem 9 *If* $\mathrm{NPMV}(k+1) = \mathrm{NPMV}(k)$, *then* $\Sigma\mathrm{MV}_3 = \mathrm{NPMV} \circ \mathrm{FP}^{\mathrm{coNPMV}[k-1]}$.

Proof of Theorem 9 In order to explain how Chang and Kadin's proof gives this result, we recall some of their definitions, with some minor modifications in notation (for greater detail, we refer the reader to their paper [CK96]). Denote the $\leq^p_m$-complete language for $\mathrm{NP}(k)$ (respectively, $\mathrm{coNP}(k)$) as $L_{\mathrm{NP}(k)}$ (respectively, $L_{\mathrm{coNP}(k)}$). For example, $L_{\mathrm{NP}(1)} = \mathrm{SAT}$ and $L_{\mathrm{NP}(2)} = \{ \langle x_1, x_2 \rangle : x_1 \in \mathrm{SAT} \text{ and } x_2 \in \overline{\mathrm{SAT}} \}$. Since, by hypothesis, $\mathrm{NP}(k) = \mathrm{coNP}(k)$, it follows that $L_{\mathrm{NP}(k)} \leq^p_m L_{\mathrm{coNP}(k)}$. The basic idea underlying the Chang and Kadin proof is that such a reduction induces a reduction from an initial segment of SAT to an initial segment of $\overline{\mathrm{SAT}}$. This is done inductively via the notion of a "hard sequence," which is a j-tuple that, together with a $\leq^p_m$-reduction from $\mathrm{NP}(k)$ to $\mathrm{coNP}(k)$, can be used to find a $\leq^p_m$-reduction from $\mathrm{NP}(k-j)$ to $\mathrm{coNP}(k-j)$.

Definition 4 *Let* $L_{\mathrm{NP}(k)} \leq^p_m L_{\mathrm{coNP}(k)}$ *via some polynomial time function* h. *Then we call* $\langle 1^m, x_1, \ldots, x_j \rangle$ *a* hard sequence *with respect to* h *for length* m *of order* j, *if either* $j = 0$ *or the following conditions hold:*

1. $1 \leq j \leq k-1$,
2. $|x_j| \leq m$,
3. $x_j \in \overline{\text{SAT}}$,
4. $\langle 1^m, x_1, \ldots, x_{j-1} \rangle$ *is a hard sequence with respect to* h, *and*
5. *for all* $y_1, \ldots, y_\ell \in \Sigma^*$ *where* $\ell = k - j$, *and for all* $1 \leq i \leq \ell$, $|y_i| \leq m$, $\pi_{\ell+1} \circ h(\langle y_1, \ldots, y_\ell, x_j, \ldots, x_1 \rangle) \in \overline{\text{SAT}}$.

6-3 A hard sequence is called *maximal* if it cannot be extended to a hard sequence of a higher order. In this case, the order of the sequence j is said to be maximal.

6-4 We can now outline the proof. Chang and Kadin's Lemma 3 [CK96] states that, given a maximal hard sequence for an appropriate (polynomially bounded) length, an NP machine can recognize an initial segment of the canonical complete language for NP^{NP}. That is, with the aid of such a sequence, we can replace a Σ_2^p machine with an NP machine. Thus, it suffices to find a maximal hard sequence to collapse the NP's of a $\Sigma\text{MV}_3 = \text{NPMV}^{\Sigma_2^p}$ machine.

6-5 Our principle observation is this: *Hard sequences of any given order can be obtained by a single query to a* coNPMV *oracle*. This result can easily be seen as follows. Define the function $H\colon 1^+ \times \text{N} \mapsto \Sigma^*$ such that $H(1^m, j) \mapsto \langle 1^m, x_1, \ldots, x_j \rangle$ if and only if $\langle 1^m, x_1, \ldots, x_j \rangle$ is a hard sequence for length m of order j. It follows from Definition 4 that the set of hard sequences is in coNP [CK96]; hence, $\text{graph}(H) \in$ coNP, so that $H \in$ coNPMV. Therefore, we can obtain a maximal hard sequence for the appropriate polynomial length $m = p(|x|)$ by querying a coNPMV oracle for a value of $H(1^m, j)$ for j varying from 1 to $k-1$. We then feed the resulting maximal hard sequence, along with the original input x, to an NPMV machine that can, via the induced reduction from coNP to NP, collapse the NP oracles in an $\text{NPMV}^{\text{NP}^{\text{NP}}}$ computation.

Proof of Theorem 9 □

7 A Remark on Counting Classes

7-1 The results of our paper show that in the context of relational structures computed by polynomial-time machines, in a sense the universal mode is more powerful than the existential one. In the context of counting classes, a similar phenomenon has been observed by Seinosuke Toda [Tod91]. In this section, we briefly show that Toda's result is a special case of one of our observations.

7-2 Recall the following general definition of counting classes from [Tod91, VW93]:

7-3 Let $\mathcal{K}$ be a class of sets. Then, $\# \cdot \mathcal{K}$ consists of those functions f for which there exist a set $A \in \mathcal{K}$ and a polynomial p such that, for all x,

$$f(x) = \left|\left\{ y \;\middle|\; |y| \leq p(|x|) \wedge \langle x, y \rangle \in A \right\}\right|.$$

It is obvious that $\# \cdot \mathrm{P} = \# \mathrm{P}$. Moreover it can be shown that $\# \cdot \mathrm{NP} = \operatorname{span} \mathrm{P}$, where $\operatorname{span} \mathrm{P}$ is the class of functions that count the number of distinct outputs of a nondeterministic polynomial-time transducer.

7-4 We have the following relationship to our classes of functions:

Proposition 2

1. $\# \cdot \mathrm{NP}$ *consists of exactly those functions* h *for which there exists a function* $f \in \mathrm{NPMV}$ *such that for all* x, $h(x) = \left|\text{set-}f(x)\right|$.

2. $\# \cdot \mathrm{coNP}$ *consists of exactly those functions* h *for which there exists a function* $f \in \mathrm{coNPMV}$ *such that for all* x, $h(x) = \left|\text{set-}f(x)\right|$.

7-5 Now we have the following surprising result, which was already proved by Toda [Tod91, Theorem 4.1.6]:

Corollary 7 $\# \cdot \mathrm{coNP} = \# \mathrm{P}^{\mathrm{NP}}$.

Proof of Corollary 7 Proof is immediate by the preceding proposition and the fact that coNPMV is the class of functions that compute witnesses for Σ_2^p computations. See the discussion before Proposition 1.

Proof of Corollary 7 □

Acknowledgments

We appreciate the contributions of J. Ramachandran to the results presented here. F.G. thanks the Computer Science Department of Boston University for its hospitality in the course of this work.

Acknowledgment of support: The authors' contributions to this paper were supported by the following grants and institutions: Stephen Fenner in part by the NSF under grants CCR 92-09833 and CCR 95-01794; Steven Homer in part by the NSF under grant NSF-CCR-9400229; Alan L. Selman in part by the NSF under grant NSF-CCR-9400229; Thomas Thierauf in part by DAAD, Acciones Integradas 1995, 322-AI-e-dr; and Heribert Vollmer by the Alexander von Humboldt foundation under a Feodor Lynen scholarship.

References

[BLS84] R. Book, T. Long, and A. Selman. Quantitative relativizations of complexity classes. *SIAM Journal on Computing*, 13(3):461–487, 1984.

[CK96] R. Chang and J. Kadin. The boolean hierarchy and the polynomial hierarchy: A closer connection. *SIAM Journal on Computing*, 25(2):340–354, 1996.

[Coo71] S. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd ACM Symposium Theory of Computing*, pages 151–158, New York, 1971. ACM Press.

[FHOS97] S. Fenner, S. Homer, M. Ogihara, and A. Selman. Oracles that compute values. *SIAM Journal on Computing*, 26(4):1043–1065, 1997.

[Kad88] J. Kadin. The polynomial time hierarchy collapses if the Boolean hierarchy collapses. *SIAM Journal on Computing*, 17(6):1263–1282, 1988.

[Kre88] M. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36:490–509, 1988.

[Kre92] M. Krentel. Generalizations of OptP to the polynomial hierarchy. *Theoretical Computer Science*, 97:183–198, 1992.

[Sel94] A. Selman. A taxonomy of complexity classes of functions. *Journal of Computer and System Sciences*, 48(2):357–381, 1994.

[Tod91] S. Toda. *Computational Complexity of Counting Complexity Classes*. Ph.D. thesis, Tokyo Institute of Technology, Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan, 1991.

[VW93] H. Vollmer and K. Wagner. The complexity of finding middle elements. *International Journal of Foundations of Computer Science*, 4:293–307, 1993.

[VW95] H. Vollmer and K. Wagner. Complexity classes of optimization functions. *Information and Computation*, 120:198–219, 1995.

[Wag90] K. Wagner. Bounded query classes. *SIAM Journal on Computing*, 19:833–846, 1990.