Graph Isomorphism is Low for $\mathrm{ZPP}^{\mathbf{NP}}$ and other Lowness results

V. Arvind[1] and J. Köbler[2]

[1] Institute of Mathematical Sciences, C.I.T Campus, Chennai 600113, India
[2] Humboldt-Universität zu Berlin, Institut für Informatik, D-10099 Berlin, Germany

Abstract. We show the following new lowness results for the probabilistic class $\mathrm{ZPP}^{\mathrm{NP}}$.

- The class $\mathrm{AM} \cap \mathrm{coAM}$ is low for $\mathrm{ZPP}^{\mathrm{NP}}$. As a consequence it follows that Graph Isomorphism and several group-theoretic problems known to be in $\mathrm{AM} \cap \mathrm{coAM}$ are low for $\mathrm{ZPP}^{\mathrm{NP}}$.
- The class IP[P/poly], consisting of sets that have interactive proof systems with honest provers in P/poly, is also low for $\mathrm{ZPP}^{\mathrm{NP}}$.

We consider lowness properties of nonuniform function classes, namely, NPMV/poly, NPSV/poly, NPMV_t/poly, and NPSV_t/poly. Specifically, we show that

- Sets whose characteristic functions are in NPSV/poly and that have program checkers (in the sense of Blum and Kannan [9]) are low for AM and $\mathrm{ZPP}^{\mathrm{NP}}$.
- Sets whose characteristic functions are in NPMV_t/poly are low for Σ_2^p.

1 Introduction

In the recent past the probabilistic class $\mathrm{ZPP}^{\mathrm{NP}}$ has appeared in different results and contexts in complexity theory research. E.g. consider the result $\mathrm{MA} \subseteq \mathrm{ZPP}^{\mathrm{NP}}$ [1, 14] which sharpens and improves Sipser's theorem $\mathrm{BPP} \subseteq \Sigma_2^p$. The proof in [1] uses derandomization techniques based on hardness assumptions [22]. Another example is the result that if $\mathrm{SAT} \in \mathrm{P/poly}$ then $\mathrm{PH} = \mathrm{ZPP}^{\mathrm{NP}}$ [21, 5], which improves the classic Karp-Lipton theorem. [1] Actually, Köbler and Watanabe in [21] prove that every self-reducible set[2] A in $(\mathrm{NP} \cap \text{co-NP})$/poly is *low* for $\mathrm{ZPP}^{\mathrm{NP}}$, i.e. $\mathrm{ZPP}^{\mathrm{NP}^A} = \mathrm{ZPP}^{\mathrm{NP}}$. This stronger result is in a sense natural, since there is usually an underlying lowness result that implies a collapse consequence result like the Karp-Lipton theorem. We may recall here that the lowness result underlying the Karp-Lipton theorem is that self-reducible sets in P/poly are low for Σ_2^p [25].

The notion of lowness was first introduced in complexity theory by Schöning in [25]. It has since then been an important conceptual tool in complexity theory, see e.g. the survey paper [17].

---

[1] The Karp-Lipton theorem states that if $\mathrm{SAT} \in \mathrm{P/poly}$ then PH collapses to Σ_2^p.

[2] By self-reducibility we mean word-decreasing self-reducibility which is adequate because standard complexity classes contained in EXP have such self-reducible complete problems.

1.1 *Lowness for* $\mathrm{ZPP}^{\mathbf{NP}}$

We recall the formal definition of lowness [25]. For a relativizable complexity class $\mathcal{C}$ such that for all sets A, $A \in \mathcal{C}^A$, let $Low(\mathcal{C})$ denote $\{A \mid \mathcal{C}^A = \mathcal{C}\}$. Clearly, $Low(\mathcal{C})$ is contained in $\mathcal{C}$ and consists of languages that are powerless as oracle for $\mathcal{C}$.

Few complexity classes have their low sets exactly characterized. These are well-known examples: $Low(\mathrm{NP}) = \mathrm{NP} \cap \text{co-NP}$, $Low(\mathrm{AM}) = \mathrm{AM} \cap \mathrm{coAM}$ [26]. For most complexity classes however, a complete characterization of the low sets appears to be a challenging open question. Regarding $Low(\Sigma_2^p)$, Schöning proved [26] that $\mathrm{AM} \cap \mathrm{coAM}$ is contained in $Low(\Sigma_2^p)$, implying that $Low(\mathrm{AM}) \subseteq Low(\Sigma_2^p)$. This containment is anomalous because $\mathrm{AM} \not\subseteq \Sigma_2^p$ in some relativized worlds [24]. Indeed, lowness appears to have other anomalous properties: it is not known to preserve containment of complexity classes, for example $\mathrm{NP} \subseteq \mathrm{PP}$ but $\mathrm{NP} \cap \text{co-NP}$ is not known to be in $Low(\mathrm{PP})$. Similarly, $\mathrm{NP} \subseteq \mathrm{MA}$ but $\mathrm{NP} \cap \text{co-NP}$ is not known to be in $Low(\mathrm{MA})$. Little is known about $Low(\mathrm{MA})$ except that it contains BPP and is contained in $\mathrm{MA} \cap \text{co-MA}$ [19].

Regarding $\mathrm{ZPP}^{\mathrm{NP}}$, it is shown in [21] that $Low(\mathrm{ZPP}^{\mathrm{NP}}) \subseteq Low(\Sigma_2^p)$. No characterization of $Low(\mathrm{ZPP}^{\mathrm{NP}})$ is known. Our aim is to show some inclusions in $Low(\mathrm{ZPP}^{\mathrm{NP}})$ as a first step.

We first show in this paper that $\mathrm{AM} \cap \mathrm{coAM}$ is low for $\mathrm{ZPP}^{\mathrm{NP}}$, i.e. $\mathrm{AM} \cap \mathrm{coAM} \subseteq Low(\mathrm{ZPP}^{\mathrm{NP}})$. Hence we have the inclusion chain

$$Low(\mathrm{MA}) \subseteq Low(\mathrm{AM}) \subseteq Low(\mathrm{ZPP}^{\mathrm{NP}}) \subseteq Low(\Sigma_2^p).$$

It follows that Graph Isomorphism and other group-theoretic problems known to be in $\mathrm{AM} \cap \mathrm{coAM}$ [4] are low for $\mathrm{ZPP}^{\mathrm{NP}}$.

We prove another lowness result for $\mathrm{ZPP}^{\mathrm{NP}}$: Let IP[P/poly] denote languages that have interactive proof systems with honest prover in P/poly. We show that $\mathrm{IP}[\mathrm{P/poly}] \subseteq Low(\mathrm{ZPP}^{\mathrm{NP}})$, improving the containment $\mathrm{IP}[\mathrm{P/poly}] \subseteq Low(\Sigma_2^p)$ shown in [3]. Our proof has a derandomization component in which the Nisan-Wigderson pseudorandom generator [22] is used to derandomize the verifier in the IP[P/poly] protocol. The rest of the proof is based on the random sampling technique as applied in [5, 18].

1.2 NP/poly $\cap$ co-NP/poly *and subclasses*

As shown in [21], self-reducible sets in $(\mathrm{NP} \cap \text{co-NP})/\mathrm{poly}$ are *low* for $\mathrm{ZPP}^{\mathrm{NP}}$. However, there are technical difficulties due to which this result does not carry over to $\mathrm{NP/poly} \cap \text{co-NP/poly}$. The best known collapse consequence of $\mathrm{NP} \subseteq \mathrm{NP/poly} \cap \text{co-NP/poly}$ (equivalently, $\mathrm{NP} \subseteq \text{co-NP/poly}$) is $\mathrm{PH} \subseteq \mathrm{ZPP}(\Sigma_2^p)$ [21].

In order to better understand this aspect of $\mathrm{NP/poly} \cap \text{co-NP/poly}$ the authors of [11] introduce two interesting subclasses of $\mathrm{NP/poly} \cap \text{co-NP/poly}$ which we discuss in Section 5. We notice firstly that $\mathrm{NP/poly} \cap \text{co-NP/poly}$ and the above-mentioned subclasses are closely connected to the function classes NPMV/poly, NPSV/poly, NPMV_t/poly, and NPSV_t/poly, which are nonuniform analogues of the function classes NPMV, NPSV, NPMV_t, and NPSV_t

introduced and studied by Selman and other researchers [27, 12]. More precisely, we note that $A \in (\mathrm{NP} \cap \text{co-NP})/\mathrm{poly}$ if and only if $\chi_A \in \mathrm{NPSV}_t/\mathrm{poly}$, where χ_A denotes the characteristic function of a language A. Similarly, $A \in \mathrm{NP}/\mathrm{poly} \cap \text{co-NP}/\mathrm{poly}$ if and only if $\chi_A \in \mathrm{NPMV}/\mathrm{poly}$. Likewise, NPSV/poly and $\mathrm{NPMV}_t/\mathrm{poly}$ capture the two new subclasses of $\mathrm{NP}/\mathrm{poly} \cap \text{co-NP}/\mathrm{poly}$ defined in [11].

We prove the following new lowness results for these classes:

- We show that self-reducible sets whose characteristic functions are in the function class $\mathrm{NPMV}_t/\mathrm{poly}$ are low for Σ_2^p (this result is essentially the lowness result underlying the collapse consequence i.e. Theorem 5.2 in [11]).
- We show that all self-checkable sets — In the program checking sense of Blum and Kannan [9]— whose characteristic functions are in NPSV/poly are low for AM.

Several proofs are omitted from this extended abstract. A full version of the paper is available as a technical report [2].

2 Preliminaries

Let $\Sigma = \{0, 1\}$. We denote the cardinality of a set X by $\|X\|$ and the length of a string $x \in \Sigma^*$ by $|x|$. The characteristic function of a language $L \subseteq \Sigma^*$ is denoted by χ_L. The definitions of standard complexity classes like P, NP, E, EXP etc. can be found in standard books [8, 23]. A relativized complexity class $\mathcal{C}$ with oracle A is denoted by either $\mathcal{C}^A$ or $\mathcal{C}(A)$. Likewise, we denote an oracle Turing machine M with oracle A by M^A or $M(A)$.

For a class $\mathcal{C}$ of sets and a class $\mathcal{F}$ of functions from 1^* to Σ^*, let $\mathcal{C}/\mathcal{F}$ [15] be the class of sets A such that there is a set $B \in \mathcal{C}$ and a function $h \in \mathcal{F}$ such that for all $x \in \Sigma^*$,

$$x \in A \Leftrightarrow \langle x, h(1^{|x|}) \rangle \in B.$$

The function h is called an *advice function* for A.

We recall definitions of AM and MA. A language L is in AM if there exist a polynomial p and a set $B \in \mathrm{P}$ such that for all x, $|x| = n$,

$$\begin{aligned} x \in A &\Rightarrow \mathrm{Prob}_{r \in_R \{0,1\}^{p(n)}}[\exists y, |y| = p(n) : \langle x, y, r \rangle \in B] \ = \ 1, \\ x \notin A &\Rightarrow \mathrm{Prob}_{r \in_R \{0,1\}^{p(n)}}[\forall y, |y| = p(n) : \langle x, y, r \rangle \in B] \leq 1/4. \end{aligned}$$

A language L is in MA if there exist a polynomial p and a set $B \in \mathrm{P}$ such that for all x, $|x| = n$,

$$\begin{aligned} x \in A &\Rightarrow \exists y, |y| = p(n) : \mathrm{Prob}_{r \in_R \{0,1\}^{p(n)}}[\langle x, y, r \rangle \in B] \geq 3/4, \\ x \notin A &\Rightarrow \forall y, |y| = p(n) : \mathrm{Prob}_{r \in_R \{0,1\}^{p(n)}}[\langle x, y, r \rangle \in B] \leq 1/4. \end{aligned}$$

Notice that we have taken the definition of AM with 1-sided error, known to be equivalent to AM with 2-sided error. Definitions for single and multiprover interactive proof systems can be found in standard texts, e.g. [23]. Let MIP denote the class of languages with multiprover interactive protocols and IP denote the class of languages with single-prover interactive protocols. We denote by MIP[$\mathcal{C}$] and IP[$\mathcal{C}$] the respective language classes where the prover complexity is bounded by FP($\mathcal{C}$), which is the set of functions that can be computed by a polynomial-time oracle transducer with oracle in $\mathcal{C}$.

3 AM $\cap$ coAM is low for $\mathrm{ZPP}^{\mathrm{NP}}$

In this section we show that AM$\cap$coAM is low for $\mathrm{ZPP}^{\mathrm{NP}}$. It follows that Graph Isomorphism and a host of group-theoretic problems known to be in AM$\cap$coAM [4] are all low for $\mathrm{ZPP}^{\mathrm{NP}}$. We recall here that it is already known that AM$\cap$coAM is low for Σ_2^p [26] and also for AM [19].

We notice first that although AM $\cap$ coAM $\subseteq \mathrm{ZPP}^{\mathrm{NP}}$ (because AM $\subseteq \mathrm{coR}^{\mathrm{NP}}$ and the equality ZPP $=$ R $\cap$ coR relativizes) and AM $\cap$ coAM is low for itself, it doesn't follow that AM $\cap$ coAM is low for $\mathrm{ZPP}^{\mathrm{NP}}$. As mentioned before, NP $\cap$ co-NP is trivially low for NP but is not known to be low for PP or MA.

Theorem 1. AM $\cap$ coAM *is low for* $\mathrm{ZPP}^{\mathrm{NP}}$.

Proof. Let L be any set in AM $\cap$ coAM. We need to show that a given $\mathrm{ZPP}^{\mathrm{NP}^L}$ machine M can be simulated in $\mathrm{ZPP}^{\mathrm{NP}}$. Consider an input x of length bounded by n to the machine M. Suppose the lengths of all the queries made to L during the computation are bounded by m. Since $L \in$ AM $\cap$ coAM, it follows from standard probability amplification techniques and quantifier swapping (cf. [26]) that there are NP sets A and B and a polynomial p such that $\forall y : |y| \leq m$, there is a subset $S \subseteq \{0,1\}^{p(m)}$ of size $||S|| \geq 2^{p(m)-1}$ with the following property:

$y \in L$ implies

$$\forall w : \langle y, w\rangle \in A \text{ and } \forall w \in S : \langle y, w\rangle \notin B$$

and $y \notin L$ implies

$$\forall w : \langle y, w\rangle \in B \text{ and } \forall w \in S : \langle y, w\rangle \notin A.$$

Notice that in the above we are using the fact that AM protocols can be assumed to have one-sided error.

In other words, a large fraction of the w's act as advice strings using which membership in L for strings of length m can be decided with an NP $\cap$ co-NP computation. Notice, however, that it would be incorrect for us to claim from here that $L \in$ (NP $\cap$ co-NP)/poly, because if we use a string from $\{0,1\}^{p(m)} - S$ as advice, the resulting combination of machines for A and B may not yield an NP $\cap$ co-NP computation for some input $y \in \Sigma^{\leq m}$. However, we observe that the above property of advice strings in S implies that $w \in S$ if and only if using w as advice yields an NP $\cap$ co-NP computation for all inputs $y \in \Sigma^{\leq m}$.

Thus, a candidate advice $w \in \Sigma^{p(m)}$ is *not* in S if and only if it satisfies the following NP predicate:

$$\exists y \in \Sigma^{\leq m} : \langle y, w \rangle \in A \cap B.$$

We now describe the $\mathrm{ZPP}^{\mathrm{NP}}$ machine N that simulates the given $\mathrm{ZPP}^{\mathrm{NP}^L}$ machine M on some input x. Machine N first randomly guesses an advice string in $w \in \Sigma^{p(m)}$ which, by assumption, is in S with probability $1/2$. A single NP query using the above NP predicate is now used to certify that $w \in S$. Using such a w as advice, N can replace the oracle L with an $\mathrm{NP} \cap \text{co-NP}$ computation when it simulates M.

Corollary 1. *Graph Isomorphism is low for* $\mathrm{ZPP}^{\mathrm{NP}}$.

The above corollary follows since Graph Isomorphism is in $\mathrm{AM} \cap \mathrm{coAM}$ [13]. The lowness result also holds for various group-theoretic problems known to be in $\mathrm{AM} \cap \mathrm{coAM}$ [4].

Notice that the previous theorem essentially shows that we can simulate $\mathrm{AM} \cap \mathrm{coAM}$ with an $\mathrm{NP} \cap \text{co-NP}$ computation using a random string in a coNP set as advice for the computation. This observation combined with the result of [21] (that self-reducible sets in $(\mathrm{NP} \cap \text{co-NP})/\mathrm{poly}$ are low for $\mathrm{ZPP}^{\mathrm{NP}}$) immediately yields the following corollary.

Corollary 2. *Self-reducible sets in* $(\mathrm{AM} \cap \mathrm{coAM})/\mathrm{poly}$ *are low for* $\mathrm{ZPP}^{\mathrm{NP}}$.

Additionally, we also have the following corollary in the average-case complexity setting. We first recall the definition of $\mathcal{AP}$ (see, e.g. [20] for a detailed treatment): $\mathcal{AP}$ is the class of decision problems A such that for every polynomial-time computable distribution there is an algorithm that decides A and is polynomial-time on the average for that distribution.

Corollary 3. *If* $\mathrm{NP} \subseteq \mathcal{AP}$ *then* $\mathrm{AM} \cap \mathrm{coAM} = \mathrm{NP} \cap \text{co-NP}$.

The proof follows from the assumption $\mathrm{NP} \subseteq \mathcal{AP}$ combined with the fact that for any set in $\mathrm{AM} \cap \mathrm{coAM}$ a large fraction of strings satisfying a coNP predicate are good advice strings, as we have already seen in the proof of Theorem 1. Thus, a ZPP computation can randomly guess such an advice string and use an $\mathcal{AP}$ algorithm for the *uniform* distribution to decide the coNP predicate. This $\mathcal{AP}$ algorithm, with its running time truncated to a suitable polynomial bound, will still accept many of the randomly picked good advice strings. This is an application of ideas from [20].

4 IP[P/poly] is low for $\mathrm{ZPP}^{\mathrm{NP}}$

The class IP[P/poly] already figures, though implicitly, in the proof of the result in [6] that if $\mathrm{EXP} \subseteq \mathrm{P/poly}$ then $\mathrm{EXP} = \mathrm{MA}$. We quickly recall the proof: Suppose $\mathrm{EXP} \subseteq \mathrm{P/poly}$. Note that each language in EXP has a multiprover interactive protocol in which the provers are in EXP. By assumption, therefore, the honest provers can be simulated by polynomial size circuits. Thus the (MIP) protocol can be simulated by an MA protocol where Merlin simply sends the circuits for the provers to Arthur in the first round. In other words, the proof shows the inclusion chain $\mathrm{EXP} \subseteq \mathrm{MIP[P/poly]} \subseteq \mathrm{MA}$. Since the MA protocol is a single prover interactive protocol, we also have $\mathrm{MIP[P/poly]} = \mathrm{IP[P/poly]} \subseteq \mathrm{MA}$.

The above collapse consequence result of [6] motivates the study of lowness properties of IP[P/poly]. Our next result states that $\mathrm{IP[P/poly]} \subseteq \mathit{Low}(\mathrm{ZPP}^{\mathrm{NP}})$, improving the containment $\mathrm{IP[P/poly]} \subseteq \mathit{Low}(\Sigma_2^p)$ shown in [3]. Our result strengthens the result of [18] that NP sets in P/poly with self-computable witnesses are low for $\mathrm{ZPP}^{\mathrm{NP}}$. IP[P/poly] contains such NP sets, but IP[P/poly] may not even be contained in NP. Although $\mathrm{IP[P/poly]} \subseteq \mathrm{MA} \subseteq \mathrm{AM}$, IP[P/poly] is not known to be closed under complement, and it is not known if IP[P/poly] is contained in coAM. Thus, $\mathrm{IP[P/poly]} \subseteq \mathit{Low}(\mathrm{ZPP}^{\mathrm{NP}})$ appears incomparable to $\mathrm{AM} \cap \mathrm{coAM} \subseteq \mathit{Low}(\mathrm{ZPP}^{\mathrm{NP}})$ shown in Theorem 1 in the previous section. Our result is also incomparable to the result in [21] that self-reducible sets in P/poly are low for $\mathrm{ZPP}^{\mathrm{NP}}$. An interesting aspect of our proof is that it combines derandomization and almost uniform random sampling.

Theorem 2. IP[P/poly] *is low for* $\mathrm{ZPP}^{\mathrm{NP}}$.

The above lowness result easily extends to $\mathrm{IP}[(\mathrm{NP} \cap \text{co-NP})/\mathrm{poly}]$ by observing that the proof relativizes in the following sense: for any oracle set A, $\mathrm{NP}^{\mathrm{IP}[\mathrm{P}^A/\mathrm{poly}]} \subseteq \mathrm{ZPP}^{\mathrm{NP}^A}$.

We conclude this section with another connection to the average-case complexity setting.

Theorem 3. *If* $\mathrm{NP} \subseteq \mathcal{AP}$ *and* $\mathrm{NP} \subseteq \mathrm{P/poly}$ *then PH collapses to* Δ_2^p.

5 Nonuniform function classes and lowness

We now study lowness properties of NPMV/poly, NPSV/poly, NPMV_t/poly, and NPSV_t/poly. These are nonuniform analogs of the function classes NPMV, NPSV, NPMV_t, and NPSV_t studied by Selman [27] and other researchers, e.g. [12]. These nonuniform classes are interesting because when restricted to characteristic functions of sets, NPSV_t/poly coincides with $(\mathrm{NP} \cap \text{co-NP})/\mathrm{poly}$ and NPMV/poly coincides with $\mathrm{NP/poly} \cap \text{co-NP/poly}$. Likewise, we note that the two subclasses of $\mathrm{NP/poly} \cap \text{co-NP/poly}$ studied in [11], namely all sets underproductively reducible to sparse sets and all sets overproductively reducible to sparse sets, also coincide with NPSV/poly and NPMV_t/poly, respectively.

Following Selman's notation in [27], a transducer is an NDTM T with a write-only output tape. On input x machine T outputs $y \in \Sigma^*$ if there is an accepting path on input x along which y is output. Hence, the function defined by T on Σ^* could be multivalued and partial. Given a multivalued function f

on Σ^* and $x \in \Sigma^*$ we use the notation

$$\textit{set-}f(x) = \{y \mid f : x \mapsto y\}$$

to denote the (possibly empty) set of function values for input x. We recall the basic definitions.

Definition 1. [10]

1. NPMV *is the class of multivalued, partial functions f for which there is a polynomial-time NDTM N such that*
   (a) *$f(x)$ is defined (i.e., set-$f(x) \neq \emptyset$) if and only if $N(x)$ has an accepting path.*
   (b) *$y \in$ set-$f(x)$ if and only if there is an accepting path of $N(x)$ where y is output.*
2. NPSV *is the class of single-valued partial functions in* NPMV.
3. NPMV_t *is the class of total functions in* NPMV.
4. NPSV_t *is the class of total single-valued functions in* NPMV.

The classes NPMV/poly, NPSV/poly, NPMV_t/poly, and NPSV_t/poly are the standard nonuniform analogs of the above classes defined as usual [15]: for $\mathcal{F} \in \{\mathrm{NPMV}, \mathrm{NPSV}, \mathrm{NPMV}_t \mathrm{NPSV}_t\}$, a multivalued partial function f is in $\mathcal{F}$/poly if there is a function $g \in \mathcal{F}$, a polynomial p, and an *advice function* $h : 1^* \mapsto \Sigma^*$ with $|h(1^n)| \leq p(n)$ for all n, such that for all $x \in \Sigma^*$,

$$\textit{set-}f(x) = \textit{set-}g(\langle x, h(1^{|x|})\rangle).$$

Before we connect these classes to NP/poly $\cap$ co-NP/poly and its subclasses defined in [11], we recall definitions from [11]: Consider polynomial-time nondeterministic oracle machines N whose computation paths can have three possible outcomes: accept, reject, or **?**. The machine N can also be viewed as a transducer which computes, for given oracle D and input x, a multivalued function. More precisely, if we identify accept with value 1 and reject with 0, and consider the **?** computation paths as rejecting paths then N^D defines a partial multivalued function: *set-*$N^D(x) \subseteq \{0,1\}$. Machine N^D is said to be *underproductive* if for each x we have $\{0,1\} \not\subseteq$ *set-*$N^D(x)$, and N is said to be *robustly underproductive* if for each oracle D and input x we have $\{0,1\} \not\subseteq$ *set-*$N^D(x)$. Likewise, N^D is *overproductive* if for each x we have *set-*$N^D(x) \neq \emptyset$, and N is said to be *robustly overproductive* if for each oracle D and input x we have *set-*$N^D(x) \neq \emptyset$.

With standard arguments we can convert a sparse set into a polynomial-size advice string and vice-versa (see, e.g. [8]). It follows that $A \in$ NP/poly $\cap$ co-NP/poly if and only if there is a sparse set S and a nondeterministic machine N such that N^S is both overproductive and underproductive and $A = L(N^S)$. Similarly, $A \in$ (NP $\cap$ co-NP)/poly if and only if there is a sparse set S and a nondeterministic machine N such that $A = L(N^S)$ and N is both robustly overproductive and robustly underproductive and $A = L(N^S)$.

Proposition 1. *Let χ_A denote the characteristic function for a set $A \subseteq \Sigma^*$:*

1. χ_A *is in* NPMV/poly *if and only if* A *is in* NP/poly $\cap$ co-NP/poly.
2. χ_A *is in* NPSV_t/poly *if and only if* A *is in* (NP $\cap$ co-NP)/poly.
3. χ_A *is in* NPSV/poly *if and only if there are a sparse set* S *and a robustly underproductive machine* N *such that* $A = L(N^S)$.
4. χ_A *is in* NPMV_t/poly *if and only if there are a sparse set* S *and a robustly overproductive machine* N *such that* $A = L(N^S)$.

By abuse of notation, we identify χ_A with A in this section. E.g. we write $A \in$ NPSV/poly when we mean $\chi_A \in$ NPSV/poly. We now turn to lowness questions for the nonuniform function classes. The classes NP/poly$\cap$co-NP/poly and (NP $\cap$ co-NP)/poly are of interest in the context of deriving strong collapse consequences from the assumption that NP (or other hard complexity classes) is contained in one of these classes. We recall the known collapse consequence result shown in [21] for NP/poly $\cap$ co-NP/poly under the assumption that NP is contained therein: If NP $\subseteq$ NP/poly $\cap$ co-NP/poly then PH collapses to $\mathrm{ZPP}^{\Sigma_2^p}$. The open question here is whether the collapse consequence can possibly be improved to $\mathrm{ZPP}^{\mathrm{NP}}$. This is one reason to consider classes that lie between NP/poly $\cap$ co-NP/poly and (NP $\cap$ co-NP)/poly.

5.1 *A lowness result for* NPMV_t/poly

It is shown in [11] that if an NP-complete problem is in NPMV_t/poly then PH collapses to Σ_2^p. In [11] the authors actually state this result in terms of overproductive reductions to sparse sets. We use ideas in their proof to show the underlying lowness result for functions: all word-decreasing self-reducible functions in NPMV_t/poly are low for Σ_2^p. We first recall the definition of word-decreasing self-reducible sets (and define its obvious extension to total single-valued functions).

Definition 2. [7] *For strings* $x, y \in \Sigma^*$, $x \prec y$ *if* $|x| < |y|$ *or* $|x| = |y|$ *and* x *is lexicographically smaller than* y. *A set* A *is* word-decreasing self-reducible *if there is a polynomial-time oracle machine* M *such that* $A = L(M^A)$, *where on any input* x *the machine* M *queries the oracle only about strings* y *such that* $y \prec x$. *Similarly, a total single-valued function* f *on* Σ^* *is* word-decreasing self-reducible *if there is a polynomial-time oracle transducer* T *such that* T^f *computes* f, *where on any input* x, *transducer* T *can query the oracle only about strings* y *such that* $y \prec x$.

The definition of lowness extends naturally to total, single-valued functions: A functional oracle f returns $f(x)$ on query x. For any relativizable complexity class $\mathcal{C}$ we say that $f \in Low(\mathcal{C})$ if $\mathcal{C}^f = \mathcal{C}$. We show next that self-reducible sets and self-reducible functions in NPMV/poly have identical lowness properties. Hence it suffices to prove lowness of self-reducible sets in NPMV/poly.

Theorem 4. *Let* $\mathcal{F}$ *contain all self-reducible functions in any of the four function classes* $\{\mathrm{NPMV/poly}, \mathrm{NPSV/poly}, \mathrm{NPMV}_t\mathrm{/poly}, \mathrm{NPSV}_t\mathrm{/poly}\}$. *Let* $\mathcal{C}$ *be the subclass of* $\mathcal{F}$ *consisting of characteristic functions (making* $\mathcal{C}$ *a language* *class, essentially). For every self-reducible function $f \in \mathcal{F}$ there is a self-reducible set $A \in \mathcal{C}$ such that f and A are polynomial-time Turing equivalent.*

Proof. Given $f \in \mathcal{F}$, we can define the corresponding set $A \in \mathcal{C}_{\mathcal{F}}$ by suitably encoding, for each x, the bits of $f(x)$ in A. We can easily ensure that the self-reducibility of f carries over to A and f and A are polynomial-time Turing equivalent.

Theorem 5. *Word-decreasing self-reducible sets in* $\mathrm{NPMV}_t/\mathrm{poly}$ *are low for* Σ_2^p*.*

Since Σ_k^p, Π_k^p, PP, $\mathrm{C_=P}$, $\mathrm{Mod}_m\mathrm{P}$, PSPACE, and EXP have many-one complete word-decreasing self-reducible sets [7], the following corollary is immediate.

Corollary 4. *If* $\mathcal{C} \in \{\Sigma_k^p, \Pi_k^p, \mathrm{PP}, \mathrm{C_=P}, \mathrm{Mod}_m\mathrm{P}, \mathrm{PSPACE}, \mathrm{EXP}\}$*, for* $k \geq 1$*, has a complete set in* $\mathrm{NPMV}_t/\mathrm{poly}$ *then* $\mathcal{C} \subseteq \Sigma_2^p$ *and* $\mathrm{PH} = \Sigma_2^p$*.*

The proof follows since for each $\mathcal{C} \in \{\Sigma_k^p, \Pi_k^p, \mathrm{PP}, \mathrm{C_=P}, \mathrm{Mod}_m\mathrm{P}, \mathrm{PSPACE}, \mathrm{EXP}\}$ and any set A complete for $\mathcal{C}$ w.r.t. polynomial-time Turing reductions we have $\Sigma_3^p \subseteq \Sigma_2^A$.

We end this section with the observation that $\mathrm{AM} \cap \mathrm{coAM}$ is contained in $\mathrm{NPMV}_t/\mathrm{poly}$. It is interesting to now compare the lowness results (Theorems 1 and 5) for these classes.

Proposition 2. *If* $L \in \mathrm{AM} \cap \mathrm{coAM}$ *then* L *is in* $\mathrm{NPMV}_t/\mathrm{poly}$*.*

Proof. Given $L \in \mathrm{AM} \cap \mathrm{coAM}$, as already observed in an earlier proof by probability amplification techniques and quantifier swapping, there are NP sets A and B and a polynomial p such that $\forall x : |x| \leq m$, there is a subset $S \subseteq \{0,1\}^{p(m)}$ of size $||S|| \geq 2^{p(m)-1}$ with the following property: $x \in L$ implies

$$\forall w : \langle x, w \rangle \in A \text{ and } \forall w \in S : \langle x, w \rangle \notin B$$

and $x \notin L$ implies

$$\forall w : \langle x, w \rangle \in B \text{ and } \forall w \in S : \langle x, w \rangle \notin A.$$

We can combine the NP machines for A and B and build a transducer I that takes pair $\langle x, w \rangle$ as input, where w is the advice string. Observe that S constitutes the set of w's that are correct advice strings. Using a $w \in S$ membership in L for strings of length m can be decided and for such advice strings the transducer I will always yield a single-valued, total computation for all inputs of length m, outputing either 1 or 0 depending on the membership of input x. Notice that the above properties also already imply L is in $\mathrm{NPMV}_t/\mathrm{poly}$, because no matter which $w \in \{0,1\}^{p(m)}$ is used as advice, $\langle x, w \rangle$ is either in the NP set A or in the NP set B and so the transducer I always outputs at least one of 0 or 1 for any advice string and any input.

5.2 *A lowness result for* NPSV/poly

In [11] it is left as an open problem to discover new lowness (or collapse consequence) results for NPSV/poly. As noted in [11], nothing better is known for NPSV/poly than the collapse consequence result: if SAT is in NPSV/poly then PH collapses to $\mathrm{ZPP}^{\Sigma_2^p}$, which holds even for the larger class NP/poly $\cap$ co-NP/poly [21].

We show that sets in NPSV/poly that are checkable, in the sense of program checking as defined by Blum and Kannan [9], are low for AM and for $\mathrm{ZPP}^{\mathrm{NP}}$. Since $\oplus$P, PP,PSPACE, and EXP have checkable complete problems, it follows that for any of these classes inclusion in NPSV/poly implies its containment in AM$\cap$coAM. This result is proved on the same lines as the Babai et al result [6]: If EXP is contained in P/poly then EXP $\subseteq$ MA.

Recall the definitions of MIP[$\mathcal{C}$] and IP[$\mathcal{C}$] for a class $\mathcal{C}$ of languages. We prove a technical lemma that immediately yields the lowness result.

Lemma 1. *If* $A \in$ NPSV/poly *then* MIP[A] $\subseteq$ AM.

Proof. Let $L \in$ MIP[A] for some set $A \in$ NPSV/poly. Let T be the nondeterministic transducer that witnesses that $A \in$ NPSV/poly. We describe an MAM protocol for L:

1. Let x be an input of length n to the protocol. Let $m = p(n)$, where p is a polynomial bounding the size of the queries to A made by the verifier during the protocol for inputs of length n.
2. **Merlin** sends advice w of length $q(m)$ to Arthur.
3. **Arthur** sends a polynomial random string r (used for simulating the original IP protocol) to Merlin.
4. **Merlin** sends back the list of successive queries to set A (generated by simulating the original IP protocol with random string r), the list of answers to those queries along with the computation paths of transducer T with advice w that certify the answers to the queries.
5. **Arthur** can verify in polynomial time that Merlin's message is all correct and accept if and only if the original IP protocol accepts.

By the fact that T computes a single-valued partial function for any advice w, although the verifier is simulating the nondeterministic transducer T, it is guaranteed that each accepting computation path has identical output and hence does identical computation. Thus, what makes the above MAM protocol work is the fact that for any advice w and query q all accepting computation paths of $T(q, w)$ output the same value. So, regardless of which computation paths are sent to Arthur by Merlin in Step 4 of the above protocol, Arthur's decision will be the same. In other words, Arthur's acceptance depends only on the random string r, hence exactly preserving the acceptance probability of the original IP protocol.

Standard techniques (cf. [4]) can be used to convert the MAM protocol to an AM protocol. This completes the proof.

We have as immediate consequence the following lowness result.

Theorem 6. *If L is a checkable set in* NPSV/poly *then* $L \in \mathrm{AM} \cap \mathrm{coAM}$ *and hence low for AM and* $\mathrm{ZPP}^{\mathrm{NP}}$.

Proof. The assumption in the theorem's statement implies that both L and $\overline{L}$ are in MIP[L] by the checker characterization theorem of [9]. Now, applying Lemma 1 yields that both L and $\overline{L}$ are in AM and the result follows.

We can derive new collapse consequences as corollary, since the classes $\oplus$P, PP, PSPACE, and EXP all have checkable complete problems. It follows that for any of these classes inclusion in NPSV/poly implies its containment in $\mathrm{AM} \cap \mathrm{coAM}$.

Corollary 5. *If any of the classes* $\oplus$P, PP, PSPACE, *and* EXP *is contained in* NPSV/poly *then it is low for* AM *and hence* PH = AM.

Notice that we have the same lowness for checkable functions in NPSV/poly.

Theorem 7. *Checkable functions in* NPSV/poly *are low for* AM *and* $\mathrm{ZPP}^{\mathrm{NP}}$.

Proof. Let f be a checkable function in NPSV/poly. We can suitably encode, for each x, the bits of $f(x)$ in a language A which is polynomial-time Turing equivalent to f and hence A is also checkable. The lowness result now follows by invoking Theorem 6.

Acknowledgements. The first author was partially supported by an Alexander von Humboldt fellowship in the year 1999, and he is grateful to Prof. Uwe Schöning for hosting his visit to Ulm university where this work was carried out.

References

1. V. Arvind and J. Köbler, *Pseudorandomness and resource-bounded measure*, Proceedings 17th Conference on the Foundations of Software Technology & Theoretical Computer Science, Springer-Verlag, LNCS 1346, pp. 235-249, 1997.
2. V. Arvind and J. Köbler, *Graph Isomorphism is Low for* $\mathrm{ZPP}^{\mathrm{NP}}$ *and other Lowness results*, ECCC Technical Report TR99-033, 1999.
3. V. Arvind, J. Köbler, and R. Schuler, *On helping and interactive proof systems*, International Journal of Foundations of Computer Science 6(2), 137-153, 1994.
4. L. Babai, *Bounded round interactive proofs in finite groups*, SIAM Journal of Discrete Mathematics, 5: 88-111, 1992.
5. N. Bshouty, R. Cleve, R. Gavaldà, S. Kannan, and C. Tamon, *Oracles and queries that are sufficient for exact learning*, Journal of Computer and System Sciences, 52, pp. 421–433 1996.
6. L. Babai, L. Fortnow, N. Nisan, and A. Wigderson, *BPP has subexponential simulations unless EXPTIME has publishable proofs*, Computational Complexity, 3, pp. 307–318, 1993.

7. J. Balcázar, *Self-reducibility*, Journal of Computer and System Sciences, 41 (1990), pp. 367–388.
8. J. L. Balcázar, J. Díaz, and J. Gabarró, *Structural Complexity I*, EATCS Monographs on Theoretical Computer Science. Springer-Verlag, second edition, 1995.
9. M. Blum and S. Kannan, Designing programs that check their work, *Journal of the ACM,* **43**:269–291, 1995.
10. R. V. Book, T. J. Long, and A. L. Selman, *Quantitative relativizations of complexity classes.* SIAM Journal on Computing, 13(3):461-487, August 1984.
11. J.Y. Cai, L.A. Hemaspaandra, and G. Wechsung, *Robust Reductions*, In *Proceedings of the 4th Annual International Computing and Combinatorics Conference*, Springer-Verlag, LNCS 1449, pp. 174-183, 1998.
12. S. Fenner, L. Fortnow, A. Naik, and J. Rogers, *Inverting onto functions*, In Proceedings of the 11th IEEE Conference on Computational Complexity, IEEE, New York, pp. 213–222, 1996.
13. O. Goldreich, S. Micali, and A. Wigderson, *Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems*, Journal of the ACM, 38:691–729, 1991.
14. O. Goldreich and D. Zuckerman, *Another proof that BPP⊆PH (and more)*, Technical Report TR97-045, Electronic Colloquium on Computational Complexity, October 1997.
15. R. M. Karp and R. J. Lipton, *Some connections between nonuniform and uniform complexity classes*, in Proceedings of the 12th ACM Symposium on Theory of Computing, ACM Press, 1980, pp. 302–309.
16. K. Ko and U. Schöning, *On circuit-size complexity and the low hierarchy in NP*, SIAM Journal on Computing, 14:41–51, 1985.
17. J. Köbler, *On the structure of low sets*, In Proceedings of the 10th Structure in Complexity Theory Conference, 246–261. IEEE Computer Society Press, 1995.
18. J. Köbler and U. Schöning, *High sets for NP*, In D. Zu and K. Ko, editors, Advances in Algorithms, Languages, and Complexity, pp. 139-156, Kluwer Acad. Publishers, 1997.
19. J. Köbler, U. Schöning, and J. Torán, *The Graph Isomorphism Problem: Its Structural Complexity*, Birkhäuser, Boston, 1993.
20. J. Köbler and R. Schuler, *Average-case intractability vs. worst-case intractability*, Proceedings of the conference on Mathematical Foundations of Computer Sciences (MFCS)¡/I¿,¡BR¿ Springer-Verlag, LNCS 1450, 493-502, 1998. ¡BR¿ ¡P¿
21. J. Köbler and O. Watanabe, *New collapse consequences of NP having small circuits*, Proceedings of the 22nd International Colloquium on Automata, Languages, and Programming, Springer-Verlag, LNCS 944, pp. 196–207, 1995.
22. N. Nisan and A. Wigderson, *Hardness vs randomness*, Journal of Computer and System Sciences, 49:149–167, 1994.
23. C. Papadimitriou, *Computational Complexity*, Addison-Wesley, 1994.
24. M. Santha, *Relativized Arthur-Merlin versus Merlin-Arthur games*, Information and Computation, 80(1), pp. 44–49, 1989.
25. U. Schöning, *A low and a high hierarchy within NP*, Journal of Computer and System Sciences, 27, pp. 14–28, 1983.
26. U. Schöning, *Probabilistic complexity classes and lowness*, Journal of Computer and System Sciences, 39, pp. 84–100, 1989.
27. A. Selman, *A taxanomy of complexity classes of functions*, Journal of Computer and System Sciences, 48(2), pp. 357–381, 1994.