New Collapse Consequences of NP Having Small Circuits

Johannes KÖBLER
Abteilung Theoretische Informatik
Universität Ulm
Oberer Eselsberg
89069 Ulm, Germany
koebler@informatik.uni-ulm.de

Osamu WATANABE[*]
Department of Computer Science
Tokyo Institute of Technology
Meguro-ku Ookayama 1-12-1
Tokyo 152, JAPAN
watanabe@cs.titech.ac.jp

Abstract

We show that if a self-reducible set has polynomial-size circuits, then it is low for the probabilistic class ZPP(NP). As a consequence we get a deeper collapse of the polynomial-time hierarchy PH to ZPP(NP) under the assumption that NP has polynomial-size circuits. This improves on the well-known result of Karp, Lipton, and Sipser [KL80] stating a collapse of PH to its second level Σ_2^{P} under the same assumption.

As a further consequence, we derive new collapse consequences under the assumption that complexity classes like UP, FewP, and $\mathrm{C_{=}P}$ have polynomial-size circuits.

Finally, we investigate the circuit-size complexity of several language classes. In particular, we show that for every fixed polynomial s, there is a set in ZPP(NP) which does not have $O(s(n))$-size circuits.

1 Introduction

The question whether intractable sets can be efficiently decided by non-uniform models of computation has motivated much work in structural complexity theory. In research from the early 1980's to the present, a variety of results has been obtained showing that this is impossible under plausible assumptions (see, e.g., the survey [HOW92]). A typical model for non-uniform computations are (families of) circuits. In the notation of Karp and Lipton [KL80], sets decidable by polynomial-size circuits are precisely the sets in P/poly, i.e., they are decidable in polynomial time with the help of a polynomial length bounded advice function [Pi79].

[*]The second author is supported in part by Grant in Aid for Scientific Research of the Ministry of Education, Science and Culture of Japan under Grant-in-Aid for Research (C) 06680308 (1994). Part of his work has been done while he was visiting Universität Ulm (supported in part by the guest scientific program of Universität Ulm).

Karp and Lipton (together with Sipser) [KL80] proved that no NP-complete set has polynomial size circuits (in symbols NP $\not\subseteq$ P/poly) unless the polynomial time hierarchy collapses to its second level. The proof given in [KL80] exploits a certain kind of self-reducibility of the well-known NP complete problem SAT. More generally, it is shown in [BBS86a, BBS86b] that every (Turing) self-reducible set in P/poly is low for the second level Σ_2^{P} of the polynomial time hierarchy. Intuitively speaking, a set is low for a relativizable complexity class if it gives no additional power when used as an oracle for that class.

In this paper, we show that every self-reducible set in P/poly is also low for the probabilistic class ZPP(NP). Since for every oracle A, $\Sigma_2^{\mathrm{P}}(A) = \exists \cdot \mathrm{ZPP}(\mathrm{NP}(A))$, lowness for ZPP(NP) implies lowness for Σ_2^{P}. As a consequence of our lowness result we get a deeper collapse of the polynomial-time hierarchy to ZPP(NP) under the assumption that NP has polynomial-size circuits. At least in some relativized world, the new collapse level is quite close to optimal: there is an oracle relative to which NP is contained in P/poly but PH does not collapse to P(NP) [He86, Wi85]. Our proof heavily uses the universal hashing technique [CW79, Si83] and builds on ideas from [Ang88, Ga92, Kö94]. A central notion used for the design of a zero error probabilistic algorithm is the concept of half-collisions introduced in the paper.

Based on our lowness result, we obtain new collapse consequences under the assumption that complexity classes like NP, UP, FewP, and $\mathrm{C}_{=}\mathrm{P}$ have polynomial-size circuits. We further obtain new *relativizable* collapses for the case that $\mathrm{Mod}_m\mathrm{P}$, PSPACE, or EXP have polynomial-size circuits.

Very recently, Bshouty, Cleve, Kannan, and Tamon [BCKT94] building on a result from [JVV86] have shown that the class of all circuits is exactly learnable in (randomized) expected polynomial time with equivalence queries and the aid of an NP oracle. This immediately implies that every set A in P/poly has an advice function in FZPP(NP(A)). More precisely, since the circuit produced by the probabilistic learning algorithm of [BCKT94] depends on the outcome of the coin flips, the FZPP(NP(A)) transducer T computes a *multi-valued* advice function, i.e., on input 0^n, T accepts with probability at least 1/2, and on every accepting path, T outputs some circuit that correctly decides all instances of length n w.r.t. A. Using the technique in [BCKT94] we are able to show that every self-reducible set A in P/poly has an advice function in FZPP(NP); thus providing an alternative way to deduce the ZPP(NP) lowness of all self-reducible sets in P/poly. However, since our main interest is in the collapse consequences of intractable problems having polynomial-size circuits we prefer to give a self-contained proof of the lowness result.

As a further application, we derive new circuit-size lower bounds. In particular, we show by relativizing proof techniques that for every fixed polynomial s, there is a set in ZPP(NP) which does not have $O(s(n))$-size circuits. This improves on the result of Kannan [Kan82] that for every polynomial s, the class $\Sigma_2^{\mathrm{P}} \cap \Pi_2^{\mathrm{P}}$ contains such a set. It further follows that in every relativized world, there exist sets in the class ZPEXP(NP) that do not have polynomial-size circuits. We mention that prior to the work of the present paper it has been shown by non-relativizing techniques that the subclass $\mathrm{MA_{exp}} \cap \text{co-}\mathrm{MA_{exp}}$ of ZPEXP(NP) contains non P/poly sets [Bu94, Th94].

The paper is organized as follows. Section 2 introduces notation and defines the self-reducibility used in the paper. In Section 3, we prove the ZPP(NP) lowness of all self-reducible sets in P/poly. In Section 4, we state the collapse consequences, and in Section 5, we derive the new circuit-size lower bounds.

2 Preliminaries and notation

All languages are over the binary alphabet $\Sigma = \{0, 1\}$. The *length* of a string $x \in \Sigma^*$ is denoted by $|x|$. $\Sigma^{\leq n}$ ($\Sigma^{<n}$) is the set of all strings of length at most n (resp., of length smaller than n). For a language A, $A^{=n} = A \cap \Sigma^n$ and $A^{\leq n} = A \cap \Sigma^{\leq n}$. The cardinality of a finite set A is denoted by $|A|$. The *characteristic function* of A is defined as $A(x) = 1$ if $x \in A$, and $A(x) = 0$, otherwise. For a class $\mathcal{C}$ of sets, co-$\mathcal{C}$ denotes the class $\{\Sigma^* - A \mid A \in \mathcal{C}\}$. To encode pairs (or tuples) of strings we use a standard polynomial-time computable pairing function denoted by $\langle \cdot, \cdot \rangle$ whose inverses are also computable in polynomial time. Where intent is clear we write $f(x_1, \ldots, x_k)$ in place of $f(\langle x_1, \ldots, x_k \rangle)$. $\mathcal{N}$ denotes the set of non-negative integers. Throughout the paper, the base of log is 2.

The textbooks [BDG, BC93, KST93, Pa94, Sch86] can be consulted for the standard notations used in the paper and for basic results in complexity theory. For definitions of probabilistic complexity classes like ZPP see also [Gi77].

An *NP machine M* is a polynomial-time nondeterministic Turing machine. Each computation path of M on some input x either accepts, rejects, or outputs "?". *M accepts x* if on input x, M has at least one accepting path, otherwise *M rejects x*. *M strongly accepts (strongly rejects) x* [Lo82] if

- at least one computation path on input x is accepting (resp., rejecting) and
- there are no rejecting (resp., accepting) computation paths on input x.

In this case, the computation of M on input x is called a *strong computation.* An NP machine that performs a strong computation on every input is called a *strong NP machine.* It is well known that exactly the sets in NP $\cap$ co-NP can be accepted by a strong NP machine [Lo82].

Next we define the kind of self-reducibility that we use in this paper.

Definition 2.1 *Let $\succ$ be an irreflexive and transitive order relation on Σ^*. A sequence $x_0, x_1, \ldots, x_k$ of strings is called a $\succ$-chain (of length k) from x_0 to x_k if $x_0 \succ x_1 \succ \cdots \succ x_k$. Relation $\succ$ is called* length checkable *if there is a polynomial q such that*

1. *for all $x, y \in \Sigma^*$, $x \succ y$ implies $|y| \leq q(|x|)$,*
2. *the language $\{\langle x, y, k \rangle \mid$ there is a $\succ$-chain of length k from x to $y\}$ is in* NP.

Definition 2.2 *A set A is* self-reducible*, if there is a polynomial-time oracle machine M and a length checkable order relation $\succ$ such that $A = L(M, A)$ and on any input x, M queries the oracle only about strings $y \prec x$.*

It is straightforward to check that the polynomially related self-reducible sets introduced by Ko [Ko83] as well as the length-decreasing and word-decreasing self-reducible sets of Balcázar [Ba90] are self-reducible in our sense. Furthermore, it is well-known that complexity classes like NP, Σ_k^{P}, Π_k^{P}, $k \geq 1$, PP, $\mathrm{C_{=}P}$, $\mathrm{Mod}_m\mathrm{P}$, $m \geq 2$, PSPACE, and EXP have many-one complete self-reducible sets (see, for example, [BDG, Ba90, OL93]).

Karp and Lipton [KL80] introduced the notion of advice functions in order to characterize non-uniform complexity classes. A function $h : \mathcal{N} \to \Sigma^*$ is called a *polynomial-length function* if for some polynomial p and for all $n \geq 0$, $|h(n)| = p(n)$. For a class $\mathcal{C}$ of sets, let $\mathcal{C}$/poly be the class of sets A such that there is a set $I \in \mathcal{C}$ and a polynomial-length function h such that

$$\forall n,\ \forall x \in \Sigma^{\leq n}\ [\ x \in A \ \Leftrightarrow\ \langle x, h(n)\rangle \in I\ ].$$

Function h is called an *advice function* for A, and I is the corresponding *interpreter set*.

In this paper, we will heavily make use of the "hashing technique", which has been very fruitful in complexity theory. Here we review some notations and facts about hash families. We also extend the notion of "collision" and introduce the concept of a "half collision" which is central to our proof technique.

Sipser [Si83] used universal hashing, originally invented by Carter and Wegman [CW79], to decide (probabilistically) whether a finite set X is large or small. A linear hash function h from Σ^m to Σ^k is given by a Boolean (k, m)-matrix (a_{ij}) and maps any string $x = x_1 \ldots x_m$ to some string $y = y_1 \ldots y_k$, where y_i is the inner product $a_i \cdot x = \sum_{j=1}^{m} a_{ij} x_j \pmod 2$ of the i-th row a_i and x.

Let $x \in \Sigma^m$, $Y \subseteq \Sigma^m$, and let h be a linear hash function from Σ^m to Σ^k. We say that x *has a collision on Y w.r.t. h* if there exists a string $y \in Y$, different from x, such that $h(x) = h(y)$. In general, for any $X \subseteq \Sigma^m$, and any family $H = \{h_1, \ldots, h_l\}$ of linear hash functions from Σ^m to Σ^k, *X has a collision on Y w.r.t. H* (*Collision*(X, Y, H) for short) if there is some $x \in X$ that has a collision on Y w.r.t. any h_i in H. That is,

$$\begin{aligned} \mathit{Collision}(X, Y, H) \ \Leftrightarrow\ & \exists\, x \in X\, \exists\, y_1, \ldots, y_l \in Y : x \notin \{y_1, \ldots y_l\} \\ & \text{and for all } i = 1, \ldots, l : h_i(x) = h_i(y_i). \end{aligned}$$

If X has a collision on itself w.r.t. H, we simply say that *X has a collision w.r.t. H*. Next we extend the notion of "collision" in the following way. For any X and $Y \subseteq \Sigma^m$, and any family $H = \{h_1, \ldots, h_l\}$ of linear hash functions, we say that *X has a half-collision on Y w.r.t. H* (*Half-Collision*(X, Y, H) for short) if there is some

$x \in X$ that has a collision on Y w.r.t. at least $\lceil l/2 \rceil$ many of the hash functions h_i in H. That is,

$$\begin{aligned} \textit{Half-Collision}(X, Y, H) \;\Leftrightarrow\; & \exists\, x \in X\, \exists\, y_1, \ldots, y_l \in Y : x \notin \{y_1, \ldots y_l\} \\ & \text{and } |\{i \mid 1 \leq i \leq l, h_i(x) = h_i(y_i)\}| \geq \lceil l/2 \rceil. \end{aligned}$$

An important relationship between collisions and half-collisions is the following one: If X has a collision w.r.t. H on $Y = Y_1 \cup Y_2$, then X must have a half-collision w.r.t. H either on Y_1 or on Y_2.

Note that the predicates $\textit{Collision}(X, Y, H)$ and $\textit{Half-Collision}(X, Y, H)$ can be decided in NP provided that membership in the sets X and Y can be tested in NP. More precisely, the sets $\{\langle v, H\rangle \mid \textit{Collision}(X_v, Y_v, H)\}$ and $\{\langle v, H\rangle \mid \textit{Half-Collision}(X_v, Y_v, H)\}$ are in NP, if the sets X_v and Y_v are succinctly represented in such a way that the languages $\{\langle x, v\rangle \mid x \in X_v\}$ and $\{\langle y, v\rangle \mid y \in Y_v\}$ are in NP.

We denote the set of all families $H = \{h_1, \ldots, h_l\}$ of l linear hash functions from Σ^m to Σ^k by $\mathcal{H}(l, m, k)$. The following theorem is proved by a pigeon-hole argument. It says that every sufficiently large set must have a collision w.r.t. any hash family.

Theorem 2.3 [Si83] *For any hash family $H \in \mathcal{H}(l, m, k)$ and any set $X \subseteq \Sigma^m$ of cardinality $|X| > l \cdot 2^k$, X must have a collision w.r.t. H.*

On the other hand, we get from the next theorem (called Coding Lemma in [Si83]) an upper bound on the collision probability for sufficiently small sets.

Theorem 2.4 [Si83] *Let $X \subseteq \Sigma^m$ be a set of cardinality at most 2^{k-1}. If we choose a hash family H uniformly random from $\mathcal{H}(k, m, k)$, then the probability that X has a collision w.r.t. H is at most $1/2$.*

We will also make use of the following extension of Theorem 2.4 which can be proved along the same lines.

Theorem 2.5 *Let $X \subseteq \Sigma^m$ be a set of cardinality at most 2^{k-s}. If we choose a hash family H uniformly random from $\mathcal{H}(l, m, k)$, then the probability that X has a collision w.r.t. H is at most $2^{k-s(l+1)}$.*

By combining Theorem 2.3 and Theorem 2.5, a rough estimation for the cardinality of a nonempty set $X \subseteq \Sigma^m$ can be obtained with high probability: choose $n \geq 0$ and for every $k = 1, \ldots, m$, randomly guess a hash family H_k from $\mathcal{H}(n + k, m, k)$; let $k_{max} \geq 0$ be the maximum $k \leq m$ such that for all $i \leq k$, X has a collision w.r.t. H_i; then we have that $|X| \leq (n + k_{max} + 1)2^{k_{max}+1}$, and with probability at least $1 - 2^{-n-1}$, $|X| > 2^{k_{max}-1}$.

Gavaldà extended Sipser's Coding Lemma (Theorem 2.4) to the case of a collection $\mathcal{C}$ of exponentially many sets. He proved that the probability that a random hash family consisting of $k(n+1)$ many functions simultaneously hashes a collection of 2^n many sets is at least $1/2$. The following theorem (letting $s = 1$ and $l = n+k$) shows that already $n + k$ many hash functions suffice.

Theorem 2.6 *Let $\mathcal{C}$ be a collection of at most 2^n subsets of Σ^m, each of which has cardinality at most 2^{k-s}. If we choose a hash family H uniformly random from $\mathcal{H}(l, m, k)$, then the probability that some $X \in \mathcal{C}$ has a collision w.r.t. H is at most $2^{n+k-s(l+1)}$.*

Proof By Theorem 2.5, we have that for every fixed $X \in \mathcal{C}$, the probability that it has a collision w.r.t. a randomly chosen hash family $H \in \mathcal{H}(l, m, k)$ is at most $2^{k-s(l+1)}$. Hence, the probability that there exists such a set $X \in \mathcal{C}$ is at most $2^{n+k-s(l+1)}$. ■

For the case of half-collisions we have the following probability bound.

Theorem 2.7 *Let $X \subseteq \Sigma^m$ and let $\mathcal{C}$ be a collection of at most 2^n subsets of Σ^m, each of which has cardinality at most 2^{k-s-2}. If we choose a hash family H uniformly random from $\mathcal{H}(l, m, k)$, then the probability that X has a half-collision on some $Y \in \mathcal{C}$ w.r.t. H is at most $|X| \cdot 2^{n-sl/2}$.*

Proof For every fixed $Y \in \mathcal{C}$ and every fixed $x \in X$, the probability that x has a collision on Y w.r.t. a randomly chosen h is at most 2^{-s-2}. Hence, the probability that x has a collision on Y w.r.t. at least half of the functions in a randomly chosen hash family $H \in \mathcal{H}(l, m, k)$ is at most

$$\sum_{i=\lceil l/2\rceil}^{l} \binom{l}{i} (2^{-s-2})^i (1 - 2^{-s-2})^{l-i} \leq 2^{-(s+2)l/2} \sum_{i=\lceil l/2\rceil}^{l} \binom{l}{i} \leq 2^{l-(s+2)l/2} = 2^{-sl/2}.$$

That is, the probability that x has a half-collision on Y w.r.t. a randomly chosen hash family H is bounded by $2^{-sl/2}$. Hence, the probability that there exists an $Y \in \mathcal{C}$ and an $x \in X$ such that x has a half-collision on Y w.r.t. H is at most $|X| \cdot 2^{n-sl/2}$. ■

3 Lowness of self-reducible sets in P/poly

In this section, we show that every self-reducible set A in (NP ∩ co-NP)/poly is low for ZPP(NP). Let I_A be an interpreter set and h_A be an advice function for A. We construct a probabilistic algorithm T_A and an NP oracle L_A having the following properties:

a) The expected running time of $T_A^{L_A}$ is polynomially bounded.

b) On every computation path on input 0^n, $T_A^{L_A}$ outputs some information that can be used to determine the membership of any x up to length n to A by some strong NP computation (in the sense of [Lo82]).

Using these properties, we can prove the lowness of A for ZPP(NP) as follows: In order to simulate any NP(A) computation, we first precompute the above mentioned information for A (up to some length) by $T_A^{L_A}$, and then by using this information, we can simulate the NP(A) computation by some NP(NP $\cap$ co-NP) computation. Note that the precomputation (performed by $T_A^{L_A}$) can be done in ZPP(NP), and since NP(NP $\cap$ co-NP) $=$ NP, the remaining computation can be done in NP. Hence, NP(A) $\subseteq$ ZPP(NP), which implies further that ZPP(NP(A)) $\subseteq$ ZPP(ZPP(NP)) ($=$ ZPP(NP) [Za82]).

We will now make the term "information" precise. For this, we need some additional notation. Let M_{self} be a polynomial-time oracle machine, let $\succ$ be a length checkable order relation, and let q be a polynomial witnessing the self-reducibility of A. We assume that $|h_A(q(n))| = p(n)$ for some fixed polynomial $p > 0$. In the following, we fix n and consider instances (to A) of length up to $q(n)$ as well as advice strings of length exactly $p(n)$.

- A *sample* is (the encoding of) a set of pairs of the form $\langle x_i, A(x_i)\rangle$, where the x_i's are instances.
- For any sample $S = \{\langle x_1, b_1\rangle, \ldots, \langle x_k, b_k\rangle\}$, let *Consistent*$(S)$ be the set $\{w \in \Sigma^{p(n)} \mid \forall i\,(1 \le i \le k) : I_A(x_i, w) = b_i\}$ of all advice strings w that are consistent with S.
- For any sample S and any instance x, let *Accept*(x,S) (resp., *Reject*(x,S)) be the set of all consistent advice strings that *accept* x (resp., *reject* x). That is, *Accept*$(x,S) = \{w \in \mathit{Consistent}(S) \mid I_A(x,w) = 1\}$ and *Reject*$(x,S) = \{w \in \mathit{Consistent}(S) \mid I_A(x,w) = 0\}$.
- Let *Correct*(x,S) be the set $\{w \in \mathit{Consistent}(S) \mid I_A(x,w) = A(x)\}$ of consistent advice strings that decide x correctly, and let *Incorrect*(x,S) be the complementary set $\{w \in \mathit{Consistent}(S) \mid I_A(x,w) \ne A(x)\}$.

Note that *Accept*(x,S) and *Reject*(x,S) form a partition of the set *Consistent*(S), and that

$$\begin{array}{lll} x \in A & \Rightarrow & \mathit{Correct}(x,S) = \mathit{Accept}(x,S) \text{ and } \mathit{Incorrect}(x,S) = \mathit{Reject}(x,S), \\ x \notin A & \Rightarrow & \mathit{Correct}(x,S) = \mathit{Reject}(x,S) \text{ and } \mathit{Incorrect}(x,S) = \mathit{Accept}(x,S). \end{array}$$

The above condition b) can now be precisely stated as follows:

b) On every computation path on input 0^n, $T_A^{L_A}$ outputs a pair (S,H) consisting of a sample S and a hash family $H \in \mathcal{H}(q(n)+k, p(n), k)$, for some k, $1 \le k \le p(n)$, such that for all x up to length n, *Consistent*(S) has a half-collision on *Correct*(x,S) w.r.t. H, but not on *Incorrect*(x,S).

```
**input** $0^n$
$S := \emptyset$
**loop**
  for $k = 1, \ldots, p(n)$, choose $H_k$ randomly from $\mathcal{H}(q(n)+k, p(n), k)$,
  $k_{max} := \max\{k \mid \forall i \leq k,\ \mathit{Consistent}(S)$ has a collision w.r.t. $H_i\}$
  $d := k_{max} - \lfloor c \log n \rfloor$
  **if** there exists an $x \in \Sigma^{\preceq n}$ such that $\mathit{Consistent}(S)$ has a half-collision
  on $\mathit{Incorrect}(x, S)$ w.r.t. $H_d$
  **then**
    use oracle $L_A$ to find such a string $x$ and to determine $A(x)$
    $S := S \cup \{\langle x, A(x)\rangle\}$
  **else exit**(**loop**) **end**
**end loop**
**output** $(S, H_d)$
```

Figure 1: The probabilistic algorithm T_A.

Once we have a pair (S, H) satisfying condition b), we can determine whether an instance x of length up to n is in A by simply checking on which one of $\mathit{Accept}(x, S)$ or $\mathit{Reject}(x, S)$, $\mathit{Consistent}(x, S)$ has a half-collision w.r.t. H. Since condition b) guarantees that the half-collision can always be found, this checking can be done by a strong NP computation. Let us now prove our main lemma.

Lemma 3.1 *For any self-reducible set A in* $(\mathrm{NP} \cap \text{co-NP})/\mathrm{poly}$, *there exist a probabilistic transducer T_A and an oracle L_A in NP satisfying the above conditions a) and b).*

Proof We use the notation introduced so far. Further, we denote by $\Sigma^{\preceq n}$ the set $\{y \mid \exists x \in \Sigma^{\leq n}, x \succeq y\}$. It is clear that $\Sigma^{\leq n} \subseteq \Sigma^{\preceq n} \subseteq \Sigma^{\leq q(n)}$ (recall that $q(n)$ is a length bound on the queries occuring in the self-reducing tree produced by M_{self} on any instance of length n). Let c be a fixed constant such that $q(n) + p(n) + 1 \leq 2^{\lfloor c \log n \rfloor - 2}$ for all sufficiently large n. (Recall that $p(n)$ is the advice length for the set of all instances of length up to $q(n)$.)

A description of T_A is given in Figure 1. Starting with the empty sample, T_A enters the main loop. During each execution of the loop, T_A first randomly guesses a series of $p(n)$ many hash families $H_k \in \mathcal{H}(q(n)+k, p(n), k), 1 \leq k \leq p(n)$. Then T_A computes the integer $d = k_{max} - \lfloor c \log n \rfloor$, where k_{max} is the maximum integer $k \in \{0, \ldots, p(n)\}$ such that $\mathit{Consistent}(S)$ has a collision w.r.t. all hash families $H_i, 1 \leq i \leq k$. (As argued below, we can assume that d is positive.)

Note, in particular, that $\mathit{Consistent}(S)$ has a collision w.r.t. H_d; thus, for every instance x, $\mathit{Consistent}(S)$ has a half-collision w.r.t. H_d on either $\mathit{Correct}(x, S)$ or $\mathit{Incorrect}(x, S)$.

If there exists a string $x \in \Sigma^{\preceq n}$ such that *Consistent*(S) has a half-collision on *Incorrect*(x, S) w.r.t. H_d, then this string is added to the sample S, and T_A reenters the loop. We will describe below how T_A uses the NP oracle L_A to find such an x (if it exists). Otherwise, the pair (S, H_d) has the desired properties as stated above, and T_A outputs the pair (S, H_d).
The intuition behind the choice of the value for d (depending on k_{max}) is as follows:

- d is still large enough to ensure that for a suitable polynomial t and for a random hash family $H \in \mathcal{H}(q(n) + d, p(n), d)$, the probability is exponentially small that *Consistent*(S) has a half-collision w.r.t. H on some set *Incorrect*(x, S) of size smaller than $c(S)/t(n)$.

- On the other hand, with high probability, d is so small that *Consistent*(S) has a collision w.r.t. every hash family $H \in \mathcal{H}(q(n) + d, p(n), d)$. This is important since in order to estimate the success probability of T_A we have to consider the *conditional* probability for *Consistent*(S) having a half-collision on X *given that Consistent*(S) has a collision w.r.t. H_d.

A more precise analysis follows. Let S be a sample and let d be the corresponding integer as determined by T_A (i.e., $d = k_{max} - \lfloor c \log n \rfloor$, where $k_{max} = p(n)$ or *Consistent*(S) does not have a collision w.r.t. some hash family $H_{k_{max}+1} \in \mathcal{H}(q(n) + k_{max} + 1, p(n), k_{max} + 1)$). We first estimate the probability that w.r.t. a uniformly at random chosen hash family $H \in \mathcal{H}(q(n) + d, p(n), d)$, *Consistent*$(S)$ has a half-collision on some set *Incorrect*(x, S) of relatively small size. Let $\mathcal{C}$ be the collection of all sets X of the form *Accept*(x, S) or *Reject*(x, S) for some $x \in \Sigma^{\leq q(n)}$ such that $|X| \leq c(S)2^{-2\lfloor c \log n \rfloor - 5}$.

Claim 1 *The probability of Consistent(S) having a half-collision on some $X \in \mathcal{C}$ w.r.t. a uniformly at random chosen hash family $H \in \mathcal{H}(q(n) + d, p(n), d)$ is at most* $2^{-q(n)-1}$.

Proof of Claim 1. By a padding trick we can assume that $c(S)$ is always larger than $2^{4\lfloor c \log n \rfloor}$. Since $k_{max} = p(n)$ or *Consistent*(S) does not have a collision w.r.t. some hash family $H_{k_{max}+1} \in \mathcal{H}(q(n) + k_{max} + 1, p(n), k_{max} + 1)$, it follows by Theorem 2.3 that $c(S) \leq (q(n) + k_{max} + 1)2^{k_{max}+1} \leq (q(n) + p(n) + 1)2^{d+\lfloor c \log n \rfloor + 1}$. Hence, we have

$$2^{4\lfloor c \log n \rfloor} \;\leq\; c(S) \;\leq\; 2^{d+2\lfloor c \log n \rfloor - 1}. \qquad (1)$$

Clearly, $|\mathcal{C}| \leq 2^{q(n)}$, and by (1), $|X| \leq c(S)2^{-2\lfloor c \log n \rfloor - 5} \leq 2^{d-6}$ for every $X \in \mathcal{C}$. Thus, it follows from Theorem 2.7 that the probability of *Consistent*(S) having a half-collision on some $X \in \mathcal{C}$ w.r.t. a uniformly at random chosen hash family $H \in \mathcal{H}(q(n) + d, p(n), d)$ is at most

$$\begin{aligned} c(S) \cdot 2^{q(n)-2(q(n)+d)} &\leq 2^{-q(n)-d+2\lfloor c \log n \rfloor} \\ &\leq 2^{-q(n)-1}. \end{aligned}$$

For the last step we used the lower bound $d \geq 2\lfloor c \log n \rfloor + 1$ which follows from inequality (1). □ *Proof of Claim 1.*

Now consider an arbitrary execution of the main loop during which S is expanded by some instance x. Since $c(S \cup \{\langle x, A(x) \rangle\}) = c(S) - |\mathit{Incorrect}(x, S)|$, the expected number of loop iterations is polynomially bounded provided that there is some fixed polynomial t such that $|\mathit{Incorrect}(x, S)| \leq c(S)/t(n)$ holds only with low probability.

Claim 2 *There is a polynomial t such that in each execution of the main loop, with probability at most 2^{-n} an instance x with $|\mathit{Incorrect}(x, S)| \leq c(S)/t(n)$ is selected.*

Proof of Claim 2. Observe that if $2^{k_{max}-1} \geq c(S)$, then $\mathit{Consistent}(S)$ must have a collision w.r.t. H_k, where integer k is chosen such that $2^{k-2} < c(S) \leq 2^{k-1}$. Thus it follows by Theorem 2.5 that

$$Pr[2^{k_{max}-1} \geq c(S)] \quad \leq \quad 2^{-q(n)-1}. \tag{2}$$

Since $q(n) + d + 1 \leq q(n) + p(n) + 1 \leq 2^{\lfloor c \log n \rfloor - 2}$, we have by the definition of d that

$$(q(n) + d + 1)2^{d+1} \quad \leq \quad 2^{k_{max}-1}. \tag{3}$$

Putting (2) and (3) together, we get that

$$Pr[(q(n) + d + 1)2^{d+1} < c(S)] \quad \geq \quad 1 - 2^{-q(n)-1}. \tag{4}$$

Let $t(n) \geq 2^{2\lfloor c \log n \rfloor + 5}$. Then, by the way collection $\mathcal{C}$ is defined, $|\mathit{Incorrect}(x, S)|$ can only be smaller than $c(S)/t(n)$ if $\mathit{Consistent}(S)$ has a half-collision on some $X \in \mathcal{C}$ w.r.t. H_d. To estimate the probability for this event, observe that, by (4), with probability at least $1 - 2^{-q(n)-1}$, $(q(n) + d + 1)2^{d+1} < c(S)$, implying (together with Theorem 2.3) that $\mathit{Consistent}(S)$ must have a collision w.r.t. any hash family in $\mathcal{H}(q(n) + d, p(n), d)$. By Claim 1 it follows that in this case $\mathit{Consistent}(S)$ has a half-collision on some $X \in \mathcal{C}$ w.r.t. H_d with probability at most $2^{-q(n)-1}$. Hence, in each execution of the main loop the total probability that $|\mathit{Incorrect}(x, S)| \leq c(S)/t(n)$ is bounded by

$$Pr[(q(n) + d + 1)2^{d+1} \geq c(S)] + Pr[(q(n) + d + 1)2^{d+1} < c(S)] \cdot 2^{-q(n)-1} \leq 2^{-n}.$$

□ *Proof of Claim 2.*

We finally show how T_A can find a string $x \in \Sigma^{\preceq n}$ such that $\mathit{Consistent}(S)$ has a half-collision on $\mathit{Incorrect}(x, S)$ w.r.t. H_d (if such an x exists). Define oracle L_A as follows:

$L_A = \{\langle 0^n, x, k, b, S, H\rangle \mid$ there is a $\succ$-chain of length k from some string $y \in \Sigma^{\leq n}$ to some string $z \leq x$ such that there is a computation path π of M_{self} on input z fulfilling the following properties:

- if a query q is answered 'yes' ('no') then *Consistent*(S) has a half-collision on *Accept*(q, S) (resp., *Reject*(q, S)) w.r.t. H,
- if π is accepting (rejecting) then *Consistent*(S)) has a half-collision on *Reject*(z, S) (resp., *Accept*(z, S)) w.r.t. H,
- if $b = 1$ then π is accepting$\}$.

If the tuple $\langle 0^n, 1^{q(n)}, 0, 0, S, H_d\rangle$ is not in T_A, then it follows by the definition of T_A that w.r.t. H_d, *Consistent*(S) does not have a half-collision on any of the sets *Incorrect*(x, S), $x \in \Sigma^{\preceq n}$.

Otherwise, by asking queries of the form $\langle 0^n, 1^{q(n)}, i, 0, S, H_d\rangle$, T_A can compute by binary search i_{max} as the maximum value $i \leq 2^{q(n)+1}$ such that $\langle 0^n, 1^{q(n)}, i, 0, S, H_d\rangle$ is in L_A (a similar idea is used in [LT91]). Knowing i_{max}, T_A can find the lexicographically smallest string x_{min} such that $\langle 0^n, x_{min}, i_{max}, 0, S, H_d\rangle$ is in L_A. Since for all $q \prec x_{min}$, *Consistent*(S) does not have a half-collision on *Incorrect*(q, S) w.r.t. H_d, it is easy to see that *Consistent*(S) must have a half-collision on *Incorrect*(x_{min}, S) w.r.t. H_d. Finally, T_A can determine $A(x_{min})$ by asking whether $\langle 0^n, x_{min}, i_{max}, 1, S, H_d\rangle$ is in L_A. ■

Theorem 3.2 *Every self-reducible set A in the class* (NP ∩ co-NP)/poly *is low for* ZPP(NP).

Proof We first show that NP$(A) \subseteq$ ZPP(NP). Let L be a set in NP(A), and let M be a deterministic polynomial-time oracle machine such that for some polynomial t,

$$L = \{x \mid \exists y \in \Sigma^{t(|x|)} : \langle x, y\rangle \in L(M, A)\}.$$

Let $s(n)$ be a polynomial bounding the length of all oracle queries of M on some input $\langle x, y\rangle$ where x is of length n. Then L can be accepted by a probabilistic oracle machine N using the following NP oracle O.

$O = \{\langle x, S, H\rangle \mid$ there is a $y \in \Sigma^{t(|x|)}$ such that M on input $\langle x, y\rangle$ has an accepting path π on which each query q is answered 'yes' ('no') only if *Consistent*(S) has a half-collision on *Accept*(q, S) (resp., *Reject*(q, S)) w.r.t. $H\,\}$.

Here is how N^O accepts L. On input x, N first simulates T_A on input $0^{s(|x|)}$ to compute a pair (S, H_d) as described above. Then N asks the query $\langle x, S, H_d\rangle$ to O to find out whether x is in L.

This proves that NP$(A) \subseteq$ ZPP(NP). Since ZPP(ZPP) = ZPP [Za82] via a proof that relativizes, it follows that ZPP(NP(A)) is also contained in ZPP(NP), showing that A is low for ZPP(NP). ■

4 Collapse consequences

As a direct consequence of Theorem 3.2 we get an improvement of Karp, Lipton, and Sipser's result [KL80] that NP is not contained in P/poly unless the polynomial-time hierarchy collapses to Σ_2^{P}.

Corollary 4.1 *If* NP *is contained in* (NP $\cap$ co-NP)/poly *then the polynomial-time hierarchy collapses to* ZPP(NP)*.*

Proof It is well-known that the NP complete set SAT is self-reducible. Thus, under the assumption that NP is contained in (NP $\cap$ co-NP)/poly, it follows that SAT is low for ZPP(NP). In other words, Σ_2^{P} = NP(NP) $\subseteq$ ZPP(NP(SAT)) $\subseteq$ ZPP(NP), implying the collapse of the polynomial-time hierarchy to ZPP(NP). ■

The collapse to ZPP(NP) in Corollary 4.1 is quite close to optimal, at least in some relativized world [He86, Wi85]: there is an oracle relative to which NP is contained in P/poly but the polynomial-time hierarchy does not collapse to P(NP).

In the rest of this section we report some other interesting collapses which can be easily derived using (by now) standard techniques, and which have also been pointed out independently by several researchers to the second author. First, it is straightforward to check that Theorem 3.2 relativizes: For any oracle B, if A is a self-reducible set in the class (NP$(B) \cap$ co-NP(B))/poly, then NP(A) is contained in ZPP(NP(B)). Consequently, Theorem 3.2 generalizes to the following result.

Theorem 4.2 *If A is a self-reducible set in the class* $(\Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}})$/poly*, then* NP$(A) \subseteq$ ZPP(Σ_k^{P})*.*

As a direct consequence of Theorem 4.2 we get an improvement of results in [AFK89, Kä91] stating (for $k = 1$) that Σ_k^{P} is not contained in $(\Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}})$/poly unless the polynomial-time hierarchy collapses to $\Sigma_{k+1}^{\mathrm{P}}$.

Corollary 4.3 *Let $k \geq 1$. If Σ_k^{P} is contained in* $(\Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}})$/poly*, then the polynomial-time hierarchy collapses to* ZPP(Σ_k^{P})*.*

Proof Since Σ_k^{P} contains complete self-reducible languages, the assumption that Σ_k^{P} is contained in $(\Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}})$/poly implies that $\Sigma_{k+1}^{\mathrm{P}}$ = NP$(\Sigma_k^{\mathrm{P}}) \subseteq$ ZPP(Σ_k^{P}). ■

A further consequence of Theorem 4.2 is the following improvement of a result due to Yap [Yap83] stating that Π_k^{P} is not contained in Σ_k^{P}/poly unless the polynomial-time hierarchy collapses to $\Sigma_{k+2}^{\mathrm{P}}$.

Corollary 4.4 *Let $k \geq 1$. If $\Pi_k^{\mathrm{P}} \subseteq \Sigma_k^{\mathrm{P}}/\mathrm{poly}$, then* PH *collapses to* $\mathrm{ZPP}(\Sigma_{k+1}^{\mathrm{P}})$.

Proof The assumption that Π_k^{P} is contained in $\Sigma_k^{\mathrm{P}}/\mathrm{poly}$ implies that $\Sigma_{k+1}^{\mathrm{P}}$ is contained in $\Sigma_k^{\mathrm{P}}/\mathrm{poly} \subseteq (\Sigma_{k+1}^{\mathrm{P}} \cap \Pi_{k+1}^{\mathrm{P}})/\mathrm{poly}$. Hence we can apply Corollary 4.3. ■

Corollary 4.5

i) *For $\mathcal{K} \in \{\mathrm{UP}, \mathrm{FewP}\}$, if $\mathcal{K} \subseteq (\mathrm{NP} \cap \text{co-NP})/\mathrm{poly}$ then $\mathcal{K}$ is low for* ZPP(NP).

ii) *For every $k \geq 1$, if $\mathrm{C_=P} \subseteq (\Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}})/\mathrm{poly}$ then $\mathrm{CH} = \mathrm{ZPP}(\Sigma_k^{\mathrm{P}})$.*

Proof

i) It is well-known that for every set A in UP (FewP), the left set of A [OW91] is word-decreasing self-reducible and in UP (resp., FewP). Thus, under the assumption that $\mathrm{UP} \subseteq (\mathrm{NP} \cap \text{co-NP})/\mathrm{poly}$ (resp., $\mathrm{FewP} \subseteq (\mathrm{NP} \cap \text{co-NP})/\mathrm{poly}$) it follows by Theorem 3.2 that the left set of A (and since A is polynomial-time many-one reducible to its left set, also A) is low for ZPP(NP).

ii) First, since $\mathrm{C_=P}$ has complete word-decreasing self-reducible languages [OL93], $\mathrm{C_=P} \subseteq (\Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}})/\mathrm{poly}$ implies $\mathrm{C_=P} \subseteq \mathrm{ZPP}(\Sigma_k^{\mathrm{P}}) \subseteq \mathrm{PH}$. Second, since $\mathrm{PH} \subseteq \mathrm{BPP}(\mathrm{C_=P})$ [TO92, Ta93], $\mathrm{C_=P} \subseteq (\Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}})/\mathrm{poly}$ implies $\mathrm{PH} \subseteq (\Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}})/\mathrm{poly}$ and therefore PH collapses to $\mathrm{ZPP}(\Sigma_k^{\mathrm{P}})$ by Corollary 4.3. Finally, since $\mathrm{CH} = \mathrm{C_=P} \cup \mathrm{C_=P}(\mathrm{C_=P}) \cup \ldots$ [To91], it follows that $\mathrm{C_=P}(\mathrm{PH}) \subseteq \mathrm{BPP}(\mathrm{C_=P})$ [TO92] $\subseteq \mathrm{PH}$, and thus we get inductively that $\mathrm{CH} \subseteq \mathrm{PH}$ ($\subseteq \mathrm{ZPP}(\Sigma_k^{\mathrm{P}})$). ■

Under certain assumptions as, for example, $\mathrm{Mod}_m\mathrm{P} \subseteq \mathrm{P}/\mathrm{poly}$ a collapse of PH to the subclass $\mathrm{MA} \cap \text{co-MA}$ of ZPP(NP) could be shown.

Theorem 4.6 [LFKN92, BFL91, BF91] *For $\mathcal{K} \in \{\mathrm{PP}, \mathrm{Mod}_m\mathrm{P}, \mathrm{PSPACE}, \mathrm{EXP}\}$, if $\mathcal{K} \subseteq \mathrm{P}/\mathrm{poly}$ then $\mathcal{K} \subseteq \mathrm{MA}$.*

However, in contrast to Theorem 4.6 which has been proved by non-relativizing techniques, and therefore is not known to relativize, the proof of the following corollary of Theorem 4.2 relativizes. Harry Buhrman pointed out that the unrelativized version of Corollary 4.7 can also be derived from Theorem 4.6.

Corollary 4.7 *For every $k \geq 1$, and in every relativized world,*

i) *For $m \geq 2$, if $\mathrm{Mod}_m\mathrm{P} \subseteq (\Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}})/\mathrm{poly}$ then $\mathrm{Mod}_m\mathrm{P} \subseteq \mathrm{PH} = \mathrm{ZPP}(\Sigma_k^{\mathrm{P}})$.*

ii) *If $\mathrm{PSPACE} \subseteq (\Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}})/\mathrm{poly}$ then $\mathrm{PSPACE} = \mathrm{ZPP}(\Sigma_k^{\mathrm{P}})$.*

iii) *If $\mathrm{EXP} \subseteq (\Sigma_k^{\mathrm{P}} \cap \Pi_k^{\mathrm{P}})/\mathrm{poly}$ then $\mathrm{EXP} = \mathrm{ZPP}(\Sigma_k^{\mathrm{P}})$.*

Proof

i) The proof is analogous to the one of part *ii)* of Corollary 4.5 using the fact that $\mathrm{Mod}_m\mathrm{P}$ has complete word-decreasing self-reducible languages [OL93], and that $\mathrm{PH} \subseteq \mathrm{BPP}(\mathrm{Mod}_m\mathrm{P})$ [TO92, Ta93].

ii) is immediate from Theorem 4.2 since PSPACE has complete (length-decreasing) self-reducible languages.

iii) is also immediate from Theorem 4.2 since EXP has complete (word-decreasing) self-reducible languages [Ba90]. ■

5 Circuit complexity

Kannan [Kan82] proved that for every fixed polynomial s, there is a set in $\Sigma_2^{\mathrm{P}} \cap \Pi_2^{\mathrm{P}}$ which does not have $O(s(n))$-size circuits. Using a padding argument, he obtained the existence of sets in NEXP(NP) ∩ co-NEXP(NP) not having polynomial-size circuits.

Theorem 5.1 [Kan82]

1. *For every polynomial s, there is a set A_s in $\Sigma_2^{\mathrm{P}} \cap \Pi_2^{\mathrm{P}}$ not having $O(s(n))$-size circuits.*

2. *For every increasing time-constructible super-polynomial function $f(n)$, there is a set A_f in* $\mathrm{NTIME}[f(n)](\mathrm{NP}) \cap \text{co-NTIME}[f(n)](\mathrm{NP})$ *not having polynomial size circuits.*

As an application of our results in Section 3, we can improve Kannan's results in every relativized world from the class $\Sigma_2^{\mathrm{P}} \cap \Pi_2^{\mathrm{P}}$ to ZPP(NP), and from the class $\mathrm{NTIME}[f(n)](\mathrm{NP}) \cap \text{co-NTIME}[f(n)](\mathrm{NP})$ to $\mathrm{ZPTIME}[f(n)](\mathrm{NP})$, respectively. Here $\mathrm{ZPTIME}[f(n)](\mathrm{NP})$ denotes the class of all sets that are accepted by some probabilistic machine relative to some oracle set $A \in \mathrm{NP}$, with zero error probability and within expected running time $O(f(n))$.

Note that for all sets in the class P/poly, we may fix the interpreter set to some appropriate one in P. Let I_{univ} denote such a fixed interpreter set. Furthermore, the class P/poly remains unchanged, if we relax the notion of an advice function h_A such that $h_A(n)$ has only to decide $A^{=n}$ correctly (instead of $A^{\leq n}$). That is, in this section, any function h_A is called an advice function for A (w.r.t. I_{univ}), if for every x, $A(x) = I_{univ}(x, h_A(|x|))$.

A sequence C_n, $n \geq 0$, of circuits is called a circuit family for A, if for every $n \geq 0$, C_n has n input gates, and for all n-bit strings $x_1 \cdots x_n$, $C_n(x_1, \ldots, x_n) = A(x_1 \cdots x_n)$. It is well-known (see, e.g., [BDG]) that I_{univ} can be chosen in such a way that advice length and circuit size (i.e., number of gates) are polynomially related to each other. More precisely, we can assume that there is a polynomial p such that the following holds for every set A.

- If h is an advice function for A w.r.t. I_{univ}, then there exists a circuit family C_n, $n \geq 0$, for A of size $|C_n| \leq p(n + |h(n)|)$.
- If C_n, $n \geq 0$, is a circuit family for A, then there exists an advice function h for A w.r.t. I_{univ} of length $|h(n)| \leq p(|C_n|)$.

Moreover, we can assume that for every polynomial-time interpreter set I there is a constant c_I such that if h is an advice function for A w.r.t. I, then there exists an advice function h' for A w.r.t. I_{univ} of length $|h'(n)| \leq |h(n)| + c_I$ for all n.

The following lemma is obtained by a direct diagonalization (cf. the corresponding result in [Kan82]). A set S is called $\mathrm{P}^{\mathcal{C}}$-printable (see [HY84]) if there is a polynomial-time oracle transducer T and an oracle set $A \in \mathcal{C}$ such that on any input 0^n, T^A outputs a list of all strings in $S^{\leq n}$.

Lemma 5.2 *For every fixed polynomial s, there is a Δ_3^{P}-printable set A_s such that every advice function h for A_s is of length $|h(n)| \geq s(n)$ for almost all n.*

Proof For a given n, let $x_1, x_2, \ldots, x_{2^n}$ be the sequence of strings of length n, enumerated in lexicographic order. Consider the two sets *Have-Advice* and *Find-A* defined as follows.

$$\langle n, a_1 \cdots a_{s(n)} \rangle \in \textit{Have-Advice} \Leftrightarrow \\ \exists\, w \in \Sigma^{<s(n)}, \forall\, i, 1 \leq i \leq \min(s(n), 2^n) : a_i = I_{univ}(x_i, w),$$

$$\langle n, a_1 \cdots a_j 1 0^{s(n)-j} \rangle \in \textit{Find-A} \Leftrightarrow \\ \exists\, a_{j+1} \cdots a_{s(n)} : \langle n, a_1 \cdots a_j a_{j+1} \cdots a_{s(n)} \rangle \notin \textit{Have-Advice}.$$

Note that for all n such that $s(n) \leq 2^n$, at least one pair of the form $\langle n, a_1 \cdots a_{s(n)} \rangle$ is not contained in *Have-Advice*. That is, there is no advice of length smaller than $s(n)$ that accepts the strings $x_1, \ldots, x_{s(n)}$ according to $a_1, \ldots, a_{s(n)}$. Let α_n denote the lexicographically smallest such pair. Then A_s is defined as follows: a string x_i of length n is in A_s if and only if $1 \leq i \leq s(n) \leq 2^n$, and the ith bit of α_n (i.e., a_i) is 1. Then clearly, for almost all n, $A_s^{=n}$ has no advice of length smaller than $s(n)$. Furthermore, *Have-Advice* is in NP and *Find-A* is in NP(NP). Note also that by using *Find-A* as an oracle, α_n is computable in polynomial time w.r.t. n. Hence, A_s is P(NP(NP))-printable. ∎

By a simple modification of Kannan's proof that for every polynomial s, the class $\Sigma_2^{\mathrm{P}} \cap \Pi_2^{\mathrm{P}}$ contains a set which does not have $O(s(n))$-size circuits we obtain the following corollary.

Corollary 5.3 *For every fixed polynomial s, there is a set A_s in* ZPP(NP) *which does not have $O(s(n))$-size circuits.*

Proof If NP does not have polynomial-size circuits, then we can take $A_s = \mathrm{SAT}$. Otherwise, PH = ZPP(NP) by Corollary 4.1, and thus the theorem easily follows from Lemma 5.2. ■

We remark that it has been observed by Harry Buhrman and independently by Thomas Thierauf that Theorem 4.6 can be used to show that the class $\mathrm{MA_{exp}} \cap \text{co-}\mathrm{MA_{exp}}$ contains non P/poly sets (see Corollary 5.4 below). Here, $\mathrm{MA_{exp}}$ denotes the exponential-time version of Babai's class MA [BM88]. That is, $\mathrm{MA_{exp}} = \mathrm{MA}[2^{n^{O(1)}}]$, where a language L is in $\mathrm{MA}[f(n)]$, if there exists a set $B \in \mathrm{DTIME}[O(n)]$ such that for all x of length n,

$$\begin{aligned} x \in L &\Rightarrow \exists y, |y| = f(n) : \Pr[\langle x, y, z\rangle \in B] > 2/3, \\ x \notin L &\Rightarrow \forall y, |y| = f(n) : \Pr[\langle x, y, z\rangle \in B] < 1/3, \end{aligned}$$

and z is chosen uniformly random from $\Sigma^{f(n)}$.

Corollary 5.4 [Bu94, Th94] $\mathrm{MA_{exp}} \cap \text{co-}\mathrm{MA_{exp}}$ *contains sets that do not have polynomial size circuits.*

Proof If EXP does not have polynomial-size circuits, then any EXP complete set is in $\mathrm{MA_{exp}} \cap \text{co-}\mathrm{MA_{exp}}$ but not in P/poly. Otherwise, $\mathrm{PH} = \mathrm{MA} \cap \text{co-MA}$ by Theorem 4.6, and we can apply the well-known upward collapse technique (see, for example, [Bo74]) to conclude that $\mathrm{NEXP(NP)} \cap \text{co-NEXP(NP)}$ collapses to $\mathrm{MA_{exp}} \cap \text{co-}\mathrm{MA_{exp}}$. Now the corollary immediately follows by Theorem 5.1. ■

It was communicated to us by Harry Buhrman [Bu94] that for any increasing time-constructible super-polynomial function $f(n)$, Corollary 5.4 can be improved from $\mathrm{MA_{exp}} \cap \text{co-}\mathrm{MA_{exp}}$ to $\mathrm{MA}[f(n)] \cap \text{co-MA}[f(n)]$. Both these results, however, are proved by non-relativizing techniques, and therefore the following corollary is incomparable to Corollary 5.4 as well as to its improvement.

Corollary 5.5 *For every increasing time-constructible super-polynomial function $f(n)$, and in every relativized world, there is a set A_f in* $\mathrm{ZPTIME}[f(n)](\mathrm{NP})$ *which does not have polynomial-size circuits.*

Proof If NP does not have polynomial-size circuits, then we can take $A_f = \mathrm{SAT}$. Otherwise, PH = ZPP(NP) by Corollary 4.1, and thus it follows from Lemma 5.2 that there is a set A in $\mathrm{ZPTIME}[n^k](\mathrm{NP})$ such that every advice function h for A is of length $|h(n)| \geq n$ for almost all n. By the proof technique of Lemma 5.2,we can assume that in all length n strings of A, 1's only occur at the $O(\log n)$ rightmost positions. Now consider the following set A_f and interpreter set I:

$$A_f = \{x \mid 0^{\lfloor f(n)^{1/k} \rfloor - n} x \in A\},$$

$$I = \{\langle y, w\rangle \mid y = 0^{\lfloor f(n)^{1/k} \rfloor - n} x, \langle x, w\rangle \in I_{univ}\}.$$

Clearly, A_f is in ZPTIME$[f(n)]$(NP) and I is in P. Furthermore, it follows for every advice function h_f for A_f that for every y of the form $0^{\lfloor f(n)^{1/k} \rfloor - n}x$, $|x| = n$,

$$\begin{aligned} y \in A &\Leftrightarrow \langle y, h_f(n) \rangle \in I \\ &\Leftrightarrow \langle y, h'_f(n) \rangle \in I_{univ} \end{aligned}$$

for a suitable advice function $h'_f(n)$ of length $|h'_f(n)| \leq |h_f(n)| + c_I$. Thus, we have for almost all n,

$$|h_f(n)| \geq |h'_f(n)| - c_I \geq |y| - c_I = \lfloor f(n)^{1/k} \rfloor - c_I.$$

This shows that the length of h_f is super-polynomial. ∎

Corollary 5.6 *In every relativized world,* ZPEXP(NP) *contains sets that do not have polynomial-size circuits.*

We notice that it is not possible to extend Corollary 5.6 by relativizing techniques to the class EXP(NP), since there exist recursive oracles relative to which all sets in EXP(NP) have polynomial size circuits [Wi85, He86].

6 Concluding remarks

An interesting question concerning complexity classes that are known to contain non P/poly sets but possibly don't have complete sets is whether an explicit non P/poly set can be *constructed* in that class. For example, by Corollary 5.4 we know that the class $\mathrm{MA}_{\mathrm{exp}} \cap$ co-$\mathrm{MA}_{\mathrm{exp}}$ must contain sets that do not have polynomial-size circuits. But we were not able to give a *constructive* proof of this fact. To our knowledge, it is not even known whether the existence of a non P/poly set can be constructively proved within the class NEXP(NP) ∩ co-NEXP(NP).

Acknowledgments

For helpful discussions and suggestions regarding this work we are very grateful to H. Buhrman, R. Gavaldà, L. Hemaspaandra, M. Ogihara, U. Schöning, R. Schuler, and T. Thierauf. We also thank H. Buhrman, L. Hemaspaandra, M. Ogihara, and T. Thierauf for permitting us to include their observations in the paper.

References

[AFK89] M. Abadi, J. Feigenbaum, and J. Kilian. On hiding information from an oracle. *Journal of Computer and System Sciences* **39** (1989) 21–30.

[Ang88] D. Angluin. Queries and concept learning. *Machine Learning* **2** (1988) 319–342.

[BF91] L. Babai, L. Fortnow. Arithmetization: A new method in structural complexity. *Computational Complexity* **1** (1991) 41–66.

[BFL91] L. Babai, L. Fortnow, C. Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity* **1** (1991) 1–40.

[BM88] L. Babai and S. Moran. Arthur-Merlin games: a randomized proof system and a hierarchy of complexity classes. *Journal of Computer and System Sciences* **36** (1988) 254–276.

[Ba90] J.L. Balcázar. Self-reducibility. *Journal of Computer and System Sciences* **41** (1990) 367–388.

[BBS86a] J.L. Balcázar, R. Book, and U. Schöning. Sparse sets, lowness and highness. *SIAM Journal on Computing* **23** (1986) 679–688.

[BBS86b] J.L. Balcázar, R. Book, and U. Schöning. The polynomial-time hierarchy and sparse oracles. *Journal of the ACM* **33(3)** (1986) 603–617.

[BDG] J.L. Balcázar, J. Díaz, J. Gabarró. *Structural Complexity Theory.* (Springer, Berlin, 1988 and 1990).

[Bo74] R. Book. Tally languages and complexity classes. *Information and Control* **26** (1974) 186–193.

[BCKT94] N.H. Bshouty, R. Cleve, S. Kannan, and C. Tamon. Oracles and queries that are sufficient for exact learning. *Proceedings 7th ACM Conference on Computational Learning Theory* (1994) 130–139.

[BC93] D.P. Bovet, P. Crescenzi. *Introduction to the Theory of Complexity.* Prentice-Hall, 1993.

[Bu94] H. Buhrman, *personal communication.*

[CW79] J.L. Carter and M.N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences* **18** (1979) 143–154.

[Ga92] R. Gavaldà. Bounding the complexity of advice functions. *Proceedings of the 7th Structure in Complexity Theory Conference* (IEEE, New York, 1992) 249–254.

[Gi77] J. Gill. Computational complexity of probabilistic complexity classes. *SIAM Journal on Computing* **6** (1977) 675–695.

[HY84] J. Hartmanis and Y. Yesha. Computation times of NP sets of different densities. *Theoretical Computer Science* **34** (1984) 17–32.

[He86] H. Heller. On relativized exponential and probabilistic complexity classes. *Information and Control* **71** (1986) 231–243.

[HOW92] L. Hemachandra, M. Ogiwara, and O. Watanabe. How hard are sparse sets? *Proceedings of the 7th Structure in Complexity Theory Conference* (IEEE, New York, 1992) 222–238.

[JVV86] M.R. Jerrum, L.G. Valiant, V.V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science* **43** (1986) 169–188.

[Kä91] J. Kämper. Non-uniform proof systems: A new framework to describe non-uniform and probabilistic complexity classes. *Theoretical Computer Science* **85(2)** (1991) 305-331.

[Kan82] R. Kannan. Circuit-size lower bounds and non-reducibility to sparse sets. *Information and Control* **55** (1982) 40–56.

[KL80] R.M. Karp and R.J. Lipton. Some connections between nonuniform and uniform complexity classes. *Proceedings 12th ACM Symposium Theory of Computing* (1980) 302–309.

[Ko83] K. Ko. On self-reducibility and weak p-selectivity. *Journal of Computer and System Sciences* **26** (1983) 209–221.

[Kö94] J. Köbler. Locating P/poly optimally in the extended low hierarchy. To appear in *Theoretical Computer Science.*

[KST93] J. Köbler, U. Schöning, J. Torán. *The Graph Isomorphism Problem: Its Structural Complexity.* Birkhäuser, Boston, 1993.

[Lo82] T.J. Long. Strong nondeterministic polynomial-time reducibilities. *Theoretical Computer Science* **21** (1982) 1–25.

[LT91] A. Lozano and J. Torán. Self-reducible sets of small density. *Mathematical Systems Theory* **24** (1991) 83–100.

[LFKN92] C. Lund, L. Fortnow, H. Karloff, N. Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM* **39(4)** (1992) 859–868.

[OL93] M. Ogiwara and A. Lozano. On sparse hard sets for counting classes. *Theoretical Computer Science* **112** (1993) 255–275.

[OW91] M. Ogiwara and O. Watanabe. On polynomial-time bounded truth-table reducibility of NP sets to sparse sets. *SIAM Journal on Computing* **20(3)** (1991) 471–483.

[Pa94] C.H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[Pi79] N. Pippenger. On simultaneous resource bounds. *Proceedings 20th Symposium on Foundations of Computer Science* (IEEE, New York, 1979) 307–311.

[Sch86] U. Schöning. *Complexity and Structure.* Lecture Notes in Computer Science, Vol. 211 (Springer, Berlin, 1986).

[Si83] M. Sipser. A complexity theoretic approach to randomness. *Proceedings 15th ACM Symposium Theory of Computing* (1983) 330–335.

[Ta93] J. Tarui. Probabilistic polynomials, AC^0 functions, and the polynomial-time hierarchy. *Theoretical Computer Science* **113** (1993) 167–183.

[Th94] T. Thierauf, *personal communication.*

[TO92] S. Toda and M. Ogiwara. Counting classes are at least as hard as the polynomial-time hierarchy. *SIAM Journal on Computing* **21** (1992) 316–328.

[To91] J. Torán. Complexity classes defined by counting quantifiers. *Journal of the ACM* **38** (1991) 753–774.

[Wi85] C. Wilson. Relativized circuit complexity. *Journal of Computer and System Sciences* **31(2)** (1985) 169–181.

[Yap83] C. Yap. Some consequences of non-uniform conditions on uniform classes. *Theoretical Computer Science* **26** (1983) 287–300.

[Za82] S. Zachos. Robustness of probabilistic computational complexity classes under definitional perturbations. *Information and Control* **54** (1982) 143–154.