Non-autoreducible Sets for NEXP

Dung T. Nguyen and Alan L. Selman

University at Buffalo, The State University of New York, NY, US
{dtn3,selman}@buffalo.edu

Abstract

We investigate autoreducibility properties of complete sets for NEXP under different polynomial-time reductions. Specifically, we show that under some polynomial-time reductions there are complete sets for NEXP that are not autoreducible. We show that settling the question whether every $\leq^p_{dtt}$-complete set for NEXP is $\leq^p_{NOR\text{-}tt}$-autoreducible either positively or negatively would lead to major results about the exponential time complexity classes.

1998 ACM Subject Classification F.1.3 Complexity Measures and Classes

Keywords and phrases Autoreducibility, NEXP, diagonalization, structural complexity

Digital Object Identifier 10.4230/LIPIcs.STACS.2014.590

1 Introduction

Autoreducibility was first introduced by Trakhtenbrot [11]. A set A is autoreducible if A is reducible to A via an oracle Turing machine M such that M never queries x on input x. Ambos-Spies [1] introduced the polynomial-time variant of autoreducibility, where the oracle Turing machine now runs in polynomial time. Each notion of polynomial-time reduction induces the corresponding notion of autoreducibility.

The main question that has drawn many researchers' attention is whether complete sets for various complexity classes are polynomial-time autoreducible. Over many years, many results about autoreducibility of complete sets of different classes have been discovered. Glaßer et al. [7] showed that all m-complete sets of the following complexity classes are many-one autoreducible: NP, PSPACE, EXP, NEXP, $\Sigma^{\mathrm{P}}_{\mathrm{k}}$, $\Pi^{\mathrm{P}}_{\mathrm{k}}$, and $\Delta^{\mathrm{P}}_{\mathrm{k}}$ for $k \geq 1$. Beigel and Feigenbaum [10] showed that all Turing complete sets for any class $\Sigma^{\mathrm{P}}_{\mathrm{k}}$, $\Pi^{\mathrm{P}}_{\mathrm{k}}$, $\Delta^{\mathrm{P}}_{\mathrm{k}}$, $k \geq 0$, are Turing autoreducible. Also, all Turing complete sets for NP are Turing autoreducible.

Resolving some open questions about autoreducibility would lead to major class separation results. Buhrman et al. [2] proved various autoreducibility results for many different complexity classes and demonstrated strong evidence that studying structural properties of the complete sets, especially the autoreducibility property, might be an important tool to separate complexity classes. For example, if there exists a Turing complete set of NEXP that is not Turing autoreducible, then EXP is different from NEXP.

We reinforce this belief with the following result. Let hypothesis A be the assertion that every $\leq^p_{dtt}$-complete set for NEXP is $\leq^p_{NOR\text{-}tt}$-autoreducible. We prove that hypothesis A is true if and only if NEXP = coNEXP. It follows immediately that $\neg A$ implies NEXP $\neq$ EXP. We see that settling hypothesis A either positively or negatively solves important problems about these classes.

With this motivation in mind, we study autoreducibility questions for NEXP. Buhrman et al. [2] extensively studied autoreducibility for EXP. It is known that under many-one, 1-tt, 2-tt, and Turing reductions, all complete sets for EXP are autoreducible. Also for any $k \geq 3$, under $\leq^p_{k\text{-}tt}$-reduction, there exists a complete set for EXP that is not autoreducible. For NEXP, it is known that all many-one complete sets are autoreducible. Moreover, Glaßer

31st Symposium on Theoretical Aspects of Computer Science (STACS'14).
Editors: Ernst W. Mayr and Natacha Portier; pp. 590–601
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

et al. [6] took a next step to show that under 2-tt, disjunctive-truth-table, and conjunctive-truth-table reductions, all complete sets for NEXP are autoreducible. We make progress in this paper by proving non-autoreducibility of complete sets for NEXP under certain polynomial-time reductions. In particular, we obtain the following results. (All definitions to follow.)

- For any positive integers s and k such that $2^s - 1 > k$, there is a $\leq^p_{s\text{-}T}$-complete set for NEXP that is not $\leq^p_{k\text{-}tt}$-autoreducible.
- There is a $\leq^p_T$-complete set for NEXP that is not $\leq^p_{tt}$-autoreducible.
- There is a $\leq^p_{3\text{-}tt}$-complete set for NEXP that is not honest $\leq^p_{3\text{-}tt}$-autoreducible.
- For any positive integer k, there is a $\leq^p_{k\text{-}tt}$-complete set for NEXP that is not weakly $\leq^p_{k\text{-}tt}$-autoreducible.

Proofs typically require intricate diagonalization arguments.

This paper is organized as follows. Section 2 contains notation and definitions about many different polynomial-time reductions and autoreducibilities. In section 3, we obtain our non-autoreducibility results for many different complete sets in NEXP. Section 4 contains our result about hypothesis A. In section 5, we will show negative results in relativized worlds for some open questions.

2 Preliminaries

Most notation and definitions are standard [9]. Strings are elements of $\{0,1\}^*$. For every string x, denote $|x|$ to be the length of x. For every Turing machine M, $L(M)$ denotes the language accepted by the machine M. We denote M^B to be an oracle Turing machine M that accesses the oracle B. Also for every input x, $M(x)$ is the outcome of the computation of M on input x; i.e., $M(x) = 1$ if and only if M accepts input x. We assume that the pairing function $\langle \ldots \rangle$ is a one-to-one, polynomial-time computable function that can take any finite number of inputs and its range does not intersect with 0^*. For every set A, the characteristic function of A is denoted by A; that is, $A(x) = 1$ if $x \in A$ and $A(x) = 0$ otherwise. Also $|A|$ denotes the cardinality of A.

For any two sets A and B, A is *Turing-reducible* to B *in polynomial time*, $A \leq^p_T B$, if there exists a deterministic polynomial-time-bounded oracle Turing machine M such that $A = L(M^B)$. Similarly, $A \leq^p_{k\text{-}T} B$ if there exists a deterministic polynomial-time-bounded oracle Turing machine M such that $A = L(M^B)$ and M asks no more than k queries for any input x. In this paper, if we do not mention explicitly the running time of a reduction, then that reduction is a polynomial time reduction. The reduction is *nonadaptive*, $A \leq^p_{tt} B$, if the queries are independent of the oracle and so they do not depend on the answers to the previous queries. Other notions of reductions are also considered. A set A is *k-truth-table-reducible* to B, $A \leq^p_{k\text{-}tt} B$, if there exists a nonadaptive oracle Turing machine M^B that accepts A such that for any input x, the computation of M^B on input x asks no more than k queries. A set A is *bounded-truth-table-reducible* to B, $A \leq^p_{btt} B$, if there exists some integer k such that $A \leq^p_{k\text{-}tt} B$. A set A is *disjunctive-truth-table reducible* to B in polynomial time, $A \leq^p_{dtt} B$, if there exists a polynomial time computable function f such that for any x, $f(x) = \langle q_1, \ldots, q_k \rangle$, and $x \in A \iff B(q_1) \vee \cdots \vee B(q_k) = 1$. Similarly, a set A is *conjunctive-truth-table reducible* to B in polynomial time, $A \leq^p_{ctt} B$, if there exists a polynomial time computable function f such that for any x, $f(x) = \langle q_1, \ldots, q_k \rangle$, and $x \in A \iff B(q_1) \wedge \cdots \wedge B(q_k) = 1$. Other notions $\leq^p_{k\text{-}dtt}$ and $\leq^p_{k\text{-}ctt}$ are defined analogously. For any k-ary Boolean function α, a set A is *α-truth-table reducible* to B in polynomial time, $A \leq^p_{\alpha tt} B$, if there exists a polynomial time computable function f such that for any x, $f(x) = \langle q_1, \ldots, q_k \rangle$, and $x \in A \iff \alpha(B(q_1), \ldots, B(q_k)) = 1$.

$\text{EXP} = \bigcup\{\text{DTIME}(2^{p(n)}) \mid p \text{ is a polynomial}\}$ is the class of languages that can be decided by a deterministic Turing machine in exponential time.

$\text{NEXP} = \bigcup\{\text{NTIME}(2^{p(n)}) \mid p \text{ is a polynomial}\}$ is the class of languages that can be decided by a nondeterministic Turing machine in exponential time.

Throughout this paper, let $\{\text{NEXP}_i\}_{i \geq 1}$ be an enumeration of all nondeterministic exponential time Turing machines. Also we assume that the computation of NEXP_j on input x has running time that is bounded by $2^{|x|^j}$.

Let $K = \{\langle i, x, l \rangle \mid \text{NEXP}_i \text{ accepts input } x \text{ within } l \text{ steps}\}$ be a canonical complete set for NEXP, where l is encoded in a binary string.

For any oracle Turing machine M^B, let $Q(M^B, x)$ denote the set of all queries of the computation of M^B on input x.

▶ **Definition 1** (Autoreducibility). For any reduction $\leq$, a set A is $\leq$-autoreducible if $A \leq A$ via an oracle Turing machine M^A such that for any x, $x \notin Q(M^A, x)$. We call M an autoreduction of A by $\leq$-reduction. The reduction $\leq$ can apply to any reductions, specifically, all those that we mention above, such as $\leq^p_T, \leq^p_{tt}, \leq^p_{k\text{-}tt}, \leq^p_{dtt}$, etc.

Honest reductions are discussed in [8] and [5]. Informally, in honest reductions, the strings queried to the oracle cannot be too short compared to the input length. In this paper, we use a stronger notion of honest reductions, where strings queried cannot be either too short or too long compared to the input length. Its formal definition is as follows.

▶ **Definition 2** (Honest truth-table reduction). Given any two sets A and B and an arbitrary positive number $c \geq 1$, we define an honest truth-table reduction $\leq^{h\text{-}c}_{tt}$ as follows: $A \leq^{h\text{-}c}_{tt} B$ if there exists a nonadaptive Turing machine M with oracle B such that M^B accepts x if and only if $x \in A$ and for any input x, all queries q made to oracle B have length satisfying $|x|^{1/c} \leq |q| \leq |x|^c$.

▶ **Definition 3** (NOR-reduction). Given any two sets A and B, we define a NOR-truth-table reduction $\leq^p_{NOR\text{-}tt}$ as follows: $A \leq^p_{NOR\text{-}tt} B$ if there exists a nonadaptive Turing machine M with oracle B such that for any input x, letting $q_1, \ldots, q_k$ be all queries of M^B on input x, then $x \in A \iff q_1 \notin B \wedge \cdots \wedge q_k \notin B$.

▶ **Definition 4** (Weak-reduction). Given any two sets A and B, we define a weak truth-table reduction $\leq^p_{tt\text{-}w}$ as follows: $A \leq^p_{tt\text{-}w} B$ if and only if there exist two polynomial computable functions f and g such that for any input x, $f(x) = \langle q_1, \ldots, q_k \rangle$, $g(x) = h(\alpha_1, \ldots \alpha_k)$ is a Boolean function with k variables $\alpha_1, \ldots \alpha_k$ such that h is neither an OR nor a NOR Boolean function, and $x \in A \iff h(B(q_1), \ldots, B(q_k)) = 1$.

3 Non-autoreducible sets for NEXP

▶ **Theorem 5.** *For any positive integers s and k such that $2^s - 1 > k$, there is a $\leq^p_{s\text{-}T}$-complete set for* NEXP *that is not $\leq^p_{k\text{-}tt}$-autoreducible.*

Proof. Let $\{M_j\}_{j \geq 1}$ be an enumeration of all $\leq^p_{k\text{-}tt}$-reductions. Assume that M_j on input x runs in time $|x|^j$. We will construct a set B such that $K \leq^p_{s\text{-}T} B$ but B is not $\leq^p_{k\text{-}tt}$-autoreducible. Recall that K, which is defined in the Preliminaries section, is a canonical complete set for NEXP.

The $\leq^p_{s\text{-}T}$-reduction from K to B will be as follows: we build a full binary tree of height s. This tree has exactly $2^s - 1$ nodes. We number the nodes from top to bottom, left to right, by using numbers $0, 1, \ldots, 2^s - 2$; i.e. the root node will be numbered 0, then its two children will be 1 and 2, etc. Then for any string x, each node i will be labeled by the pair $\langle x, i\rangle$. From now on, for every such x, $\mathcal{T}(x)$ is such a query tree, and for every node $\mathcal{N}$, $\mathcal{N}$ is referred as a node itself or its label interchangeably. Also for any two nodes $\mathcal{N}_1$ and $\mathcal{N}_2$ such that one node is an ancestor of another node, denote $\mathcal{P}(\mathcal{N}_1, \mathcal{N}_2)$ to be a unique path from $\mathcal{N}_1$ to $\mathcal{N}_2$. For every node $\mathcal{N}$, denote the left path $\mathcal{L}(\mathcal{N})$ to be a path from $\mathcal{N}$ to a leaf node by just traversing left. The right path $\mathcal{R}(\mathcal{N})$ is defined similarly. Those labels are possible queries that can be asked to the oracle B by this reduction. Specifically, start at the root node, and if the current query is node $\mathcal{N}$, if the answer is YES, i.e. $\mathcal{N} \in B$, then the next query will be $\mathcal{N}$'s left child; otherwise the right child will be asked. The reduction accepts if and only if the last query (certainly, it is one of the leaf nodes) belongs to B. Define the sequence $\{y_n\}_{n\geq 0}$ such that $y_0 = 1$ and $y_{n+1} = 2^{y_n^n} + 1$ for every $n \geq 0$. Now we construct such a set B that satisfies the above reduction. At the same time, we want to diagonalize against all M_n^B such that M_n^B accepts 0^{y_n} if and only if $0^{y_n} \notin B$. The set B is constructed in each stage as follows. Initially we set $B = \emptyset$.

At stage n, suppose that the set B has been constructed such that all strings of length up to y_{n-1}^{n-1} have already been encoded into B appropriately to make the above reduction work. We will encode all strings of length between $y_{n-1}^{n-1} + 1$ and y_n^n into B in this stage.

Compute Q that is the set of all queries q of M_n on input 0^{y_n} such that $q = \langle x, i\rangle$, $i \leq 2^s - 1$, and $y_{n-1}^{n-1} + 1 \leq |x| \leq y_n^n$. Denote P to be the set of all x such that $\langle x, i\rangle \in Q$ for some $0 \leq i \leq 2^s - 1$. And for each $x \in P$, denote P^x to be the set of all $\langle x, i\rangle$ such that $\langle x, i\rangle \in Q$ and $0 \leq i \leq 2^s - 1$.

For each x in P, consider set P^x. Notice that $|P^x| \leq k < 2^s - 1$. Consider the query tree $\mathcal{T}(x)$:

- **Case 1:** If all leaf nodes are in P^x, then there are some internal nodes such that they are not in P_x. Let $\mathcal{N}$ be the smallest node in the set of those nodes. Put $\mathcal{N}$ into B if and only if $x \in K$. Also for every node $\mathcal{N}'$ in $\mathcal{L}(\mathcal{N})$ and $\mathcal{N}' \neq \mathcal{N}$, add $\mathcal{N}'$ to B. Finally for every node $\mathcal{N}'$ in the path $\mathcal{P}(Root, \mathcal{N})$, add $\mathcal{N}'$ to B if and only if its left child node is in the path.
- **Case 2:** If there are some leaf nodes that are not in P^x, let $\mathcal{N}$ be the smallest node in the set of those nodes. Add $\mathcal{N}$ to B if and only if $x \in K$. For every node $\mathcal{N}'$ in $\mathcal{P}(Root, \mathcal{N})$, add $\mathcal{N}'$ to B if and only if its left child is in that path.

For every $x \notin P$ such that $y_{n-1}^{n-1} + 1 \leq |x| \leq y_n^n$, put $\langle x, 2^s - 1\rangle$ into B if and only if $x \in K$.

After all those steps are done, put 0^{y_n} into B if and only if M_n^B rejects 0^{y_n}.

That is how B is constructed. It is straightforward to see that the construction satisfies two properties: $K \leq_{s\text{-}T}^p B$ and B is not $\leq_{k\text{-}tt}^p$-autoreducible.

▶ **Claim 6.** $B \in \mathrm{NEXP}$

Proof. Notice that all elements of B have one of two forms 0^* and $\langle x, i\rangle$ where $0 \leq i \leq 2^s - 1$. For any input of any other form, it just rejects immediately.

Given an input b, consider the following cases:

- $b = 0^{y_n}$ for some n (otherwise, $b \notin B$). Then by the construction,

$$0^{y_n} \in B \iff M_n^B \text{ rejects } 0^{y_n}.$$

So if we know how to resolve all queries made to oracle B then it is easy to determine whether M_n^B accepts 0^{y_n} in exponential time. Now notice that in the above construction, for every query q, it can be resolved by considering the query tree and it does not depend on the membership of some x in K. In this case membership in B can be answered deterministically in exponential time.

- $b = \langle x, i \rangle$ for some $0 \le i \le 2^s - 1$. By considering the query tree $\mathcal{T}(x)$, there are two cases:
  - The membership of b in B can be determined straightforwardly, based on the above construction, and does not depend on whether $x \in K$ or not.
  - $b \in B \iff x \in K$. In this case, we can simulate the machine to accept K on an input x. Notice that $|x| < |b|$, so it can be done nondeterministically in exponential time.

Thus, $B \in$ NEXP ◀

Hence, B is a $\le^p_{s\text{-}T}$-complete set for NEXP that is not $\le^p_{k\text{-}tt}$-autoreducible. ◀

Glaßer et al. [6] showed that every $\le^p_{2\text{-}tt}$-complete set for NEXP is $\le^p_{2\text{-}tt}$-autoreducible. Theorem 5 is somehow "tight" in case $s = 2$ and $k = 2$. The following corollary separates the notions of $\le^p_{2\text{-}T}$ and $\le^p_{2\text{-}tt}$.

▶ **Corollary 7.** *There is a $\le^p_{2\text{-}T}$-complete set for* NEXP *that is not $\le^p_{2\text{-}tt}$-complete.*

It has been known that there is a Turing complete set for EXP that is not $\le^p_{tt}$-autoreducible [4]. We want to remark that Buhrman et al. [3] showed that there is a set that is Turing complete but not $\le^p_{tt}$-complete for NEXP. Moreover, their construction technique can be adapted to show that for any positive integers s and k such that $2^{s-2} > k$, there is a $\le^p_{s\text{-}T}$-complete set for NEXP that is not $\le^p_{k\text{-}tt}$-autoreducible, which is weaker than what Theorem 5 states. By adding a minor trick to the proof in Theorem 5 or cleverly adapting the technique in [3], we can separate the Turing-completeness notion from the $\le^p_{tt}$-autoreducibility notion in NEXP, as opposed to Turing-completeness versus $\le^p_{tt}$-completeness in [3].

▶ **Corollary 8.** *There is a Turing complete set for* NEXP *that is not $\le^p_{tt}$-autoreducible.*

Proof. Notice that in this case, the M_n autoreduction will not ask just k queries on input 0^{y_n}, but it can ask up to y_n^n queries, because its running time on input 0^{y_n} is bounded by y_n^n. Another modification is that the reduction from K to B will now need to ask more queries, say $|x|^2$ adaptive queries; that also means the query tree will have height $|x|^2$. With this trick in mind, in the construction algorithm of B at stage n, for every x in P, $|P^x| \le y_n^n = (2^{y_{n-1}^{n-1}} + 1)^n < 2^{y_{n-1}^{2(n-1)}} < 2^{|x|^2}$. So the number of nodes in the query tree $\mathcal{T}(x)$ will be bigger than the number of queries of M_n on input 0^{y_n}. In cases 1 and 2 the construction will work similarly. ◀

Now we consider the more difficult question of whether every $\le^p_{3\text{-}tt}$-complete set for NEXP is $\le^p_{3\text{-}tt}$-autoreducible. Notice that the above technique cannot be used, because the number of options to encode every x in K into B is no more than the number of queries of M_n^B on input 0^{y_n}; both are equal 3 in this case. This difficulty arises because we have no "room" for the encoding and diagonalization at the same time. We need to use a different technique to resolve that issue.

▶ **Theorem 9.** *For any number c, there is a $\le^p_{2\text{-}T}$-complete set for* NEXP *that is not $\le^{h\text{-}c}_{3\text{-}tt}$-autoreducible.*

Proof. Let $\{M_i\}_{i \ge 1}$ be a standard enumeration of all $\le^{h\text{-}c}_{3\text{-}tt}$-autoreductions clocked such that M_i runs in time n^i. We will construct a $\le^p_{2\text{-}T}$-complete set B for NEXP incrementally in each stage and diagonalize against all autoreductions M_i. We define the sequence $\{y_n\}_{n \ge 1}$ recursively as follows: $y_1 = 1$ and $y_{n+1} = \max(y_n^n, y_n^{c^2}) + 1$ for all $n \ge 1$.

In each stage, we construct B such that the following procedure is the $\leq^p_{2\text{-}T}$-reduction that reduces K to B. Given any input x, ask a query 0^m to oracle B, where m is a number that is bounded by some polynomial in $|x|$. If the answer is YES, then accept x if and only if $\langle 0, x\rangle \in B$. If the answer is NO, then accept x if and only if $\langle 1, x\rangle \in B$. Obviously if B satisfies this condition, then B is $\leq^p_{2\text{-}T}$-hard for NEXP.

The detail of how B is constructed will be as follows.

- Initially $B = \emptyset$.
- Suppose at stage n, the set B is constructed up to length $y_n - 1$. At this stage, we will add appropriate strings of length between y_n and $y_{n+1} - 1$ to accomplish two things: encoding K into B and diagonalize, using the string $0^{y^c_n}$, against the autoreduction M_n that asks no more than 3 queries. Therefore, in the following steps, if M_n asks more than 3 queries, then the diagonalization task will be skipped to the next stage.

Consider the following case where queries of M^B_n on input $0^{y^c_n}$ are $\langle 0, q_1\rangle$, $\langle 1, q_2\rangle$, and $\langle 1, q_3\rangle$ and the Boolean truth-table function is $f(a, b_1, b_2)$. In other words, M^B_n accepts $0^{y^c_n}$ if and only if $f(B(\langle 0, q_1\rangle), B(\langle 1, q_2\rangle), B(\langle 1, q_3\rangle)) = 1$. (Lack of space does not permit a complete proof of this Theorem.)

▶ **Lemma 10.** *For any Boolean function $f(a, b_1, b_2)$, at least one of the following statements must be true:*

- *There exist two Boolean functions $g_1(a)$ and $g_2(a)$, where $g_1(a)$ and $g_2(a)$ are one of $a, 0$, or 1, such that $f(a, g_1(a), g_2(a)) = 0$ for every a.*
- *There exists a Boolean function $h(b_1, b_2)$, where $h(b_1, b_2)$ is one of $0, 1, b_1, b_2, b_1 \wedge b_2$, or $b_1 \vee b_2$, such that $f(h(b_1, b_2), b_1, b_2) = 1$ for every b_1 and b_2.*

Suppose that we have $f(b_1 \vee b_2, b_1, b_2) = 1$, for every b_1 and b_2 (in this case, we are considering Statement 2 in the above lemma). Then if we set $B(\langle 1, q_2\rangle) = 1$ if $q_2 \in K$, $B(\langle 1, q_3\rangle) = 1$ if $q_3 \in K$, and $B(\langle 0, q_1\rangle) = B(\langle 1, q_2\rangle) \vee B(\langle 1, q_3\rangle)$. Also $B(0^{y^c_n}) = 0$. It is easy to verify that $0^{y^c_n} \notin B$ and M^B_n accepts $0^{y^c_n}$. So the diagonalization can be achieved by this fact.

Moreover by this setting, the reduction $K \leq^p_{2\text{-}T} B$ can be obtained correctly too. Notice that $0^{y^c_n} \notin B$. Thus, $q_2 \in K$ if and only if $\langle 1, q_2\rangle \in B$. Similarly for $\langle 1, q_3\rangle$. This fact is correctly reflected in the above setting.

Last but not least, we need B to be in NEXP. Consider whether $\langle 1, q_2\rangle \in B$. Notice that it is equivalent to the question whether $q_2 \in K$, which can be solved nondeterministically in exponential time. A more difficult question is whether $\langle 0, q_1\rangle$ is in B. By B's construction, $B(\langle 0, q_1\rangle) = B(\langle 1, q_2\rangle) \vee B(\langle 1, q_3\rangle)$. By this fact, $\langle 0, q_1\rangle$ is in B if one of the two strings q_2 and q_3 is in B. This condition can also be solved nondeterministically in exponential time. In summary, B is in NEXP.

By our construction we obtain three properties: $\leq^p_{2\text{-}T}$-hardness of B, B is in NEXP, and B is not autoreducible. That is, B is the set that we want to construct to prove this theorem. ◀

We note that Lemma 10 cannot be generalized to a Boolean function of 4 variables a_1, a_2, b_1, b_2 or more because we found a counterexample in that case. We obtained the counterexample by writing a program to list all possible Boolean functions of 4 variables, and then for each function checking whether it satisfies the two statements in Lemma 10. So the proof of Theorem 9 cannot be generalized to work with $\leq^p_{k\text{-}tt}$-reductions for $k \geq 4$. Nevertheless, the following theorem will show non-autoreducibility for $\leq^p_{k\text{-}tt}$-reductions if we reduce the power of the $\leq^p_{k\text{-}tt}$-autoreduction by not allowing the truth-table function to be an OR or a NOR.

▶ **Theorem 11.** *For any positive integer k, there is a $\leq^p_{k\text{-}tt}$-complete set for* NEXP(EXP) *that is not weakly $\leq^p_{k\text{-}tt\text{-}w}$-autoreducible.*

Proof. Let $\{M_j\}_{j\geq 1}$ be an enumeration of polynomial-time weak $\leq^p_{k\text{-}tt}$-autoreductions. For each $j \geq 1$, assume that M_j on input x runs in time $|x|^j$. Denote $\alpha_1, \ldots, \alpha_k$ to be the lexicographically first k strings of length $\lceil \log k \rceil$. We will construct a set B with the following property: $x \in K \iff$ there exists a j, $1 \leq j \leq k$, and $\langle \alpha_j, x\rangle \in B$, which ensures that $K \leq^p_{k\text{-}tt} B$, and then B is $\leq^p_{k\text{-}tt}$-hard for NEXP. We also need B so that for any $n \geq 1$, the following property holds: $0^{y_n} \in B \iff M_n^B$ rejects input 0^{y_n}, which ensures that M_n is not an autoreduction of B. (The value of y_n will be chosen later in the proof) Then we can conclude that B is not autoreducible.

We construct B in stages. In each stage, we will encode K into B and diagonalize against all weak $\leq^p_{k\text{-}tt}$-reductions using the string 0^{y_n} simultaneously to obtain those above two properties.

Before going into detail of how B is constructed, we define the sequence $\{y_n\}_{n\geq 0}$ such that $y_0 = 1$ and $y_{n+1} = 2^{y_n^n} + 1$ for every $n \geq 0$. B is constructed in each stage as follows.

Initially we set $B = \emptyset$. At stage n, suppose that B is already constructed up to strings of length y_{n-1}^{n-1}. We will encode appropriately all strings of length between $y_{n-1}^{n-1} + 1$ and y_n^n into B.

Let Q be the set of all queries q of M_n on input 0^{y_n} such that $|q| > y_{n-1}^{n-1}$. Let $P = \{x \mid$ there exists a $1 \leq j \leq k$ such that $\langle \alpha_j, x\rangle \in Q\}$. For every $x \in P$, denote $P^x = \{\langle \alpha_j, x\rangle \mid \langle \alpha_j, x\rangle \in Q\}$.

Now we consider the following cases:

- If $|P^x| < k$ for all x, then for every $x \in P$, denote t to be the smallest number such that $\langle \alpha_t, x\rangle \notin Q$. Put $\langle \alpha_t, x\rangle$ into B if and only if $x \in K$. Also for every $x \notin P$, put $\langle \alpha_1, x\rangle$ into B if and only if $x \in K$. Finally, put 0^{y_n} into B if and only if M_n^B rejects 0^{y_n}.
- If $|P^x| = k$ for some x, consider the Boolean truth-table function g of M_n on input 0^{y_n}, we have two following cases:
  - If $g(0, 0, \ldots, 0) = 0$, then let $c_1, \ldots, c_k$ be the lexicographically smallest non-zero value such that $g(c_1, \ldots, c_k) = 0$. For every c_i such that $c_i = 1$, put $\langle \alpha_i, x\rangle$ into B if and only if $x \in K$. Also for every $x \notin P$, put $\langle \alpha_1, x\rangle$ into B if and only if $x \in K$. Finally put 0^{y_n} into B.
  - If $g(0, 0, \ldots, 0) = 1$, then let $c_1, \ldots, c_k$ be the lexicographically smallest non-zero value such that $g(c_1, \ldots, c_k) = 1$. For every c_i such that $c_i = 1$, put $\langle \alpha_i, x\rangle$ into B if and only if $x \in K$. Also for every $x \notin P$, put $\langle \alpha_1, x\rangle$ into B if and only if $x \in K$.

This concludes the construction of B. The following lemma claims the time complexity of B.

▶ **Lemma 12.** $B \in$ NEXP.

Proof. Given an input b, one of the following cases can happen:

- **Case 1:** If b has the form 0^*: if $|b| \neq y_n$ for all n then reject. Otherwise, compute the set Q of all queries when running a Turing machine M_n^B on input 0^{y_n}. Notations of P and P^x are defined similarly to B's construction above.
  - If $|P^x| < k$ for all x, then simulate the Turing machine M_n^B on input 0^{y_n}. Whenever a query q is asked, the answer from oracle B will be resolved as follows:
    - If $|q| > y_{n-1}^{n-1}$, then answer NO.
    - Otherwise, check whether $q \in B$ recursively.

- If $|P^x| = k$ for some x. Then let g be a Boolean truth-table function of M_n^B on input 0^{y_n}.
  - If $g(0, \dots, 0) = 0$ then accept.
  - Otherwise, reject.
- **Case 2:** $b = \langle \alpha_i, x \rangle$ for some α_i (if $b \neq \langle \alpha_j, x \rangle$ for all j then just reject)
  Compute the number n such that $y_{n-1}^{n-1} < |b| \leq y_n^n$.
  Consider sets Q, P, and P^x as above when running M_n^B on input 0^{y_n}. We have the following cases:
  - If $|P^y| < k$ for every $y \in P$: If $b \in Q$ then reject. Otherwise, accept if and only if $i = 1$ and $x \in K$.
  - If $|P^y| = k$ for some $y \in P$: If $x \neq y$ then accept if and only if $i = 1$ and $x \in K$. Otherwise, if $x = y$, let g be a Boolean truth-table function of M_n on input 0^{y_n}. Consider the two following cases:
    - If $g(0, \dots, 0) = 0$. Let $c_1, \dots, c_k$ be the lexicographically smallest non-zero value such that $g(c_1, \dots, c_k) = 0$. Accept if and only if $c_i = 1$ and $x \in K$.
    - If $g(0, \dots, 0) = 1$. Let $c_1, \dots, c_k$ be the lexicographically smallest non-zero value such that $g(c_1, \dots, c_k) = 1$. Accept if and only if $c_i = 1$ and $x \in K$.

Now we will analyze the running time of the above tasks. The most expensive tasks will be described as follows:
- The number n can be determined in polynomial time in terms of length of input b.
- The query set Q and the truth-table function g can be computed in time y_n^n, which is no more than $O(2^{|b|^2})$.
- In case 1, to recursively check whether the query q of length smaller than y_{n-1}^{n-1} belongs to B or not deterministically takes time $2^{2^{y_{n-1}^{n-1}}}$, which is no more than $2^{y_n} = 2^{|b|}$ (Recall that in this case, $b = 0^{y_n}$).
- Determining whether x belongs to K can be done nondeterministically in $2^{|x|} < 2^{|b|}$.

We conclude that $B \in$ NEXP. ◀

▶ **Lemma 13.** $K \leq_{k\text{-}tt}^p B$.

Proof. In B's construction, for every x that is in K, we encode at least one of the following strings $\langle \alpha_1, x \rangle \dots \langle \alpha_k, x \rangle$ into B. Strings that do not belong to K are not encoded into B. It follows that $K \leq_{k\text{-}tt}^p B$. ◀

It is not hard to see that B is not weakly $\leq_{k\text{-}tt}^p$-autoreducible, so by Lemma 12 and Lemma 13, B is a $\leq_{k\text{-}tt}^p$-complete set for NEXP that is not weakly $\leq_{k\text{-}tt}^p$-autoreducible. ◀

The above proof also yields the following corollary:

▶ **Corollary 14.** *For any positive integer k, there is a $\leq_{k\text{-}dtt}^p$-complete set for* NEXP(EXP) *that is not weakly $\leq_{k\text{-}tt}^p$-autoreducible.*

4 Implications

We begin with the following theorem.

▶ **Theorem 15.** *Every $\leq_{dtt}^p$-complete set for* EXP *is $\leq_{NOR\text{-}tt}^p$-autoreducible.*

Glaßer et al. [6] also showed that every $\leq_{dtt}^p$-complete set for EXP is $\leq_{dtt}^p$-autoreducible. Then by Theorem 15, Corollary 14 is somehow “tight” for EXP.

Algorithm 1 Algorithm to decide B. Input is of the form $\langle 0^i, x\rangle$.

```
Q := Q(M_i, ⟨0^i, x⟩) // Set of all queries of M_i on input ⟨0^i, x⟩
If (x ∉ Q) Then
   If (x ∉ A) Then
       Accept
   Else
       Reject
   EndIf
Else
   Reject
EndIf
```

Algorithm 2 Autoreduction algorithm for A. Input string is x.

```
Q := {q_1, ..., q_k} := Q(M_j, ⟨0^j, x⟩)
If x ∉ Q Then
   If ((q_1 ∉ A)&&(q_2 ∉ A)&&...&&(q_k ∉ A)) Then
      Accept
   Else
      Reject
   EndIf
Else
   Reject
EndIf
```

Proof. Let A be a $\leq^p_{dtt}$-complete set for EXP. We will show that A is also $\leq^p_{NOR\text{-}tt}$-autoreducible.

Let $\{M_i\}_{i\geq 1}$ be a standard enumeration of all $\leq^p_{dtt}$-reductions such that M_i runs in time $p_i(n) = n^i$ on inputs of size n.

Consider a set B containing elements of the form $\langle 0^i, x\rangle$ that are decided by Algorithm 1. Obviously $B \in$ EXP.

Since A is the $\leq^p_{dtt}$-complete set for EXP, $B \leq^p_{dtt} A$ by some disjunctive truth-table reduction M_j. For any x, if x is one of queries of M_j on input $\langle 0^j, x\rangle$, then $\langle 0^j, x\rangle \notin B$. This fact implies that for all queries q, including x, $q \notin A$. Then $x \notin A$. If x is not one of the queries $q_1, \ldots, q_k$ of M_j on input $\langle 0^j, x\rangle$, then $x \in A \iff \langle 0^j, x\rangle \notin B \iff q_i \notin A$ for all i.

Based on that observation, we have the autoreduction algorithm for A described in Algorithm 2.

Observe that this is a $\leq^p_{NOR\text{-}tt}$-autoreduction. Thus A is $\leq^p_{NOR\text{-}tt}$-autoreducible. ◀

Recall that every $\leq^p_{k\text{-}dtt}$-complete set for NEXP is $\leq^p_{k\text{-}dtt}$-autoreducible [6]. Also every $\leq^p_{k\text{-}dtt}$-complete set for EXP is both $\leq^p_{k\text{-}dtt}$-autoreducible [6] and $\leq^p_{NOR\text{-}k\text{-}tt}$-autoreducible. We want to know whether the same holds for NEXP; that is, whether every $\leq^p_{k\text{-}dtt}$-complete set for NEXP is also $\leq^p_{NOR\text{-}k\text{-}tt}$-autoreducible. Settling this question would lead to important complexity class results.

▶ **Theorem 16.** *For any positive integer k, every $\leq^p_{k\text{-}dtt}$-complete set for* NEXP *is $\leq^p_{NOR\text{-}k\text{-}tt}$-autoreducible if and only if* NEXP = coNEXP.

Proof. Suppose every $\leq^p_{k\text{-}dtt}$-complete set for NEXP is $\leq^p_{NOR\text{-}k\text{-}tt}$-autoreducible. Notice that K, the canonical complete set of NEXP, is also $\leq^p_{k\text{-}dtt}$-complete. By the assumption, K is $\leq^p_{NOR\text{-}k\text{-}tt}$-autoreducible.

Algorithm 3 NOR-Autoreduction algorithm for A. Input string is x.

```
⟨q_1, ..., q_k⟩ ← f(x)
For i := 1 to k do
   If (x = q_i) Then
      Reject and Terminate
   EndIf
EndFor
If ((q_1 ∉ A)&&...&&(q_k ∉ A)) Then
   Accept
Else
   Reject
EndIf
```

Let f be the autoreduction of K. That is, for every x, $f(x) = \langle q_1, \ldots, q_k \rangle$, $x \neq q_i$ for all i, and $x \in K \iff q_1 \notin K \wedge \cdots \wedge q_k \notin K$. We have the following fact:

$$x \in \overline{K} \iff x \notin K \iff q_1 \in K \vee \cdots \vee q_k \in K.$$

So $\bar{K} \leq^p_{k\text{-}dtt} K$. Because $K \in$ NEXP, we have $\overline{K} \in$ NEXP. Therefore, NEXP = coNEXP.

To prove the other direction, suppose NEXP = coNEXP. Let A be any $\leq^p_{k\text{-}dtt}$-complete set for NEXP. We show that A is also $\leq^p_{NOR\text{-}k\text{-}tt}$-autoreducible. Note that $\overline{A} \in$ NEXP. Hence, $\overline{A} \leq^p_{k\text{-}dtt} A$ by some polynomial-time function f. In other words, for any x, $f(x) = \langle q_1, \ldots, q_k \rangle$ and $x \in \overline{A} \iff q_1 \in A \vee q_2 \in A \vee \cdots \vee q_k \in A$.

Rewriting this, we have $x \in A \iff q_1 \notin A \wedge q_2 \notin A \wedge \cdots \wedge q_k \notin A$. Observe that if there is some i, $i = 1, \ldots, k$ such that $q_i = x$ then $x \notin A$. Because otherwise, it contradicts to the preceding fact. Based on these observations, we have the $\leq^p_{NOR\text{-}k\text{-}tt}$-autoreduction for A described in Algorithm 3. Hence, A is $\leq^p_{NOR\text{-}k\text{-}tt}$-autoreducible. ◀

▶ **Corollary 17.** *For any positive integer k, if there is a $\leq^p_{k\text{-}dtt}$-complete set for* NEXP *that is not $\leq^p_{NOR\text{-}k\text{-}tt}$-autoreducible, then* NEXP $\neq$ EXP.

Proof. The proof follows directly from either Theorem 15 or Theorem 16. ◀

In the following section, we will show a partial result about NOR-autoreducibility for a $\leq^p_{dtt}$-complete set for NEXP in the relativized world.

5 Relativization

While the question whether every $\leq^p_{dtt}$-complete set for NEXP is $\leq^p_{NOR\text{-}tt}$-autoreducible is still open, we can prove that it does not hold in a relativized world.

▶ **Theorem 18.** *Relative to some oracle B, there is a $\leq^{p^B}_m$-complete set for* NEXP^B *that is not $\leq^{p^B}_{NOR\text{-}tt}$-autoreducible.*

Proof. Let $\{M^B_j\}_{j\geq 1}$ be an enumeration of polynomial-time $\leq^{p^B}_{NOR\text{-}tt}$-autoreductions. Notice that M^B_j can now access oracle B.

Let $\{\text{NEXP}^B_i\}_{i\geq 1}$ be an enumeration of all nondeterministic exponential time oracle Turing machines. For each $j \geq 1$, suppose that n^j bounds the running time of M^B_j and 2^{n^j} bounds the running time of NEXP^B_j. Let $K^B = \{\langle i, x, l\rangle \mid \text{NEXP}^B_i$ accepts input x within l steps$\}$, where l is encoded in a binary string, be a canonical complete set for NEXP^B.

We will construct sets A and B with the property $x \in K^B \iff \langle 0, x\rangle \in A$, which ensures that $K^B \leq_m^p A$, and then A is $\leq_m^p$-hard for NEXP^B. We also need A and B so that for any $n \geq 1$, the following property holds: $0^{y_n} \in A \iff M_n^{B,A}$ rejects input 0^{y_n} (the value of y_n will be chosen later in the proof). These properties guarantee that M_n^B is not an autoreduction of A. Then we can conclude that A is not autoreducible for NEXP^B.

We construct A and B together in stages. In each stage, we encode K^B into A and diagonalize against all $\leq_{NOR\text{-}tt}^{p^B}$-reductions using the string 0^{y_n} simultaneously to obtain those above two properties.

We define the sequence $\{y_n\}_{n\geq 0}$ such that $y_0 = 1$ and $y_{n+1} = y_n^n + 1$ for every $n \geq 0$.

Suppose at stage n that the set A has already been constructed up to length $y_n - 1$. At this stage, we will construct A for strings of length between y_n and $y_{n+1} - 1$. Now consider all queries q of M_n^B on input 0^{y_n} made to oracle A when $|q| \geq y_n$ and $q = \langle 0, x\rangle$ for some j. Let Q be the set of all such queries q.

Consider the following cases:

1. If there is a query q' such that $|q'| < y_n$ and $q' \in A$. Then put 0^{y_n} into A and $\langle 0^{2^{y_n}}, 0^{y_n}\rangle$ into B. Finally, for all strings $s = \langle 0, x\rangle$ and $y_n \leq |s| < y_{n+1}$, put s into A if and only if $x \in K^B$.
2. Otherwise, ignore all queries of length smaller than y_n. For every $q' = \langle 0, x\rangle \in Q$ such that $x \in K^B$, choose any accepting path of K^B on input x and denote $Q^{q'}$ to be the set of all queries made in that path. Consider the following cases:
   a. If no such q' exists, then for all strings $s = \langle 0, x\rangle$ and $y_n \leq |s| < y_{n+1}$, put s into A if and only if $x \in K^B$.
   b. Otherwise, let P be the union of $Q^{q'}$ for all such q'. Notice that there are no more than y_n^n such q', and for every q', $|Q^{q'}| \leq 2^{|x|} < 2^{y_n^n}$. Then, $|P| < y_n^n 2^{y_n^n} < 2^{2^{y_n}}$. Therefore, there exists a string t of length 2^{y_n} such that $t \notin P$. Put $\langle t, 0^{y_n}\rangle$ into B. Put 0^{y_n} into A. Finally, for all strings of $s = \langle 0, x\rangle$ and $y_n \leq |s| < y_{n+1}$, put s into A if and only if $x \in K^B$.

This concludes the construction of sets A and B.

Now we will briefly show that A belongs to NEXP^B. To determine membership of an input 0^y, we just need to guess one string t of length 2^y and ask one query $\langle t, 0^y\rangle$ to oracle B; accept if and only if the answer is YES. For other input of the form $\langle 0, x\rangle$, accept if and only if $x \in K^B$. So $A \in \mathrm{NEXP}^B$.

To see that A is not reduced to itself by any $\leq_{NOR\text{-}tt}^{p^B}$-autoreduction, we will show that for any M_n, $M_n^{A,B}$ accepts 0^{y_n} if and only if $0^{y_n} \notin A$. In case (1), because there is one query q' such that $q' \in A$, by the NOR-tt reduction, $M_n^{A,B}$ rejects 0^{y_n}. Notice that putting $\langle 0^{2^{y_n}}, 0^{y_n}\rangle$ into B does not affect the membership of q' in A. In case (2a), $M_n^{A,B}$ accepts 0^{y_n} and in this case 0^{y_n} is not put into A, and then it makes M_n^B not reduce A to itself. In case (2b), $M_n^{A,B}$ does not accept 0^{y_n} and notice that putting $\langle t, 0^{y_n}\rangle$ into B does not affect the memberships of all q' in K^B. And finally 0^{y_n} is added to A to make M_n^B not reduce A to itself.

It is easy to see that $K^B \leq_m^p A$ because we encode all strings $x \in K^B$ by $\langle 0, x\rangle$ into A and nothing else, except the strings of form 0^*. Hence, A is the many-one complete set for NEXP^B that is not $\leq_{NOR\text{-}tt}^{p^B}$-autoreducible. ◀

We note that Theorem 16 actually relativizes. So we have the following familiar corollary:

▶ **Corollary 19.** *There is a set B such that relative to the oracle B, $\mathrm{NEXP}^B \neq \mathrm{coNEXP}^B$.*

Buhrman et al. [2] showed that relative to some oracle, there is a $\leq^p_{2\text{-}T}$-complete set for EXP that is not Turing autoreducible. Their technique also works for NEXP. I.e., we have the following theorem:

▶ **Theorem 20.** *Relative to some oracle, there is a $\leq^p_{2\text{-}T}$-complete set for* NEXP *that is not Turing autoreducible.*

6 Open Questions

We know for any positive integers s and k such that $2^s - 1 > k$ that there is a $\leq^p_{s\text{-}T}$-complete set for NEXP that is not $\leq^p_{k\text{-}tt}$-autoreducible. We do not know what happens when $2^s - 1 \leq k$. It is not known whether every Turing-complete set is Turing-autoreducible. Referring to Theorem 9, the situation for $\leq^p_{k\text{-}tt}$-reductions for $k \geq 4$ is still open.

Acknowledgements. Our thanks to Nils Wisiol and Benedikt Budig for useful discussions and feedback on drafts of this paper. Special thanks to Leen Torenvliet for his extensive comments and corrections of an earlier version.

References

1 K. Ambos-Spies. P-mitotic sets. In E. Börger, G. Hasenjäger, and D. Roding, editors, *Logic and Machines, Lecture Notes in Computer Science 177*, pages 1–23. Springer-Verlag, 1984.
2 H. Buhrman, L. Fortnow, D. van Melkebeek, and L. Torenvliet. Using autoreducibility to separate complexity classes. *SIAM Journal on Computing*, 29(5):1497–1520, 2000.
3 H. Buhrman, S. Homer, and L. Torenvliet. Completeness for nondeterministic complexity classes. *Mathematical Systems Theory*, 24(3):179–200, 1991.
4 H. Buhrman and L. Torenvliet. On the structure of complete sets. In *IEEE Structure in Complexity Theory Conference, 1994., Proceedings of the Ninth Annual*, pages 118–133, 1994.
5 R. Downey, S. Homer, W. Gasarch, and M. Moses. On honest polynomial reductions, relativizations, and P=NP. In *IEEE Structure in Complexity Theory Conference*, pages 196–207. IEEE Computer Society, 1989.
6 C. Glaßer, D. Nguyen, C. Reitwießner, A. Selman, and M. Witek. Autoreducibility of complete sets for log-space and polynomial-time reductions. In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *ICALP (1)*, volume 7965 of *Lecture Notes in Computer Science*, pages 473–484. Springer, 2013.
7 C. Glaßer, M. Ogihara, A. Pavan, A. Selman, and L. Zhang. Autoreducibility, mitoticity, and immunity. *J. Comput. Syst. Sci.*, 73(5):735–754, 2007.
8 S. Homer. Minimal degrees for polynomial reducibilities. *J. ACM*, 34(2):480–491, 1987.
9 R. Ladner, N. Lynch, and A. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science*, 1:103–123, 1975.
10 J. Feigenbaum R. Beigel. On being incoherent without being very hard. *Computational Complexity*, 2:1–17, 1992.
11 B. Trahtenbrot. On autoreducibility. *Dokl. Akad. Nauk SSSR*, 192, 1970. Translation in Soviet Math. Dokl. 11: 814– 817, 1970.