Nontriviality for Exponential Time w.r.t. Weak Reducibilities

Klaus Ambos-Spies

University of Heidelberg, Institut für Informatik, Im Neuenheimer Feld 294, D-69120 Heidelberg, Germany.

Timur Bakibayev

Al-Farabi Kazakh National University, 71 Al-Farabi ave., Almaty 050038, Kazakhstan.

Abstract

A set A is nontrivial for the linear exponential time class $\mathrm{E} = \mathrm{DTIME}(2^{lin})$ if $A \in \mathrm{E}$ and the sets from E which can be reduced to A are not from a single level $\mathrm{DTIME}(2^{kn})$ of the linear exponential hierarchy. Similarly, a set A is nontrivial for the polynomial exponential time class $\mathrm{EXP} = \mathrm{DTIME}(2^{poly})$ if $A \in \mathrm{EXP}$ and the sets from EXP which can be reduced to A are not from a single level $\mathrm{DTIME}(2^{n^k})$ of the polynomial exponential hierarchy (see [2]). Here we compare the strength of the nontriviality notions with respect to the underlying reducibilities where we consider the polynomial-time variants of many-one, bounded truth-table, truth-table, and Turing reducibilities. Surprisingly, the results obtained for E and EXP differ. While the above reducibilities yield a proper hierarchy of nontriviality notions for E, nontriviality for EXP under many-one reducibility and truth-table reducibility coincides.

1. Introduction

A set A is *nontrivial* for $\mathrm{E} = \mathrm{DTIME}(2^{lin})$ (or E-*nontrivial* for short) if there are arbitrarily complex sets from E which can be reduced to A, i.e., if for any $k \geq 1$ there is a set $B \in \mathrm{E}$ reducible to A which is 2^{kn}-complex (i.e., $B \notin \mathrm{DTIME}(2^{kn})$). Similarly, a set A is *nontrivial* for $\mathrm{EXP} = \mathrm{DTIME}(2^{poly})$ if for any $k \geq 1$ there is a set B in $\mathrm{EXP} \setminus \mathrm{DTIME}(2^{n^k})$ which can be reduced to A. Nontriviality, which was introduced by the authors in [2], was inspired by Lutz's concept of weak completeness. While a set $A \in \mathrm{C}$ is complete for a complexity class C in the classical sense if *all* problems in C can be reduced to A, Lutz [11] proposed to call a set $A \in \mathrm{C}$ *weakly complete* for C if a *nonnegligible part* of problems in C can be reduced to A. Lutz formalized the idea of weak completeness for the exponential time classes E and EXP by introducing some resource bounded (pseudo) measures on these classes and by calling a subclass of E and EXP negligible if it has measure 0 in E and EXP, respectively. As one can easily show, weakly complete sets for E and EXP in the sense of Lutz [11] are E-nontrivial and EXP-nontrivial, respectively, and in [2] it is argued that E-nontriviality and EXP-nontriviality are the *weakest* meaningful weak completeness notions for the corresponding exponential time classes.

While weak completeness generalizes completeness by relaxing the requirement that all sets from the considered class $\mathcal{C}$ can be reduced, the classical approach for generalizing completeness is to relax (i.e., to weaken) the underlying reducibility. So one might replace the polynomial time bounded many-one reducibility (p-m-reducibility for short) on which completeness (as well as weak completeness and nontriviality) is usually based by more general polynomial-time reducibilities like the polynomial time variants of bounded truth-table reducibility (p-btt) or truth-table reducibility (p-tt) or Turing reducibility (p-T). As Watanabe [12] has shown, these more general reducibilities also yield more general completeness notions for the exponential time classes. For Lutz's weak completeness notions for E and EXP the corresponding separations have been obtained by Ambos-Spies, Mayordomo and Zheng [4]. Moreover, there it has been shown that there are no trade-offs between the two types of generalizations of completeness, i.e., completeness under a weaker reducibility does not imply weak completeness under a stronger reducibility and vice versa.

Here we generalize these results in the literature by addressing the corresponding questions for nontriviality (instead of weak completeness) where we also consider the question of possible trade-offs: If arbitrarily complex sets from E - or even all sets from E - can be reduced to a set $A \in$ E by some weaker reducibility, can we also reduce arbitrarily complex sets from E to A by some stronger reducibility (and, similarly, for EXP)?

For the investigation of these questions, the following phenomenon has to be taken into account. While, by a simple padding argument, hardness for E and EXP coincide (whence a set $A \in$ E is E-complete if and only if it is EXP-complete), surprisingly, for Lutz's weak completeness only one direction holds. Namely any weakly E-complete set is weakly EXP-complete but there are sets in E which are weakly EXP-complete but not weakly E-complete (see Juedes and Lutz [9]). For the still weaker nontriviality notions, E-nontriviality and EXP-nontriviality are in fact independent (see Ambos-Spies and Bakibayev [3]), i.e., there are sets in E which are E-nontrivial but not EXP-nontrivial and vice versa.

This difference in the nontriviality notions for E and EXP is also manifested in a quite surprising way in our results here. While for E the hierarchy of the nontriviality notions under the weak polynomial time reducibilities completely mirrors Watanabe's separation results for the corresponding completeness notions, for EXP nontriviality under truth-table reducibility and nontriviality under many-one reducibility coincide.

The outline of the paper is as follows. In Section 2 we show that, for E and EXP, nontriviality under truth-table reducibility is stronger than nontriviality under Turing reducibility. In fact we show that there is a T-complete set A for E which is neither tt-nontrivial for E nor tt-nontrivial for EXP. So the fact that *all* sets in E can be recovered from a set A by a Turing reduction does not imply that there are arbitrarily complex sets in E which can be recovered from A by some truth-table reductions. In Section 4 we give corresponding separations of many-one, bounded truth-table and truth-table reducibilities for E whereas in Section 3 we prove the coincidence of EXP-nontriviality under many-one reducibility with EXP-nontriviality under truth-table reducibility.

We assume familiarity with the basic notions of structural complexity theory (see e.g. Balcázar et al. [5] and [6] for unexplained notation). All reducibilities considered here are polynomial-time bounded. For a survey of the polynomial-time reducibilities see Ladner, Lynch and Selman [10].

In the following a set A is a set of binary strings, i.e., $A \subseteq \{0,1\}^*$, and we write $A(x) = 1$ if $x \in A$ and $A(x) = 0$ if $x \notin A$. For a binary string x we let $x(i)$ denote the $(i+1)$th bit of x, i.e., $x = x(0)x(1)\dots x(n-1)$ where n is the length of x. For the exponential time classes we use the following abbreviations: $\mathrm{E}_k = \mathrm{DTIME}(2^{kn})$ and $\mathrm{EXP}_k = \mathrm{DTIME}(2^{k^n})$. So $\mathrm{E} = \bigcup_{k\geq 1} \mathrm{E}_k$ and $\mathrm{EXP} = \bigcup_{k\geq 1} \mathrm{EXP}_k$. A set $A \in \mathrm{E}$ is r-E-*nontrivial* if, for any $k \geq 1$, there is a set $B_k \in \mathrm{E} \setminus \mathrm{E}_k$ such that $B_k \leq^p_r A$; and A is r-E-*trivial* otherwise. Similarly, $A \in \mathrm{EXP}$ is r-EXP-*nontrivial* if, for any $k \geq 1$, there is a set $B_k \in \mathrm{EXP} \setminus \mathrm{EXP}_k$ such that $B_k \leq^p_r A$; and A is r-EXP-*trivial* otherwise.

The current paper is the extended version of the authors' conference paper [1] presented at TAMC 2010.

2. Turing Completeness vs. Truth-Table Nontriviality

We start with separating nontriviality (for E and EXP) under Turing and truth-table reducibilities.

Theorem 2.1. *There is a T-E-complete set A such that A is tt-trivial for* E *and* EXP.

Proof. Fix an m-complete set C for E such that $C \in \mathrm{E}_1$. It suffices to define a set A such that

$$C \leq^p_T A, \tag{1}$$

$$A \in \mathrm{E}_1, \mathit{and} \tag{2}$$

$$\forall\, B\ (B \leq^p_{tt} A \Rightarrow B \in \mathrm{E}_6) \tag{3}$$

hold. Namely, (1) and (2) guarantee that A is T-complete for E while, by (3), A is tt-trivial for E and EXP.

We first describe a framework for constructing sets which will guarantee (1) and (2), and then we define a set A within this framework which will satisfy condition (3) too.

In order to guarantee (1) we define a p-Turing reduction of C to A as follows. For any string $z \neq \lambda$, let

$$CODE(z) = \{\langle z, y\rangle : |y| \leq 3|z|^2 + 1\}$$

be the set of z-*codes* where the pairing function $\langle, \rangle$ is defined by $\langle z, y\rangle = 0^{4|z|}1zy$. Then, in the course of the construction of A, we define a string $code(z)$ of length $3|z|^2 + 1$ such that the last bit of $code(z)$ is the value of $C(z)$, i.e.,

$$C(z) = code(z)(3|z|^2), \tag{4}$$

and we put a z-code $\langle z, y\rangle$ into A if and only if y is an initial segment of $code(z)$ thereby guaranteeing

$$A \cap CODE(z) = \{\langle z, y\rangle : y \sqsubseteq code(z)\}. \tag{5}$$

Obviously, this ensures that $C \leq^p_T A$ since, by (5), A can compute $code(z)$ by a standard prefix search, and, by (4), $code(z)$ gives the value of $C(z)$.

Strings will be put into A only by the above coding procedure. So

$$A = \bigcup_{z\in\{0,1\}^+} \{0^{4|z|}1zy : y \sqsubseteq code(z)\} = \bigcup_{z\in\{0,1\}^+} \{\langle z, y\rangle : y \sqsubseteq code(z)\}. \tag{6}$$

Now, for a string z of length $n \geq 1$, $code(z)$ will consist of n segments of length $3n$ and the final coding bit, i.e.,

$$code(z) = v_1^z \ \ldots \ v_n^z \ C(z) \quad \text{where } n = |z| \text{ and } |v_1^z| = \cdots = |v_n^z| = 3n. \tag{7}$$

Moreover, these segments will be chosen so that

$$v_1^z \ldots v_m^z \ (1 \leq m \leq |z|) \ \text{ can be computed in } O(poly(|z|) \cdot 2^{4m}) \text{ steps.} \tag{8}$$

Note that, by $C \in \mathrm{E}_1$, (7) and (8) guarantee that

$$code(z) \text{ can be computed in } O(poly(|z|) \cdot 2^{4|z|}) \leq O(2^{5|z|}) \text{ steps.} \tag{9}$$

This allows us to argue that (2) holds, i.e., that $A \in \mathrm{E}_1$, as follows. Given a string x, it follows from (6) that x is in A if and only if there is a string z such that $x = 0^{4|z|}1zy$ for some initial segment y of $code(z)$. But, by the above observation on the complexity of $code(z)$ and by $|x| \geq 5|z|$, this can be decided in $O(poly(|x|) + 2^{5|z|}) \leq O(2^{|x|})$ steps.

Having described the frame for the construction, we now show how, for given z of length $n \geq 1$, the segments v_m^z $(1 \leq m \leq n)$ of $code(z)$ can be chosen so that (8) is satisfied and such that, for the corresponding set A defined according to (6) and (7), A satisfies condition (3). Since, by the preceding discussion, A will satisfy (1) and (2) too, this will complete the proof.

We start with some notation. Fix a standard enumeration $\{M_e : e \geq 0\}$ of the polynomial-time bounded oracle Turing machines such that, for any oracle X, the run time of M_e^X on inputs of length n is bounded by $p_e(n)$ (uniformly in e and n) where the polynomials p_e are chosen such that $n \leq p_e(n) \leq p_{e+1}(n)$ and $p_e(n)^2 < 2^n$ for all e and n with $e \leq n$. Let $Q_e(x)$ be the set of oracle queries made by $M_e^{\emptyset}$ on input x. Note that, for e and x such that $e \leq |x|$, $Q_e(x)$ consists of less than $p_e(|x|) < 2^{|x|}$ strings, each having length less than $p_e(|x|) < 2^{|x|}$, and $Q_e(x)$ can be computed in time $p_e(|x|) < 2^{|x|}$. Finally, note that if M_e describes a p-tt-reduction then M_e is nonadaptive, i.e., the query set of M_e on input x does not depend on the oracle set whence $Q_e(x)$ is the query set of $M_e^A(x)$.

Now, given a string z of length $n \geq 1$, the segments $v_1^z, \ldots, v_n^z$ of $code(z)$ are inductively defined as follows. Given m with $1 \leq m \leq n$ and the strings $v_1^z, \ldots v_{m-1}^z$, let v_m^z be the least string v of length $3n$ such that

$$\forall\, e < m \ \forall\, x \in \{0,1\}^m \ \forall\, y \in Q_e(x) \ (0^{4|z|}1zv_1^z \ldots v_{m-1}^z v \not\sqsubseteq y). \tag{10}$$

In order to show that v_m^z is well defined (i.e., that there is a string v satisfying (10)) and that the segments v_m^z of $code(z)$ satisfy (8), we first observe that the set $Q = \bigcup_{e<m, |x|=m} Q_e(x)$ of the strings y which are not allowed to extend $0^{4|z|}1zv_1^z \ldots v_{m-1}^z v_m^z$ has cardinality less than 2^{2m} and can be listed in time $O(2^{2m})$. Note that there are m numbers $e < m$ and 2^m strings x of length m. Moreover, as observed above, for each such e and x, $|Q_e(x)| < p_e(m)$. So, by choice of the polynomials p_e (and by $e < m$),

$$|Q| < m \cdot 2^m \cdot p_e(m) \leq p_m(m)^2 \cdot 2^m \leq 2^{2m}.$$

The existence of a listing of Q in time $O(2^{2m})$ follows by a similar argument from the observation that each of the sets $Q_e(x)$ can be listed in time $\leq p_e(m)$.

Now the existence of a string v of length $3n$ as in (10) is immediate since there are 2^{3n} strings v of length $3n$ whereas there are less than 2^{2n} strings y which have to be avoided as extensions of $0^{4|z|}1zv_1^z \dots v_{m-1}^z v$.

Condition (8) is established by induction on m. Given m and $v_1^z, \dots, v_{m-1}^z$ it suffices to show that v_m^z can be computed in $O(poly(n) \cdot 2^{4m})$ steps. Since Q can be listed in time $O(2^{2m})$ and since $z, v_1^z, \dots, v_{m-1}^z$ are given, in $poly(n) \cdot 2^{2m}$ steps we can list the set Q' of all strings w of length $3n$ such that $0^{4|z|}1zv_1^z \dots v_{m-1}^z w$ is an initial segment of any string y in Q. So, by sorting Q', in $O(poly(n) \cdot 2^{4m})$ steps we can find the least string v of length $3n$ such that $v \notin Q'$ and, obviously, v_m^z is the least such string.

It remains to show that (3) is satisfied. So fix a set B such that $B \leq_{tt}^p A$. We have to show that $B \in \mathrm{E}_6$.

Fix e such that M_e is nonadaptive and $B = M_e^A$. Then, given a string x where w.l.o.g. $e < |x|$, $B(x)$ can be computed in $O(2^{6|x|})$ steps by simulating $M_e^A(x)$ as follows. Since M_e is nonadaptive, $Q_e(x)$ is the query set of this computation. So knowing $A(y)$ for all strings $y \in Q_e(x)$ allows us to compute $M_e^A(x)$ in polynomial time. Hence, by $|Q_e(x)| \leq 2^{|x|}$, it suffices to compute $A(y)$ for a given $y \in Q_e(x)$ in $O(2^{5|x|})$ steps.

In order to compute $A(y)$, first decide whether y is an element of a code set $CODE(z)$ and if so compute the unique z such that $y \in CODE(z)$ and the unique w such that $y = 0^{4|z|}1zw$. Since $|y| < p_e(|x|)$, this can be done in $poly(|x|)$ steps. Now if y is not in any code set then, by (6), $y \notin A$. If $y = 0^{4|z|}1zw$ is a z-code then, by (6), $y \in A$ iff $y \sqsubseteq 0^{4|z|}1z code(z)$. For deciding the latter, distinguish the following two cases. If $|z| \leq |x|$ then, by (9), $code(z)$ can be computed in $O(2^{5|z|}) \leq O(2^{5|x|})$ steps. Finally, if $|x| < |z|$ then, by $e < |x| < |z|$ and by choice of $v_{|x|}^z$ (see (10)), $y \sqsubseteq 0^{4|z|}1z code(z)$ if and only if $y \sqsubseteq 0^{4|z|}1zv_1^z \dots v_{|x|}^z$. Moreover, by (8), $v_1^z \dots v_{|x|}^z$ can be computed in $O(poly(|z|) \cdot 2^{4|x|})$ steps, and, by $|z| < |y| < p_e(|x|)$, $O(poly(|z|) \cdot 2^{4|x|}) \leq O(2^{5|x|})$.

This completes the proof. □

3. Collapse of Truth-Table Nontriviality for EXP

In contrast to the hierarchy theorems for EXP-completeness by Watanabe [12] and for weak EXP-completeness by Ambos-Spies, Mayordomo and Zheng [4], here we show that tt-nontriviality for EXP and m-nontriviality for EXP coincide.

Theorem 3.1. *For any set $A \in \mathrm{EXP}$ the following are equivalent.*

(i) A is m-nontrivial for EXP*.*

(ii) A is tt-nontrivial for EXP*.*

Proof. For a proof of the nontrivial direction assume that A is tt-nontrivial for EXP and fix $k \geq 1$. It suffices to show that there is a set B such that $B \leq_m^p A$ and $B \notin \mathrm{EXP}_k$. (Note that, by $A \in \mathrm{EXP}$ and by downward closure of EXP under $\leq_m^p$, $B \leq_m^p A$ will imply that $B \in \mathrm{EXP}$.)

By tt-nontriviality of A, fix a set C such that $C \in \mathrm{EXP} \setminus \mathrm{EXP}_{k+1}$ and $C \leq_{tt}^{p} A$. Moreover, fix a nonadaptive oracle Turing machine M such that $C \leq_{tt}^{p} A$ via M and let p be a polynomial time-bound for M. For any input string x let $q(x,0),\ldots,q(x,n_x)$ be the list of oracle queries of M on input x (with empty oracle) in order of appearance. Finally, define the set B by

$$B = \{\langle x, z_n \rangle : n \leq n_x \ \&\ q(x,n) \in A\}$$

where z_n is the nth string with respect to the length-lexicographical ordering and the coded pair $\langle x, y\rangle$ is defined by $\langle x, y\rangle = 1^{|x|}0xy$.

We claim that B has the required properties. Obviously, $B \leq_m^p A$ via f where f is defined by

$$f(y) = \begin{cases} q(x,n) & \text{if } y = 1^{|x|}0xz_n \ \&\ n \leq n_x \\ x_0 & \text{otherwise} \end{cases}$$

where x_0 is a fixed string in the complement of A.

It remains to show that $B \notin \mathrm{EXP}_k$. For a contradiction assume $B \in \mathrm{EXP}_k$. Then, for given x, $C(x)$ can be computed by the following procedure.

- Compute the queries $q(x,0),\ldots,q(x,n_x)$ by running $M^{\emptyset}$ on input x.
  (This can be done in $p(|x|)$ steps.)
- For $n \leq n_x$ compute $A(q(x,n))$ by using the identity

  $$A(q(x,n)) = B(\langle x, z_n\rangle) = B(1^{|x|}0xz_n).$$

  (Note that $n \leq n_x < p(|x|)$ and that the length of z_n is logarithmic in n whence $|1^{|x|}0xz_n| \leq 3|x| + O(1)$. So, by assumption on B, this part of the procedure can be completed in $O(p(|x|) \cdot 2^{(3|x|)^k})$ steps.)
- Finally, using the values $A(q(x,n))$ $(n \leq n_x)$, simulate the computation of M with oracle A on input x in order to get $C(x) = M^A(x)$.
  (This can be done in $p(|x|)$ steps.)

So $C(x)$ can be computed in

$$O(p(|x|) \cdot 2^{(3|x|)^k}) \leq O(2^{|x|^{k+1}})$$

steps. It follows that $C \in \mathrm{EXP}_{k+1}$ contrary to assumption.

This completes the proof. □

For a tt-E-nontrivial set $A \in \mathrm{E}$ we can modify the above argument as follows. Given $k \geq 1$, take a set C such that $C \in \mathrm{E} \setminus \mathrm{E}_{4k}$ and $C \leq_{tt}^{p} A$, and let B be the set obtained from C as above. Then one can show as above that $B \leq_m^p A$ and $B \notin \mathrm{E}_k$. We cannot argue, however, that B is in E. So the above proof of Theorem 3.1 cannot be converted into a proof of the corresponding claim for E in place of EXP. In fact, as we will show next, Theorem 3.1 fails for E.

4. Separating Nontriviality for E Under Truth-Table Type Reducibilities

We now separate the nontriviality notions for E under the different truth-table type reducibilities. In order to separate E-nontriviality under bounded truth-table reducibility from E-nontriviality under many-one reducibility, we give some stronger separation results by showing that E-nontriviality (or even E-completeness) under bounded truth-table reductions of norm $k+1$ does not imply E-nontriviality under bounded truth-table reductions of norm k.

Theorem 4.1. *(a) Let $k \geq 1$. There is a $(k+1)$-tt-complete set for* E *which is k-tt*-E-*trivial.*

(b) There is a tt-complete set for E *which is btt*-E-*trivial.*

Proof. Since the proofs of the two parts are very similar, we give a detailed proof of part (a) and give some hints how this proof has to be changed in order to prove part (b).

(a) By a slow diagonalization argument, we construct a set $A \in \mathrm{DTIME}(2^{n^2})$ such that

$$\text{A is } (k+1)\text{-tt-hard for E} \tag{11}$$

and

$$\forall B \in \mathrm{E}\ (B \leq^p_{k-tt} A \Rightarrow B \in \mathrm{DTIME}(2^n)). \tag{12}$$

Then any set $\hat{A} \in \mathrm{E}$ with $\hat{A} =^p_m A$ (as, for instance, $\hat{A} = \{0^{|x|^2}1x : x \in A\}$) will be $(k+1)$-*tt*-complete for E but *k*-*tt*-E-trivial.

We first explain how condition (11) is satisfied. Fix an E-complete set C such that $C \in \mathrm{E}_1$. Then it suffices to ensure that $C \leq^p_{(k+1)-tt} A$. This is achieved by guaranteeing

$$x \in C \Leftrightarrow |A \cap CODE(x)| \text{ is odd} \tag{13}$$

for all strings x and for

$$CODE(x) = \{xz^k_0, ..., xz^k_k\}$$

where z^k_n is the $(n+1)$th string of length k in lexicographical order.

Note that $|CODE(x)| = k+1$. Moreover, for any string $y \in CODE(x)$, $|y| = |x| + k$ and, for any strings x and x',

$$x < x' \Rightarrow \forall y \in CODE(x)\ \forall y' \in CODE(x')\ (y < y') \tag{14}$$

holds. So, in particular, $CODE(x) \cap CODE(x') = \emptyset$ for $x \neq x'$. By construction we will have

$$A \subseteq CODE \text{ where } CODE = \bigcup_{x \in \{0,1\}^*} CODE(x). \tag{15}$$

Moreover, for the $(s+1)$th string z_s with respect to the length-lexicographical ordering, $A \cap CODE(z_s)$ will be defined at stage s of the construction of A below.

Our strategy for satisfying (12) is much less straightforward, and, for implementing it, we will need a speed-up argument. We start with some notation.

We model a p-k-tt-reduction by a pair $(\overrightarrow{g}, h)$ where $\overrightarrow{g} = (g_1, \ldots, g_k)$ is a k-tuple of polynomial-time computable selector functions $g_i : \{0,1\}^* \rightarrow \{0,1\}^*$ and h is a polynomial-time computable evaluator function $h : \{0,1\}^* \times \{0,1\}^k \to \{0,1\}$. Then $X \leq^p_{k-tt} Y$ via $(\overrightarrow{g}, h)$ if

$$\forall\, x\, [X(x) = h(x, Y(g_1(x)), \ldots, Y(g_k(x)))].$$

We fix an enumeration $\{(\overrightarrow{g_e}, h_e) : e \geq 0\}$ of all p-k-tt-reductions (where $\overrightarrow{g_e}$ is the k-tuple $(g_{e,1}, ..., g_{e,k})$) such that, for a simultaneous time bound p_e of $g_{e,1}, \ldots, g_{e,k}$ and h_e,

$$\forall\, e \geq 0\, \forall\, x\, [|x| > e \Rightarrow p_e((|x| + k)^2) \leq 2^{|x|}]$$

holds. Since we will only consider reductions to A, by (15) we may assume that all queries are elements of $CODE$, and, w.l.o.g., we may assume that, for any input x the corresponding queries are ordered, i.e., that for all e and x

$$g_{e,1}(x), \ldots, g_{e,k}(x) \in CODE \;\text{ and }\; g_{e,1}(x) < \cdots < g_{e,k}(x)$$

hold. Finally, let $\{E_e : e \geq 0\}$ be an enumeration of E such that, for x with $|x| > e$, $E_e(x)$ can be (uniformly) computed in time $2^{e|x|}$.

Then, in order to satisfy (12), it suffices to meet the requirements

$$\Re'_e : \;\text{ If } E_{e_0} \leq^p_{k-tt} A \text{ via } (\overrightarrow{g_{e_1}}, h_{e_1}) \text{ then } E_{e_0} \in \text{DTIME}(2^n). \tag{16}$$

for all numbers $e = \langle e_0, e_1 \rangle \geq 0$. The strategy for meeting these requirements is based on the following observation. Assume that $E_{e_0} \leq^p_{k-tt} A$ via $(\overrightarrow{g_{e_1}}, h_{e_1})$ i.e., that

$$E_{e_0}(x) = h_{e_1}(x, A(g_{e_1,1}(x)), \ldots, A(g_{e_1,k}(x))) \tag{17}$$

for all strings x. Then, by $A \in \text{DTIME}(2^{n^2})$, $E_{e_0} \in \text{DTIME}(2^n)$ can be established by using this identity as long as, for almost all x, there are no relevant queries $g_{e_1,i}(x)$ with $|g_{e_1,i}(x)|^2 > |x|$ (where a query $g_{e_1,i}(x)$ is irrelevant if the value of $A(g_{e_1,i}(x))$ is not needed for computing $h_{e_1}(x, A(g_{e_1,1}(x)), \ldots, A(g_{e_1,k}(x))))$. So in order to meet requirement $\Re'_e$ it suffices to ensure (by diagonalization) that E_{e_0} is not p-k-tt-reducible to A via $(\overrightarrow{g_{e_1}}, h_{e_1})$ if there are relevant queries $g_{e_1,i}(x)$ such that $|g_{e_1,i}(x)|^2 > |x|$ for infinitely many strings x.

A naive implementation of this strategy, however, will fail since the complexity of the required diagonalization process is not compatible with making A computable in time $O(2^{n^2})$ while, on the other hand, the latter is crucial for the above given argument that the requirement will still be met if no diagonalization candidate is found. This conflict will be resolved by a speed-up argument. For growing indices e the search for a diagonalization witness x for requirement $\Re'_e$ will be more and more strictly bounded. To be more precise, the diagonalization attempt of $\Re'_e$ will only ensure that, for all but finitely many x, there is no relevant query $g_{e_1,i}(x)$ such that $|g_{e_1,i}(x)|^2 \geq 2^e \cdot |x|$. This reduction of the search space will imply that the complexity of the diagonalization procedure required by $\Re'_e$ will be decreasing in e. Since the requirements are finitary, this will allow us to argue that, for any e, we get a speeded up algorithm for computing A which runs in time $O(2^{2^{-e} \cdot n^2})$ (where this algorithm retrieves the information on the finite impact of the first $e+1$ requirements on the construction of A for free by using a finite table which allows it to omit the time consuming actions related to the first $e+1$ requirements which are responsible for the higher complexity of the actual construction).

Hence, in case that the above diagonalization attempt of $\Re'_e$ fails, the obtained bound on the relevant queries will still ensure that E_{e_0} can be computed by (17) in time $O(2^n)$. So requirement $\Re'_e$ will be met.

Having explained the ideas underlying our strategy for satisfying (12), we now describe the strategy formally. We will ensure that

$$\forall\, \alpha > 0\ (A \in \mathrm{DTIME}(2^{\alpha \cdot n^2})) \tag{18}$$

holds (where α is a real number), and, for $e \geq 0$ where $e = \langle e_0, e_1 \rangle$, we will meet the requirement

$$\begin{array}{ll} \Re_e: & \text{If } E_{e_0} \leq^p_{k-tt} A \text{ via } (\overrightarrow{g_{e_1}}, h_{e_1}) \text{ then, for almost all } x \text{ and all } 1 \leq i \leq k \\ & \text{such that } i \text{ is } (e_1, x)\text{-critical, } |x| > 2^{-e} \cdot |g_{e_1,i}(x)|^2. \end{array}$$

where a number i ($1 \leq i \leq k$) is (e_1, x)-*critical* if there are bits $j_i, \ldots, j_k$ and $j'_i, \ldots, j'_k$ such that

$$\begin{array}{c} h_{e_1}(x, A(g_{e_1,1}(x)), ..., A(g_{e_1,i-1}(x)), j_i, ..., j_k) \neq \\ h_{e_1}(x, A(g_{e_1,1}(x)), ..., A(g_{e_1,i-1}(x)), j'_i, ..., j'_k). \end{array} \tag{19}$$

In order to show that this will guarantee (12), let $B \in \mathrm{E}$ be given such that $B \leq^p_{k-tt} A$. Fix $e = \langle e_0, e_1 \rangle$ such that $B = E_{e_0}$ and $E_{e_0} \leq^p_{k-tt} A$ via $(\overrightarrow{g_{e_1}}, h_{e_1})$. Then, by requirement $\Re_e$, we may fix n_0 such that, for all x with $|x| \geq n_0$ and for all i such that i is (e_1, x)-critical, $|x| > 2^{-e}|g_{e_1,i}(x)|^2$ holds. Now, given x with $|x| \geq n_0$, let

$$y_i = \begin{cases} A(g_{e_1,i}(x)) & \text{if } |x| > 2^{-e}|g_{e_1,i}(x)|^2 \\ 0 & \text{otherwise} \end{cases}$$

for $1 \leq i \leq k$. Then by assumption and by choice of n_0

$$B(x) = E_0(x) = h_{e_1}(x, A(g_{e_1,1}(x)), ..., A(g_{e_1,k}(x))) = h_{e_1}(x, y_1, \ldots, y_k).$$

So, in order to show that $B \in \mathrm{DTIME}(2^n)$ it suffices to show that the strings y_i can be computed in $O(2^{|x|})$ steps. But this is immediate by definition of y_i since, by (18) (for $\alpha = 2^{-e}$), $A \in \mathrm{DTIME}(2^{2^{-e} \cdot n^2})$.

We now turn to the construction of A and describe stage s of the construction at which $A \cap CODE(z_s)$ is defined.

We say that requirement $\Re_e$ *requires attention* at stage s if $e < |z_s|$, $\Re_e$ is not satisfied at any stage $t < s$, and either

$$\begin{array}{l} \text{There is an } \Re_e\text{-commitment } (y_i, j_i), ..., (y_k, j_k) \text{ at the end of stage} \\ s - 1 \text{ such that } y_i \in CODE(z_s). \end{array} \tag{20}$$

or there is no $\Re_e$-commitment at the end of stage $s - 1$ and

$$\begin{array}{c} \exists x\ \exists i\ (1 \leq i \leq k\ \&\ i \text{ is } (e_1, x)\text{-critical } \&\ |x| \leq 2^{-e}(|z_s| + k)^2\ \& \\ g_{e_1,i}(x) \in CODE(z_s)\ \&\ [i > 1 \Rightarrow g_{e_1,i-1}(x) \notin CODE(z_s)]) \end{array} \tag{21}$$

holds. (It will be explained below when $\Re_e$ will be satisfied and what an $\Re_e$-commitment will be. Note that (21) can be decided at stage s: By $g_{e_1,i}(x) \in CODE(z_s)$ and

$g_{e_1,i-1}(x) \notin CODE(z_s)$, the question of whether i is (e_1, x)-critical or not depends only on the part of A defined prior to stage s. Also note that $|x| \leq 2^{-e}(|z_s| + k)^2$ and $g_{e_1,i}(x) \in CODE(z_s)$ imply that $|x| \leq 2^{-e}(|g_{e_1,i}(x)|)^2$ since the elements of $CODE(z_s)$ have length $|z_s| + k$.)

Now, if some requirement requires attention, then fix e minimal such that $\Re_e$ requires attention. Declare that $\Re_e$ is *active* at stage s and distinguish the following cases.

If $\Re_e$ requires attention via (20) then let $(y_i, j_i), ..., (y_k, j_k)$ be the $\Re_e$-commitment at the end of stage $s-1$. Otherwise define $(y_i, j_i), ..., (y_k, j_k)$ as follows. Fix x and i as in (21) minimal, let $y_i = g_{e_1,i}(x), ..., y_k = g_{e_1,k}(x)$ and fix $j_i, ..., j_k$ minimal such that

$$E_{e_0}(x) \neq h_{e_1}(x, A(g_{e_1,1}(x)), ..., A(g_{e_1,i-1}(x)), j_i, ..., j_k). \tag{22}$$

In either case call $(y_i, j_i), ..., (y_k, j_k)$ the *critical sequence* of $\Re_e$ at stage s and proceed as follows. Let

$$\mathrm{P}_s = \{y_r : i \leq r \leq k \;\&\; j_r = 1 \;\&\; y_r \in CODE(z_s)\}$$

$$\mathrm{N}_s = \{y_r : i \leq r \leq k \;\&\; j_r = 0 \;\&\; y_r \in CODE(z_s)\}$$

and fix $p \leq k$ minimal such that $y_p \notin CODE(z_s)$ (if there is no such p then let $p = k+1$).

Define $A \cap CODE(z_s)$ by

$$A \cap CODE(z_s) = \begin{cases} \mathrm{P}_s \cup \{y\} & \text{if } |\mathrm{P}_s| \text{ even and } C(z_s) = 1 \\ & \text{or } |\mathrm{P}_s| \text{ odd and } C(z_s) = 0 \\ \mathrm{P}_s & \text{otherwise} \end{cases}$$

where y is the least element of $CODE(z_s)$ such that $y \notin P_s \cup N_s$. (Note that $|\mathrm{P}_s \cup \mathrm{N}_s| \leq k$ and $|CODE(z_s)| = k + 1$.)

Moreover, if $p \leq k$ then let $(y_p, j_p), ..., (y_k, j_k)$ be the $\Re_e$-*commitment* at stage s, and if $p = k + 1$ then declare $\Re_e$ to be *satisfied*. Cancel all $\Re_{e'}$-commitments where $e < e'$. (If $e < e'$ and the current $\Re_{e'}$-commitment is cancelled then we say that requirement $\Re_{e'}$ is *injured* by requirement $\Re_e$.)

If no requirement requires attention then let

$$A \cap CODE(z_s) = \begin{cases} \emptyset & \text{if } C(z_s) = 0 \\ \{z_s 0^k\} & \text{if } C(z_s) = 1 \end{cases}$$

Finally, in any case, if there is an $\Re_e$-commitment $(y_i, j_i), ..., (y_k, j_k)$ at the end of stage $s - 1$ and $\Re_e$ is neither active nor injured at stage s then the $\Re_e$-commitment $(y_i, j_i), ..., (y_k, j_k)$ is in force at the end of stage s too.

This completes the construction of the set A.

Note that the definition of $A \cap CODE(z_s)$ at stage s ensures that (13) (hence (11)) holds. So, in order to show that A has the required properties, it suffices to show that all requirements $\Re_e$ are met and that (18) holds. We do this by establishing a series of claims.

Claim 1. Every requirement $\Re_e$ is active at most finitely often.

Proof. The proof is by induction. Fix e and, by inductive hypothesis, choose s_0 such that no requirement $\Re_{e'}$ with $e' < e$ becomes active after stage s_0. Then $\Re_e$ will not be injured after stage s_0.

Now, for a contradiction, assume that $\Re_e$ is active at infinitely many stages $s > s_0$, say at stages $s_1 < s_2 < s_3 \dots$ Then $\Re_e$ is not satisfied at any of these stages since otherwise it will cease to require attention. So, by construction, at the end of any stage s_n, $n \geq 1$, there will be some commitment $(y_p, j_p), ..., (y_k, j_k)$ attached to $\Re_e$ and, since $\Re_e$ is not injured after stage s_0, no such commitment will be cancelled. So at the following stage at which $\Re_e$ will become active, i.e, at stage s_{n+1}, $\Re_e$ will require attention via (20). But then, by construction, the commitment attached to $\Re_e$ at the end of stage s_{n+1} will be a proper suffix of $(y_p, j_p), ..., (y_k, j_k)$. So this can happen only finitely often contrary to assumption.

Claim 2. Every requirement $\Re_e$ requires attention at most finitely often.

Proof. By Claim 1 fix a stage s_0 such that no requirement $\Re_{e'}$ with $e' \leq e$ is active after stage s_0. Then $\Re_e$ will not require attention at any stage $s > s_0$ (since otherwise $\Re_e$ or some $\Re_{e'}$ with $e' < e$ will become active at stage s contrary to choice of s_0).

Claim 3. If requirement $\Re_e$ is satisfied at some stage s then $\Re_e$ is met.

Proof. Assume that $\Re_e$ is satisfied at stage s. Fix $s' \leq s$ minimal such that $\Re_e$ is active at stage s' and $\Re_e$ is not injured at any stage t with $s' \leq t \leq s$. Then $\Re_e$ requires attention via (21) at stage s'. So there is a string x, a number $1 \leq i \leq k$, and a sequence $(y_i, j_i), ..., (y_k, j_k)$, namely the critical sequence of $\Re_e$ at stage s', such that $y_i = g_{e_1,i}(x), ..., y_k = g_{e_1,k}(x)$, and (22) holds. Now, in order to show that $\Re_e$ is met it suffices to show that

$$E_{e_0}(x) \neq h_{e_1}(x, A(g_{e_1,1}(x)), ..., A(g_{e_1,k}(x))) \tag{23}$$

holds. We do this by distinguishing the following two cases.

First assume that $s' = s$. Then $y_i, ..., y_k \in CODE(z_{s'})$ and we let $A(g_{e_1,m}(x)) = j_m$ for $i \leq m \leq k$ at stage s. So (23) is immediate by (22).

Finally, assume that $s' < s$. Let $s' = s_1 < s_2 < ... < s_n = s$ be the stages t, $s' \leq t \leq s$, at which $\Re_e$ requires attention. Since $\Re_e$ is not injured, $\Re_e$ becomes active at these stages. So, by construction, there are numbers $i = p_0 < p_1 < p_2 < ... < p_n = k + 1$ such that $(y_{p_m}, j_{p_m}), ..., (y_k, j_k)$ is the $\Re_e$-commitment at the end of stage s_m and $A(y_q)$ is set to j_q at stage s_m for $p_{m-1} \leq q < p_m$. So (23) follows from (22) in this case too.

Claim 4. Every requirement $\Re_e$ is met.

Proof. For a contradiction assume that $\Re_e$ is not met, and fix e_0 and e_1 such that $e = \langle e_0, e_1 \rangle$. Then $E_{e_0} \leq^p_{k-tt} A$ via $(\overrightarrow{g_{e_1}}, h_{e_1})$ and

$$\exists^\infty x \; \exists i \; (1 \leq i \leq k \;\&\; i \text{ is } (e_1, x)\text{-critical} \;\&\; |x| \leq 2^{-e} \cdot |g_{e_1,i}(x)|^2). \tag{24}$$

Also note that, by Claim 3, $\Re_e$ is never satisfied and, by Claim 2, we may fix s_0 such that $e < |z_{s_0}|$ and such that no requirement $\Re_{e'}$ with $e' \leq e$ will require attention after stage s_0. So, in particular, $\Re_e$ neither requires attention nor is injured at any stage $\geq s_0$. It follows, by construction, that there is no $\Re_e$-commitment at the end of stage $s_0 - 1$. (Otherwise this commitment will be of the form $(y_i, j_i), ..., (y_k, j_k)$ where $y_i \in CODE(z_{s'})$

for some $s' > s_0 - 1$, in which case $\Re_e$ would require attention at stage s' unless it were to be injured at a stage t with $s_0 - 1 < t \leq s'$. But either would contradict the choice of s_0.) It follows by choice of s_0 that there will be no $\Re_e$-commitments at any stage $s \geq s_0 + 1$. Hence $\Re_e$ will require attention at any stage $s \geq s_0$ such that (21) holds.

So, in order to get the desired contradiction, it suffices to show that there is a stage $s \geq s_0$ such that (21) holds. But the existence of such a stage easily follows from (24). Namely, by (24), there is a string x such that, for some i ($1 \leq i \leq k$), i is (e_1, x)-critical, $|x| \leq 2^{-e} \cdot |g_{e_1,i}(x)|^2$ and $|g_{e_1,i}(x)| > |z_{s_0}| + k$. By our convention on the values of the selection functions g_{e_i} and by choice of the code sets, the latter implies that $g_{e_1,i}(x) \in CODE(z_{s'})$ for some $s' > s_0$. So (21) holds at stage $s' > s_0$.

Claim 5. (18) *holds.*

Proof. It suffices to show that $A \in \mathrm{DTIME}(2^{e^{-1} \cdot n^2})$ for any given $e \geq 4$. An algorithm for computing A within this time bound is based on the following ideas. Fix a stage s_0 such that no requirement $\Re_{e'}$ with $e' \leq e$ requires attention after stage s_0, and put all the relevant information on the first $s_0 + 1$ stages of the construction (like how A is defined on the sets $CODE(z_s)$ for $s \leq s_0$ and which requirements have been satisfied by the end of stage s_0) in a finite table. Then compute $A(x)$ as follows. Find s such that $x \in CODE(z_s)$ (if x is in none of the code sets, $A(x) = 0$). Then $A(x)$ is determined at stage s of the construction. So, if $s \leq s_0$, $A(x)$ can be obtained by table look-up. If $s > s_0$ it suffices to simulate the construction of A at the stages $s_0 + 1, \ldots, s$ (using the information about the stages $\leq s_0$ stored in the table). The crucial observation is that in this simulation all actions related to the requirements $\Re_{e'}$ with $e' \leq e$ may be omitted since these requirements will not require attention after stage s_0 hence have no impact on the actual construction. This modification will sufficiently decrease the complexity of the construction in order to argue that the algorithm meets the given time bound.

In the following we give the argument and the somewhat tedious complexity analysis in detail. Fix $e \geq 4$ and let $A \restriction n$ be the initial segment of (the characteristic sequence of) A of length n, i.e., $A \restriction n = A(z_0) \ldots A(z_{n-1}) = \{z_m \in A : m < n\}$, let $SAT(s)$ be the set of indices e' such that requirement $\Re_{e'}$ is satisfied by the end of stage s, and let $com(e', s)$ be the $\Re_{e'}$-commitment at the end of stage s (if any, and let $com(e', s) = \lambda$ otherwise).

Then, in order to show that $A \in \mathrm{DTIME}(2^{e^{-1} \cdot n^2})$, it suffices to show that there is an inductive procedure which, given

$$A \restriction z_s 0^k = A \cap \bigcup_{s' < s} CODE(z_{s'}),$$

$$SAT(s-1), \text{ and } \tag{25}$$

$$com(e', s-1) \text{ (for } e' < |z_{s-1}|\text{; note that } com(e', s-1) = \lambda \text{ for } e' \geq |z_{s-1}|\text{)},$$

computes

$$A \cap CODE(z_s) \text{ (hence } A \restriction z_{s+1} 0^k\text{)},$$

$$SAT(s), \text{ and } \tag{26}$$

$$com(e', s) \text{ (for } e' < |z_s|\text{)}$$

in $O(2^{(e+1)^{-1}\cdot|z_s|^2})$ steps. Namely, by this inductive procedure, $A \cap CODE(z_s)$ can be computed in

$$O(\sum_{s'=0}^{s} 2^{(e+1)^{-1}\cdot|z_{s'}|^2}) \leq O(2^{|z_s|}2^{(e+1)^{-1}\cdot|z_s|^2}) \leq O(2^{e^{-1}\cdot|z_s|^2})$$

steps. Since $A \subseteq CODE$ and since, for given x, $x \in CODE$ can be decided in $poly(|x|)$ steps, and if so the unique corresponding z_s with $x \in CODE(z_s)$ can be found in $poly(|x|)$ steps too, the claim follows by the fact that $|x| \geq |z_s|$ for $x \in CODE(z_s)$.

Now, in order to show that there is a procedure which computes (26) from (25) in time $O(2^{(e+1)^{-1}\cdot|z_s|^2})$, we give a procedure $\mathcal{P}$ which, given (25), in $O(2^{(e+1)^{-1}\cdot|z_s|^2})$ steps tells whether any requirement is active at stage s and, if so, gives the index e' of the active requirement and its critical sequence $(y_i, j_i), ..., (y_k, j_k)$ at stage s. Note that this is sufficient since, by construction, the parameters in (26) can be computed from these parameters and from the parameters in (25) in $poly(|z_s|)$ steps.

For giving the procedure $\mathcal{P}$ we first observe that (by using a finite table look-up) we may consider only sufficiently large stages s. Hence, by Claim 2, we may assume that no requirement $\Re_{e'}$ with $e' \leq e$ requires attention after stage $s-1$. So fix such an s and assume that (25) is given. Then the procedure $\mathcal{P}$ works as follows.

1. First, $\mathcal{P}$ determines the indices e' such that $\Re_{e'}$ requires attention at stage s. Moreover, in case that $\Re_{e'}$ requires attention via (21), $\mathcal{P}$ in addition computes the least corresponding witnesses x and i together with the least sequences $j_i, \dots, j_k$ and $j'_i, \dots, j'_k$ witnessing that i is (e'_1, x)-critical (i.e. satisfying (19)).

   Note that $\Re_{e'}$ may require attention at stage s only if $e' < |z_s|$ and $e' \notin SAT(s-1)$ (by construction) and if $e' \geq e+1$ (by assumption on s). In order to decide whether for such a number e' the requirement $\Re_{e'}$ requires attention, $\mathcal{P}$ distinguishes the following cases.

   If $com(e', s-1) \neq \lambda$, say $com(e', s-1) = (y_i, j_i), ..., (y_k, j_k)$ (note that the value of $com(e', s-1)$ is given by (25)), then $\Re_{e'}$ requires attention (via (20)) if and only if $y_i \in CODE(z_s)$, and the latter can be decided in $poly(|z_s|)$ steps.

   If $com(e', s-1) = \lambda$ then $\Re_{e'}$ requires attention if and only if there are numbers x and i as in (21) (for $e' = \langle e'_0, e'_1 \rangle$ in place of $e = \langle e_0, e_1 \rangle$). In order to find the least such x and the least corresponding i (if any), $\mathcal{P}$ runs the following subroutines $\mathcal{Q}(e', x)$ for x with $|x| \leq 2^{-e'}(|z_s| + k)^2$ which, for the given x, will find the least corresponding i as in (21) if it exists.

   For $i = 1, \dots, k$ in order, $\mathcal{Q}(e', x)$ computes $g_{e'_1,i}(x)$ and checks whether $g_{e'_1,i}(x)$ is in $CODE(z_s)$. If there is no such i then there is no i corresponding to x as in (21). Otherwise, for the least such i, $\mathcal{Q}(e', x)$ decides whether i is (e'_1, x)-critical by checking for all pairs of $(k+1-i)$-tuples of bits $j_i, \dots, j_k$ and $j'_i, \dots, j'_k$ whether (19) (again with $e' = \langle e'_0, e'_1 \rangle$ in place of $e = \langle e_0, e_1 \rangle$) holds. If such a pair is found, $\mathcal{Q}(e', x)$ returns x, i to the procedure $\mathcal{P}$ together with the least pair $j_i, \dots, j_k$ and $j'_i, \dots, j'_k$ witnessing that i is (e'_1, x)-critical.

The time complexity of the subroutine $\mathcal{Q}(e',x)$ is bounded by $O(2^{|z_s|})$. This is shown as follows. By our choice of the enumeration of the *p-k-tt*-reductions and by $|x| \leq 2^{-e'}(|z_s|+k)^2$, $g_{e'_1,i}(x)$ can be computed in

$$p_{e'_1}(|x|) \leq p_{e'_1}((|z_s|+k)^2) \leq 2^{|z_s|}$$

steps. Moreover, since $CODE \in \mathrm{P} \subseteq \mathrm{E}_1$ and since the elements of $CODE(z_s)$ have length $|z_s|+k$, $g_{e'_1,i}(x) \in CODE(z_s)$ can be decided in $O(2^{|z_s|})$ steps too. So, the first part of the procedure can be completed in $O(k \cdot 2^{|z_s|}) = O(2^{|z_s|})$ steps. If $g_{e'_1,i}(x) \in CODE(z_s)$ for some i, fix i minimal with this property. Then, in order to complete procedure $\mathcal{Q}(e',x)$, for a constant number of sequences of bits $j_i,...,j_k$ the value of

$$h_{e'_1}(x, A(g_{e'_1,1}(x)),...,A(g_{e'_1,i-1}(x)), j_i,...,j_k)$$

has to be computed where the values of $g_{e'_1,1}(x),\dots,g_{e'_1,i-1}(x)$ have been previously computed and the values of $A(g_{e'_1,1}(x)),...,A(g_{e'_1,i-1}(x))$ are provided by (25). So, again by choice of the enumeration of the *p-k-tt*-reductions, this part of the procedure can be completed in $O(2^{|z_s|})$ steps too.

Since the subroutine $\mathcal{Q}(e',x)$ will be called only for strings x such that $|x| \leq 2^{-e'}(|z_s|+k)^2$, it follows that the decision whether requirement $\Re_{e'}$ requires attention can be done in $O(2^{2^{-e'}(|z_s|+k)^2} \cdot 2^{|z_s|})$ steps. Since there are at most $|z_s|$ requirements $\Re_{e'}$ which may require attention and since, for each such e', $e' \geq e+1$, the first part of procedure $\mathcal{P}$ can be completed in

$$O(|z_s| \cdot 2^{2^{-(e+1)}(|z_s|+k)^2} \cdot 2^{|z_s|}) \leq O(2^{2^{-(e+1)} \cdot |z_s|^2 + O(|z_s|)}) \leq O(2^{(e+1)^{-1} \cdot |z_s|^2})$$

steps.

2. If no requirement requires attention then $\mathcal{P}$ is done. Otherwise, by part 1 of the procedure, fix the least e' such that $\Re_{e'}$ requires attention at stage s and, if $\Re_{e'}$ requires attention via (21), fix the least corresponding witnesses x and i together with the least sequences $j_i,\dots,j_k$ and $j'_i,\dots,j'_k$ as in (19).

   Then, in either case, $\Re_{e'}$ becomes active at stage s. Moreover, if $\Re_{e'}$ requires attention via (20) then the critical sequence of $\Re_{e'}$ at stage s is just the $\Re_{e'}$-commitment $com(e',s-1) = (y_i,j_i),...,(y_k,j_k)$ at the end of stage $s-1$ which is given by (25).

   Finally, if $\Re_{e'}$ requires attention via (21) then the critical sequence of $\Re_{e'}$ at stage s is the sequence $(g_{e'_1,i}(x),j_i),...,(g_{e'_1,k}(x),j_k)$ if

$$E_{e'_0}(x) \neq h_{e'_1}(x, A(g_{e'_1,1}(x)),...,A(g_{e'_1,i-1}(x)), j_i,...,j_k) \tag{27}$$

   holds, and the critical sequence is $(g_{e'_1,i}(x),j'_i),...,(g_{e'_1,k}(x),j'_k)$ otherwise (for the above given x, i, $j_i,\dots,j_k$ and $j'_i,\dots,j'_k$).

As we have seen in part 1 of the procedure already, the right hand side of (27) can be computed in $O(2^{|z_s|})$ steps (from (25)). On the other hand, by $|x| \leq 2^{-e'}(|z_s|+k)^2$ and by choice of the enumeration $\{E_e : e \geq 0\}$ of E, $E_{e'_0}(x)$ can be computed in

$$O(2^{e'_0|x|}) \leq O(2^{e'|x|}) \leq O(2^{e' \cdot 2^{-e'}(|z_s|+k)^2}) \leq O(2^{e' \cdot 2^{-e'} \cdot |z_s|^2 + O(|z_s|)})$$

steps. Moreover, by $e+1 \geq 5$, $(e+1) \cdot 2^{-(e+1)} < (e+1)^{-1}$, hence

$$(e+1) \cdot 2^{-(e+1)} \cdot n^2 + O(n) < (e+1)^{-1} \cdot n^2$$

for sufficiently large n. So, by $e' \geq e+1$,

$$O(2^{e' \cdot 2^{-e'} \cdot |z_s|^2 + O(|z_s|)}) \leq O(2^{(e+1)^{-1}|z_s|^2}).$$

It follows that the second part of the procedure $\mathcal{P}$ can be completed in time $O(2^{(e+1)^{-1}|z_s|^2})$ too.

This completes the proof of Claim 5 and the proof of part (a) of the theorem.

(b) For a proof of the second part of the theorem it suffices to construct a set A in DTIME(2^{n^2}) such that

$$\text{A is } tt\text{-hard for E} \tag{28}$$

and

$$\forall B \in \text{E } (B \leq^p_{btt} A \Rightarrow B \in \text{DTIME}(2^n)). \tag{29}$$

In order to ensure (28), we fix an E-complete set C such that $C \in \text{E}_1$ and ensure that $C \leq^p_{tt} A$ as follows. For any string x we guarantee (13) where now the coding set is defined by

$$CODE(x) = \{xz_0^{|x|}, ..., xz_{|x|}^{|x|}\}.$$

Note that $|CODE(x)| = |x| + 1$. Moreover, for any string $y \in CODE(x)$, $|y| = 2|x|$ and, for any strings x and x', (14) holds. As in the proof of part (a), the constructed set A will satisfy (15), and $A \cap CODE(z_s)$ will be defined at stage s of the construction.

The above coding will be flexible enough to allow us to diagonalize against *btt*-reductions in order to satisfy (29). We now fix an enumeration $\{(\overrightarrow{g_{(k,e)}}, h_{(k,e)}) : e \geq 0\}$ of all *p*-*btt*-reductions (i.e., of all *k*-*tt*-reductions for all $k \geq 1$) with properties similar to the enumeration of the *p*-*k*-*tt*-reductions in part (a), say by letting $(\overrightarrow{g_{(k,e)}}, h_{(k,e)})$ be the *k*-*tt*-reduction $(\overrightarrow{g_e}, h_e)$ defined there.

Then (just as in part (a)), in order to satisfy (29), it suffices to satisfy (18) and to meet the requirements

$\Re_{\langle k,e \rangle}$: If $E_{e_0} \leq^p_{k-tt} A$ via $(\overrightarrow{g_{(k,e_1)}}, h_{(k,e_1)})$ then, for almost all x and all $1 \leq i \leq k$ such that i is (e_1, x)-critical, $|x| > 2^{-e} \cdot |g_{(k,e_1),i}(x)|^2$.

for all numbers $k \geq 1$ and $e = \langle e_0, e_1 \rangle \geq 0$ where (e_1, x)-criticalness is defined as in the proof of part (a).

The construction of A and the proof of correctness are obtained by straightforward changes of the corresponding parts of the proof of part (a) of the theorem.

□

We conclude our analysis of E-nontriviality under the weak reducibilities with the observation that 1-*tt*-nontriviality for E coincides with m-nontriviality for E. The corresponding observations for E-completeness and weak E-completeness have been made by Homer et al. [8] and Ambos-Spies et al. [4], respectively.

Lemma 4.2. *Let $A \in \mathrm{E}$ be 1-tt-nontrivial for* E*. Then A is m-nontrivial for* E.

PROOF. Given k, we have to show that there is a set $B \in \mathrm{E} \setminus \mathrm{E}_k$ such that $B \leq^p_m A$. By 1-*tt*-nontriviality of A we may pick $C \in \mathrm{E} \setminus \mathrm{E}_k$ such that $C \leq^p_{1\text{-}tt} A$, say $C \leq^p_{1\text{-}tt} A$ via the selector function $g : \{0,1\}^* \to \{0,1\}^*$ and the evaluator $h : \{0,1\}^* \times \{0,1\} \to \{0,1\}$, i.e., $C(x) = h(x, A(g(x)))$. Then,

$$C(x) = \begin{cases} A(g(x)) & \text{if } h(x,0) < h(x,1) \\ 1 - A(g(x)) & \text{if } h(x,0) > h(x,1) \\ 0 & \text{if } h(x,0) = h(x,1) = 0 \\ 1 & \text{if } h(x,0) = h(x,1) = 1. \end{cases}$$

Now let

$$B(x) = \begin{cases} 1 - C(x) & \text{if } h(x,0) > h(x,1) \\ C(x) & \text{otherwise.} \end{cases}$$

Then, as one can easily check, $B \in \mathrm{E} \setminus \mathrm{E}_k$. Moreover, $B \leq^p_m A$ via the function f defined by

$$f(x) = \begin{cases} g(x) & \text{if } h(x,0) \neq h(x,1) \\ y_0 & \text{if } h(x,0) = h(x,1) = 0 \\ y_1 & \text{if } h(x,0) = h(x,1) = 1. \end{cases}$$

where y_0 and y_1 are fixed strings such that $y_0 \notin A$ and $y_1 \in A$. □

5. Summary of Results

Our results on the relations among completeness and nontriviality under the common polynomial-time reducibilities can be summarized as follows.

Theorem 5.1. *For $A \in \mathrm{E}$ the following and (up to transitive closure) only the following implications hold in general:*

$$\begin{array}{ccc}
\boxed{\begin{array}{c} A\ m\text{-E-}\mathit{complete} \\ \Updownarrow \\ A\ 1\text{-}tt\text{-E-}\mathit{complete} \end{array}} & \Rightarrow & \boxed{\begin{array}{c} A\ m\text{-E-}\mathit{nontrivial} \\ \Updownarrow \\ A\ 1\text{-}tt\text{-E-}\mathit{nontrivial} \end{array}} \\
\Downarrow & & \Downarrow \\
A\ btt\text{-E-}\mathit{complete} & \Rightarrow & A\ btt\text{-E-}\mathit{nontrivial} \\
\Downarrow & & \Downarrow \\
A\ tt\text{-E-}\mathit{complete} & \Rightarrow & A\ tt\text{-E-}\mathit{nontrivial} \\
\Downarrow & & \Downarrow \\
A\ T\text{-E-}\mathit{complete} & \Rightarrow & A\ T\text{-E-}\mathit{nontrivial}
\end{array}$$

Proof. Note that the downwards implications and the implications from left to right are immediate by definitions (and by the time hierarchy theorem). The unique upwards arrows in the first and the second columns hold by Homer et al. [8] and Lemma 4.2, respectively. Finally, Theorems 2.1 and 4.1 imply that no other implications hold. □

Theorem 5.2. *For $A \in$ EXP the following and (up to transitive closure) only the following implications hold in general:*

A m-EXP-*complete*		A m-EXP-*nontrivial*
$\Updownarrow$		$\Updownarrow$
A 1-tt-EXP-*complete*		A 1-tt-EXP-*nontrivial*
$\Downarrow$		$\Updownarrow$
A btt-EXP-*complete*		A btt-EXP-*nontrivial*
$\Downarrow$		$\Updownarrow$
A tt-EXP-*complete*	$\Rightarrow$	A tt-EXP-*nontrivial*
$\Downarrow$		$\Downarrow$
A T-EXP-*complete*	$\Rightarrow$	T-EXP-*nontrivial*

Proof. Again the downwards implications and the implications from left to right are immediate by definitions (and by the time hierarchy theorem) while the unique upwards arrow in the first column holds by Homer et al. [8]. The upwards implications in the second column are justified by Theorem 3.1. Finally, Theorems 2.1 and the separation results for the exponential time completeness notions in Watanabe [12] imply that no other implications hold. (In place of [12] we may also apply Theorem 4.1 which, by the coincidence of hardness for E and EXP, implies the required results from [12].) □

Here we have not looked at E- or EXP-nontriviality under the strong reducibilities, i.e., at the reducibilities strengthening many-one reducibility. Berman [7] has shown that E-completeness under many-one reducibility coincides with E-completeness under length-increasing one-one reducibility and the corresponding fact for weak E- (and EXP-) completeness has been shown in [4]. It can be easily shown that E- (and EXP-) nontrivality under many-one reducibility coincides with E- (and EXP-) nontrivality under length-increasing many-one reducibility. The question whether nontriviality under many-one reducibility and nontriviality under one-one reducibility coincide, however, is open. We have obtained such a collapse only under the strong hypothesis that P = PSPACE. (All these results can be found in the doctoral thesis of the second author.)

References

[1] Ambos-Spies, K., Bakibayev, T.: Nontriviality for exponential time w.r.t. weak reducibilities. Proceedings TAMC 2010, 84-93, Lecture Notes in Comput. Sci., 6108, Springer, Berlin, 2010

[2] Ambos-Spies, K., Bakibayev, T.: Weak completeness notions for exponential time. Proceedings ICALP 2010, Part I, 503–514, Lecture Notes in Comput. Sci., 6198, Springer, Berlin, 2010

[3] Ambos-Spies, K., Bakibayev, T.: Comparing nontriviality for E and EXP. Theory of Computing Systems (to appear)

[4] Ambos-Spies K., Mayordomo E., Zheng X.: A Comparison of Weak Completeness Notions. Proceedings of the 11th Annual IEEE Conference on Computational Complexity, 171–178 (1996)

[5] Balcázar, J.L., Díaz, J., Gabarró, J.: Structural complexity I. Second edition. Springer, Berlin (1995)

[6] Balcázar, J.L., Díaz, J., Gabarró, J.: Structural complexity II. Springer, Berlin (1990)

[7] Berman, L.: On the structure of complete sets: almost everywhere complexity and infinitely often speedup. Proceedings of the 17th Annual Symposium on Foundations of Computer Science, 76–80 (1976)
[8] Homer, S., Kurtz, S., Royer, J.: On 1-truth-table-hard languages. Theoret. Comput. Sci. 115, 383–389 (1993)
[9] Juedes, D.W., Lutz, J.H.: Weak completeness in E and E_2. Theoret. Comput. Sci. 143, 149–158 (1995)
[10] Ladner, R. E., Lynch, N. A., Selman, A. L: A comparison of polynomial time reducibilities. Theoret. Comput. Sci. 1, 103–123 (1975)
[11] Lutz, J.H.: Weakly hard problems. SIAM J. Comput. 24, 1170–1189 (1995)
[12] Watanabe, O.: A comparison of polynomial time completeness notions. Theoret. Comput. Sci. 54, 249–265 (1987)