Bi-Immunity Separates Strong NP-Completeness Notions

A. Pavan [a,*,1], Alan L. Selman [b]

[a] *Department of Computer Science, Iowa State University, Ames, IA 50011*

[b] *Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY 14260.*

Abstract

We prove that if for some $\epsilon > 0$, NP contains a set that is DTIME(2^{n^ϵ})-bi-immune, then NP contains a set that is 2-Turing complete for NP (hence 3-truth-table complete) but not 1-truth-table complete for NP. Thus this hypothesis implies a strong separation of completeness notions for NP. Lutz and Mayordomo [LM96] and Ambos-Spies and Bentzien [ASB00] previously obtained the same consequence using strong hypotheses involving resource-bounded measure and/or category theory. Our hypothesis is weaker and involves no assumptions about stochastic properties of NP.

Key words: Turing completeness, many-one completeness, bi-immunity.
1991 MSC: 68Q15

---

1 Introduction

We obtain a strong separation of polynomial-time completeness notions under the hypothesis that for some $\epsilon > 0$, NP contains a set that is DTIME(2^{n^ϵ})-bi-immune. We prove under this hypothesis that NP contains a set that is $\leq^{\rm P}_{2-T}$-complete (hence $\leq^{\rm P}_{3-tt}$-complete) for NP but not $\leq^{\rm P}_{1-tt}$-complete for NP. In addition, we prove that if for some $\epsilon > 0$, NP $\cap$ *co*-NP contains a set that is DTIME(2^{n^ϵ})-bi-immune, then NP contains a set that is $\leq^{\rm P}_{2-tt}$-complete for NP

* Corresponding author.
Email addresses: `pavan@cs.iastate.edu` (A. Pavan), `selman@cse.buffalo.edu` (Alan L. Selman).

[1] Work done while the author was at University at Buffalo.

but not $\leq^{\mathrm{P}}_{1-tt}$-complete for NP. (We review common notation for polynomial-time reducibilities in the next section.)

The question of whether various completeness notions for NP are distinct has a very long history [LLS75], and has always been of interest because of the surprising phenomenon that no natural NP-complete problem has ever been discovered that requires anything other than many-one reducibility for proving its completeness. This is in contrast to the situation for NP-hard problems. There exist natural, combinatorial problems that are hard for NP using Turing reductions that have not been shown to be hard using nonadaptive reductions [JK76]. The common belief is that NP-hardness requires Turing reductions, and this intuition is confirmed by the well-known result that if $\mathrm{P} \neq \mathrm{NP}$, then there are sets that are hard for NP using Turing reductions that are not hard for NP using many-one reductions [SG77].

There have been few results comparing reducibilities within NP, and we have known very little concerning various notions of NP-completeness. The first result to distinguish reducibilities within NP is an observation of Wilson in one of Selman's papers on p-selective sets [Sel82]. It is a corollary of results there that if $\mathrm{NE} \cap co\text{-}\mathrm{NE} \neq \mathrm{E}$, then there exist sets A and B belonging to NP such that $A \leq^{\mathrm{P}}_{postt} B$, $B \leq^{\mathrm{P}}_{tt} A$, and $B \not\leq^{P}_{postt} A$, where $\leq^{\mathrm{P}}_{postt}$ denotes positive truth-table reducibility. Regarding completeness, Longpré and Young [LY90] proved that there are $\leq^{\mathrm{P}}_{m}$-complete sets for NP for which $\leq^{\mathrm{P}}_{T}$-reductions to these sets are *faster*, but they did not prove that the completeness notions differ. Lutz and Mayordomo [LM96] were the first to give technical evidence that $\leq^{\mathrm{P}}_{T}$- and $\leq^{\mathrm{P}}_{m}$-completeness for NP differ. They proved that if the p-measure of NP is not zero, then there exists a $\leq^{\mathrm{P}}_{2-T}$-complete language for NP that is not $\leq^{\mathrm{P}}_{m}$-complete. Ambos-Spies and Bentzien [ASB00] extended this result significantly. They used an hypothesis of resource-bounded category theory that asserts that "NP has a p-generic language, " which is weaker than the hypothesis of Lutz and Mayordomo, to separate nearly all NP-completeness notions for the bounded truth-table reducibilities, including the consequence obtained by Lutz and Mayordomo.

Here we prove that the consequence of Lutz and Mayordomo follows from the hypothesis that NP contains a $\mathrm{DTIME}(2^{n^\epsilon})$-bi-immune language. This hypothesis is weaker than the genericity hypothesis in the sense that the genericity hypothesis implies the existence of a 2^{n^ϵ}-bi-immune language in NP. Indeed, there exists a $\mathrm{DTIME}(2^{n^\epsilon})$-bi-immune language, in EXP, that is not p-generic [PS02]. Notably, our hypothesis, unlike either the measure or genericity hypotheses, involves no stochastic assumptions about NP.

Pavan and Selman [PS02] proved that if for some $\epsilon > 0$, $\mathrm{NP} \cap co\text{-}\mathrm{NP}$ contains a set that is $\mathrm{DTIME}(2^{n^\epsilon})$-bi-immune, then there exists a $\leq^{\mathrm{P}}_{T}$-complete set for NP that is not $\leq^{\mathrm{P}}_{m}$-complete. The results that we present here are significantly sharper. Also, they introduced an Hypothesis H from which it follows that there exists a $\leq_T^{\mathrm{P}}$-complete set for NP that is not $\leq_{tt}^{\mathrm{P}}$-complete. We do not need to state this hypothesis here. Suffice it to say that if for some $\epsilon > 0$, UP$\cap$*co*-UP contains a DTIME(2^{n^ϵ})-bi-immune set, then Hypothesis H is true. Thus, we may partially summarize the results of the two papers as follows:

(1) If for some $\epsilon > 0$, NP contains a DTIME(2^{n^ϵ})-bi-immune set, then NP contains a set that is $\leq_{2-T}^{\mathrm{P}}$-complete (hence $\leq_{3-tt}^{\mathrm{P}}$-complete) that is not $\leq_{1-tt}^{\mathrm{P}}$-complete.
(2) If for some $\epsilon > 0$, NP $\cap$ *co*-NP contains a DTIME(2^{n^ϵ})-bi-immune set, then NP contains a set that is $\leq_{2-tt}^{\mathrm{P}}$-complete that is not $\leq_{1-tt}^{\mathrm{P}}$-complete.
(3) If for some $\epsilon > 0$, UP $\cap$ *co*-UP contains a DTIME(2^{n^ϵ})-bi-immune set, then NP contains a set that is $\leq_T^{\mathrm{P}}$-complete that is not $\leq_{tt}^{\mathrm{P}}$-complete.

2 Preliminaries

We use standard notation for polynomial-time reductions [LLS75] and we assume that readers are familiar with Turing, $\leq_T^{\mathrm{P}}$, and many-one, $\leq_m^{\mathrm{P}}$, reducibilities. Given any positive integer $k > 0$, a *k-Turing* reduction ($\leq_{k\text{-}T}^{P}$) is a Turing reduction that on each input word makes at most k queries to the oracle. A set A is *truth-table* reducible to a set B ($A \leq_{tt}^{\mathrm{P}} B$) if there exist polynomial-time computable functions g and h such that on input x, $g(x)$, for some $m \geq 0$, is (an encoding of) a set of queries $Q = \{q_1, q_2, \cdots, q_m\}$, and $x \in A$ if and only if $h(x, B(q_1), \cdots, B(q_m)) = 1$. For a constant $k > 0$, A is *k-truth-table* reducible to B ($A \leq_{k\text{-}tt}^{P} B$) if for all x, $\|Q\| = k$. Given a polynomial-time reducibility $\leq_r^{\mathrm{P}}$, recall that a set S is $\leq_r^{\mathrm{P}}$-complete for NP if $S \in$ NP and every set in NP is $\leq_r^{\mathrm{P}}$-reducible to S.

A language is DTIME$(T(n))$-*complex* if L does not belong to DTIME$(T(n))$ almost everywhere; that is, every Turing machine M that accepts L runs in time greater than $T(|x|)$, for all but finitely many words x. A language L is *immune* to a complexity class **C**, or **C**-*immune*, if L is infinite and no infinite subset of L belongs to **C**. A language L is *bi-immune* to a complexity class **C**, or **C**-*bi-immune*, if both L and $\overline{L}$ are **C**-*immune*. Balcázar and Schöning [BS85] proved that for every time-constructible function T, L is DTIME$(T(n))$-complex if and only if L is bi-immune to DTIME$(T(n))$. We will use the following property of bi-immune sets. See Balcázar *et al.* [BDG90] for a proof.

Proposition 1 *Let L be a* DTIME$(T(n))$*-bi-immune language and A be an infinite set in* DTIME$(T(n))$*. Then both $A \cap L$ and $A \cap \overline{L}$ are infinite.*

3 Separation Results

Our first goal is to separate $\leq_{2-T}^{\mathrm{P}}$-completeness from $\leq_m^{\mathrm{P}}$-completeness under the assumption that NP contains a DTIME(2^{2n})-bi-immune language.

Theorem 2 *If* NP *contains a* DTIME(2^{2n})*-bi-immune language, then* NP *contains a* $\leq_{2-T}^{\mathrm{P}}$*-complete set* S *that is not* $\leq_m^{\mathrm{P}}$*-complete.*

Proof.

Let L be a DTIME(2^{2n})-bi-immune language in NP. Let $k > 0$ be a positive integer such that $L \in \mathrm{DTIME}(2^{n^k})$. Let M decide L in 2^{n^k} time. Define

$$\begin{aligned} t_1 &= 2^k, \text{ and, for } i \geq 1, \\ t_{i+1} &= (t_i)^{k^2}, \end{aligned}$$

and, for each $i \geq 1$, define

$$I_i = \{x \mid t_i^{1/k} \leq |x| < t_i^k\}.$$

Observe that $\{I_i\}_{i \geq 1}$ partitions $\Sigma^* - \{x \mid |x| < 2\}$. Define the following sets:

$$\begin{aligned} E &= \cup_{i \text{ even}} I_i, \\ O &= \cup_{i \text{ odd}} I_i, \\ L_e &= L \cap E, \\ L_o &= L \cap O, \\ \mathrm{PadSAT} &= \mathrm{SAT} \cap E. \end{aligned}$$

Since L belongs to NP, L_e and L_o also belong to NP. We can easily see that PadSAT is NP-complete.

We now define our $\leq_{2-T}^{\mathrm{P}}$-complete set S. To simplify the notation we use a three letter alphabet.

$$S = 0(L_e \cup \mathrm{PadSAT}) \cup 1(L_e \cap \mathrm{PadSAT}) \cup 2L_e.$$

It is easy to see that S is $\leq_{2-T}^{\mathrm{P}}$-complete: To determine whether a string x belongs to PadSAT, first query whether $x \in L_e$. If $x \in L_e$, then $x \in \mathrm{PadSAT}$ if and only if $x \in (L_e \cap \mathrm{PadSAT})$, and, if $x \notin L_e$, then $x \in \mathrm{PadSAT}$ if and only if $x \in (L_e \cup \mathrm{PadSAT})$. The same reduction, since it consists of three distinct queries, demonstrates also that S is $\leq_{3-tt}^{\mathrm{P}}$-complete for NP.

The rest of the proof is to show that S is not $\leq_m^{\mathrm{P}}$-complete for NP. So assume otherwise and let f be a polynomial-time computable many-one reduction of

L_o to S. We will show this contradicts the hypothesis that L is DTIME(2^{2n})-bi-immune.

We need the following lemmas about L_o. Note that $L_o \subseteq O$.

Lemma 3 *Let A be an infinite subset of O that can be decided in 2^{2n} time. Then both the sets $A \cap L_o$ and $A \cap \overline{L_o}$ are infinite.*

Proof. Since A is a subset of O, a string x in A belongs to L_o if and only if it belongs to L. Thus $A \cap L_o$ is infinite if and only if $A \cap L$ is infinite. Similarly, $A \cap \overline{L_o}$ is infinite if and only if $A \cap \overline{L}$ is infinite. Since A can be decided in 2^{2n} time, and L is 2^{2n}-bi-immune, by Proposition 1, both the sets $A \cap L$ and $A \cap \overline{L}$ are infinite. Thus, $A \cap L_o$ and $A \cap \overline{L_o}$ are infinite. ∎

Lemma 4 *Let A belong to* DTIME(2^{n^k})*, and suppose that g is a $\leq_m^{\mathrm{P}}$-reduction from L_o to A. Then the set*

$$T = \{x \in O \mid |g(x)| < |x|^{1/k}\}$$

is finite.

Proof.

It is clear that $T \in \mathrm{P}$. Recall that M is a deterministic algorithm that correctly decides L. Let N decide A in 2^{n^k} time. The following algorithm correctly decides L and runs in 2^n time on all strings belonging to T: On input x, if x does not belong to T, then run M on x. If $x \in T$, then $x \in L$ if and only if $x \in L_o$, so run N on $g(x)$ and accept if and only if N accepts $g(x)$. N takes $2^{|g(x)|^k}$ steps on $g(x)$. Since $|g(x)| < |x|^{1/k}$, N runs in $2^{|x|}$ time. Thus, the algorithm runs in 2^n steps on all strings belonging to T. Unless T is finite, this contradicts the fact that L is DTIME(2^{2n})-bi-immune. ∎

Next we show that the reduction, f from L_o to S, should map almost all the strings of O to strings of form by, where $y \in E$ and $b \in \{0, 1, 2\}$.

Lemma 5 *Let*

$$A = \{x \mid x \in O, f(x) = by, \text{ and } y \in O\}.$$

Then A is finite.

Proof. It is easy to see that A belongs to P. Both PadSAT and L_e are subsets of E. Thus if a string by belongs to S, where $b \in \{0, 1, 2\}$, then $y \in E$. For every string x in A, $f(x) = by$ and $y \in O$. Thus $by \notin S$, which implies, since f is a many-one reduction from L_o to S, that $x \notin L_o$. Thus $A \cap L_o$ is empty. Since $A \subseteq O$, if A were infinite, then this would contradict Lemma 3, so A is finite. ∎

Thus, for all but finitely many x, if $x \in O$ and $f(x) = by$, then $y \in E$. Now we consider the following set B,

$$B = \{x \mid |x| = t_i \text{ and } i \text{ is odd}\}.$$

Observe that $B \in \mathrm{P}$ and that B is an infinite subset of O. Thus, by Lemma 3, $B \cap L_o$ is an infinite set. Since, for all strings x, $x \in L_o \Leftrightarrow f(x) \in S$, it follows that f maps infinitely many of the strings in B into S. The rest of the proof is dedicated to showing a contradiction to this fact. In particular, we define the sets

$$\begin{aligned} B_0 &= \{x \in B \mid f(x) = 0y\}, \\ B_1 &= \{x \in B \mid f(x) = 1y\}, \text{ and} \\ B_2 &= \{x \in B \mid f(x) = 2y\}, \end{aligned}$$

and we prove that each of these sets is finite.

Lemma 6 *B_0 is finite.*

Proof. Assume B_0 is infinite. Let

$$C = \{x \in B_0 \mid f(x) = 0y \text{ and } y \in E\}.$$

Since B_0 is a subset of O, by Lemma 5, for all but finitely strings in B_0, if $f(x) = 0y$, then $y \in E$. Thus B_0 is infinite if and only if C is infinite.

Consider the following partition of C.

$$\begin{aligned} C_1 &= \{x \in C \mid f(x) = 0y, |y| < |x|^{1/k}\}, \\ C_2 &= \{x \in C \mid f(x) = 0y, |x|^{1/k} \le |y| < |x|^k\}, \\ C_3 &= \{x \in C \mid f(x) = 0y, |y| \ge |x|^k\}. \end{aligned}$$

We will show that each of the sets C_1, C_2, and C_3 is finite.

Claim 7 *C_1 is finite.*

Proof. Since $S \in \mathrm{DTIME}(2^{n^k})$, the claim follows from Lemma 4. ∎

Claim 8 *C_2 is the empty set.*

Proof. Assume that $x \in C_2$. Since $C_2 \subseteq C \subseteq B$, $|x| = t_i$, for some odd i. So, $|x|^{1/k} \le |y| < |x|^k$ implies that $t_i^{1/k} \le |y| < t_i^k$, which implies $y \in I_i$. Since i is odd, $y \in O$. However, by definition of C, $y \in E$. Thus, $C_2 = \emptyset$. ∎

Claim 9 *C_3 is finite.*

Proof. Observe that $C_3 \in \mathrm{P}$. Suppose C_3 is infinite. Define $C_4 = C_3 - L_o$. We first show, under the assumption C_3 is infinite, that C_4 is infinite. Suppose C_4 is finite. Then the set $C_5 = C_3 \cap L_o$ differs from C_3 by a finite set. Thus, since $C_3 \in \mathrm{P}$, $C_5 \in \mathrm{P}$ also. At this point, we know that C_5 is an infinite subset of O that belongs to P, and that C_5 is a subset of L_o. Thus, $C_5 \cap \overline{L_o}$ is empty, which contradicts Lemma 3. Thus, C_4 is an infinite subset of C_3.

Let

$$F = \{y \in E \mid \exists x\ [x \in O, x \notin L_o, f(x) = 0y, \text{ and } |y| \geq |x|^k]\}.$$

The following implications show that F is infinite:

$$\begin{array}{c} C_4 \text{ is infinite} \\ \Rightarrow \\ \exists^\infty x\ [x \in O, x \notin L_o, f(x) = 0y, |y| \geq |x|^k, y \in E] \\ \Rightarrow \\ \exists^\infty y \in E\ [\exists x\ x \in O, x \notin L_o, f(x) = 0y, |y| \geq |x|^k]. \end{array}$$

For each string y in F, there exists a string $x \in O - L_o$ such that $f(x) = 0y$. Since f is a many-one reduction from L_o to S, $f(x) = 0y \notin S$. Thus $y \notin L_e \cup \mathrm{PadSAT}$, and so $y \notin L_e$. However, since $y \in E$, we conclude that $y \notin L$. Thus, F is an infinite subset of $\overline{L}$.

Now we contradict the fact that L is $\mathrm{DTIME}(2^{2n})$-bi-immune by showing that F is decidable in time 2^{2n}. Let y be an input string. First decide, in polynomial time, whether y belongs to E. If $y \notin E$, then $y \notin F$. If $y \in E$, compute the set of all x such that $|x| \leq |y|^{1/k}$, $x \in O$, and $f(x) = 0y$. Run M on every string x in this set until M rejects one of them. Since $x \in O$, M rejects a string x only if $x \notin L_o$. If such a string is found, then $y \in F$, and otherwise $y \notin F$. There are at most $2 \times 2^{|y|^{1/k}}$ many x's such that $|x| \leq |y|^{1/k}$ and $f(x) = 0y$. The time taken to run M on each such x is at most $2^{|x|^k} \leq 2^{|y|}$. Thus, the total time to decide whether $y \in F$ is at most $2^{|y|} \times 2^{|y|^{1/k}} \times 2 \leq 2^{2|y|}$. Thus, F is decidable in time 2^{2n}.

We conclude that F must be a finite set. Therefore, C_4 is finite, from which it follows that C_3 is finite. ∎

Each of the claims is established. Thus, $C = C_1 \cup C_2 \cup C_3$ is a finite set, and this proves that B_0 is a finite set. ∎

Lemma 10 *B_1 is a finite set.*

Proof. Much of the proof is similar to the proof of Lemma 6. Assume that B_1 is infinite. This time, define

$$C = \{x \in B_1 \mid f(x) = 1y \text{ and } y \in E\}.$$

By Lemma 5, C is infinite if and only if B_1 is infinite. Thus, by our assumption, C is infinite. Partition C as follows.

$$\begin{aligned} C_1 &= \{x \in C \mid f(x) = 1y, |y| < |x|^{1/k}\} \\ C_2 &= \{x \in C \mid f(x) = 1y, |x|^{1/k} \leq |y| < |x|^k\} \\ C_3 &= \{x \in C \mid f(x) = 1y, |y| \geq |x|^k\} \end{aligned}$$

As in the proof of Lemma 6, we can show that C_1 is a finite set and C_2 is empty. Now we proceed to show that C_3 is also a finite set.

Claim 11 *C_3 is finite.*

Proof. Assume C_3 is infinite and observe that $C_3 \in \mathrm{P}$. Define $C_4 = C_3 \cap L_o$. Now we show that C_4 is infinite. If C_4 is finite, then $C_5 = C_3 - L_o$ contains all but finitely many strings of C_3. Thus, since C_3 belongs to P, C_5 also belongs to P. Thus C_5 is an infinite subset of O that belongs to P, for which $C_5 \cap L_o$ is empty. That contradicts Lemma 3. Thus, C_4 is infinite.

Consider the following set:

$$F = \{y \in E \mid \exists x[x \in L_o, f(x) = 1y, |y| \geq |x|^k]\}$$

The following implications show that F is infinite.

$$\begin{gathered} C_4 \text{ is infinite} \\ \Rightarrow \\ \exists^\infty x\ [x \in L_o, f(x) = 1y, |y| \geq |x|^k, y \in E] \\ \Rightarrow \\ \exists^\infty y[\exists x\ f(x) = 1y, |y| \geq |x|^k, x \in L_o, y \in E]. \end{gathered}$$

For each string $y \in F$, there exists a string $x \in L_o$ such that $f(x) = 1y$. Since f is a $\leq_m^{\mathrm{P}}$-reduction from L_o to S, $f(x) = 1y \in S$, so $y \in L_e \cap \mathrm{PadSAT}$. In particular, $y \in L_e \subseteq L$. Therefore, F is an infinite subset of L. However, as in the proof of Claim 9, we can decide whether $y \in F$ in $2^{2|y|}$ steps, which contradicts the fact that L is DTIME(2^{2n})-bi-immune: Let y be an input string. First decide whether $y \in E$, and if not, then reject. If $y \in E$, then search all strings x such that $|x| \leq |y|^{1/k}$, $x \in O$, and $f(x) = 1y$. For each such x, run M on x to determine whether $x \in L \cap O = L_o$. If an $x \in L_o$ is found, then

$y \in F$, and otherwise $y \notin F$. The proof that this algorithm runs in 2^{2n} steps is identical to the argument in the proof of Claim 9.

Therefore, F is finite, from which it follows that C_4 is finite, and so C_3 must be finite. ∎

Now we know that C is finite. This proves that B_1 is finite, which completes the proof of Lemma 10. ∎

Lemma 12 *B_2 is a finite set.*

Proof. Assume B_2 is infinite. Then

$$C = \{x \in B \mid f(x) = 2y, \text{ and } y \in E\}$$

is infinite. We partition C into

$$\begin{aligned} C_1 &= \{x \in C \mid f(x) = 2y, |y| < |x|^{1/k}\} \\ C_2 &= \{x \in C \mid f(x) = 2y, |x|^{1/k} \leq |y| < |x|^k\} \\ C_3 &= \{x \in C \mid f(x) = 2y, |y| \geq |x|^k\} \end{aligned}$$

The proofs that C_1, C_2, and C_3 are finite are identical to the arguments in the proof of Lemma 10. (In particular, it suffices to define F as in the proof of Lemma 10.) ∎

Now we have achieved our contradiction, for we have shown that the each of the sets B_1, B_2, and B_3 are finite. Therefore, f cannot map infinitely many of the strings in B into S, which proves that f cannot be a $\leq_m^{\mathrm{P}}$-reduction from L_o to S. Therefore, S is not $\leq_m^{\mathrm{P}}$-complete. ∎

Next we show that NP has a $\mathrm{DTIME}(2^{n^\epsilon})$-bi-immune set if and only if NP has a $\mathrm{DTIME}(2^{n^k})$-bi-immune set using a reverse padding trick [ASTZ97].

Theorem 13 *Let $0 < \epsilon < 1$ and k be any positive integer.* NP *has a* $\mathrm{DTIME}(2^{n^\epsilon})$*-bi-immune set if and only if* NP *has a* $\mathrm{DTIME}(2^{n^k})$*-bi-immune set.*

Proof.

The implication from right to left is obvious. Let $L \in \mathrm{NP}$ be a $\mathrm{DTIME}(2^{n^\epsilon})$-bi-immune set. Define

$$L' = \{x \mid 0^{n^{k/\epsilon}} x \in L, |x| = n\}$$

and observe that $L' \in \mathrm{NP}$. We claim that L' is $\mathrm{DTIME}(2^{n^k})$-bi-immune. Suppose otherwise. Then there exists an algorithm M that decides L' and M runs in 2^{n^k} steps on infinitely many strings. Consider the following algorithm for L:

input y;
if $y = 0^{n^{k/\epsilon}}x$ ($|x| = n$)
 then run M on x
  and accept y if and only if M accepts x
 else run a machine that decides L;

Since M runs in 2^{n^k} time on infinitely many x, the above algorithm runs in time $2^{|x|^k}$ steps on infinitely many strings of the form $y = 0^{|x|^{k/\epsilon}}x$. Observe that $|y| \geq |x|^{n^{k/\epsilon}}$. Thus, the above algorithm runs in $2^{|y|^\epsilon}$ steps on infinitely many y. This contradicts the DTIME(2^{n^ϵ})-bi-immunity of L. ∎

Corollary 14 *If* NP *contains a* 2^{n^ϵ}*-bi-immune language, then* NP *contains a* $\leq^{\mathrm{P}}_{2-T}$*-complete set* S *that is not* $\leq^{\mathrm{P}}_{m}$*-complete.*

The proof of the next theorem shows that we can extend the proof of Theorem 2 to show that the set S defined there is not $\leq^{\mathrm{P}}_{1-tt}$-complete. Thus, we arrive at our main result.

Theorem 15 *If* NP *contains a* 2^{n^ϵ}*-bi-immune language, then* NP *contains a* $\leq^{\mathrm{P}}_{2-T}$*-complete set* S *that is not* $\leq^{\mathrm{P}}_{1-tt}$*-complete.*

Proof. The proof is a variation of the proof of Theorem 2, and we demonstrate the interesting case only. Assume that the set S defined there is $\leq^{\mathrm{P}}_{1-tt}$-complete and let (g, h) be a 1-truth-table reduction from L_o to S. Recall that, for each string x, $g(x)$ is a query to S and that

$$x \in L_o \Leftrightarrow h(x, S(g(x))) = 1.$$

The function h on input x implicitly defines four possible truth-tables. Let us define the sets

$$\begin{aligned}
T &= \{x \mid h(x,1) = 1 \text{ and } h(x,0) = 1\},\\
F &= \{x \mid h(x,1) = 0 \text{ and } h(x,0) = 0\},\\
Y &= \{x \mid h(x,1) = 1 \text{ and } h(x,0) = 0\},\\
N &= \{x \mid h(x,1) = 0 \text{ and } h(x,0) = 1\}.
\end{aligned}$$

Each of the sets T, F, Y, and N belongs to P. Also, $T \subseteq L_o$, $F \subseteq \overline{L_o}$, for all strings $x \in Y$,

$$x \in L_o \Leftrightarrow g(x) \in S,$$

and for all strings $x \in N$,

$$x \in L_o \Leftrightarrow g(x) \in \overline{S}.$$

It follows immediately that T and F are finite sets. Now, as we did in the proof of Theorem 2, we consider the set $B = \{x \mid |x| = t_i \text{ and } i \text{ is odd}\}$. Recall that $B \in \mathrm{P}$ and that B is an infinite subset of O. For all but finitely many strings $x \in B$, either $x \in Y$ or $x \in N$. In order to illustrate the interesting case, let us assume that $B^N = B \cap N$ is infinite. Note that $B^N \in \mathrm{P}$ and that B^N is an infinite subset of O. By Lemma 1, $B^N \cap \overline{L_o}$ is infinite. For all $x \in B^N$, $x \in \overline{L_o} \Leftrightarrow x \in S$. Thus, g maps infinitely many of the strings in B^N into S. Similar to our earlier analysis, we contradict this by showing that each of the following sets is finite:

$$B_0 = \{x \in B^N \mid g(x) = 0y\},$$
$$B_1 = \{x \in B^N \mid g(x) = 1y\},$$
$$B_2 = \{x \in B^N \mid g(x) = 2y\}.$$

Here we will demonstrate that B_0 is finite. The other cases will follow similarly.

Define $A = \{x \in B_0 \mid g(x) = by, \text{ and } y \in O\}$. Again we need to show that A is a finite set, but we need a slightly different proof from that for Lemma 5. Note that $A \in \mathrm{P}$. If $g(x) = 0y \in S$, then $y \in E$. Thus, $x \in A \Rightarrow g(x) \notin S \Rightarrow x \in L_o$. Thus $A \subseteq L_o$, from which it follows that A is finite. Hence, the set

$$C = \{x \in B_0 \mid g(x) = 0y \text{ and } y \in E\}$$

is an infinite set. As earlier, we partition C into the sets

$$C_1 = \{x \in C \mid f(x) = 0y, |y| < |x|^{1/k}\},$$
$$C_2 = \{x \in C \mid f(x) = 0y, |x|^{1/k} \leq |y| < |x|^k\},$$
$$C_3 = \{x \in C \mid f(x) = 0y, |y| \geq |x|^k\},$$

and we show that each of these sets is finite. To show that C_1 is finite, we show more generally, as in the proof of Lemma 4, that $V = \{x \in B^N \mid |g(x)| < |x|^{1/k}\}$ is a finite set. (The critical fact is that for $x \in V$, $x \in S \Leftrightarrow x \in \overline{L_o} \Leftrightarrow x \notin L$, because $V \subseteq O$.) Also, it is easy to see that $C_2 = \emptyset$.

We need to show that C_3 is finite. Assume that C_3 is infinite. Noting that $C_3 \in \mathrm{P}$, the proof of Claim 11 (not Claim 9!) shows that the set $C_4 = C_3 \cap L_o$ is infinite. Then,

$$\exists^\infty x[x \in C_4, g(x) = 0y, |y| < |x|^{1/k}]$$
$$\Rightarrow$$
$$\exists^\infty x[x \in B^N, x \in L_o, y \in E, g(x) = 0y, |y| < |x|^{1/k}]$$
$$\Rightarrow$$
$$\exists^\infty y \exists x[x \in B^N, x \in L_o, y \in E, g(x) = 0y, |y| < |x|^{1/k}].$$

Thus, the set

$$U = \{y \mid \exists x[x \in B^N, x \in L_o, y \in E, g(x) = 0y, |y| < |x|^{1/k}]\}$$

is infinite. For each string $y \in U$, there exists $x \in B^N \cap L_o$ such that $g(x) = 0y$. For each such x, $g(x) = 0y \in \overline{S}$. Thus, $y \notin L_e \cup \text{PadSAT}$, so, in particular, $y \notin L_e$. However, $y \in E$, so $y \in L$. Thus, U is an infinite subset of $\overline{L}$.

Now we know that C is finite, from which it follows that B_0 is a finite set. In a similar manner we can prove that B_1 and B_2 are finite, which completes the proof of the case that B^N is infinite. The other possibility, that $B^Y = B \cap Y$ is infinite can be handled similarly. ∎

There is no previous work that indicates a separation of $\leq^{\mathrm{P}}_{2-tt}$-completeness from $\leq^{\mathrm{P}}_{1-tt}$-completeness for NP. Our next result accomplishes this, but with a stronger hypothesis.

Theorem 16 *If* NP $\cap$ co-NP *contains a* 2^{n^ϵ}*-bi-immune set, then* NP *contains a* $\leq^{\mathrm{P}}_{2-tt}$*-complete set that is not* $\leq^{\mathrm{P}}_{1-tt}$*-complete.*

Proof. The hypothesis implies the existence of a 2^{n^k}-bi-immune language L in NP $\cap$ co-NP. Let

$$S = 0(L_e \cap \text{PadSAT}) \cup 1((E - L_e) \cap \text{PadSAT}).$$

Since L belongs to NP $\cap$ co-NP, S belongs to NP. Since both PadSAT and L_e are subsets of E, for any string x

$$x \in \text{PadSAT} \Leftrightarrow (x \in L_e \cap \text{PadSAT}) \vee (x \in (E - L_e) \cap \text{PadSAT}).$$

Thus S is 2-tt-complete for NP. The rest of the proof is similar to the proof of Theorem 15. ∎

4 Acknowledgments

The authors benefited from conversations about this work with Osamu Watanabe, and with Lance Fortnow and Jack Lutz.

References

[ASB00] K. Ambos-Spies and L. Bentzien. Separating NP-completeness under strong hypotheses. *Journal of Computer and System Sciences*, 61(3):335–361, 2000.

[ASTZ97] K. Ambos-Spies, A. Terwijn, and X. Zheng. Resource bounded randomness and weakly complete problems. *Theoretical Computer Science*, 172(1):195–207, 1997.

[BDG90] J. Balcázar, J. Diaz, and J. Gabarró. *Structural Complexity II*. Springer-Verlag, Berlin, 1990.

[BS85] J. Balcázar and U. Schöning. Bi-immune sets for complexity classes. *Mathematical Systems Theory*, 18(1):1–18, June 1985.

[JK76] D. Johnson and S. Kashdan. Lower bounds for selection in $x+y$ and other multi sets. Technical Report 183, Pennsylvania State Univ., University Park, PA, 1976.

[LLS75] R. Ladner, N. Lynch, and A. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science*, 1:103–123, 1975.

[LM96] J. Lutz and E. Mayordomo. Cook versus Karp-Levin: Separating completeness notions if NP is not small. *Theoretical Computer Science*, 164:141–163, 1996.

[LY90] L. Longpré and P. Young. Cook reducibility is faster than Karp reducibility. *Journal of Computer and System Sciences*, 41:389–401, 1990.

[PS02] A. Pavan and A. Selman. Separation of NP-completeness notions. *SIAM Journal on Computing*, 31(3): 906–918, 2002.

[Sel82] A. Selman. Reductions on NP and P-selective sets. *Theoretical Computer Science*, 19:287–304, 1982.

[SG77] I. Simon and J. Gill. Polynomial reducibilities and upward diagonalizations. *Proceedings of the Ninth Annual ACM Symposium on Theory of Computing*, pages 186–194, 1977.