RELATING EQUIVALENCE AND REDUCIBILITY TO SPARSE SETS

ERIC ALLENDER[*]
DEPARTMENT OF COMPUTER SCIENCE
RUTGERS UNIVERSITY
NEW BRUNSWICK, NJ 08903

LANE A. HEMACHANDRA[†]
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ROCHESTER
ROCHESTER, NY 14627

MITSUNORI OGIWARA[‡]
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ELECTRO-COMMUNICATIONS
TOKYO 182

OSAMU WATANABE
DEPARTMENT OF COMPUTER SCIENCE
TOKYO INSTITUTE OF TECHNOLOGY
TOKYO 152

Abstract. For various polynomial-time reducibilities r, this paper asks whether being r-reducible to a sparse set is a broader notion than being r-equivalent to a sparse set. Although distinguishing equivalence and reducibility to sparse sets, for many-one or 1-truth-table reductions, would imply that $\mathrm{P} \neq \mathrm{NP}$, this paper shows that for k-truth-table reductions, $k \geq 2$, equivalence and reducibility to sparse sets provably differ. Though Gavaldà and Watanabe have shown that, for any polynomial-time computable unbounded function $f(\cdot)$, some sets $f(n)$-truth-table reducible to sparse sets are not even Turing equivalent to sparse sets, this paper shows that extending their result to the 2-truth-table case would provide a proof that $\mathrm{P} \neq \mathrm{NP}$. Additionally, this paper studies the relative power of different notions of reducibility, and proves that disjunctive and conjunctive truth-table reductions to sparse sets are surprisingly powerful, refuting a conjecture of Ko.

[*] RESEARCH SUPPORTED IN PART BY THE NATIONAL SCIENCE FOUNDATION UNDER GRANT CCR-9000045 AND BY THE INTERNATIONAL INFORMATION SCIENCE FOUNDATION UNDER GRANT 90-1-3-227.

[†] RESEARCH SUPPORTED IN PART BY THE NATIONAL SCIENCE FOUNDATION UNDER GRANTS CCR-8996198 AND CCR-8957604, AND BY THE INTERNATIONAL INFORMATION SCIENCE FOUNDATION UNDER GRANT 90-1-3-228.

[‡] RESEARCH DONE IN PART WHILE AT THE TOKYO INSTITUTE OF TECHNOLOGY.

1. Introduction

Computer science is the study of information—coding information, decoding information, organizing information, and accessing information. Sets whose information content is small, intuitively the structurally simplest of sets, have played a central rôle in the development of the theory of computing. Sparse sets—sets with at most polynomially many elements of each length—are one natural notion of "sets of small information content," and, indeed, sparse sets have been essential to recent advances in computational complexity theory ([15], see also [24,25]).

However, in complexity theory it is common to investigate notions sufficiently robust so as to be invariant under polynomial-time reductions. Thus, an even more natural notion of "small information content" is $\mathrm{R}^p_T(\mathrm{SPARSE})$, the class of sets that polynomial-time Turing reduce to sparse sets.[1] The sense in which $\mathrm{R}^p_T(\mathrm{SPARSE})$ sets are of small information content can be crisply formalized: $\mathrm{R}^p_T(\mathrm{SPARSE})$ is precisely the class—more commonly referred to as P/poly—of sets having polynomial-sized (nonuniform) circuits (Meyer, see [7]).

$\mathrm{R}^p_T(\mathrm{SPARSE})$ has been intensely studied, both in terms of the question "NP $\subseteq$ $\mathrm{R}^p_T(\mathrm{SPARSE})$?" [19,16,10,18], and in terms of the robustness of $\mathrm{R}^p_T(\mathrm{SPARSE})$. $\mathrm{R}^p_T(\mathrm{SPARSE})$ is indeed quite robust; in addition to its characterization in terms of small circuits, $\mathrm{R}^p_T(\mathrm{SPARSE})$ is easily noted equivalent to $\mathrm{R}^p_T(\mathrm{TALLY})$, $\mathrm{R}^p_{tt}(\mathrm{TALLY})$, and $\mathrm{R}^p_{tt}(\mathrm{SPARSE})$ (see [6]). Nonetheless, Book and Ko showed that there were limits to the robustness of $\mathrm{R}^p_T(\mathrm{SPARSE})$; they initiated the study of the classes of languages reducible to sparse (and tally) sets under various weak notions of polynomial-time reducibility, and proved that such classes differed both from $\mathrm{R}^p_T(\mathrm{SPARSE})$ and from each other ([9], see also the earlier related work on two subclasses of $\mathrm{R}^p_T(\mathrm{SPARSE})$: "almost polynomial time" [26] and the P-close sets [32,28]).

Tang and Book initiated an analogous study of the classes of languages *equivalent* to sparse (and tally) sets under various notions of polynomial-time reducibility, and proved that in many cases such classes differed from each other [31]. Additionally, equivalence has been used by Balcázar and Book to characterize completely a natural subset of $\mathrm{R}^p_T(\mathrm{SPARSE})$, namely the sets with self-producible circuits [4].

[1] Though formal definitions will be given in Section 2, it is useful to introduce some notation here. For a given reducibility $\leq^p_r$, we define: (1) $\mathrm{R}^p_r(\mathrm{SPARSE})$ as the class of sets L such that, for some sparse set S, $L \leq^p_r S$, and (2) $\mathrm{R}^p_r(\mathrm{TALLY})$ as the class of sets L such that, for some tally set T, $L \leq^p_r T$.

The study of equivalence to sparse sets and the study of reducibility to sparse sets have each yielded a flurry of results [9,31,10,16,18,20,3,1,21]. Nonetheless, many of the most basic questions have remained unanswered and, in some cases, unasked.

In particular, the relationships between equivalence and reducibility to sparse sets have remained wholly unknown. The first results along this line are those of the present paper and the companion paper of Gavaldà and Watanabe (see Theorem C below). The present paper asks, for the case of bounded truth-table reductions, whether reducibility to sparse sets is a broader notion than equivalence to sparse sets. We provide answers to this question and indicate some areas in which further progress is unlikely until longstanding open problems in complexity theory are resolved. Among our results are the following.[2]

Theorem A $\mathrm{R}^p_{2-tt}(\mathrm{SPARSE}) \not\subseteq \mathrm{E}^p_{f(n)-tt}(\mathrm{SPARSE})$*, for any* $f(n) = n^{o(1)}$.

Theorem B *If* $\mathrm{P} = \mathrm{NP}$ *then* $\mathrm{R}^p_{1-tt}(\mathrm{SPARSE}) = \mathrm{E}^p_{1-tt}(\mathrm{SPARSE})$*. If* $\mathrm{P} = \mathrm{NP}$ *then* $\mathrm{R}^p_m(\mathrm{SPARSE}) = \mathrm{E}^p_m(\mathrm{SPARSE}) \bigcup \{\Sigma^*\}$.

Theorem A implies that reducibility and equivalence to sparse sets differ sharply for $\leq^p_{2-tt}$ reductions. In contrast, Theorem B indicates that proving an analogous result for $\leq^p_{1-tt}$ or $\leq^p_m$ reductions would involve proving $\mathrm{P} \neq \mathrm{NP}$. Theorem A raises the issue of the strength of reducibility that will suffice to provide equivalence to some sparse set for sets bounded truth-table equivalent to sparse sets. That is, what is the cost paid—in terms of increase in strength of reduction—to achieve equivalence? Gavaldà and Watanabe have shown that for truth-table reductions whose number of queries is unbounded, an extremely heavy price is exacted. We show that the Gavaldà-Watanabe result cannot be extended to the 2-truth-table case without providing a proof that $\mathrm{P} \neq \mathrm{NP}$.

Theorem C [13] *Let* $f(n)$ *be any unbounded polynomial-time computable function. Then* $\mathrm{R}^p_{f(n)-tt}(\mathrm{SPARSE}) \not\subseteq \mathrm{E}^p_T(\mathrm{SPARSE})$.

Theorem D *Let* $\epsilon > 0$*. If* $\mathrm{P} = \mathrm{NP}$ *then* $\mathrm{R}^p_{2-tt}(\mathrm{SPARSE}) \subseteq \mathrm{E}^p_{n^\epsilon-tt}(\mathrm{SPARSE})$.

Theorem D shows that Theorem A is optimal, and provides an upper bound, conditioned upon the assumption that $\mathrm{P} = \mathrm{NP}$, for the complexity of equivalence. We also provide unconditional upper bounds.

Finally, we turn to one of the key open questions about the structure of $\mathrm{R}^p_r(\mathrm{SPARSE})$

[2] Notation: given a notion of reducibility $\leq^p_r$, we define: (1) $\mathrm{E}^p_r(\mathrm{SPARSE})$ as the class of sets L such that, for some sparse set S, $L \leq^p_r S$ and $S \leq^p_r L$, and (2) $\mathrm{E}^p_r(\mathrm{TALLY})$ as the class of sets L such that, for some tally set T, $L \leq^p_r T$ and $T \leq^p_r L$. An $f(n)$–tt reduction is a truth-table reduction that, on inputs of size n, generates at most $f(n)$ queries. See Section 2 for full definitions of these and other notions.

classes. Ko [20, p. 65] conjectures that for each $k > 0$ it holds that $\mathrm{R}^p_{k-tt}(\mathrm{SPARSE}) \not\subseteq \mathrm{R}^p_{dtt}(\mathrm{SPARSE})$. We refute this conjecture by proving the following result.

Theorem E $\mathrm{R}^p_{btt}(\mathrm{SPARSE}) \subseteq \mathrm{R}^p_{dtt}(\mathrm{SPARSE})$.

This may be interpreted as saying that disjunctive truth-table reductions to sparse sets are surprisingly powerful. We prove related results showing that the power of conjunctive truth-table and many-one reductions, in the nondeterministic model of Ladner, Lynch, and Selman [22], is also substantial.

Theorem F

1. $\mathrm{R}^{NP}_m(\mathrm{SPARSE}) = \mathrm{R}^{NP}_{btt}(\mathrm{SPARSE})$.
2. $\mathrm{R}^{NP}_{ctt}(\mathrm{SPARSE}) = \mathrm{R}^{NP}_{T}(\mathrm{SPARSE})$.

The paper is organized as follows. Section 2 reviews notations and definitions. Section 3 studies the relationship between reductions and equivalences for the cases of many-one and 1-truth-table reducibilities. Section 4 studies the case of k-truth-table reductions, $k \geq 2$—a case that differs sharply from those of Section 3. Section 5 investigates the interrelationships between reducibility classes and their seemingly restrictive (but surprisingly powerful) disjunctive and conjunctive versions. Section 6 presents open problems and conclusions.

2. Preliminaries

Let Σ be a fixed finite alphabet. Let $|y|$ denote the length of string $y \in \Sigma^*$, and let $||S||$ denote the cardinality of set $S \subseteq \Sigma^*$. Let $X \triangle Y$ denote $(X - Y) \bigcup (Y - X)$. For a set T, we define $T^{=n} = \{y \mid y \in T \text{ and } |y| = n\}$ and $T^{\leq n} = \{y \mid y \in T \text{ and } |y| \leq n\}$.

Let $\langle \cdot, \cdot \rangle_2$ denote a pairing function over finite strings (equivalently, over $\{0, 1, \ldots\}$ via the standard correspondence between strings and natural numbers), with the standard nice computability, invertibility, and other properties (e.g., for all $a, b \in \Sigma^*$ it holds that $|\langle a, b \rangle_2| = |a| + |b|$). Let $\langle y \rangle$ denote $\langle 1, y \rangle_2$, and, for every $k \geq 2$, let $\langle y_1, y_2, \ldots, y_k \rangle$ denote $\langle k, \langle y_1, \langle y_2, \langle \ldots, \langle y_{k-1}, y_k \rangle_2 \ldots \rangle_2 \rangle_2 \rangle_2 \rangle_2$. This function is polynomial-time computable and polynomial-time invertible, and unambiguously codes a variable number of arguments.[3]

[3] For notational convenience, when speaking of functions f of more than one argument, we will freely write $f(y_1, \ldots)$ as a shorthand for $f(\langle y_1, \ldots \rangle)$.

We adopt the standard notions of reducibility, as introduced by Ladner, Lynch, and Selman. (We make slight alterations in the definitions; these alterations do not effect the reductions defined.)

DEFINITION 2.1. [**22**]

1. *A* tt-condition *is a finite string of the form $\langle y_1, y_2, \ldots\rangle$, where each y_i is a member of Σ^*.*

2. *A* tt-condition generator *is a recursive mapping from Σ^* into the set of tt-conditions.*

3. *A* tt-condition evaluator *is a recursive mapping from $\{\langle x, \sigma_1, \cdots, \sigma_n\rangle \mid x \in \Sigma^*$ and $(\forall i)[\sigma_i \in \{0, 1\}]\}$ into $\{0, 1\}$. (We will use the convention of Footnote 3 often on arguments of tt-condition evaluators.)*

4. *Let e be a tt-condition evaluator. A tt-condition $\langle y_1, \ldots.., y_k\rangle$ is* e-satisfied on input x by B $\subseteq \Sigma^*$ *if and only if $e(x, \chi_B(y_1), \ldots, \chi_B(y_k)) = 1$.*

5. *Let g be a tt-condition generator and e be a tt-evaluator. Let $\lambda = (g, e)$. We will say that $\lambda^S(x)$* accepts *if the tt-condition generated by $g(x)$ is e-satisfied on input x by S.*

6. *We say that $A \leq_{tt}^{p} B$ if there exist a polynomial-time computable generator g and a polynomial-time computable evaluator e such that, for all x, $x \in A \iff g(x)$ is e-satisfied on input x by B. If $A \leq_{tt}^{p} B$ we say that A is truth-table reducible to B in polynomial time.*

7. *We say that $A \leq_{f(n)-tt}^{p} B$ if $A \leq_{tt}^{p} B$ via a generator g and evaluator e such that on each input x, it holds that $g(x) \in \{\langle y_1, \ldots, y_k\rangle \mid y_i \in \Sigma^*$ and $k \leq f(|x|)\}$. For the special cases $\leq_{k-tt}^{p}$—that is, $f(n) = k$, $k \in \{1, 2, \ldots\}$—we will without loss of generality assume that the evaluator asks* exactly *k questions.*

8. *We say that $A \leq_{btt}^{p} B$ (A is polynomial-time bounded truth-table reducible to B) if $A \leq_{k-tt}^{p} B$ for some k.*

9. *We say that $A \leq_{ctt}^{p} B$ (A is polynomial-time conjunctive truth-table reducible to B) if $A \leq_{tt}^{p} B$ via a generator g and evaluator e such that the evaluator e has the property that for every x, $e(x, \sigma_1, \ldots, \sigma_k) = 1 \iff (\forall i : 1 \leq i \leq k)[\sigma_i = 1]$.*

10. *We say that $A \leq^p_{dtt} B$ (A is polynomial-time disjunctive truth-table reducible to B) if $A \leq^p_{tt} B$ via a generator g and evaluator e such that the evaluator e has the property that for every x, $e(x, \sigma_1, \ldots, \sigma_k) = 0 \iff (\forall i : 1 \leq i \leq k)[\sigma_i = 0]$.*

In addition to the above reductions, we will also be concerned with nondeterministic reduction types. We defer the definitions of such reductions to Section 5.

Having defined the above types of reductions, we can now speak of the class of sets reducible or equivalent to a certain class of sets via a certain type of reduction. Such notions were first investigated in a systematic way by Book and Ko [9] and Tang and Book [31]. We modify their nomenclature to allow a uniform notation for all reduction types.

DEFINITION 2.2.

1. *Let $\mathcal{C}$ be a class of sets and let $\leq^t_r$ be a reducibility. We define $\mathrm{R}^t_r(\mathcal{C}) = \{A \,\Big|\, (\exists B)[B \in \mathcal{C}$ and $A \leq^t_r B]\}$.*

2. *Let $\mathcal{C}$ be a class of sets and let $\leq^t_r$ be a reducibility. We define $\mathrm{E}^t_r(\mathcal{C}) = \{A \,\Big|\, (\exists B)[B \in \mathcal{C}$ and $A \leq^t_r B$ and $B \leq^t_r A]\}$.*

3. Many-One and 1-Truth-Table Reductions

We first note that, if P = NP, then all sets many-one reducible to sparse sets are in fact many-one equivalent to sparse sets.

THEOREM 3.1. *If* P = NP *then* $\mathrm{R}^p_m(\mathrm{SPARSE}) = \mathrm{E}^p_m(\mathrm{SPARSE}) \bigcup \{\Sigma^*\}$.

Proof: Suppose $L \leq^p_m S$, S sparse, via many-one reduction $g(\cdot)$, and $L \neq \Sigma^*$. Define $S' = \{\langle 0^l, x\rangle \,\Big|\, x \in S$ and $(\exists y)[y \in L$ and $|y| = l$ and $g(y) = x]\}$ (see [23] for a similar "multiple-copy" approach). First, note that $L \leq^p_m S'$, as $y \in L \iff \langle 0^{|y|}, g(y)\rangle \in S'$. Second, note that $S' \leq^p_m L$ if P = NP. This is because, when asked whether $\langle 0^l, x\rangle \in S'$, we may use the fact that P = NP to determine whether there exists a y such that $|y| = l$ and $g(y) = x$. If not, reject $\langle 0^l, x\rangle$ by mapping onto an element out of L. If so, use the P = NP assumption to find one such y, call it y', and map to asking whether $y' \in L$. Finally, note that, immediately from the definition of S' and the fact that S is sparse, that S' is sparse. Thus, $L \in \mathrm{E}^p_m(\mathrm{SPARSE})$. ▮

The proof of Theorem 3.1 can easily be modified to the case of 1-truth-table reductions. We need only change the definition of S' to $S' = \{\langle 0^l, x\rangle \,\Big|$ either (1) $(\exists y)[y \in L$ and $|y| = l$

and the truth-table for input y accepts if and only if $x \in S$], or (2) $(\exists y)[y \notin L$ and $|y| = l$ and the truth-table for input y accepts if and only if $x \notin S]\}$.

THEOREM 3.2. *If* P = NP *then* $\mathrm{R}^p_{1-tt}(\mathrm{SPARSE}) = \mathrm{E}^p_{1-tt}(\mathrm{SPARSE})$.

We say that a truth-table reduction, with truth-table condition generator g, is *honest* if there exists a polynomial $q(\cdot)$ such that whenever y is one of the query strings generated by $g(x)$, it holds that $q(|y|) \geq |x|$. Theorem 3.2 in fact gives honest equivalence.

Note that Theorem 3.1 does not establish that $\mathrm{R}^p_m(\mathrm{SPARSE}) = \mathrm{E}^p_m(\mathrm{SPARSE}) \bigcup \{\Sigma^*\}$ is equivalent to the claim that P=NP. We now note that analogous questions about tally sets are indeed equivalent to important open questions in complexity theory.

First, we present some definitions. A function f is *weakly invertible* if there is a polynomial-time computable function h such that $f(h(x)) = x$ for all $x \in range(f)$. Let E denote $\bigcup_{k \geq 0} \mathrm{DTIME}[2^{kn}]$, and let NE denote $\bigcup_{k \geq 0} \mathrm{NTIME}[2^{kn}]$.

It is shown in [3] that the following are equivalent:

1. Every NE predicate is E-solvable.
2. Every honest polynomial-time computable function $f : \Sigma^* \to 0^*$ is weakly invertible.
3. $\mathrm{E}^p_m(\mathrm{TALLY}) \bigcup \{\Sigma^*\} = \mathrm{E}^p_{1-tt}(\mathrm{TALLY})$.
4. $\mathrm{E}^p_m(\mathrm{TALLY}) \bigcup \{\Sigma^*\} = \mathrm{E}^p_{btt}(\mathrm{TALLY})$.

Condition 1 above is the natural "witness-finding" analog of the E = NE question. Impagliazzo and Tardos [17] have recently shown that there are relativized worlds in which Condition 1 fails to hold, yet E = NE. Their work provides a relativized refutation of a conjecture of Sewelson [30], whose thesis forms the protasis of the [3,17] research stream.

We note that the above equivalence can be extended to include classes of the form $\mathrm{R}^p_{btt}(\mathrm{TALLY})$, and, equivalently, $\mathrm{R}^p_m(\mathrm{TALLY})$. The following result, alluded to in [2], was observed independently by Fu Bin [8].

THEOREM 3.3. *Every NE predicate is E-solvable if and only if* $\mathrm{R}^p_{btt}(\mathrm{TALLY}) = \mathrm{E}^p_m(\mathrm{TALLY}) \bigcup \{\Sigma^*\}$.

Proof Under the assumption that $\mathrm{R}^p_{btt}(\mathrm{TALLY}) = \mathrm{E}^p_m(\mathrm{TALLY}) \bigcup \{\Sigma^*\}$, it follows immediately that $\mathrm{R}^p_{btt}(\mathrm{TALLY}) \subseteq \mathrm{E}^p_m(\mathrm{TALLY}) \bigcup \{\Sigma^*\} \subseteq \mathrm{E}^p_{btt}(\mathrm{TALLY}) \subseteq \mathrm{R}^p_{btt}(\mathrm{TALLY})$. Thus, $\mathrm{E}^p_m(\mathrm{TALLY}) \bigcup \{\Sigma^*\} = \mathrm{E}^p_{btt}(\mathrm{TALLY})$, and by the result of [3] mentioned above, it follows that every NE predicate is E-solvable.

Conversely, assume, via the above-mentioned equivalence of [3], that every honest polynomial-time computable function $f : \Sigma^* \to 0^*$ is weakly invertible, and let $L \leq_m^p T$, for some tally set T, via many-one reduction $g(\cdot)$. As in the proof of Theorem 3.1, define $T' = \{0^{\langle l, i+1\rangle} \mid 0^i \in T \text{ and } (\exists y)[y \in L \text{ and } |y| = l \text{ and } g(y) = 0^i]\}$. Then the function f defined by:

$$f(x) = \begin{cases} 0^{\langle |x|, i\rangle} & \text{if } g(x) = 0^i \\ 0^{\langle |x|, 0\rangle} & \text{if } g(x) \not\in \{0^i \mid i \geq 0\} \end{cases}$$

is a many-one reduction from L to T'. Furthermore, under the assumption that f is weakly invertible (and assuming that $L \neq \Sigma^*$), it is easy to see that $T' \leq_m^p L$. Thus, under this assumption, $\mathrm{R}_m^p(\mathrm{TALLY}) = \mathrm{E}_m^p(\mathrm{TALLY}) \bigcup \{\Sigma^*\}$, and thus—via the fact that $\mathrm{R}_m^p(\mathrm{TALLY}) = \mathrm{R}_{btt}^p(\mathrm{TALLY})$ [9]—it holds that $\mathrm{R}_{btt}^p(\mathrm{TALLY}) = \mathrm{E}_m^p(\mathrm{TALLY}) \bigcup \{\Sigma^*\}$. ▮

4. Bounded-Truth-Table Reductions

4.1. A Lower Bound

Gavaldà and Watanabe have proven that for any unbounded polynomial-time computable function $f(n)$, $\mathrm{R}_{f(n)-tt}^p(\mathrm{SPARSE}) \not\subseteq \mathrm{E}_T^p(\mathrm{SPARSE})$ [13]; their techniques do not seem to apply to the classes of sets reducible to sparse sets via $\leq_{k-tt}^p$ reductions, for constants k. However, Theorem 4.1 establishes, for the case of bounded truth-table reductions, a wide separation between reducibility and equivalence.

THEOREM 4.1. *Let $h(n) = n^{o(1)}$. Then $\mathrm{R}_{2-tt}^p(\mathrm{SPARSE}) \not\subseteq \mathrm{E}_{h(n)-tt}^p(\mathrm{SPARSE})$.*

Proof: For the purposes of this proof, we change our assumptions about the pairing function, and now assume that one of the properties of the pairing function $\langle \cdot, \cdot \rangle_2$ of Section 2 is that $(\forall x, y)[|\langle x, y\rangle_2| = 2|x| + |y|]$. Note that in this proof (and only in this proof) we will use both our standard pairing function $\langle \cdots \rangle$ and its constituent function $\langle \cdot, \cdot \rangle_2$.

Let us define an operator A such that, for any set S, $A(S) = \{\langle x, y\rangle_2 \mid |x| = |y| \text{ and } x \text{ lexicographically precedes } y \text{ and } (x \in S \text{ or } y \in S)\}$. Note that for any sparse set S, $A(S) \in \mathrm{R}_{2-tt}^p(\mathrm{SPARSE})$. We will construct a sparse set S so that $A(S) \not\in \mathrm{E}_{h(n)-tt}^p(\mathrm{SPARSE})$.

In the following, for each $k \geq 1$, we use p_k to denote the polynomial $n^k + k$. Consider some enumeration $\{f_k\}_{k \geq 1}$ of $\leq_{h(n)-tt}^p$ reductions; without loss of generality, we may assume that, for all $k \geq 1$ and $x \in \Sigma^*$, the length of queries asked by $f_k(x)$ is bounded by $p_k(|x|)$. Let $C_{\langle i,\,j,\,l\rangle}$ denote the condition that, for each set W with census function bounded by p_l, either f_i is not a $\leq^p_{h(n)-tt}$ reduction from $A(S)$ to W, or f_j is not a $\leq^p_{h(n)-tt}$ reduction from W to $A(S)$.

Let us introduce some notations so that we may state the condition $C_{\langle i,\,j,\,l\rangle}$ more precisely. For any polynomial p, we say that set L is *p-sparse* if the census of L is bounded by p. For any set L, let $f_i^{-1}(L)$ denote the set $\{x \,\big|\,$ the truth-table condition of $f_i(x)$ evaluates to true when given L as the oracle$\}$. Then we can now restate $C_{\langle i,\,j,\,l\rangle}$ as the disjunction of the following two conditions:

I: $f_j^{-1}(A(S))$ is not p_l-sparse.
II: $A(S) \neq f_i^{-1}(f_j^{-1}(A(S)))$; that is, some v exists such that
$v \in A(S) \iff v \notin f_i^{-1}(f_j^{-1}(A(S)))$.

We will build our set S in stages, where stage $\langle i,j,l\rangle$ will guarantee that $C_{\langle i,\,j,\,l\rangle}$ is satisfied. (Initially S is the empty set.) Note that this suffices to prove that S has the desired properties.

Stage $\langle i,j,l\rangle$:
Choose n large enough so that:
(i) interference with previous stages is avoided,
(ii) $(2^n/(2p_l \circ p_i(3n))^{h(3n)}) - h(3n)h(p_i(3n)) - 1 > 0$, and
(iii) $h(p_i(3n)) < n$.
(Note that such an n always exists since $h(n) = n^{o(1)}$.)
Case I: If there is a set $D \subseteq \Sigma^{=n}$, $||D|| \leq h(3n)h(p_i(3n)) + 1$, such that $f_j^{-1}(A(S \bigcup D))$ is not p_l-sparse, then set S to $S \bigcup D$.
Case II: If there is a set $D \subseteq \Sigma^{=n}$, $||D|| \leq h(3n)h(p_i(3n)) + 1$, such that $A(S \bigcup D) \neq f_i^{-1}(f_j^{-1}(A(S \bigcup D)))$, then set S to $S \bigcup D$.
(The construction fails if neither Case I nor Case II holds.)

If the above construction is completed, then the constructed set S clearly satisfies our purpose, that is, S is sparse and satisfies condition $C_{\langle i,\,j,\,l\rangle}$ for every $\langle i,j,l\rangle$. Thus, it remains only to show that the construction can be completed. That is, if Case I fails, then Case II must hold.

Consider any stage $\langle i,j,l\rangle$ and any sufficiently large n such that Case I does not hold. For such $\langle i,j,l\rangle$ and n, we show that Case II holds with some D. In the following discussion, let i, j, l, and n be fixed; let h denote $h(3n)$, and let h' denote $h(p_i(3n))$.

This paragraph gives an informal overview of the proof, in order to make the construction easier to understand. If Case I and Case II both fail, then there are sparse sets (call them W_1 and W_2) such that the following hold:

1. f_i is a $\leq^p_{h-tt}$ reduction from $A(S)$ to W_1 and f_j is a $\leq^p_{h-tt}$ reduction from W_1 to $A(S)$.

2. f_i is a $\leq^p_{h-tt}$ reduction from $A(S \bigcup \{0^n\})$ to W_2 and f_j is a $\leq^p_{h-tt}$ reduction from W_2 to $A(S \bigcup \{0^n\})$.

That is, only a small number of strings ($W_1 \cup W_2$) are sensitive to the presence or absence of 0^n in the set we are constructing. It follows that there is some string, w_1, that is queried by a large fraction of the strings in the set $\{\langle 0^n, y\rangle_2 \mid y \in \Sigma^*\}$ (recall that at this point, n is fixed). Thus w_1 may be thought of as being "influential" in some sense, and we can define Y_1 to be the (large) set of strings that are influenced by w_1. Let Z_1 be the (small) set of strings queried by the reduction f_j on input w_1. By setting membership for all the strings in Z_1, we completely determine membership for w_1, which means that there must be some string w_2 and some large subset $Y_2 \subseteq Y_1$ such that w_2 influences Y_2. We continue in this way until we arrive at a non-empty set of strings, each of which is influenced by (and thus queries) $x_1, x_2, \ldots, x_{h+1}$. But this is a contradiction, since no string makes more than h queries. This informal argument is made precise below.

We construct sets D_1, ..., D_{k_0}, D_1^+, ..., $D_{k_0}^+$ so that either D_{k_0} or $D_{k_0}^+$ satisfies Case II. The construction proceeds as follows:

Basis:

Set $Y_0 = \Sigma^{=n} - \{0^n\}$, $D_0 = \emptyset$, and $Z_0 = \emptyset$.

Definition of D_k and D_k^+ $(1 \leq k)$:

Set $D_k = D_{k-1} \bigcup Z_{k-1}$, $D_k^+ = D_k \bigcup \{0^n\}$.

Set $A_k = A(S \bigcup D_k)$, $A_k^+ = A(S \bigcup D_k^+)$.

For each $w \in \Sigma^*$, set $Q_k(w) = \{y \in Y_{k-1} \mid f_i(\langle 0^n, y\rangle_2)$ queries $w\}$.

Set $C_k = \{w \mid Q_k(w) \neq \emptyset$ and $w \in f_j^{-1}(A_k) \triangle f_j^{-1}(A_k^+)\}$;

if C_k is empty, then terminate the construction.

Set w_k to be a string in C_k such that $||Q_k(w_k)|| \geq ||Y_{k-1}||/2p_l \circ p_i(3n)$;

if such w_k does not exist, then terminate the construction.

Set $Z_k = \{z \in \Sigma^{=n} - \{0^n\} \mid f_j(w_k) \text{ queries } \langle 0^n, z\rangle_2\}$.
Set $Y_k = Q_k(w_k) - Z_k$.

Now we show, in the following claims, that the construction terminates at some k_0, $1 \le k_0 \le h+1$. Note that the construction terminates either because C_{k_0} is empty or because no w_{k_0} exists. For each case, we prove that either $A_{k_0} \neq f_i^{-1}(f_j^{-1}(A_{k_0}))$ or $A^+_{k_0} \neq f_i^{-1}(f_j^{-1}(A^+_{k_0}))$ occurs; that is, either D_{k_0} or $D^+_{k_0}$ satisfies Case II.

Before going further, let us explain the purpose of each of the sets in this construction. For each k, Y_k is a (large) set of strings that queries each of $\{w_1, w_2, \ldots, w_k\}$. Z_k is the set of strings queried by w_k, and $D_k = \bigcup_{s \le k} Z_s$. (The strings $w_1, w_2, \ldots$ are chosen to be "influential," and the sets D_k are constructed so as to eliminate the influence of these strings.) D^+_k is just $D_k \cup \{0^n\}$, and the sets A_k and A^+_k are constructed from D_k and D^+_k using the $A(\cdot)$ operator. C_k is the set of those strings that are sensitive to the difference between A_k and A^+_k, under reduction f_j.

Claim 1 states some properties that are immediate from the construction; its proof is omitted.

Claim 1

(1) For any k, $1 \le k \le h+1$, such that A_k and A^+_k are defined:

(a) $||D_k|| \le ||D^+_k|| \le (k-1)h' + 1$,

(b) $\{\langle 0^n, z\rangle_2 \mid z \in \bigcup_{1 \le s < k} Z_s\} \subseteq A_k \subseteq A^+_k$,

(c) $||Y_{k-1}|| \ge (2^n/(2p_l \circ p_i(3n))^{k-1}) - (k-1)h' - 1 > 0$, and

(d) $A^+_k - A_k \supseteq \{\langle 0^n, y\rangle_2 \mid y \in Y_{k-1}\} \neq \emptyset$.

(2) Let k, $1 \le k \le h+1$, be any index such that Y_k is defined. For every $y \in Y_k$, $f_i(\langle 0^n, y\rangle_2)$ queries w_1, ..., w_k.

The set C_k is the set of strings w such that (i) w is queried by $f_i(\langle 0^n, y\rangle_2)$ for some $y \in Y_{k-1}$, and (ii) $f_j(w)$ evaluates differently between oracle A_k and A^+_k. The following property of C_k is central to our construction.

Claim 2 Let k, $1 \le k \le h+1$, be any index such that C_k is defined. For every s, $1 \le s < k$, $w_s \notin C_k$.

Proof of Claim 2 Note that every $\langle 0^n, z\rangle_2$ (except $\langle 0^n, 0^n\rangle_2$) that is queried by $f_j(w_s)$ is in A_k. Thus, the truth-table value of $f_j(w_s)$ does not vary by changing the oracle from A_k to A_k^+.

End of Proof of Claim 2 ▮

Claim 3 Suppose that the construction does not terminate at h. Then C_{h+1} is empty; thus, the construction terminates at most at $h+1$.

Proof of Claim 3 It follows from Claim 1, Part (2), that for every $y \in Y_h$, $f_i(\langle 0^n, y\rangle_2)$ queries w_1, ..., w_h. Since f_i is a $\leq_{h-tt}^p$ reduction, $\{w_1, ..., w_h\}$ is exactly the set of queries that are asked by $f_i(\langle 0^n, y\rangle_2)$ for some $y \in Y_h$. (Recall that $|\langle x, y\rangle_2| = 3n$ whenever $x \in \Sigma^{=n}$ and $y \in \Sigma^{=n}$.) Thus, $C_{h+1} \subseteq \{w_1, ..., w_h\}$. On the other hand, from Claim 2, none of w_1, ..., w_h belongs to C_{h+1}. Therefore, $C_{h+1} = \emptyset$.

End of Proof of Claim 3 ▮

Claim 4 Suppose that the construction terminates at k_0. Then either $A_{k_0} \neq f_i^{-1}(f_j^{-1}(A_{k_0}))$ or $A_{k_0}^+ \neq f_i^{-1}(f_j^{-1}(A_{k_0}^+))$.

Proof of Claim 4 First suppose that the construction terminates at k_0 because $C_{k_0} = \emptyset$. Since $k_0 \leq h+1$, it follows from Claim 1, Part (1.d), that $\langle 0^n, y_0\rangle_2 \in A_{k_0}^+ - A_{k_0}$ for some $y_0 \in Y_{k_0-1}$. On the other hand, since $C_{k_0} = \emptyset$, the truth-table values of $f_i(\langle 0^n, y_0\rangle_2)$ relative to $f_j^{-1}(A_{k_0})$ and $f_j^{-1}(A_{k_0}^+)$ are the same. Hence, either $A_{k_0} \neq f_i^{-1}(f_j^{-1}(A_{k_0}))$ or $A_{k_0}^+ \neq f_i^{-1}(f_j^{-1}(A_{k_0}^+))$.

Next we show that if no w_{k_0} exists (and thus the construction terminates at k_0), then either $A_{k_0} \neq f_i^{-1}(f_j^{-1}(A_{k_0}))$ or $A_{k_0}^+ \neq f_i^{-1}(f_j^{-1}(A_{k_0}^+))$. We prove the contrapositive, i.e., for any k, $1 \leq k \leq h$, if $A_k = f_i^{-1}(f_j^{-1}(A_k))$ and $A_k^+ = f_i^{-1}(f_j^{-1}(A_k^+))$, then w_k certainly exists.

We show that $Y_{k-1} = \{z \mid (\exists w \in C_k)[z \in Q_k(w)]\}$ and $||C_k|| \leq 2p_l \circ p_i(3n)$, thereby proving that some $w_k \in C_k$ exists such that $||Q_k(w_k)|| \geq ||Y_{k-1}||/2p_l \circ p_i(3n)$.

Consider any $y \in Y_{k-1}$. Since $\langle 0^n, y\rangle_2$ is in $A_k \triangle A_k^+$ (from Claim 1, Part (1.d)), and $A_k = f_i^{-1}(f_j^{-1}(A_k))$ and $A_k^+ = f_i^{-1}(f_j^{-1}(A_k^+))$ (from the assumption), $f_i(\langle 0^n, y\rangle_2)$ must query some w_y that is in $f_j^{-1}(A_k) \triangle f_j^{-1}(A_k^+)$. Recall that C_k is the set of strings in $f_j^{-1}(A_k) \triangle f_j^{-1}(A_k^+)$ that are queried by $f_i(\langle 0^n, y\rangle_2)$ for some $y \in Y_{k-1}$. Hence, w_y is in C_k. Thus, for each $y \in Y_{k-1}$, there is some $w_y \in C_k$ such that $f_i(\langle 0^n, y\rangle_2)$ queries w_y, i.e., $y \in Q_k(w_y)$; in other words, $Y_{k-1} = \{z \mid (\exists w \in C_k)[z \in Q_k(w)]\}$.

Recall that we are assuming that $f_j^{-1}(A(S \bigcup D))$ is p_l-sparse for any $D \subseteq \Sigma^{=n}$, $||D|| \leq hh'+1$. Hence, both $f_j^{-1}(A_k)$ and $f_j^{-1}(A_k^+)$ are p_l-sparse; then clearly $C_k \subseteq f_j^{-1}(A_k) \triangle f_j^{-1}(A_k^+)$ is $2p_l$-sparse. Note that each $w \in C_k$ is queried by $f_i(\langle 0^n, y\rangle_2)$ for some $y \in \Sigma^{=n}$ and that the length of such a string is bounded by $p_i(|\langle 0^n, y\rangle_2|) \leq p_i(3n)$. (Recall that $|\langle x, y\rangle_2| = 3n$ for every x and $y \in \Sigma^{=n}$.) Thus, $||C_k|| \leq 2p_l(p_i(3n))$.

End of Proof of Claim 4 ▮

End of Proof of Theorem 4.1 ▮

The following is an immediate corollary.

COROLLARY 4.2. *For every $k \geq 2$, it holds that* $\mathrm{R}^p_{k-tt}(\mathrm{SPARSE}) \neq \mathrm{E}^p_{k-tt}(\mathrm{SPARSE})$.

4.2. Upper Bounds

Corollary 4.2 establishes that, for all $k \geq 2$, $\mathrm{R}^p_{k-tt}(\mathrm{SPARSE}) \neq \mathrm{E}^p_{k-tt}(\mathrm{SPARSE})$. Gavaldà and Watanabe [13] have proven that, for any unbounded polynomial-time computable function $f(n)$, $\mathrm{R}^p_{f(n)-tt}(\mathrm{SPARSE}) \not\subseteq \mathrm{E}^p_T(\mathrm{SPARSE})$. Both these results suggest that equivalence exacts a price; in order to achieve equivalence to sets reducible to sparse sets, one must use a more powerful type of reduction.

It is natural to seek the exact price that equivalence extracts. This section shows that, unless P $\neq$ NP, every set 2-truth-table reducible to a sparse set is truth-table equivalent to a sparse set. It follows that the result of Gavaldà and Watanabe cannot be extended to 2-truth-table reductions without providing a proof that P $\neq$ NP.

THEOREM 4.3. *If* P = NP *then* $\mathrm{R}^p_{2-tt}(\mathrm{SPARSE}) \subseteq \mathrm{E}^p_{tt}(\mathrm{SPARSE})$.

Proof: Let $L \leq^p_{2-tt} S$, S sparse, via truth-table generator g and evaluator e [22]. Under our hypothesis that P = NP, we construct a sparse set $\widehat{S}$ such that $L \equiv^p_{tt} \widehat{S}$. Let $\{\tau_i \mid 1 \leq i \leq 16\}$ represent the sixteen truth-tables of arity two. Let $H_i = \{x \mid$ the truth table that $e(x, \cdot, \cdot)$ uses is table $\tau_i\}$. For each i, $1 \leq i \leq 16$, we will define a sparse set S_i and truth-table reductions $\lambda_i = (g_i, e_i)$ and $\gamma_i = (g'_i, e'_i)$ such that:

1. $(\forall i : 1 \leq i \leq 16)(\forall x \in H_i)[x \in L \iff \lambda_i^{S_i}(x)$ accepts] and
2. $(\forall i : 1 \leq i \leq 16)(\forall y)[y \in S_i \iff \gamma_i^{H_i \bigcap L}(y)$ accepts] and
3. $(\forall i : 1 \leq i \leq 16)(\forall y)[g'_i(y)$ queries only strings in H_i].[4]

[4] We assume that each H_i is non-empty; the case where some H_i are empty can easily be dealt with by

Set $\widehat{S} = \{\langle i, j\rangle \mid j \in S_i \text{ and } 1 \leq i \leq 16\}$. By the above three conditions, $L \leq_{tt}^{p} \widehat{S}$ via the reduction that, on input x, determines which H_i contains x and uses λ_i modified so that each query z to S_i becomes a query $\langle i, z\rangle$ to $\widehat{S}$. Clearly, $\widehat{S} \leq_{tt}^{p} L$ via (g'', e''), where $g''(\langle i, z\rangle) = g_i'(z)$ and $e''(\langle i, z\rangle, \cdots) = e_i'(z, \cdots)$, for $1 \leq i \leq 16$, and as noted in footnote 4 for other i. Thus, $\widehat{S} \equiv_{tt}^{p} L$.

Figure 4.2 lists the sixteen truth-tables of size two. We proceed to define the sets $S_1, \ldots, S_{16}$.

Note that without loss of generality we may assume:

Assumption 4.4. *g is length-increasing and $g(x) = \langle b, c\rangle \Rightarrow |b| = |c|$.* This is simply because if $A \leq_{2-tt}^{p} B$, B sparse, via truth-table generator h, then $A \leq_{2-tt}^{p} B'$ via truth-table generator h', where $B' = \{\langle 0^l, y\rangle \mid y \in S\}$ and if $h(x)$ outputs $\langle q_1, q_2\rangle$ then $h'(x)$ outputs $\langle\langle 0^{|q_2|+|x|+1}, q_1\rangle, \langle 0^{|q_1|+|x|+1}, q_2\rangle\rangle$. Note that B' is sparse and h' maintains the properties asserted in Assumption 4.4 (recall that $|\langle a, b\rangle| = |a| + |b|$). We assume these properties throughout this proof.

Tables 1 and 16 are trivial; let $S_1 = S_{16} = \emptyset$. Tables 6, 8, 9, and 11 are 1-truth-table reductions; thus the construction of S_6, S_8, S_9, and S_{11} is essentially handled by Theorem 3.2. Similarly, Table 4 represents conjunctive 2-truth-table reductions and is essentially handled by the same result, since every set that conjunctive bounded truth-table reduces to a sparse set in fact many-one reduces to some sparse set.

Let us say that 2-truth-table a is the *complement* of 2-truth-table b if a and b differ on each possible response; for example, Tables 4 and 13 are complementary. Suppose we have proven that: (**) if A reduces to sparse set B via a 2-truth-table reduction that always uses Table τ, then $A \in \mathrm{E}_{tt}^{p}(\mathrm{SPARSE})$. It follows immediately that we have also proven: if A reduces to sparse set B via a 2-truth-table reduction that always uses Table *complement*(τ), then $A \in \mathrm{E}_{tt}^{p}(\mathrm{SPARSE})$. This is so because $\widehat{A}$ 2-truth-table reduces to $\widehat{B}$ via truth-table *complement*(τ) if and only if $\overline{\widehat{A}}$ 2-truth-table reduces to $\widehat{B}$ via truth-table τ. Thus, if we have established (**), we can conclude that $(\exists \text{ sparse set } C)[\overline{\widehat{A}} \equiv_{tt}^{p} C]$, and thus $\widehat{A} \equiv_{1-tt}^{p} \overline{\widehat{A}} \equiv_{tt}^{p} C$, so $\widehat{A} \equiv_{tt}^{p} C$. Thus it follows that the case of Table 13 follows immediately from that of Table 4. Below, we will use complementarity to reduce our work.

Consider the case of Table 15 (2-disjunctive reductions). Let $\widehat{S}_{15}$ represent all strings

using vacuous truth-table reductions. For example, if $H_7 = \emptyset$, then set $S_7 = \emptyset$ and reduce S_7 to L via the truth-table evaluator that always rejects. Similarly, when resolving the membership in $\widehat{S}$ of elements of the form $\langle i, j\rangle$, with $i \notin \{1, 2, \ldots, 16\}$, we can also use a vacuous reduction.

Table Number	First Query Answered “no”		First Query Answered “yes”	
	2nd Ans. “no”	2nd Ans. “yes”	2nd Ans. “no”	2nd Ans. “yes”
1	0	0	0	0
2	1	0	0	0
3	0	0	1	0
4	0	0	0	1
5	0	1	0	0
6	1	0	1	0
7	1	0	0	1
8	1	1	0	0
9	0	0	1	1
10	0	1	1	0
11	0	1	0	1
12	1	0	1	1
13	1	1	1	0
14	1	1	0	1
15	0	1	1	1
16	1	1	1	1

FIG. 1. *Truth-tables of arity two.*

in S that are queried by some truth-table reduction from a member of H_{15}. Recalling Assumption 4.4, let polynomial $q(n)$ strictly upper-bound the number of elements in $\widehat{S}_{15}$ of length at most n. We will say that a string x is *busy* if there are more than $q(|x|)$ distinct strings w (each necessarily of the same length as x) that satisfy the condition:

> there exists a string $\alpha_w \in H_{15} \bigcap L$ such that the (unordered) pair of strings queried by $g(\alpha_w)$ is $\{x, w\}$.

All busy strings are in $\widehat{S}_{15}$. However, there may also be strings in $\widehat{S}_{15}$ that are not busy. We now define $S_{15} = \{\langle 0^l, z\rangle \bigm| (\exists y, w)[|y| = l$ and the (unordered) pair of strings queried by $g(y)$ is $\{z, w\}$ and z is busy$]\} \bigcup \{\langle 0^l, y, z\rangle \bigm| (\exists w)[|w| = l$ and $w \in L$ and the (unordered) pair of strings queried by $g(w)$ is $\{y, z\}$ and neither y nor z is busy$\}]$.

Clearly, for strings in H_{15}, membership in L can be tested via $\leq^p_{3-tt}$ reduction to S_{15}, and clearly S_{15} is sparse.

> **Claim 1** If P = NP then S_{15} truth-table reduces to L via a truth-table reduction that queries only members of H_{15}.
>
> **Proof of Claim 1**
>
> There are two cases, corresponding to the two types of strings in S_{15}. In the first case, we are asked whether a string $\langle 0^l, z\rangle$ is in S_{15}. Use our assumption that P = NP to find[5] if possible (if not, then reject) more than $q(|z|)$ strings (not necessarily of length l) $\alpha_i \in H_{15}$, with each α_i mapping to $\{z, w_i\}$, with all the w_i's distinct, and use our P = NP assumption to find an appropriate y (of length l and in H_{15}). Then, via a truth-table query to L, check whether all the α_i are in L and accept if and only if all are.
>
> In second case, we are asked whether a string $\langle 0^l, y, z\rangle$ is in S_{15}. Use our assumption that P = NP to attempt to find more than $q(|z|)$ strings $\alpha_i \in H_{15}$, with each α_i mapping to $\{z, w_i\}$ with all the w_i's distinct. Also, use our assumption that P = NP to attempt to find more than $q(|z|)$ strings $\beta_i \in H_{15}$, with each β_i mapping to $\{y, v_i\}$ with all the v_i's distinct. Finally, use our P = NP assumption to find a string $w \in H_{15}$, of length l, such that

[5] Via binary search, in the standard fashion, using a test set such as $\{\langle z, prefix, \alpha_1, \ldots\rangle \bigm|$ there exists a string $\widehat{\alpha} \in H_{15}$ whose prefix is *prefix* and that differs from all the α_i and $g(\widehat{\alpha})$ yields the pair $\{z, p\}$ and this pair is not yielded by $g(\alpha_i)$ for any $i\}$.

> $g(w)$ maps to the pair $\{y, z\}$. Now, we make a truth-table query to L, inquiring about the membership of w, the α_i's, and the β_i's. We accept if and only if (1) $w \in L$ and (2) either we failed to find the requisite number of α_i's or some of the α_i's found are not in L and (3) either we failed to find the requisite number of β_i's or some of the β_i's found are not in L.
>
> **End of Proof of Claim 1** ▮

Note that, by the earlier complementation argument, solving Table 15 implicitly solves Table 2.

Consider now the case of Table 10 (exclusive or). Let $\widehat{S}_{10}$ represent all strings in S that are queried by some truth-table reduction from a member of H_{10}. Recalling Assumption 4.4, let polynomial $q(n)$ strictly upper-bound the number of elements in $\widehat{S}_{10}$ of length at most n. We will say that a string x is *heavy* if there are more than $q(|x|)$ distinct strings w (each necessarily of the same length as x) that satisfy the condition:

> there exists a string[6] $\alpha_w \in H_{10} \bigcap L$ such that the (unordered) pair of strings queried by $g(\alpha_w)$ is $\{x, w\}$.

All heavy strings are in $\widehat{S}_{10}$. However, there may also be strings in $\widehat{S}_{10}$ that are not heavy. We now define $S_{10} = \{\langle 0^l, z\rangle \,\Big|\, z \text{ is heavy}\} \bigcup \{\langle 0^l, w, z\rangle \,\Big|\, w \text{ is heavy and } (\exists y)[|y| = l \text{ and } y \notin L$ and the (unordered) pair of strings queried by $g(y)$ is $\{w, z\}]\} \bigcup \{\langle 1^l, w, z\rangle \,\Big|\, (\exists y)[|y| = l$ and $y \in L$ and the (unordered) pair of strings queried by $g(y)$ is $\{w, z\}$ and neither w nor z is heavy$]\}$.

Clearly, for strings in H_{10}, membership in L can be tested via $\leq^p_{5-tt}$ reduction to S_{10}, and clearly S_{10} is sparse.

> **Claim 2** If P = NP then S_{10} truth-table reduces to L via a truth-table reduction that queries only members of H_{10}.
>
> **Proof of Claim 2**
>
> There are three cases, corresponding to the three types of strings in S_{10}.
>
> **Case 1:** In the first case, we are asked whether a string $\langle 0^l, z\rangle$ is in S_{10}. Use our assumption that P = NP to find (as before) as many α_i as possible

[6] Unlike the case of disjunctive reductions, in the exclusive-or case a string in $H_{10} - L$ may map to two strings in the sparse set, and we wish *not* to allow such cases to contribute towards heaviness.

(but no more than $2q(|z|)+1$) such that $\alpha_i \in H_{10}$ and $g(\alpha_i)$ queries the (unordered) pair $\{z,\, w_i\}$ and $j \neq k \;\Rightarrow\; w_j \neq w_k$. If we have found $\leq q(|z|)$ such α_i's, then reject. Otherwise, make a truth-table query to L regarding the α_i's, and see if more than $q(|z|)$ of the α_i's are in L, and accept if and only if this is the case.

Note: the above strategy works since (1) if z is heavy, then there are no more than $q(|z|)$ values w_i such that some *nonmember* of $H_{10} - L$ maps to $\{z,\, w_i\}$ (as these w_i's must be in $\widehat{S}_{10}$), and (2) if z is not heavy, there can be at most $q(|z|)$ distinct values w_i such that some *member* of $L \bigcap H_{10}$ maps to $\{z,\, w_i\}$.

Case 2: In the second case, we are asked whether a string $\langle 0^l, w,\ z\rangle$ is in S_{10}. Check whether w is heavy as in Case 1. Also, use our P = NP assumption to find a y as in the definition of S_{10}, and use L to check whether $y \notin L$. Accept if and only if an appropriate y was found and $y \notin L$ and w is heavy. (Note that all the above can be done via a single round of truth-table queries to L.)

Case 3: In the third case, we are asked whether a string $\langle 1^l, w,\ z\rangle$ is in S_{10}. Check that w is not heavy and that z is not heavy as in Case 1, except exchanging criteria (that is, if there are less than or equal to $q(|z|)$ values α_i then we find a string "not heavy," otherwise a string is "not heavy" if and only if no more than $q(|z|)$ of the α_i's are in L). Also, use our P = NP assumption to obtain y as in the definition of S_{10}, and use L to verify that $y \in L$. Accept if and only if an appropriate y exists and $y \in L$ and w is not heavy and z is not heavy. (Again, note that all the above can be done via a truth-table query to L.)

End of Proof of Claim 2 ∎

Note that, by the earlier complementation argument, solving Table 10 implicitly solves Table 7.

Consider now the case of Table 3. Let $\widehat{S}_3$ represent all strings in S that are queried by some truth-table reduction from a member of H_3. Recalling Assumption 4.4, let polynomial $q(n)$ strictly upper-bound the number of elements in $\widehat{S}_3$ of length at most n. We will say that a string x is *top-heavy* if there are more than $q(|x|)$ distinct strings w (each necessarily of the same length as x) that satisfy the condition:

> there exists a string $\alpha_w \in H_3 \bigcap L$ such that the (ordered) pair of strings queried by $g(\alpha_w)$ is (x, w).

All top-heavy strings are in $\widehat{S}_3$. However, there may also be strings in $\widehat{S}_3$ that are not top-heavy. We now define $S_3 = \{\langle 0^l, z\rangle \bigm| z \text{ is top-heavy}\} \bigcup \{\langle 0^l, z', z''\rangle \bigm| z' \text{ is not top-heavy and } (\exists y)[|y| = l \text{ and } y \in L \bigcap H_3 \text{ and } g(y) = \langle z', z''\rangle]\} \bigcup \{\langle 1^l, z, z'\rangle \bigm| z \text{ is top-heavy and } (\exists w)[|w| = l \text{ and } w \in H_3 - L \text{ and } g(w) = \langle z, z'\rangle]\}$.

Clearly, for strings in H_3, membership in L can be tested via $\leq^p_{3-tt}$ reduction to S_3, and clearly S_3 is sparse.

Claim 3 If P = NP then S_3 truth-table reduces to L via a truth-table reduction that queries only members of H_3.

Proof of Claim 3

There are three cases, corresponding to the three types of strings in S_3.

Case 1: In the first case, we are asked whether a string $\langle 0^l, z\rangle$ is in S_3. Use our assumption that P = NP to find (as before) as many α_i as possible (but no more than $2q(|z|) + 1$) such that $\alpha_i \in H_3$ and $g(\alpha_i) = \langle z, w_i\rangle$ and $j \neq k \Rightarrow w_j \neq w_k$. If we have found more than $q(|z|)$ such α_i's that are in L, then accept, otherwise reject.

Note: the above strategy works since if z is top-heavy, then there are no more than $q(|z|)$ values α_i as above such that $\alpha_i \in H_3 - L$.

Case 2: In the second case, we are asked whether a string $\langle 0^l, z', z''\rangle$ is in S_3. Check whether z' is *not* top-heavy as in Case 1, except flipping our notions of acceptance and rejection. Also, use our P = NP assumption to find y as in the definition of S_3 (reject if there is no such y). Accept if and only if y is in L and z' is not top-heavy.

Case 3: In the third case, we are asked whether a string $\langle 1^l, z, z'\rangle$ is in S_3. Check whether z is top-heavy as in Case 1. Also, use our P = NP assumption to find w as in the definition of S_3 (reject if there is no such w). Accept if and only if w is not in L and z is top-heavy.

End of Proof of Claim 3 ▮

Note that, by the earlier complementation argument, and by symmetry, and by both complementation and symmetry, solving Table 3 implicitly solves Tables 14, 5, and 12.

End of Proof of Theorem 4.3 ∎

In fact, a careful inspection of the proof of Theorem 4.3 reveals that various stronger statements than Theorem 4.3 have been implicitly proven. These improvements show, among other things, that Theorem 4.1 cannot be improved. Also, the power of unbounded-truth-table reductions, and the strength of the P = NP assumption, are both used only in one direction. Thus we have:

THEOREM 4.5. *Let $\epsilon > 0$. If $L \in \mathrm{R}^p_{2-tt}(\mathrm{SPARSE})$ then there exists a sparse set S' such that:*

- *$L \leq^p_{5-tt} S'$, and*
- *if $\mathrm{P} = \mathrm{NP}$ then $S' \leq^p_{n^\epsilon-tt} L$.*

Proof: The proof of Theorem 4.3 provides a set $\widehat{S}$ such that $L \leq^p_{5-tt} \widehat{S}$, and if $\mathrm{P} = \mathrm{NP}$ then $\widehat{S} \leq^p_{n^k-tt} L$ for some k. Now let $S' = \{x10^{|x|^l} : x \in \widehat{S}\}$ for some l such that $k/l < \epsilon$; it is immediate that $L \leq^p_{5-tt} S'$, and if $\mathrm{P} = \mathrm{NP}$ then $S' \leq^p_{n^\epsilon-tt} L$. ∎

Of course, the assumption that P = NP is a very strong one. However, though the P = NP assumption gives polynomial-time computations access to the full power of the polynomial-hierarchy, in fact the above proofs used the P = NP assumption only to give polynomial-time computations access to the power of NP (and in particular, the power to find sets of inverses of honest polynomial-time many-one functions). Thus the above proof in fact proves Theorem 4.6 below, whose oracle access mechanism is exactly that used in defining the extended-low-two sets [5]—a mechanism that appears in other applications also [14]. Of particular note is that the set L is—as in Theorem 4.3 but unlike Theorem 4.7—queried only polynomially often.

THEOREM 4.6. *If $L \in \mathrm{R}^p_{2-tt}(\mathrm{SPARSE})$ then there exists a sparse set S' such that:*

- *$L \leq^p_{5-tt} S'$, and*
- *$S' \in \mathrm{P}^{\mathrm{NP} \oplus L}$.*

The above results are all conditioned upon the assumption that P = NP or the essentially equivalent use of an NP oracle. In fact, we can outright eliminate such assumptions, at the cost of acquiescing to relatively powerful reductions that are allowed to access the set L far more than polynomially often. Thus, the following theorem neither implies nor is implied by Theorem 4.6.

THEOREM 4.7. *If $L \in \mathrm{R}^p_{2-tt}(\mathrm{SPARSE})$ then there exists a sparse set S' such that:*

- $L \leq^p_{5-tt} S'$, *and*
- $S' \in \mathrm{DP}^L$.

Here, DP, difference polynomial time, is the class of sets—first studied by Papadimitriou and Yannakakis [27]—that can be represented as the difference of two NP sets; DP sets are crucial to the normal-form structure of the boolean hierarchy [11] and appear naturally in many settings [12]. Informally, we may describe Theorem 4.7 as stating that all sets 2-truth-table reducible to sparse sets are DP-equivalent to sparse sets. We omit the proof, as it is based on a detailed analysis similar to that of Theorem 4.3.

Finally, we note that all the theorems of this section yield not only equivalence but indeed honest equivalence.

5. On the Power of Conjunctive and Disjunctive Reductions

In this section, we will show several inclusions among classes of sets that are reducible to sparse sets. We first show the following lemma.

LEMMA 5.1. $\mathrm{R}^p_{1-tt}(\mathrm{SPARSE}) \subseteq \mathrm{R}^p_{dtt}(\mathrm{SPARSE})$.

Proof Let L be a set that is $\leq^p_{1-tt}$ reducible to a sparse set S. We will show that $L \leq^p_{dtt} U$ for some sparse set U. To prove this, we need to define some notation. For a string x and $n \geq 1$, we will use x_n to denote the nth symbol in x. For two strings x and y and a set A, xAy denotes the set $\{xwy \,\big|\, w \in A\}$. Let # be a special symbol not in Σ. For two sets A and B, $A \oplus B$ denotes the set $0A \bigcup 1B$. Since S is sparse, there exists a polynomial p_0 such that for every $n \geq 0$ it holds that $||S^{\leq n}|| \leq p_0(n)$.

Let $T = \{0z0 \,\big|\, z \in S\} \bigcup 01^*$. It is not hard to see that for every $x \in \Sigma^*$, $x \in S \iff 0x0 \in T$ and for every $n \geq 1$, $T^{=n} \neq \emptyset$. Moreover, for every $n \geq 0$,

$$||T^{\leq n}|| \;\leq\; ||S^{\leq n-2}|| + n \;\leq\; ||S^{\leq n}|| + n \;\leq\; p_0(n) + n.$$

Therefore, T is sparse.

Now define U to be the set of strings of the form $\#^n u \# b$ such that:

(1) $u \in \Sigma^{\leq n}$ and $b \in \Sigma$,

(2) $ub\Sigma^* \bigcap T^{=n} = \emptyset$; that is, ub is not a prefix of any string in $T^{=n}$, and

(3) $u\Sigma^* \bigcap T^{=n} \neq \emptyset$; that is, u is a prefix of some string in $T^{=n}$.

Then, for every $n \geq 0$:

$$\begin{aligned}
||U^{\leq n}|| &\leq ||\{\#^m u\#b \in U \,\big|\, 1 \leq m \leq n\}|| \\
&\leq ||\{\#^m u\#b \,\big|\, 1 \leq m \leq n \text{ and } b \in \Sigma \text{ and } u\Sigma^* \bigcap T^{=m} \neq \emptyset\}|| \\
&\leq 2n||\{w \,\big|\, w \text{ is a prefix of some string in } T^{\leq n}\}|| \\
&\leq 2n^2||T^{\leq n}|| \\
&\leq 2n^2(p_0(n) + n).
\end{aligned}$$

Therefore, U is sparse.

We now establish the following claim.

Claim 1 Let $x \in \Sigma^*$ and $y = 0x0$. Then, $y \notin T$ if and only if $\{\#^{|y|}y_1 \cdots y_k\#y_{k+1} \,\big|\, 1 \leq k \leq |y| - 1\} \bigcap U \neq \emptyset$ and $y \in T$ if and only if $\#^{|y|}y\#0 \in U$.

Proof of Claim 1 Let x be any string and let $y = 0x0$. Furthermore, let n denote $|y|$ $(= |x| + 2)$. First suppose that $y \in T$. Since $|y0| > n$ and $y \in T^{=n}$, $\#^n y\#0 \in U$. Furthermore, for every $i, 1 \leq i \leq n$, $y_1 \cdots y_i$ is clearly a prefix of y. Thus, for every $k, 1 \leq k < n$, $\#^n y_1 \cdots y_k\#y_{k+1}$ is not in U.

On the other hand, suppose that $y \notin T$. Clearly, $\#^n y\#0 \notin U$ because $y \notin T^{=n}$. Furthermore, since for every $n \geq 1$ it holds that $T^{=n} \neq \emptyset$ and $T^{=n} \subseteq 0\Sigma^*$, there exists a unique $m, 1 \leq m < n - 1$, such that $y_1 \cdots y_m$ is a prefix of some string in $T^{=n}$ but $y_1 \cdots y_{m+1}$ is not a prefix of any string in $T^{=n}$. So, clearly, $\#^n y_1 \cdots y_m\#y_{m+1} \in U$.

From the above considerations, $y \in T$ if and only if $\#^n y\#0 \in U$ and $y \notin T$ if and only if for some $k, 1 \leq k < n$, $\#^n y_1 \cdots y_k\#y_{k+1} \in U$. This proves the claim.

End of Proof of Claim 1 ▮

Since $x \in S$ if and only if $y = 0x0 \in T$, from Claim 1, we have:

$$\begin{aligned}
x \in S &\iff \#^{|y|}y\#0 \in U \qquad \text{and} \\
x \notin S &\iff (\exists k : 1 \leq k \leq |y| - 1)[\#^{|y|}y_1 \cdots y_k\#y_{k+1} \in U].
\end{aligned}$$

Therefore, $S \oplus \bar{S} \leq_{dtt}^{p} U$. Since $L \leq_{1-tt}^{p} S$ implies that $L \leq_{m}^{p} S \oplus \bar{S}$, $L \leq_{dtt}^{p} U$ and this proves the theorem.

End of Proof of Lemma 5.1 ▮

From Lemma 5.1, we obtain the following theorem.

THEOREM 5.2. $\mathrm{R}_{btt}^{p}(\mathrm{SPARSE}) \subseteq \mathrm{R}_{dtt}^{p}(\mathrm{SPARSE})$.

Proof Let L be a set $\leq_{k-tt}^{p}$ reducible to a sparse set S for some $k \geq 0$ via a polynomial-time computable function f. To establish the theorem, we have only to show that there is another sparse set A to which L is $\leq_{dtt}^{p}$ reducible. Since f witnesses that $L \leq_{k-tt}^{p} S$, without loss of generality (see [22]), we may assume the following: For every $x \in \Sigma^*$,

(a) $f(x)$ is of the form $b_{11} \cdots b_{1k}\$ \cdots \$b_{m1} \cdots b_{mk}\$w_1\$ \cdots \$w_k$, where (1) \$ is a new symbol not in $\{0, 1, \#\}$, (2) for every $i, 1 \leq i \leq m$ and $j, 1 \leq j \leq k$, $b_{ij} \in \{0, 1\}$, and (3) for every $i, 1 \leq i \leq m$, $w_i \in \Sigma^*$, and

(b) $x \in L$ if and only if $(\exists i : 1 \leq i \leq m)(\forall j : 1 \leq j \leq k)[\chi_S(w_j) = \text{true} \iff b_{ij} = 0]$, where χ_S is the characteristic function of S; that is, for every w, $\chi_S(w) =$ true if $w \in S$ and false otherwise.

Since $S \oplus \bar{S}$ is $\{0s \mid s \in S\} \bigcup \{1s \mid s \in \bar{S}\}$, it is not hard to see that the condition (b) is equivalent to the following:

(b1) $x \in L$ if and only if $(\exists i : 1 \leq i \leq m)(\forall j : 1 \leq j \leq k)[b_{ij}w_j \in S \oplus \bar{S}]$.

Now recall that we showed in Lemma 5.1 that there is a sparse set U to which $S \oplus \bar{S}$ is $\leq_{dtt}^{p}$ reducible. Let g be a $\leq_{dtt}^{p}$ reduction from $S \oplus \bar{S}$ to U. Then, without loss of generality, we may assume that for every $y \in \Sigma^*$,

(c) $g(y)$ is of the form $z_1\$ \cdots \$z_m$, where $m = p(|y|)$ for some polynomial p and

(d) $y \in S \oplus \bar{S}$ if and only if $\{z_1, \cdots, z_m\} \bigcap U \neq \emptyset$.

For each y, let $\sigma(y)$ denote the set of all strings $\{z_1, \cdots, z_m\}$ that g outputs upon input y. Then, the condition (b1) is equivalent to the following:

(b2) $x \in L$ if and only if $(\exists i : 1 \leq i \leq m)(\forall j : 1 \leq j \leq k)[\sigma(b_{ij}w_j) \bigcap U \neq \emptyset]$.

For each $i, 1 \leq i \leq m$, let $H(i)$ denote the set:

$$\{u_1\$ \cdots \$u_k \mid (\forall j : 1 \leq j \leq k)[u_j \in \sigma(b_{ij}w_j)]\}.$$

Also, define:

$$A = \{u_1\$\cdots\$u_k \,\Big|\, (\forall j : 1 \le j \le k)[u_j \in U]\}.$$

Then, it is not hard to see that the condition (b2) is equivalent to:

(b3) $x \in L$ if and only if $(\exists i : 1 \le i \le m)(\exists v \in H(i))[v \in A]$.

Since f and g are polynomial-time computable and k is a constant, there is a polynomial q such that $||\{z \,\big|\, (\exists i : 1 \le i \le m)[z \in H(i)]\}|| \le q(|x|)$.

Furthermore, it is easy to see that the set $\{z \,\big|\, (\exists i : 1 \le i \le m)[z \in H(i)]\}$ is polynomial-time computable in $|x|$. So, let h be a function that computes $v_1\$\cdots\$v_n$ so that $v_1, \cdots, v_n$ is an enumeration of all strings in $H(i)$ for some $i, 1 \le i \le m$. h is polynomial-time computable. $x \in L$ if and only if $\{v_1, \cdots, v_n\} \bigcap A \ne \emptyset$. Thus, h witnesses $L \le_{dtt}^{p} A$. Finally, since U is sparse and k is a constant, clearly A is sparse. Therefore, $L \in \mathrm{R}_{dtt}^{p}(\mathrm{SPARSE})$, and this proves the theorem. ▮

Next we consider the classes of sets that are reducible to sparse sets via polynomial-time nondeterministic Turing machines. The following definitions are due to Ladner, Lynch, and Selman.

Definition 5.3. *[22]*

(1) A set A is polynomial-time nondeterministic many-one reducible to a set B (denoted $A \le_m^{NP} B$) if there exists a polynomial-time nondeterministic Turing machine M such that for every $x \in \Sigma^*$,

(1A) for each computation path of M on x, M outputs some string, and

(1B) $x \in A$ if and only if there exists some string $y \in B$ that M outputs for some computation path on input x.

(2) A set A is polynomial-time nondeterministic Turing reducible to a set B (denoted $A \le_T^{NP} B$) if there exists a polynomial-time nondeterministic oracle Turing machine M such that for every $x \in \Sigma^*$, $x \in A$ if and only if there exists an accepting computation path of M on x relative to B.

(3) A set A is polynomial-time nondeterministic bounded truth-table reducible to a set B (denoted $A \le_{btt}^{NP} B$) if there exist $k \ge 0$ and a polynomial-time nondeterministic Turing machine M such that for every $x \in \Sigma^*$,

(3A) for each computation path of M on x, M outputs a string of the form $(\alpha, y_1, \cdots, y_k)$, where α is a k-truth-table and $y_1, \cdots, y_k \in \Sigma^*$, and

(3B) $x \in A$ if and only if there exists some output $(\alpha, y_1, \cdots, y_k)$ of M on x for some computation path such that $\alpha(\chi_B(y_1), \cdots, \chi_B(y_k)) = \text{true}$, where χ_B is the characteristic function of B.

(4) A set A is polynomial-time nondeterministic truth-table reducible to a set B, denoted by $A \leq_{tt}^{NP} B$, if there exists a polynomial-time nondeterministic Turing machine M and a polynomial-time computable truth-table evaluator such that $x \in A$ if and only if M, on input x, computes on some computation path a tt-condition y that is e-satisfied by B.

(5) A set A is polynomial-time nondeterministic conjunctive truth-table reducible to a set B, denoted by $A \leq_{ctt}^{NP} B$, if there exists a polynomial-time nondeterministic Turing machine M such that for every $x \in \Sigma^*$,

(5A) for each computation path of M on x, M outputs a string of the form $(y_1, \cdots, y_k)$, where $y_1, \cdots, y_k \in \Sigma^*$, and

(5B) $x \in A$ if and only if there exists some output $(y_1, \cdots, y_k)$ of M on x for some computation path such that $\{y_1, \cdots, y_k\} \subseteq B$.

(6) A set A is polynomial-time nondeterministic disjunctive truth-table reducible to a set B, denoted by $A \leq_{dtt}^{NP} B$, if there exists a polynomial-time nondeterministic Turing machine M such that for every $x \in \Sigma^*$,

(6A) for each computation path of M on x, M outputs a string of the form $(y_1, \cdots, y_k)$, where $y_1, \cdots, y_k \in \Sigma^*$, and

(6B) $x \in A$ if and only if there exists some output $(y_1, \cdots, y_k)$ of M on x for some computation path such that $\{y_1, \cdots, y_k\} \bigcap B \neq \emptyset$.

DEFINITION 5.4. $\mathrm{R}_m^{NP}(\mathrm{SPARSE})$ *(*$\mathrm{R}_T^{NP}(\mathrm{SPARSE})$, $\mathrm{R}_{btt}^{NP}(\mathrm{SPARSE})$, $\mathrm{R}_{tt}^{NP}(\mathrm{SPARSE})$, $\mathrm{R}_{ctt}^{NP}(\mathrm{SPARSE})$, $\mathrm{R}_{dtt}^{NP}(\mathrm{SPARSE})$*) denotes the class of sets that are $\leq_m^{NP}$ (respectively, $\leq_T^{NP}$, $\leq_{btt}^{NP}$, $\leq_{tt}^{NP}$, $\leq_{ctt}^{NP}$, $\leq_{dtt}^{NP}$) reducible to some sparse set.*

We may also use Lemma 5.1 to obtain the following theorem. It is important to emphasize that the results of this section depend crucially on the fact that we are reducing to the class of *sparse* sets. In particular, the following theorem should be contrasted with the fact that there are classes $\mathcal{C}$, and indeed single sets, such that $\mathrm{R}_{btt}^{NP}(\mathcal{C})$ and $\mathrm{R}_m^{NP}(\mathcal{C})$ differ [22].

THEOREM 5.5. $\mathrm{R}_m^{NP}(\mathrm{SPARSE}) = \mathrm{R}_{btt}^{NP}(\mathrm{SPARSE})$.

Proof To prove this, we will show that $\mathrm{R}_{btt}^{NP}(\mathrm{SPARSE}) \subseteq \mathrm{R}_{dtt}^{NP}(\mathrm{SPARSE})$. Ladner, Lynch, and Selman [22, Theorem 4.1, Part (iii)] have shown that for every pair of sets A and B, it holds that $A \leq_m^{NP} B$ if and only if $A \leq_{dtt}^{NP} B$. It follows immediately that $\mathrm{R}_{dtt}^{NP}(\mathrm{SPARSE}) \subseteq \mathrm{R}_m^{NP}(\mathrm{SPARSE})$. Thus, it suffices to show that $\mathrm{R}_{btt}^{NP}(\mathrm{SPARSE}) \subseteq \mathrm{R}_{dtt}^{NP}(\mathrm{SPARSE})$.

Let L be a set that, for some k, is $\leq_{k-tt}^{NP}$ reducible to a sparse set S via polynomial-time nondeterministic Turing machine M. Without loss of generality, we may assume that there is a polynomial p such that for every $x \in \Sigma^*$, each computation path of M on x has length exactly $p(|x|)$. Define $A = \{x\#y \,\big|\, y \in \Sigma^{=p(|x|)}$ and $M(x)$ on computation path y has output of the form $(\alpha, y_1, \cdots, y_k)$ such that $\alpha(\chi_S(y_1), \cdots, \chi_S(y_k)) = \mathrm{true}\}$. It is not hard to see that $A \leq_{k-tt}^{p} S$, and for every $x \in \Sigma^*$, $x \in L$ if and only if for some $y \in \Sigma^{=p(|x|)}$ it holds that $x\#y \in A$. Since $A \leq_{k-tt}^{p} S$, from Theorem 5.2, there exist a sparse set S' and a polynomial-time computable function f such that $A \leq_{dtt}^{p} S'$ is witnesses by f. Consider a machine N that, on input $x \in \Sigma^*$, nondeterministically guesses $y \in \Sigma^{=p(|x|)}$ and outputs $f(x\#y)$. Clearly, the machine N witnesses $L \leq_{dtt}^{NP} S'$. ▮

Next we prove the following theorem.

THEOREM 5.6. $\mathrm{R}_{ctt}^{NP}(\mathrm{SPARSE}) = \mathrm{R}_T^{NP}(\mathrm{SPARSE})$.

Proof Let L be a set in $\mathrm{R}_T^{NP}(\mathrm{SPARSE})$. Thus, there exists a polynomial-time nondeterministic oracle Turing machine and a sparse set S such that for every x, $x \in L$ if and only if M on input x relative to S accepts. Here, without loss of generality, we may assume the following: There exist two polynomials p and q such that for every x,

(1) M on input x has exactly $p(|x|)$ nondeterministic steps for each computation path, and

(2) for every computation path and for every oracle set X, M on x queries the oracle set exactly $q(|x|)$ times.

We will encode each computation path of M on input x as a string of length $p(|x|)$.

Moreover, since S is sparse, there exist a polynomial-time computable function f and a sparse set U such that $S \oplus \bar{S} \leq_{dtt}^{p} U$ via f.

Now consider the following nondeterministic Turing machine M':

(Description of M')

1. On input x, M' nondeterministically guesses a string $w \in \Sigma^{=p(|x|)}$ and $b_1, \cdots, b_{q(|x|)} \in \{0, 1\}$. M' simulates the computation of M on input x for the

computation path w in the following way: whenever the ith query y_i is made, instead of querying to the oracle M' regards the answer to the query as YES if $b_i = 0$ and NO otherwise, and M' stores the query string on its tape. If the simulation of M on x terminates at an accepting state, then M' proceeds to the next step. Otherwise, M' outputs a fixed string not in U and halts.

2. For each $i, 1 \leq i \leq q(|x|)$, M' looks up the table of query strings computed in the previous step, computes $f(b_i y_i)$, and nondeterministically picks a string z_i in the output of f.

3. Finally, M' outputs $z_1\$ \cdots \$z_k$ and halts.

(End of the Description of M')

From the above description, as in the proof of the previous theorem, it is not hard to see that (1) M' runs in time polynomial in $|x|$ and (2) $x \in L$ if and only if M' on x outputs some $(z_1, \cdots, z_{q(|x|)})$ such that $\{z_1, \cdots, z_{q(|x|)}\} \subseteq U$. Therefore, M' witnesses that $L \in \mathrm{R}_{ctt}^{NP}(\mathrm{SPARSE})$, thus proving the theorem. ▮

6. Conclusions and Open Problems

This paper addressed the question of whether reducibility to sparse sets is a broader notion than equivalence to sparse sets. For the many-one and 1-truth-table cases, we showed that differentiating reducibility from equivalence would yield a proof that P $\neq$ NP. In contrast, for the k-truth-table case, $k \geq 2$, reducibility *is* a provably broader notion than equivalence.

Nonetheless, there are limits on how much broader it can be. Gavaldà and Watanabe have proven that for every nice unbounded function f, some sets $f(n)$-truth-table reducible to sparse sets are not Turing equivalent (or even strong-nondeterministic equivalent) to any sparse set. However, we showed that their result can not be extended to the 2-truth-table case without yielding a proof that P $\neq$ NP. In particular, if P $=$ NP then all sets 2-truth-table reducible to sparse sets are truth-table equivalent to sparse sets.

Finally, we addressed the power of disjunctive and conjunctive reductions to sparse sets. Refuting a conjecture of Ko [20], we proved that all sets bounded truth-table reducible to sparse sets are indeed disjunctive truth-table reducible to sparse sets. Relatedly, for nondeterministic reductions to sparse sets, we proved that bounded truth-table reductions are no stronger than many-one reductions, and that Turing reductions are no stronger than conjunctive truth-table reductions.

A number of questions remain open. Regarding Section 5, though we refuted Ko's conjecture about disjunctive reductions, Ko's other conjectures have as yet been neither proven nor refuted. Regarding Section 4.2, can our proof be generalized from the 2-truth-table case to the bounded truth-table case?

A particularly interesting issue is that, even in the wake of the Gavaldà and Watanabe's study of the case of Turing reductions, many of the same questions remain open for the Turing case. Gavaldà and Watanabe [13] show that not all sets Turing reducible to sparse sets are even strong-nondeterministic equivalent to sparse sets. This is essentially an NP $\bigcap$ coNP lower bound on the strength of the reduction needed to achieve equivalence. A moment's thought reveals—via [29, Lemma 5.6]—an upper bound of Σ_2^p; that is, every set Turing reducible to a sparse set is $\equiv_T^{\Sigma_2^p}$ to some other sparse set.[7] However, the exact location of the optimal strength of reduction needed to achieve equivalence has not yet been pinpointed more accurately than the range (NP $\bigcap$ coNP, Σ_2^p].

Acknowledgments

We are grateful to Ronald Book for making our collaboration possible, to Russell Impagliazzo, Sanjay Jain, Robert Szelepcsényi, and Jozef Vyskoč for helpful comments and conversations, and to two anonymous referees for helpful suggestions. We thank the Tokyo Institute of Technology for hosting a workshop on computational complexity, in August 1990, at which this work was done in part. We are particularly grateful to Ricard Gavaldà and an anonymous referee for pointing out an error in an earlier version of this paper.

REFERENCES

[1] E. Allender and L. Hemachandra, *Lower bounds for the low hierarchy*, Journal of the ACM. To appear.

[2] ———, *Lower bounds for the low hierarchy*, in Proceedings of the 16th International Colloquium on Automata, Languages, and Programming, Springer-Verlag *Lecture Notes in Computer Science #372*, July 1989, pp. 31–45.

[7] It is important to note that we are *not* asserting that every set A that is Turing-reducible to a sparse set is Turing reducible to some sparse set in $\Sigma_2^{p,\,A}$; the best bound on such sets seems to be $\Delta_3^{p,\,A}$, via extending [29, Lemma 5.6] by first taking prefixes and then using adaptive search to find the lexicographically first suitable sparse set (circuit). The somewhat subtle point at work here is that in some cases equivalence allows us to trade off quantifiers between different directions of the equivalence, but reduction allows no such trade-offs.

[3] E. Allender and O. Watanabe, *Kolmogorov complexity and the degrees of tally sets*, Information and Computation, 86 (1990), pp. 160–178.

[4] J. Balcázar and R. Book, *Sets with small generalized Kolmogorov complexity*, Acta Informatica, 23 (1986), pp. 679–688.

[5] J. Balcázar, R. Book, and U. Schöning, *Sparse sets, lowness and highness*, SIAM Journal on Computing, 15 (1986), pp. 739–746.

[6] J. Balcázar, J. Díaz, and J. Gabarró, *Structural Complexity I*, EATCS Monographs in Theoretical Computer Science, Springer-Verlag, 1988.

[7] L. Berman and J. Hartmanis, *On isomorphisms and density of NP and other complete sets*, SIAM Journal on Computing, 6 (1977), pp. 305–322.

[8] F. Bin, Sept. 1989. Personal communication.

[9] R. Book and K. Ko, *On sets truth-table reducible to sparse sets*, SIAM Journal on Computing, 17 (1988), pp. 903–919.

[10] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung, *The boolean hierarchy II: Applications*, SIAM Journal on Computing, 18 (1989), pp. 95–111.

[11] ———, *The boolean hierarchy I: Structural properties*, SIAM Journal on Computing, 17 (1988), pp. 1232–1252.

[12] J. Cai and G. Meyer, *Graph minimal uncolorability is* D^{P}*-complete*, SIAM Journal on Computing, 16 (1987).

[13] R. Gavaldà and O. Watanabe, *On the computational complexity of small descriptions*, in Proceedings of the 6th Structure in Complexity Theory Conference, IEEE Computer Society Press, June/July 1991, pp. 89–101.

[14] J. Hartmanis and L. Hemachandra, *Robust machines accept easy sets*, Theoretical Computer Science, 74 (1990), pp. 217–226.

[15] J. Hartmanis and S. Mahaney, *An essay about research on sparse NP complete sets*, in Proceedings of the 9th Symposium on Mathematical Foundations of Computer Science, Springer-Verlag *Lecture Notes in Computer Science #88*, Sept. 1980, pp. 40–57.

[16] N. Immerman and S. Mahaney, *Relativizing relativized computations*, Theoretical Computer Science, 68 (1989), pp. 267–276.

[17] R. Impagliazzo and G. Tardos, *Decision versus search problems in super-polynomial time*, in Proceedings of the 30th IEEE Symposium on Foundations of Computer Science, IEEE Computer Society Press, October/November 1989, pp. 222–227.

[18] J. Kadin, $\mathrm{P}^{\mathrm{NP}[\log n]}$ *and sparse Turing-complete sets for NP*, Journal of Computer and System Sciences, 39 (1989), pp. 282–298.

[19] R. Karp and R. Lipton, *Some connections between nonuniform and uniform complexity classes*, in Proceedings of the 12th ACM Symposium on Theory of Computing, April 1980, pp. 302–309.

[20] K. Ko, *Distinguishing conjunctive and disjunctive reducibilities by sparse sets*, Information and Computation, 81 (1989), pp. 62–87.

[21] ———, *On adaptive versus nonadaptive bounded query machines*, Theoretical Computer Science, 82 (1991), pp. 51–69.

[22] R. Ladner, N. Lynch, and A. Selman, *A comparison of polynomial time reducibilities*, Theoretical Computer Science, 1 (1975), pp. 103–124.

[23] S. Mahaney, *Sparse complete sets for NP: Solution of a conjecture of Berman and Hartmanis*, Journal of Computer and System Sciences, 25 (1982), pp. 130–143.

[24] ———, *Sparse sets and reducibilities*, in Studies in Complexity Theory, R. Book, ed., John Wiley and Sons, 1986, pp. 63–118.

[25] ———, *The isomorphism conjecture and sparse sets*, in Computational Complexity Theory, J. Hartmanis, ed., American Mathematical Society, 1989, pp. 18–46. Proceedings of Symposia in Applied Mathematics #38.

[26] A. Meyer and M. Paterson, *With what frequency are apparently intractable problems difficult?*, Tech. Report MIT/LCS/TM-126, MIT Laboratory for Computer Science, Cambridge, MA, 1979.

[27] C. Papadimitriou and M. Yannakakis, *The complexity of facets (and some facets of complexity)*, Journal of Computer and System Sciences, 28 (1984), pp. 244–259.

[28] U. Schöning, *Complete sets and closeness to complexity classes*, Mathematical Systems Theory, 19 (1986), pp. 29–42.

[29] ———, *Complexity and Structure*, Springer Verlag *Lecture Notes in Computer Science #211*, 1986.

[30] V. Sewelson, *A Study of the Structure of NP*, PhD thesis, Cornell University, Ithaca, NY, August 1983. Available as Cornell Department of Computer Science Technical Report #83-575.

[31] S. Tang and R. Book, *Reducibilities on tally and sparse sets*, Theoretical Informatics and Applications (RAIRO), 25 (1991), pp. 293–302.

[32] Y. Yesha, *On certain polynomial-time truth-table reducibilities of complete sets to sparse sets*, SIAM Journal on Computing, 12 (1983), pp. 411–425.