Monotonous and randomized reductions to sparse sets*

V. Arvind
Department of Computer Science
Institute of Mathematical Sciences, C.I.T. Campus
Madras 600113, India

J. Köbler
Universität Ulm
Theoretische Informatik
D-89069 Ulm, Germany

M. Mundhenk
Universität Trier
Lehrstuhl IV – Informatik
D-54286 Trier, Germany

December 1, 1994

Abstract

An oracle machine is called *monotonous*, if after a negative answer the machine does not ask further queries to the oracle. For example, one truth-table, conjunctive, and Hausdorff reducibilities are monotonous. We study the consequences of the existence of sparse hard sets for different complexity classes under monotonous and randomized reductions. We prove trade-offs between the randomized time complexity of NP sets that reduce to a set B via such reductions and the density of B as well as the number of queries made by the monotonous reduction. As a consequence, bounded Turing hard sets for NP are not co-rp reducible to a sparse set unless RP = NP. We also prove similar results under the apparently weaker assumption that some solution of the promise problem $(1\mathrm{SAT}, \mathrm{SAT})$ reduces via the mentioned reductions to a sparse set.

1 Introduction

An important area of research in structural complexity theory concerns reductions to sparse sets, i.e. sets which only contain a polynomially bounded number of strings up to each length. This study has its roots in a conjecture by L. Berman and J. Hartmanis [BH77] that there are no sparse NP-complete sets under many-one

*Parts of this work have been presented at FST & TCS 1992 [AKM92].

reductions. Mahaney settled the conjecture by proving that if any NP-complete set many-one reduces to a sparse set then P = NP [Mah82]. Related work has been done in [Ber78, For79, Yap83, Yes83]; see Section 4 for a detailed discussion. From a different perspective, the possible existence of sparse Turing-hard sets for NP was studied in [KL80]. This question is equivalent to NP-complete problems having nonuniform polynomial-size circuits. Karp, Lipton, and Sipser proved that if NP has sparse Turing-hard sets then the polynomial-time hierarchy collapses to Σ_2^p [KL80]. It is also known that the existence of sparse Turing-complete sets for NP would collapse the polynomial-time hierarchy to $\mathrm{P}^{\mathrm{NP}[\log]}$ [Kad87].

The main purpose of this paper is to investigate monotonous and randomized reductions to sparse sets and to use the left set method to derive unlikely complexity class inclusions from the assumption that intractable sets reduce to sparse sets under these reductions. Discovering unlikely consequences of the existence of sparse hard sets for different kinds of polynomial-time truth-table reducibilities has become an active research area since the breakthrough result of Ogiwara and Watanabe [OW91] showing that NP does not have sparse hard sets under bounded Turing reductions unless P = NP. The proof relies on the notion of left sets, which are NP sets with a special self-reducibility structure. The left set method turned out to be a well suited tool to prove collapse results concerning sparse sets. Using this method similar results were obtained for polynomial-time conjunctive reductions [AHH+93, RR92]. Also the proof in [AHH+93] showing that no bounded Turing hard set for NP conjunctively reduces to a sparse set unless P = NP uses the left set technique. Furthermore, it makes use of the fact that the sets in $R_{bT}^p(R_c^p(\mathrm{SPARSE}))$ are monotonously reducible to a sparse set. The reason for this is that the class $R_c^p(\mathrm{SPARSE})$ has the algebraic structure of a set ring (i.e. it is closed under union and intersection). In this paper we investigate consequences of NP sets reducing by monotonous reductions to sets in the class $R_m^{co\text{-}rp}(\mathrm{SPARSE})$, which also forms a set ring. We prove as the main result that no bounded Turing hard set for NP co-rp reduces to a sparse set unless RP = NP.

The paper is organized as follows: In Section 3 we consider monotonous, non-adaptive, and positive oracle machines and show how the many-one, conjunctive, and Hausdorff reducibilities can be characterized by them.

Section 4 contains an overview of results concerning reductions to sparse sets. In particular, we make a brief tour describing for different types of reducibilities collapse consequences for the polynomial time hierarchy under the assumption that there are sparse hard sets for NP. In the overview we also touch upon certain other related issues concerning the complexity of sparse sets.

In Section 5 we prove our main result. We consider the case that an NP set A reduces to some set B via the composition of a Hausdorff and a co-rp many-one reduction. Similar to the results of [HL91] for the deterministic truth-table case, we derive interesting trade-offs between the density of the set B, the number $k(n)$ of queries in the Hausdorff reduction, and the randomized time complexity of A. As a special case we obtain that no bounded Turing hard set for NP co-rp many-one reduces to a sparse set unless RP = NP. This extends the result in [RR92] that an NP-complete set is not $\leq_m^{co\text{-}rp}$ reducible to a sparse set unless RP = NP.

In Section 6 we consider the problem of reducing some solution of the promise problem (1SAT, SAT) to sparse sets. In particular, we show that the conclusion RP = NP can be derived from the apparently weaker assumption that some solution of the promise problem (1SAT, SAT) is in $R_{bT}^p(R_m^{co\text{-}rp}(\mathrm{SPARSE}))$.

2 Notations

Our standard alphabet is $\Sigma = \{0, 1\}$. The set $\bigcup_{0 \leq i \leq n} \Sigma^i$ of all strings in Σ^* of length up to n is denoted by $\Sigma^{\leq n}$. For any set $A \subseteq \Sigma^*$, $A^{\leq n} = A \cap \Sigma^{\leq n}$, and $A^{=n} = A \cap \Sigma^n$. χ_A denotes the characteristic function of A. The length of a string x is denoted by $|x|$, and the cardinality of a set A is denoted by $\|A\|$.

A subset T of 0^* is called a tally set. The density function of a set A is defined as $density_A(n) = \|A^{\leq n}\|$. A set S is called sparse if its density function is bounded above by a polynomial. We use TALLY and SPARSE to denote the classes of tally and sparse sets, respectively. For a class of languages $\mathcal{C}$, co-$\mathcal{C}$ is the class of all sets whose complements are in $\mathcal{C}$, and $\bigcup \mathcal{C}$ denotes the union of all sets in $\mathcal{C}$. $\langle \cdot, \cdot \rangle$ denotes a standard polynomial-time invertible pairing function such that $\langle 0^i, 0^j \rangle \in 0^*$ for all $i, j \geq 0$. Such a pairing function can be extended in a standard fashion to encode arbitrary sequences $(x_1, \ldots, x_k)$ of strings into a string $\langle x_1, \ldots, x_k \rangle$. Where intent is clear we write $f(x_1, \ldots, x_k)$ in place of $f(\langle x_1, \ldots, x_k \rangle)$.

The reducibilities discussed in this paper are the standard polynomial-time reducibilities defined by Ladner, Lynch, and Selman [LLS75], the Hausdorff reducibility introduced by Wagner [Wag87], and the co-rp many-one reducibility (cf. [AM77, CKR91, Sch93]).

Definition 2.1 *Let A and B be sets.*

1. *A* truth-table reduces *to B ($A \leq_{tt}^p B$), if there exist polynomial-time functions f and g such that for all x, $f(x)$ computes a list of queries $\langle x_1, \ldots, x_m \rangle$, $g(x)$ is a boolean function in m variables, and $x \in A$ if and only if $g(x)$ evaluates to true on $(\chi_B(x_1), \ldots, \chi_B(x_m))$.*

2. *A* conjunctively reduces *to B ($A \leq_c^p B$), if $A \leq_{tt}^p B$ via f and g as above where $g(x)$ is the and-function. The definition of the* disjunctive reducibility *($A \leq_d^p B$) is analogous. As usual, the boolean and-function on zero variables evaluates to true, and the boolean or-function on zero variables evaluates to false.*

3. *A* Hausdorff reduces *to B ($A \leq_{hd}^p B$), if $A \leq_{tt}^p B$ via f and g as above such that for all x, if $f(x) = \langle x_1, \ldots, x_m \rangle$, then $\chi_B(x_1) \geq \chi_B(x_2) \geq \cdots \geq \chi_B(x_m)$, and g is the m-place parity function. I.e. $x \in A$ if and only if $\chi_B(x_i) > \chi_B(x_{i+1})$ for some odd i, $1 \leq i \leq m - 1$.*

For any function $h : \mathbb{N} \to \mathbb{N}$ and reducibility $\leq_r^p$, we use $A \leq_{h(n)\text{-}r}^p B$ to denote that $A \leq_r^p B$ via a reduction that on any input x asks at most $h(|x|)$ queries to the oracle. We write $A \leq_{br}^p B$, if $A \leq_{h(n)\text{-}r}^p B$ for some constant function h, saying that A bounded r reduces to B. The class $\{A \mid \exists B \in \mathcal{C} : A \leq_r^p B\}$ of sets which $\leq_r^p$ reduce to a set in the class $\mathcal{C}$ is denoted by $R_r^p(\mathcal{C})$.

Next we define the polynomial-time randomized reducibility that we use in this paper. In a co-rp many-one reduction from A to B the queries are randomly generated, and unlike the deterministic case above, the outcome depends on the membership of an exponential number of queries in B. We require that the probability of the reduction being correct is 1 for instances in A, but for instances not in A it can be as small as $1/poly$.

Definition 2.2 *$A \leq_m^{co\text{-}rp} B$ if there exist a polynomial-time function f and polynomials p and q such that for all x,*

$$\begin{aligned} x \in A &\Rightarrow \mathrm{Prob}[f(x,w) \in B] = 1, \text{ and} \\ x \notin A &\Rightarrow \mathrm{Prob}[f(x,w) \in B] \leq 1 - 1/p(|x|), \end{aligned}$$

where the string w is chosen uniformly at random from the set $\Sigma^{q(|x|)}$.

Observe that for every set B, $R_c^p(B) \subseteq R_m^{co\text{-}rp}(B)$ and $R_m^{co\text{-}rp}(R_m^{co\text{-}rp}(B)) \subseteq R_m^{co\text{-}rp}(B)$.

RTIME($t(n)$) denotes the class of sets A accepted by $O(t(n))$ time bounded randomized Turing machines (cf.[Gi77]) that have zero error probability for inputs not in A (and error probability at most 1/2 for instances in A). RP = RTIME($n^{O(1)}$).

For further notations we refer to [BDG].

3 Reducibilities and oracle machine properties

In this section we investigate how the restricted truth-table reducibilities defined earlier can be expressed by means of combinations of different restrictions on oracle machines. Unless otherwise specified, all the oracle machines considered here are polynomially time bounded.

Definition 3.1

1. *An oracle machine M is called* non-adaptive*, if M does not use the oracle to compute its queries. In a sense, the process of computing the queries to the oracle is independent of the oracle.*

2. *An oracle machine M is called* monotonous w.r.t. an oracle set B*, if for every input x, the sequence of queries $y_1, \ldots, y_m$ produced by $M^B(x)$ is monotonous w.r.t. B, i.e. $\chi_B(y_i) \geq \chi_B(y_{i+1})$ for $i = 1, \ldots, m-1$. M is called monotonous, if M is monotonous w.r.t. any set B. We use $A \leq_h^p B$ to denote that $A = L(M, B)$ for an oracle machine M which is monotonous w.r.t. B.*

3. [Sel82] *An oracle machine M is called* positive, *if for all sets B, B' it holds that $B \subseteq B'$ implies $L(M,B) \subseteq L(M,B')$.*

It is well-known [LLS75] that $A \leq_{tt}^{p} B$ if and only if $A = L(M,B)$ for a non-adaptive oracle machine M. We give further characterizations of reducibilities in terms of the three oracle machine properties defined above. These characterizations shed a new light on the Hausdorff reducibility and its composition with the conjunctive reducibility.

Proposition 3.2 *Let A, B be sets with $B \neq \emptyset$. Then $A \leq_{hd}^{p} B$ if and only if $A = L(M,B)$ for a non-adaptive oracle machine M which is monotonous w.r.t. B.*

Proof Clearly, any Hausdorff reduction $A \leq_{hd}^{p} B$ can be performed by a non-adaptive oracle machine which is monotonous w.r.t. B. For the converse, assume that $A = L(M,B)$ for a non-adaptive oracle Turing machine M which is monotonous w.r.t. B. Let $y_1, \ldots, y_m$ be the sequence of oracle queries of M on input x, and let $y_{i_1}, \ldots, y_{i_k}$ be the subsequence of maximal length such that for all $j = 1, \ldots, k$, $M^{\{y_{i_1}, \ldots, y_{i_{j-1}}\}}(x) \neq M^{\{y_{i_1}, \ldots, y_{i_j}\}}(x)$. Consider the polynomial-time function f defined as follows:

$$f(x) = \begin{cases} \langle y_{i_1}, \ldots, y_{i_k} \rangle, & M^{\emptyset}(x) \text{ rejects}, \\ \langle y_+, y_{i_1}, \ldots, y_{i_k} \rangle, & \text{otherwise}, \end{cases}$$

where y_+ is a fixed string in B. It is easy to verify that $A \leq_{hd}^{p} B$ via f. ∎

Proposition 3.3 *Let A, B be sets with $B \neq \Sigma^*$. Then $A \leq_{c}^{p} B$ if and only if $A = L(M,B)$ for a monotonous, positive oracle machine M.*

Proof Assume that $A \leq_{c}^{p} B$ via f. Consider the following oracle machine M. On input x, M first computes $f(x) = \langle y_1, \ldots, y_m \rangle$. Then M asks consecutively the queries $y_1, \ldots, y_m$ as long as the answers are positive. In the case of a negative answer M immediately rejects without asking further queries, otherwise M accepts. Clearly M is monotonous w.r.t. any oracle set. M is also positive, since an input is only accepted if all queries are answered positively by the oracle.

To prove the reverse direction, assume that $A = L(M,B)$ for a positive and monotonous oracle machine M. Let y_0 be a fixed string in $\overline{B}$, and consider the polynomial-time function f computed by the following algorithm.

> **input** x
> **if** $M^{\Sigma^*}(x)$ rejects **then**
>   **output** $f(x) = \langle y_0 \rangle$
> **else**
>   let $y_1, \ldots, y_m$ be the queries asked by $M^{\Sigma^*}(x)$, and
>   let $i \geq 0$ be the least index such that $M^{\{y_1, \ldots, y_i\}}(x)$ accepts
>   **output** $f(x) = \langle y_1, \ldots, y_i \rangle$
> **end**

First assume that $f(x) = \langle y_1, \ldots, y_i \rangle$ and that $\{y_1, \ldots, y_i\} \subseteq B$. By the definition of f it follows that $M^{\{y_1,\ldots,y_i\}}(x)$ accepts. Therefore, since M is positive, also $M^B(x)$ accepts.

Now assume that $M^B(x)$ accepts. Since M is positive, it follows that $M^{\Sigma^*}(x)$ accepts. Let $y_1, \ldots, y_m$ be the queries asked by $M^{\Sigma^*}(x)$, and let $f(x) = \langle y_1, \ldots, y_i \rangle$. By way of a contradiction let $j \leq i$ be the least index such that $y_j \notin B$. By the definition of f it follows that the first j queries of $M^B(x)$ are $y_1, \ldots, y_j$, and that $M^{\{y_1,\ldots,y_{j-1}\}}(x)$ rejects. Since M is monotonous, also $M^B(x)$ rejects, a contradiction. ∎

The next proposition should be compared with the characterization of the composition of the Hausdorff and conjunctive reducibilities in terms of the non-monotonic Hausdorff reducibility given in [AKM].

Proposition 3.4 *For every set B, $R^p_{hd}(R^p_c(B)) = R^p_h(B)$.*

Proof If $A \in R^p_{hd}(R^p_c(B))$ via a Hausdorff reduction function f and a conjunctive reduction function g, then $x \in A$ can be easily decided knowing the maximum initial subsequence s of $(y^1_1, \ldots, y^1_{k_1}, \ldots, y^m_1, \ldots, y^m_{k_m})$ containing only positive queries, where $f(x) = \langle y_1, \ldots, y_m \rangle$, and $g(y_j) = \langle y^j_1, \ldots, y^j_{k_j} \rangle$.

For the converse, assume that $A = L(M, B)$ for a monotonous oracle machine M. By Proposition 3.2, it suffices to show that $A = L(M', B^\omega_{and})$, where M' is the following non-adaptive oracle machine that is monotonous w.r.t. B^ω_{and}, and $B^\omega_{and} = \{\langle z_1, \ldots, z_n \rangle \mid \text{for all } i = 1, \ldots, n : z_i \in B\} \in R^p_c(B)$.

> M' on input x simulates $M^{\Sigma^*}(x)$ and collects all the queries $y_1, \ldots, y_m$. Then M' asks the queries $\langle y_1 \rangle, \langle y_1, y_2 \rangle, \ldots, \langle y_1, \ldots, y_m \rangle$ and accepts if and only if $M^{\{y_1,\ldots,y_i\}}(x)$ accepts, where $i \in \{0, \ldots, m\}$ is the maximum index such that $\langle y_1, \ldots, y_i \rangle$ gets a positive answer from the oracle.

Clearly, M' is non-adaptive and monotonous w.r.t. B^ω_{and}. ∎

As a straightforward consequence of the above proofs we get

Proposition 3.5 *For every class $\mathcal{C}$: if $\mathcal{C}$ is closed downward under $\leq^p_c$ reducibility, then $R^p_{(k(n)-1)\text{-}h}(\mathcal{C}) \subseteq R^p_{k(n)\text{-}hd}(\mathcal{C})$.*

Finally we characterize the many-one reducibility by oracle machines which are at the same time non-adaptive, positive, and monotonous.

Proposition 3.6 *Let A, B be sets with $B \neq \emptyset$ and $B \neq \Sigma^*$. Then $A \leq^p_m B$ if and only if $A = L(M, B)$ for a non-adaptive, positive, monotonous oracle machine M.*

Proof It is immediate that $\leq^p_m$ has the three properties. For the converse, assume $A = L(M, B)$ for a non-adaptive, positive, monotonous oracle machine

M. Let $y_1, \ldots, y_m$ be the queries of $M(x)$, and assume that M does not decide x independently of the oracle answers. Since M is monotonous, we have $\chi_B(y_1) \geq \cdots \geq \chi_B(y_m)$, and since M is positive, there exists an index i, $1 \leq i \leq m$, such that $M^{\{y_1,\ldots,y_j\}}(x)$ accepts if and only if $j \geq i$. Thus, $x \in A \iff y_i \in B$. ■

The characterizations of reducibilities performed by monotonous oracle machines are summarized in the following table.

oracle machines being			perform exactly
non-adaptive	positive	monotonous	reductions of type
-	-	✓	$\leq^p_{hd} \circ \leq^p_c$
✓	-	✓	$\leq^p_{hd}$
-	✓	✓	$\leq^p_c$
✓	✓	✓	$\leq^p_m$

Of special interest in the present paper are reductions where the number of queries is bounded by a constant. It is well-known that the closure of any class under bounded Turing reductions is the same as its closure under non-adaptive bounded reductions. The following Theorem states sufficient properties of a class to have the same closure under bounded Turing reductions and monotonous non-adaptive bounded reductions. A class $\mathcal{C}$ of sets is said to be a *set ring* if it includes $\emptyset$ and Σ^* and is closed under union and intersection.

Theorem 3.7 [AKM] *Let $\mathcal{C}$ be a set ring which is closed under many-one reductions. Then $R^p_{bT}(\mathcal{C}) = R^p_{bhd}(\mathcal{C})$.*

Using the fact that every sparse set is in $R^p_c(\text{TALLY})$ [BLS93] it is easy to see that $R^p_c(\text{SPARSE})$ forms a set ring, and therefore $R^p_{bhd}(R^p_c(\text{SPARSE})) = R^p_{bT}(R^p_c(\text{SPARSE}))$ [AKM]. As shown in the next theorem, also the reduction class $R^{co\text{-}rp}_m(\text{SPARSE})$ forms a set ring giving the following characterization.

Theorem 3.8 $R^p_{bhd}(R^{co\text{-}rp}_m(\text{SPARSE})) = R^p_{bT}(R^{co\text{-}rp}_m(\text{SPARSE}))$.

Proof We need to show that $R^{co\text{-}rp}_m(\text{SPARSE})$ is a set ring. Since SPARSE $\subseteq R^{co\text{-}rp}_m(\text{TALLY})$ [BLS93, Sch93], it suffices to show that $R^{co\text{-}rp}_m(\text{TALLY})$ is a set ring. Assume that $A \leq^{co\text{-}rp}_m T_1$ and $B \leq^{co\text{-}rp}_m T_2$, for sets $T_1, T_2 \in$ TALLY, via polynomial-time functions f, g and polynomials p and q (we can assume that there are uniform polynomials corresponding to both reduction functions), i.e.

$$\begin{aligned} x \in A &\Rightarrow \text{Prob}[f(x,w) \in T_1] = 1, \text{ and} \\ x \notin A &\Rightarrow \text{Prob}[f(x,w) \in T_1] \leq 1 - 1/p(|x|), \end{aligned}$$

and

$$\begin{aligned} x \in B &\Rightarrow \text{Prob}[g(x,w) \in T_2] = 1, \text{ and} \\ x \notin B &\Rightarrow \text{Prob}[g(x,w) \in T_2] \le 1 - 1/p(|x|), \end{aligned}$$

where w is chosen uniformly at random from the set $\Sigma^{q(|x|)}$. Consider the two tally sets

$$T_{or} = \{\langle a,b \rangle \mid a \in T_1 \text{ or } b \in T_2\} \text{ and } T_{and} = \{\langle a,b \rangle \mid a \in T_1 \text{ and } b \in T_2\},$$

and define the reduction function h as follows. For strings $w_1, w_2 \in \Sigma^*$ of the same length, $h(x, w_1 w_2) = \langle f(x,w_1), g(x,w_2) \rangle$. Then we have that

$$\begin{aligned} x \in A \cup B &\Rightarrow \text{Prob}[h(x,w) \in T_{or}] = 1, \text{ and} \\ x \notin A \cup B &\Rightarrow \text{Prob}[h(x,w) \in T_{or}] \le 1 - (1/p(|x|))^2, \end{aligned}$$

and

$$\begin{aligned} x \in A \cap B &\Rightarrow \text{Prob}[h(x,w) \in T_{and}] = 1, \text{ and} \\ x \notin A \cap B &\Rightarrow \text{Prob}[h(x,w) \in T_{and}] \le 1 - 1/p(|x|), \end{aligned}$$

where the string w is chosen uniformly at random from the set $\Sigma^{2q(|x|)}$. This shows that $R_m^{co\text{-}rp}(\text{TALLY})$ is closed under union and intersection. ■

4 Overview on reductions to sparse sets

There has been over a decade of research investigating consequences of the existence of hard NP sets in various sparse reduction classes. In this section we give a brief historical account leading to some of the most recent results in this area. In order to show the relationships between the various considered sparse and tally reduction classes we also give a brief summary of inclusion relationships between the most important of these reduction classes. This overview is not meant to be comprehensive about reductions to sparse sets. A more complete survey on the complexity of sparse sets can be found in [HOW92].

4.1 Reductions to sparse sets

As mentioned in the introduction, the study of reductions to sparse sets was started by the conjecture of L. Berman and J. Hartmanis [BH77] that there are no sparse NP-complete sets under $\le_m^p$ reductions. The first result was P. Berman's proof that P = NP if some tally set is NP-complete [Ber78]. This result was followed by Fortune's proof that if there is a sparse set that is complete for co-NP then P = NP [For79]. Both results were proved by giving a polynomial-time algorithm for SAT under the assumption that SAT reduces to a tally set (respectively co-sparse set in the case of Fortune's result). The main idea in the algorithm was to carry out a depth-first search on the self-reduction tree for SAT formulas. The self-reduction tree, which could have exponentially many nodes, is pruned using the assumption that SAT reduces to a tally set (or co-sparse set), so that only a polynomially bounded number of the nodes in the tree need to be examined.

Theorem 4.1

1. [Ber78] *If* SAT $\leq_m^p$*-reduces to a tally set, then* P = NP.
2. [For79] *If* $\overline{\mathrm{SAT}}$ $\leq_m^p$*-reduces to a sparse set, then* P = NP.

However, the ideas of Berman and Fortune directly did not work to resolve the sparseness conjecture. Finally, Mahaney settled the conjecture by proving his well-known result.

Theorem 4.2 [Mah82] *If* SAT $\leq_m^p$*-reduces to a sparse set, then* P = NP.

The proof of Mahaney's theorem was essentially based on the depth-first search with pruning of the self-reduction tree for SAT formulas which was used by Fortune in part 2 of Theorem 4.1. But the crux of the proof was a census argument. Given the exact census (up to some suitable length) of the sparse NP set to which SAT is assumed to reduce as advice information, Mahaney argued that a many-one reduction of SAT to the sparse set can be modified to a many-one reduction of $\overline{\mathrm{SAT}}$ to the sparse set. Since the census can take only polynomially many possible values the algorithm in Fortune's proof can be used repeatedly for each possible value of the census (one of which is the correct value) and, when run for the correct census value, it would detect the satisfiability of the input formula by constructing a satisfying truth assignment for it, where the truth assignment is determined by a root-to-leaf path in the self-reduction tree.

Around the same time but motivated more algorithmically, Karp, Lipton, and Sipser investigated the possibility of NP-complete sets being recognizable by nonuniform polynomial-size circuits. They obtained also a negative consequence of this assumption in the form of a collapse of the polynomial-time hierarchy PH to the second level.

Theorem 4.3 [KL80] *If* SAT *has nonuniform polynomial-size circuits (i.e.* SAT *is in* P/*poly), then* PH $= \Sigma_2^p$.

The results of Mahaney, and of Karp and Lipton tie up due to the following connection between polynomial-size circuits and sparse sets. It is known that the class of sets with nonuniform polynomial-size circuits coincides with the class of sets polynomial-time Turing (or even truth-table) reducible to sparse sets.

Theorem 4.4

1. [BK88] $R_T^p(\mathrm{SPARSE}) = R_{tt}^p(\mathrm{TALLY})$.

2. [BH77] $R_T^p(\mathrm{SPARSE}) = \mathrm{P}/poly$.

Interestingly, the existence of sparse sets that are *complete* for NP under polynomial-time Turing reductions implies a collapse of PH to Θ_2^p. This was proved by Kadin [Kad87], some years later, applying also a census argument. His argument, in a nutshell, is that the density function of a sparse set in NP can be computed making logarithmically many queries to a suitable NP oracle. Further, given specific values of the density function, an NP base machine accessing the sparse NP set as oracle can easily be modified to an NP machine without oracle which accepts the same language.

Theorem 4.5 [Kad87] *If there is a sparse Turing-complete set for* NP*, then* $\mathrm{PH} = \Theta_2^p$.

Immerman and Mahaney [IM89] showed that the result of Karp and Lipton is optimal for relativizable proof techniques. Thus, after the results of Mahaney and of Karp and Lipton, the natural question was for which reductions whose strengths lie between many-one and Turing reductions does the existence of sparse sets hard for NP imply P = NP. Several results followed in quick succession whose proofs are essentially based on the depth-first search with pruning technique of Fortune [For79]. We summarize these results below.

Theorem 4.6

1. [Yes83] *If* $\overline{\mathrm{SAT}}$ $\leq_{pos\text{-}bT}^p$*-reduces to a sparse set, then* P = NP.[1]

2. [Yes83] *If* SAT $\leq_{pos\text{-}bT}^p$*-reduces to a sparse* NP *set, then* P = NP.

3. [Ukk83, Yap83] *If* $\overline{\mathrm{SAT}}$ $\leq_c^p$*-reduces to a sparse set, then* P = NP.

4. [Yap83] *If* SAT $\leq_c^p$*- and* $\leq_d^p$*-reduces to a sparse* NP *set, then* P = NP.

The existing methods were not adequate to handle more flexible reducibilities, in particular the bounded Turing reducibility. After a gap of several years, the bounded Turing reducibility case was resolved in a breakthrough paper in the area by Ogiwara and Watanabe [OW91]. They showed that if there is a sparse set that is hard for NP under bounded Turing reductions then P = NP. Their proof exploits a new self-reducibility structure in certain NP sets called left sets. Given an NP set A and a polynomial-time computable relation associated with A, there is a corresponding left set *Left* (A) which is in NP. For any set $A \in \mathrm{NP}$ it holds that $A \leq_m^p Left\,(A)$. For NP-complete sets A it also holds that $Left\,(A) \leq_m^p A$.

[1] A $\leq_{pos\text{-}bT}^p$-reduction is a $\leq_{btt}^p$-reduction where the "formula" which evaluates the answers of the oracle is positive, i.e. it contains no negation symbol.

Theorem 4.7 [OW91] *For any set A in* NP, *if Left* (A) $\leq_{bT}^{p}$*-reduces to a sparse set, then Left* $(A) \in$ P. *Therefore, if* SAT $\leq_{bT}^{p}$*-reduces to a sparse set, then* P = NP.

The left set method turned out to be a powerful and convenient method to prove collapse results under the assumption that there is a sparse set that is hard for NP. In [AHH+93] Theorem 4.7 was extended to a more general reducibility.

Theorem 4.8

1. [AHH+93, RR92] *For any set A in* NP, *if Left* (A) $\leq_{c}^{p}$*-reduces to a sparse set, then Left* $(A) \in$ P. *Therefore, if* SAT $\leq_{c}^{p}$*-reduces to a sparse set, then* P = NP.
2. [RR92] *If* SAT $\leq_{m}^{co\text{-}rp}$*-reduces to a sparse set, then* RP = NP.
3. [AHH+93] *For any set A in* NP, *if Left* $(A) \in R_{bT}^{p}(R_{c}^{p}(\mathrm{SPARSE}))$, *then Left* $(A) \in$ P. *Therefore, if* SAT $\in R_{bT}^{p}(R_{c}^{p}(\mathrm{SPARSE}))$, *then* P = NP.

Saluja [Sal93] proved that the left-set technique cannot yield a collapse of P and NP under the assumption that NP $\subseteq R_{d}^{p}(\mathrm{SPARSE})$.

Finally, we take a brief look at consequences of other complexity classes like PP, $\mathrm{C_{=}P}$, PSPACE, UP, and $\mathrm{Mod}_k\mathrm{P}$ being reducible to sparse sets. It turns out that similar results as for NP will always hold for the complexity classes PP, $\mathrm{C_{=}P}$, and PSPACE. The now standard argument [OL91, AHH+93] for these classes is as follows: Assume that for some truth-table reducibility r, PP $\subseteq R_{r}^{p}(\mathrm{SPARSE})$ (respectively, *co*-$\mathrm{C_{=}P} \subseteq R_{r}^{p}(\mathrm{SPARSE})$), and that NP $\subseteq R_{r}^{p}(\mathrm{SPARSE})$ implies P = NP. Since PP and *co*-$\mathrm{C_{=}P}$ contain NP, P = NP follows. Further, PP and $\mathrm{C_{=}P}$ have complete sets that are one word-decreasing self-reducible [Ba90, OL91], and every one word-decreasing self-reducible set in $R_{T}^{p}(\mathrm{SPARSE})$ is in Σ_2^p [Ba90]. Thus, PP = P (respectively, $\mathrm{C_{=}P}$ = P) follows. The argument for the class PSPACE is similar using the result [KL80] that PSPACE $\subseteq R_{T}^{p}(\mathrm{SPARSE})$ implies PSPACE $\subseteq \Sigma_2^p$. We formalize this observation into the following general theorem.

Theorem 4.9 [OL91, AHH+93] *If for some truth-table reducibility r it holds that* NP $\subseteq R_{r}^{p}(\mathrm{SPARSE})$ *implies that* P = NP, *it follows for any class* $\mathcal{K} \in \{\mathrm{PP}, \text{co-}\mathrm{C_{=}P}, \mathrm{PSPACE}\}$ *that* $\mathcal{K} \subseteq R_{r}^{p}(\mathrm{SPARSE})$ *implies* $\mathcal{K} = \mathrm{P}$.

The above theorem yields the following results (from [OL91, AHH+93]) as a direct consequence of the corresponding results for NP.

Corollary 4.10

1. [OL91] *For any class* $\mathcal{K} \in \{\mathrm{PP}, \mathrm{C_{=}P}, \mathrm{PSPACE}\}$, *if* $\mathcal{K} \subseteq R_{bT}^{p}(\mathrm{SPARSE})$ *then* $\mathcal{K} = \mathrm{P}$.
2. [AHH+93] *For any* $\mathcal{K} \in \{\mathrm{PP}, \mathrm{C_{=}P}, \mathrm{PSPACE}\}$, *if* $\mathcal{K} \subseteq R_{bT}^{p}(R_{c}^{p}(\mathrm{SPARSE}))$ *then* $\mathcal{K} = \mathrm{P}$.

For the $\mathrm{Mod}_k\mathrm{P}$ classes, there are similar results exploiting the special word-decreasing self-reducibility structure of certain complete sets for these classes [OL91, AHH+93].

Theorem 4.11

1. [OL91] *For all* $k \geq 2$*, if* $\mathrm{Mod}_k\mathrm{P} \subseteq R^p_{bT}(\mathrm{SPARSE})$ *then* $\mathrm{Mod}_k\mathrm{P} = \mathrm{P}$.

2. [AHH+93] *For all* $k \geq 2$*, if* $\mathrm{Mod}_k\mathrm{P} \subseteq R^p_{c}(\mathrm{SPARSE})$ *then* $\mathrm{Mod}_k\mathrm{P} = \mathrm{P}$.

It is an open question whether $\mathrm{Mod}_k\mathrm{P} \subseteq R^p_{bT}(R^p_c(\mathrm{SPARSE}))$ implies $\mathrm{Mod}_k\mathrm{P} = \mathrm{P}$.

4.2 The complexity of small descriptions

If a set A is reducible to a sparse set, does it follow that A is reducible to some sparse set that has a "simple" description relative to A? In this subsection we discuss this well-studied question and state applications (in the form of collapse results) of certain specific answers to this question. This study originates in the notions of equivalence and reducibility to sparse sets (see for example [TB91, AHOW92, GW93]). It concerns the complexity (relative to A) of small descriptions for sets A which are reducible to sparse sets. Gavaldà and Watanabe [GW93] obtained an important lower bound for the case of Turing reductions by constructing a set B that is Turing reducible to a sparse set but is not Turing reducible to any sparse set in $\mathrm{NP}(B) \cap \text{co-NP}(B)$. This result implies that the class of sets Turing equivalent to some sparse set is a proper subclass of $\mathrm{P}/poly$ resolving what was a long-standing open question.

Theorem 4.12 [GW93] *There is a set* B *that is in* $R^p_T(\mathrm{SPARSE})$ *but is not Turing reducible to a sparse set in* $\mathrm{NP}(B) \cap \text{co-NP}(B)$.

The separation of equivalence and reduction classes for restricted truth-table reducibilities is further investigated in [AHOW92]. In a broader setting it is of interest to know for various classes of sets that reduce to sparse sets, the complexity of the easiest sparse sets to which such sets reduce. This question is first investigated in [AHH+93] where upper bounds for the relative complexity of sparse descriptions are proved for certain truth-table reducibilities.

Theorem 4.13 [AHH+93] *Any set* A *that disjunctively reduces (respectively, bounded disjunctively reduces, 2-truth-table reduces) to a sparse set in fact disjunctively reduces (respectively, bounded disjunctively reduces, 2-truth-table reduces) to a sparse set that is in* $\Delta^p_2(A)$ *(respectively,* $\Theta^p_2(A)$, $\Theta^p_2(A)$*).*

The notion of small descriptions is formalized in [GW93, AKM]. Let $\leq_r$ be a reducibility. A sparse set S is a sparse r-description for a set A if $A \leq_r S$. For every set A in $R_r(\mathrm{SPARSE})$, we are interested in finding upper bounds for the complexity of sparse r-descriptions relative to A. A sparse r-description satisfying the established upper bound is called a *simple* sparse r-description (with respect to that upper bound).

Simple sparse descriptions for a set A can be used to derive lowness properties for A. In order to prove the lowness of a set A that reduces to a sparse set, first a suitable bound for the complexity of a sparse description for A is derived. Using this description, a deterministic enumeration technique similar to that of Mahaney [Mah82] or a census technique similar to that of Kadin [Kad87] is used to replace the sparse oracle S (and thus A). For self-reducible sets stronger (unrelativized) simplicity results and consequently, unrelativized lowness results can be derived. In [AKM] this approach is extensively used and several new lowness results are proved for sets that reduce to sparse sets for reducibilities of different strengths. These lowness results are based on suitably obtained simple descriptions for the concerned set. We state some of these results that in turn yield collapse results.

Theorem 4.14

1. [AKM] *For every set* $A \in R_h^p(\mathrm{SPARSE})$ *there is a sparse set* $S \in \mathrm{NP}(A)$ *such that* $A \in R_h^p(S)$.

2. [AHH+93] *For every set* $A \in R_d^p(\mathrm{SPARSE})$ *there is a sparse set in* $\Delta_2^p(A)$ *to which* A *disjunctively reduces.*

Theorem 4.15 [AKM]

1. *For every word-decreasing*[2] *self-reducible set* A *in* $R_d^p(\mathrm{SPARSE})$ *there exists an* $\mathrm{FP}^{\mathrm{NP}}$*-printable sparse set* S *such that* $A \leq_d^p S$ *and thus* A *is low for* Δ_2^p.

2. *For every word-decreasing self-reducible set* A *in* $R_h^p(\mathrm{SPARSE})$ *there exists an* $\mathrm{FP}^{\mathrm{NP}}$*-printable sparse set* S *such that* $A \leq_h^p S$ *and thus* A *is low for* Δ_2^p.

Theorem 4.15 yields the following interesting collapse results.

Theorem 4.16 [AKM]

1. *If an* NP*-complete set monotonously reduces to a sparse set then* $\mathrm{PH} = \Delta_2^p$.

2. *If an* NP*-complete set disjunctively reduces to a sparse set then* $\mathrm{PH} = \Delta_2^p$.

A skeletal inclusion structure between some subclasses of $R_T^p(\mathrm{SPARSE})$ is given in Figure 1 (see [BK88, Ko89, AHOW92, AHH+93, BLS93, GW93, Mu93] for results). The inclusion structure is interesting in the light of different collapse consequences for the different subclasses of $R_T^p(\mathrm{SPARSE})$ (assuming that NP is contained in that subclass).

[2]These self-reducibility results hold for a notion of self-reducibility defined in [AKM] which generalizes the word-decreasing and polynomially related self-reducibilities defined in [Ba90] and [Ko83] respectively.

5 Collapses

In this section we consider monotonous reductions composed with randomized reductions to sparse sets. As a consequence of the main theorem we show that if a bounded Turing hard set for NP reduces to a sparse set via a co-rp many-one reduction, then NP = RP. This extends the result proved in [RR92] that NP $\subseteq R_m^{co\text{-}rp}$(SPARSE) implies NP = RP.

For the proof we need the following folklore result on amplifying randomized reductions having one-sided error.

Lemma 5.1 *If $A \leq_m^{co\text{-}rp} B$ then for every polynomial p there exist a polynomial-time function f computing sets of strings and a polynomial q such that*

$$\begin{aligned} x \in A &\Rightarrow \text{Prob}[f(x,w) \subseteq B] = 1, \text{ and} \\ x \notin A &\Rightarrow \text{Prob}[f(x,w) \subseteq B] \leq 2^{-p(|x|)}, \end{aligned}$$

where w is chosen uniformly at random from the set $\Sigma^{q(|x|)}$.

Now we are ready to prove our main result. Suppose that an NP-complete set A is in $R^p_{k(n)\text{-}h}(R_m^{co\text{-}rp}(B))$ for some set B. The theorem below brings out an interesting trade-off between the number $k(n)$ of queries in the monotonous reduction and the density of the set B. (If $A \leq^p_{k(n)\text{-}h} B$ we say that A $k(n)$-monotonous reduces to B.)

Theorem 5.2 *Let k, c_B be non-decreasing, polynomial-time functions. If there exists a set B such that $density_B(n) \leq c_B(n)$ and $R_m^{co\text{-}rp}(B)$ contains a $k(n^{O(1)})$-monotonous hard set for* NP, *then*

$$\text{NP} \subseteq \bigcup_{j \geq 0} \text{RTIME}(n^j \cdot c_B(n^j)^{O(k(n^j))}).$$

Proof Let A be some NP set, and let q be a polynomial and P_A be a polynomial-time set such that $A = \{x \mid \exists w \in \Sigma^{q(|x|)} : \langle x, w\rangle \in P_A\}$. Let $w_{max}(x)$ denote the lexicographically greatest $w \in \Sigma^{q(|x|)}$ such that $\langle x, w\rangle \in P_A$. We apply the left set technique developed in [OW91] and adapted for Hausdorff representations in [AHH+93] combined with probability amplification to device a randomized algorithm that on input $x \in A$ computes $w_{max}(x)$ with high probability. As in the deterministic setting the algorithm performs a breadth-first search through the tree of witness prefixes for an input x. More specifically, let

$$L(A) = \{\langle x, y\rangle \mid \exists u, v : |y| = |u|, y \leq u, \langle x, uv\rangle \in P_A\}$$

be the set of all pairs $\langle x, y\rangle$ such that $x \in A$ and y is lexicographically smaller than the length $|y|$ prefix of w_{max}. Since $L(A)$ is in NP it follows by the assumption of the theorem and by Proposition 3.5 that $L(A)$ is reducible to some set in $R_m^{co\text{-}rp}(B)$ via a Hausdorff reduction that on input $\langle x, y\rangle$, asks $k(|x|^c)$ queries of length at most $|x|^c$ for an FP function k and a suitable constant c. Our algorithm uses the information provided by the reduction of $L(A)$ to B to eliminate with very high probability only such prefixes that don't lead to $w_{max}(x)$:

input x, $|x| = n$
$N := \{\varepsilon\}$
for $i := 1$ **to** $q(n)$ **do**

- Expand N to $\{y0 \mid y \in N\} \cup \{y1 \mid y \in N\}$
- In case the size of N exceeds $(c_B(n^c)+1)^{k(n^c)+1}$ use the randomized procedure described below to prune N back to that size retaining the length i prefix of w_{max} with very high probability

end
if there is a $w \in N$ such that $\langle x, w\rangle \in P_A$ **then** *accept* **else** *reject* **end**

It is clear that the algorithm rejects every instance $x \notin A$ with probability 1. The main part of the proof consists in implementing the randomized pruning procedure such that the algorithm accepts every instance $x \in A$ with probability at least 3/4.

Let I_n denote the index set $\{1, \ldots, k(n^c)\}$, and for any $i \in I_n \cup \{0\}$ let $I_n^{\le i}$ ($I_n^{>i}$) denote the subset $\{j \in I_n \mid j \le i\}$ (resp., $\{j \in I_n \mid j > i\}$). Further, let p be a polynomial such that for all n, $(1-2^{-p(n)})^{q(n)\cdot k(n^c)} \ge 3/4$. From the definition of the Hausdorff reducibility and using Lemma 5.1 it follows that there is a polynomial s and a polynomial-time function f such that for all x, y, $|x| = n$,

- there exists an $i \in I_n \cup \{0\}$ such that for all $j \in I_n^{\le i}$, $\mathrm{Prob}[f(j,x,y,w) \subseteq B] = 1$ and for all $j \in I_n^{>i}$, $\mathrm{Prob}[f(j,x,y,w) \subseteq B] \le 2^{-p(n)}$,
- $\langle x, y\rangle \in L(A)$ if and only if i is odd,

where w is chosen uniformly at random from $\Sigma^{s(n)}$. Moreover, by combining $f(j,x,y,w)$ with all the queries in the sets $f(l,x,y,w)$, $l \le j$, we can assume that $f(j,x,y,w) \subseteq B$ implies $f(j-1,x,y,w) \subseteq B$, for $j = 2, \ldots, k(n^c)$.

In the sequel let x be an arbitrary but fixed instance in A. For simplicity, we denote $w_{max}(x)$ by w_{max} and $|x|$ by n. Let $N = \{y_1, \ldots, y_t\}$ be a lexicographically ordered set of prefixes (all of same length) that includes the prefix, say y_h, of w_{max} of that length. We use some crucial properties of the function f for the design of a randomized procedure that prunes N to a subset of size at most $(c_B(n^c)+1)^{k(n^c)+1}$, and retains y_h with probability at least $(1-2^{-p(n)})^{k(n^c)}$.

PRUNE(x, N), $N = \{y_1, \ldots, y_t\}$
guess randomly $w_1, \ldots, w_t \in \Sigma^{s(n)}$
for $i := 1$ **to** $k(n^c)$ **do**
  compute an index set $J_i \subseteq \{1, \ldots, t\}$ of candidates for h,
  where h is the index of the prefix y_h of w_{max} in N
end
return $\{y_j \mid j \in \bigcup_{i=1}^{k(n^c)} J_i\}$

The above procedure computes for every $i = 1, \ldots, k(n^c)$ an index set J_i of size at most $(c_B(n^c)+1)^i$ such that h is contained in some J_i with probability at least

$(1-2^{-p(n)})^{k(n^c)}$. Let $J_0 = \{0\}$, then the sets J_i, $i = 1, \ldots, k(n^c)$ are computed as follows. If i is odd,

$J_i := \emptyset$
for each $j \in J_{i-1}$ **do**
  $Q := \emptyset$
  for $k := j+1$ **to** t **do**
    if $f(i,x,y_k,w_k) \not\subseteq \bigcup_{l=j+1}^{k-1} f(i,x,y_l,w_l)$ **and** $\|Q\| \le c_B(n^c)$ **then**
      $J_i := J_i \cup \{k-1\}$
      $Q := Q \cup f(i,x,y_k,w_k)$
    end
  end
  if $\|Q\| \le c_B(n^c)$ **then** $J_i := J_i \cup \{t\}$ **end**
end

and if i is even,

$J_i := \emptyset$
for each $j \in J_{i-1}$ **do**
  $Q := \emptyset$
  for $k := j-1$ **downto** 1 **do**
    if $f(i,x,y_k,w_k) \not\subseteq \bigcup_{l=k+1}^{j-1} f(i,x,y_l,w_l)$ **and** $\|Q\| \le c_B(n^c)$ **then**
      $J_i := J_i \cup \{k\}$
      $Q := Q \cup f(i,x,y_k,w_k)$
    end
  end
  if $\|Q\| \le c_B(n^c)$ **then** $J_i := J_i \cup \{0\}$ **end**
end

Claim *The set returned by procedure* PRUNE *is of size at most* $(c_B(n^c)+1)^{k(n^c)+1}$ *and contains* y_h *with probability at least* $(1-2^{-p(n)})^{k(n^c)}$.

Proof It is straightforward to show that $\|J_i\| \le (\|Q\|+1)\|J_{i-1}\|$ for $i = 1, \ldots, k(n^c)$, and thus the cardinality of $\bigcup_{i=1}^{k(n^c)} J_i$ is at most $(c_B(n^c)+1)^{k(n^c)+1}$.

The strategy behind the computation of the index sets J_i is as follows. Let r_1 be the maximum index r such that $\mathrm{Prob}[f(1,x,y_k,w) \subseteq B] = 1$ for all $k = 1, \ldots, r$. Since for $k = 1, \ldots, h$, the pair $\langle x, y_k\rangle$ is in $L(A)$ it follows by the properties of the Hausdorff reducibility that $h \le r_1 \le t$. If $r_1 = t$ then $f(1,x,y_k,w_k) \subseteq B$ for $k = 1, \ldots, t$, and thus $r_1 = t$ is included into J_1 with probability 1. Otherwise, if $r_1 < t$ then with probability at least $1 - 2^{-p(n)}$, the string w_{r_1+1} is chosen such that $f(1,x,y_{r_1+1},w_{r_1+1}) \not\subseteq B$. Since $f(1,x,y_k,w_k) \subseteq B$ for $k = 1, \ldots, r_1$, i.e. $\|\bigcup_{k=1}^{r_1} f(i,x,y_k,w_k)\| \le c_B(n^c)$, but $f(1,x,y_{r_1+1},w_{r_1+1}) \not\subseteq B$, it follows that r_1 is included into J_1 with probability at least $1-2^{-p(n)}$. Now, if $r_1 = h$ then the probability that $r_1 = h$ is included into J_1 is at least $1-2^{-p(n)}$.

Otherwise, if $r_1 > h$ then assume that the algorithm includes r_1 into J_1, and let l_2 be the least index l such that $\mathrm{Prob}[f(2,x,y_k,w) \subseteq B] = 1$ for all $k = l,\ldots,r_1$. Since for $k = h+1,\ldots,r_1$, the pair $\langle x, y_k\rangle$ is not in $L(A)$ it follows by the properties of the Hausdorff reducibility that $1 \le l_2 \le h+1$. If $l_2 = 1$ then $f(2,x,y_k,w_k) \subseteq B$ for $k = 1,\ldots,r_1$, and thus $l_2 - 1 = 0$ is included into J_2 with probability 1. Otherwise, if $l_2 > 1$ then with probability at least $1-2^{-p(n)}$, the string w_{l_2-1} is chosen such that $f(1,x,y_{l_2-1},w_{l_2-1}) \not\subseteq B$. Since $f(2,x,y_k,w_k) \subseteq B$ for $k = l_2,\ldots,r_1$, i.e. $\|\bigcup_{k=l_2}^{r_1} f(i,x,y_k,w_k)\| \le c_B(n^c)$, but $f(2,x,y_{l_2-1},w_{l_2-1}) \not\subseteq B$, it follows that $l_2 - 1$ is included into J_2 with probability at least $1-2^{-p(n)}$. Now, if $l_2 - 1 = h$ then the probability that r_1 and $l_2 - 1 = h$ are included into J_1 and J_2, respectively, is at least $(1-2^{-p(n)})^2$.

In general, if $i = 2j$ $(i = 2j+1)$ and $l_i - 1 < h$ (resp., $r_i > h$) then assume that $r_1, l_2, \ldots, l_i$ (resp., $r_1, l_2, \ldots, r_i$) were included into $J_1, J_2, \ldots, J_i$, respectively, and let r_{i+1} (resp., l_{i+1}) be the maximum index r (resp., minimum index l) such that $\mathrm{Prob}[f(i+1,x,y_k,w) \subseteq B] = 1$ for all $k = l_i,\ldots,r$ (resp., for all $k = l,\ldots,r_i$). Since for $k = l_i,\ldots,h$ (resp., for $k = h+1,\ldots,r_i$), the pair $\langle x,y_k\rangle$ is (resp., is not) in $L(A)$, it follows by the properties of the Hausdorff reducibility that $h \le r_{i+1} \le r_{i-1}$ (resp., $l_{i-1} \le l_{i+1} \le h+1$). If $r_{i+1} = r_{i-1}$ (resp., $l_{i+1} = l_{i-1}$) then r_{i+1} (resp., l_{i+1}) is included into J_{i+1} with probability 1. Otherwise, with probability at least $1-2^{-p(n)}$, the string $w_{r_{i+1}+1}$ (resp., $w_{l_{i+1}-1}$) is chosen such that $f(i+1,x,y_{r_{i+1}+1},w_{r_{i+1}+1}) \not\subseteq B$ (resp., $f(i+1,x,y_{l_{i+1}-1},w_{l_{i+1}-1}) \not\subseteq B$). Thus it follows that r_{i+1} (resp., l_{i+1}) is included into J_{i+1} with probability at least $1-2^{-p(n)}$, implying that the probability that $r_1, l_2, \ldots, r_{i+1}$ (resp., $r_1, l_2, \ldots, l_{i+1}-1$) are included into $J_1, J_2, \ldots, J_{i+1}$, respectively, is at least $(1-2^{-p(n)})^{i+1}$.

This completes the proof of the claim since by the properties of the Hausdorff reducibility it holds for some $i \le k(n^c)$ that $h = r_i$ or $h = l_i - 1$, depending on i being odd or even. □

By the Claim, the set N contains w_{max} after the execution of the for-loop in the main program with probability at least $(1-2^{-p(n)})^{k(n^c)q(n)}$, which is more than $3/4$ by the choice of p. Finally, to check the running time of the algorithm, observe that the main for-loop is executed $q(n)$ times, and that the size of N never exceeds $2\cdot(c_B(n^c)+1)^{k(n^c)+1}$. ■

From Theorem 5.2 we can derive the following immediate consequences.

Corollary 5.3 *If* NP *is contained in* $R^p_{bT}(R^{co\text{-}rp}_m(\mathrm{SPARSE}))$, *then* NP = RP *and* PH ⊆ BPP.

Proof Since by Theorem 3.8, $R^p_{bhd}(R^{co\text{-}rp}_m(\mathrm{SPARSE})) = R^p_{bT}(R^{co\text{-}rp}_m(\mathrm{SPARSE}))$, NP = RP follows directly from Theorem 5.2. That NP ⊆ BPP implies PH ⊆ BPP is stated in [Ko82]. ■

Along the same lines as Theorem 5.2 (but without the probability analysis) we can prove the following trade-off result.

Theorem 5.4 *Let k, c_B be non-decreasing, polynomial-time functions. If there exists a set B such that $density_B(1^n) \leq c_B(n)$ and $R_c^p(B)$ contains a $k(n)^{O(1)}$-monotonous hard set for* NP*, then*

$$\mathrm{NP} \subseteq \bigcup_{j \geq 0} \mathrm{DTIME}(n^j \cdot c_B(n^j)^{O(k(n^j))}).$$

Theorems 5.2 and 5.4 yield the following corollaries which are similar to the results in [HL91] regarding truth-table reductions of NP-complete sets to sets of different densities.

Corollary 5.5 *If B is a set of density $O(\log n)$ such that an* NP*-complete set is reducible to a set in $R_c^p(B)$ (resp., $R_m^{co\text{-}rp}(B)$) by a $O(\log n)/\log(\log n)$-monotonous reduction then* P = NP *(resp.,* RP = NP*).*

Corollary 5.6 *If an* NP*-complete set is reducible to a set in R_c^p(SPARSE) (resp., $R_m^{co\text{-}rp}$(SPARSE)) by a $O(\log n)$-monotonous reduction then* $\mathrm{NP} \subseteq \mathrm{DTIME}(2^{O(\log^2 n)})$ *(resp.,* $\mathrm{NP} \subseteq \mathrm{RTIME}(2^{O(\log^2 n)})$*).*

An interesting point to note in the above corollaries is that the number of queries in the conjunctive reduction is unbounded and it plays no role in the trade-off. The trade-off is purely between the density of B and the number of queries in the monotonous reductions.

Finally, we consider consequences for $\mathcal{K} \in \{\mathrm{PP}, \mathrm{PSPACE}, \mathrm{C_=P}\}$ being contained in $R_{bT}^p(R_m^{co\text{-}rp}(\mathrm{SPARSE}))$. Using the facts that $R_{bT}^p(R_m^{co\text{-}rp}(\mathrm{SPARSE})) \subseteq R_T^p(\mathrm{SPARSE})$, and $\mathcal{K} \subseteq R_T^p(\mathrm{SPARSE})$ implies $\mathcal{K} \subseteq \Sigma_2^p$ (see Section 4), the following theorem is obtained as a consequence of Corollary 5.3.

Theorem 5.7 *For $\mathcal{K} \in \{\mathrm{PP}, \mathrm{PSPACE}, \mathrm{C_=P}\}$, if a bounded Turing hard set for $\mathcal{K}$ co-rp many-one reduces to a sparse set then $\mathcal{K} \subseteq$* BPP.

6 Promise problems and randomized reductions to sparse sets

In this section we investigate consequences of some solution of the promise problem (1SAT, SAT) reducing to a sparse set, where 1SAT is the set of boolean formulas having at most one satisfying assignment. In particular, we show that no solution of the promise problem (1SAT, SAT) bounded Turing reduces to a set in $R_m^{co\text{-}rp}$(SPARSE) unless NP = RP. We first give the definition of promise problems and state its relation to randomized reductions.

Definition 6.1 [ESY84] *A promise problem is a pair of sets (Q,R). A set L is called a solution of the promise problem (Q,R) if for all $x \in Q$, $x \in L \Leftrightarrow x \in R$.*

Observe that a solution for the promise problem $(1\mathrm{SAT}, \mathrm{SAT})$ has to agree with SAT in the formulas having a unique satisfying assignment as well as in the unsatisfiable formulas. Let USAT be the set of formulas having a unique satisfying assignment. The well known result of Valiant and Vazirani stating the NP-hardness of USAT under (a different kind of) randomized reductions [VV86] has the following implication for the promise problem $(1\mathrm{SAT}, \mathrm{SAT})$.

Theorem 6.2 [VV86] *If there is a solution of the promise problem* $(1\mathrm{SAT}, \mathrm{SAT})$ *in* RP *then* $\mathrm{NP} = \mathrm{RP}$.

We now improve Corollary 5.3 by weakening the assumption that NP is contained in $R^p_{bT}(R^{co\text{-}rp}_m(\mathrm{SPARSE}))$.

Theorem 6.3 *If there is a solution in* $R^p_{bT}(R^{co\text{-}rp}_m(\mathrm{SPARSE}))$ *for the promise problem* $(1\mathrm{SAT}, \mathrm{SAT})$ *then* $\mathrm{NP} = \mathrm{RP}$.

Proof Let $L \in R^p_{bT}(R^{co\text{-}rp}_m(\mathrm{SPARSE})) = R^p_{bhd}(R^{co\text{-}rp}_m(\mathrm{SPARSE}))$ be a solution of the promise problem $(1\mathrm{SAT}, \mathrm{SAT})$. Then we have for all $x \in 1\mathrm{SAT}$, $x \in L \Leftrightarrow x \in \mathrm{SAT}$. The natural (prefix) left set associated with SAT is the set

$$L(\mathrm{SAT}) = \{\langle x, y\rangle \mid \exists u, v : |y| = |u|, y \leq u, uv \text{ is a satisfying assignment for } x\}$$

of all pairs $\langle x, y\rangle$ such that $x \in \mathrm{SAT}$ and y is lexicographically smaller than the length $|y|$ prefix of the maximum satisfying assignment for x. We first show that the promise problem $(Q, L(\mathrm{SAT}))$ has a solution $L' \in R^p_{bhd}(R^{co\text{-}rp}_m(\mathrm{SPARSE}))$, where $Q = \{\langle x, y\rangle \mid x \in 1\mathrm{SAT}\}$.

By the definition of $L(\mathrm{SAT})$ it is clear that $L(\mathrm{SAT})$ is accepted by some NP machine which on inputs $\langle x, y\rangle$, $x \in 1\mathrm{SAT}$, has at most one accepting path. Thus there is a (parsimonious) many-one reduction function g from $L(\mathrm{SAT})$ to SAT such that $g(x, y) \in 1\mathrm{SAT}$ for all pairs $\langle x, y\rangle$ for which $x \in 1\mathrm{SAT}$. Now define $L' = \{\langle x, y\rangle \mid g(x, y) \in L\}$. Clearly g many-one reduces L' to L, implying that $L' \in R^p_{bhd}(R^{co\text{-}rp}_m(\mathrm{SPARSE}))$.

Furthermore, since L is a solution of $(1\mathrm{SAT}, \mathrm{SAT})$, and since for all $\langle x, y\rangle \in Q$, $g(x, y) \in 1\mathrm{SAT}$, it follows for all $\langle x, y\rangle \in Q$ that $g(x, y) \in L$ if and only if $g(x, y) \in \mathrm{SAT}$. Since g many-one reduces both $L(\mathrm{SAT})$ to SAT and L' to L, we have for all $\langle x, y\rangle \in Q$ that $\langle x, y\rangle \in L' \Leftrightarrow \langle x, y\rangle \in L(\mathrm{SAT})$, i.e., L' is a solution for the promise problem $(Q, L(\mathrm{SAT}))$.

Consider a modification of the algorithm described in the proof of Theorem 5.2 which uses the reduction of L' to a sparse set B (instead of $L(\mathrm{SAT})$ to B) to guide the search for the maximum satisfying assignment w_{max}. We claim that on input $x \in 1\mathrm{SAT} \cap \mathrm{SAT}$ this algorithm computes with high probability the unique satisfying assignment w_{max} for x. This is a consequence of the fact that on input $x \in 1\mathrm{SAT} \cap \mathrm{SAT}$ the algorithm considers only pairs $\langle x, y\rangle$ in Q, implying that $\langle x, y\rangle \in L' \Leftrightarrow \langle x, y\rangle \in L(\mathrm{SAT})$. Hence the set accepted by the algorithm is an RP solution for the promise problem $(1\mathrm{SAT}, \mathrm{SAT})$ and by Theorem 6.2 it follows that $\mathrm{NP} = \mathrm{RP}$. ∎

Regarding the possible existence of solutions for $(1\mathrm{SAT}, \mathrm{SAT})$ in the deterministic reduction class $R^p_{bT}(R^p_c(\mathrm{SPARSE}))$ we get the following result.

Theorem 6.4 *If there is a solution of* $(1\mathrm{SAT}, \mathrm{SAT})$ *in* $R^p_{bT}(R^p_c(\mathrm{SPARSE}))$ *then* $(1\mathrm{SAT}, \mathrm{SAT})$ *has a solution in* P*, implying that* Few = P *and* USAT $\in$ co-NP*.*

Proof The first implication follows along the lines of the previous theorem. The consequence Few = P follows from [To91] using the containment Few $\subseteq \mathrm{P}^{\mathrm{FewP}}$ [KSTT], and USAT $\in$ co-NP follows from [JT93]. ∎

References

[AM77] L. Adleman and K. Manders. Reducibility, randomness, and intractability. *Proc. 9th ACM Symp. on Theory of Computing*, 1977, 151-163.

[AHOW92] E. Allender, L. Hemachandra, M. Ogiwara, and O. Watanabe. Relating equivalence and reducibility to sparse sets. *SIAM Journal on Computing*, 21(3):529-539, 1992.

[AHH+93] V. Arvind, Y. Han, L.A. Hemachandra, J. Köbler, A. Lozano, M. Mundhenk, M. Ogiwara, U. Schöning, R. Silvestri, and T. Thierauf. Reductions to sets of low information content. In *Complexity Theory, Current Research*, Cambridge University Press, 1–45, 1993.

[AKM92] V. Arvind, J. Köbler, and M. Mundhenk. On bounded truth-table, conjunctive, and randomized reductions to sparse sets. *Proc. 12th Conf. FSTTCS,* Lecture Notes in Computer Science, #652:140-151, Springer Verlag, 1992.

[AKM] V. Arvind, J. Köbler, and M. Mundhenk. Upper bounds on the complexity of sparse and tally descriptions. *Mathematical Systems Theory,* to appear.

[Ba90] J. Balcázar. Self-reducibility. *Journal of Computer and System Sciences*, 41:367-388, 1990.

[BDG] J.L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I,II.* EATCS Monographs on Theoretical Computer Science, Springer Verlag, 1988.

[Ber78] P. Berman. Relationship between density and deterministic complexity of NP-complete languages. *Proceedings of the 5th International Colloquium on Automata, Languages, and Programming*, Lecture Notes in Computer Science, #62:63–71, Springer Verlag, 1978.

[BH77] L. Berman and J. Hartmanis. On isomorphisms and density of NP and other complete sets. *SIAM Journal on Computing*, 6(2):305–322, 1977.

[BK88] R. Book and K. Ko. On sets truth-table reducible to sparse sets. *SIAM Journal on Computing*, 17(5):903-919, 1988.

[BLS93] H. Buhrman, L. Longpré, and E. Spaan. Sparse reduces conjunctively to tally. *Proceedings of the 8th Structure in Complexity Theory Conference*, IEEE Computer Society Press, 1993.

[CKR91] R. Chang, J. Kadin, and P. Rohatgi. Connections between the complexity of unique satisfiability and the threshold behavior of randomized reductions. *Proceedings of the 6th Structure in Complexity Theory Conference*, 255-269, IEEE Computer Society Press, 1991.

[ESY84] S. Even, A. Selman, and Y. Yacobi. The complexity of promise problems with applications to public-key cryptography. *Information and Control*, 61:114-133, 1984.

[For79] S. Fortune. A note on sparse complete sets. *SIAM Journal on Computing*, 8(3):431–433, 1979.

[GW93] R. Gavaldà and O. Watanabe. On the computational complexity of small descriptions. *SIAM Journal on Computing*, December 1993, to appear.

[Gi77] J. Gill. Computational complexity of probabilistic complexity classes. *SIAM Journal on Computing*, 6:675-695, 1977.

[HL91] S. Homer and L. Longpré. On reductions of NP sets to sparse sets. *Proc. 6th Structure in Complexity Theory Conference*, 79–88. IEEE Computer Society Press, 1991.

[HOW92] L. Hemachandra, M. Ogiwara, and O. Watanabe. How hard are sparse sets? *Proc. 7th Structure in Complexity Theory Conference*. IEEE Computer Society Press, 1992.

[IM89] N. Immerman and S. Mahaney. Relativizing relativized computations. *Theoretical Computer Science*, 68:267–276, 1989.

[JT93] B. Jenner and J. Torán. Computing functions with parallel queries to NP. *Proceedings of the 8th Structure in Complexity Theory Conference*, 280-291. IEEE Computer Society Press, May 1993.

[Kad87] J. Kadin. $\mathrm{P}^{\mathrm{NP}[\log n]}$ and sparse Turing-complete sets for NP. *Journal of Computer and System Sciences*, 39(3):282-298, 1989.

[KL80] R. Karp and R. Lipton. Some connections between nonuniform and uniform complexity classes. *Proceedings of the 12th ACM Symposium on Theory of Computing,* 302-309, 1980.

[Ko82] K. Ko. Some observations on the probabilistic algorithms and NP-hard problems. *Information Processing Letters* 14:39-43, 1982.

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

[Ko89] K. Ko. Distinguishing conjunctive and disjunctive reducibilities by sparse sets. *Information and Computation*, 81(1):62–87, 1989.

[KSTT] J. Köbler, U. Schöning, S. Toda, J. Torán. Turing machines with few accepting computations and low sets for PP. *Journal of Computer and System Sciences*, 44(2):272-286, 1992.

[LLS75] R. Ladner, N. Lynch, and A. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science*, 1(2):103-124, 1975.

[Mah82] S. Mahaney. Sparse complete sets for NP: Solution of a conjecture of Berman and Hartmanis. *Journal of Computer and System Sciences*, 25(2):130-143, 1982.

[Mu93] M. Mundhenk. Hausdorff-Reduktionen zu Mengen mit geringem Informationsgehalt. *PhD dissertation, Universität Ulm*, 1993.

[OL91] M. Ogiwara and A. Lozano. On one query self-reducible sets. *Theoretical Computer Science,* 112:255-276, 1993.

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

[RR92] D. Ranjan and P. Rohatgi. Randomized reductions to sparse sets. *Proceedings of the 7th Structure in Complexity Theory Conference*, IEEE Computer Society Press, 239-242, 1992.

[Sal93] S. Saluja. Relativized limitations of the left set technique and closure classes of sparse sets. *Proceedings of the 8th Structure in Complexity Theory Conference*, 215-222. IEEE Computer Society Press, 1993.

[Sch93] U. Schöning. On random reductions from sparse sets to tally sets. *Information Processing Letters*, 46:239-241, 1993.

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

[TB91] S. Tang and R. Book. Reducibilities on tally and sparse sets. *Theoretical Informatics and Applications*, 25:293-302, 1991.

[To91] S. Toda. On polynomial time truth-table reducibilities of intractable sets to P-selective sets. *Mathematical Systems Theory*, 24(2):69-82, 1991.

[Ukk83] E. Ukkonen. Two results on polynomial time truth-table reductions to sparse sets. *SIAM Journal on Computing*, 12(3):580–587, 1983.

[VV86] L.G. Valiant and V.V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.

[Wag87] K.W. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science*, 51:53-80, 1987.

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

[Yes83] Y. Yesha. On certain polynomial-time truth-table reducibilities of complete sets to sparse sets. *SIAM Journal on Computing*, 12(3):411–425, 1983.

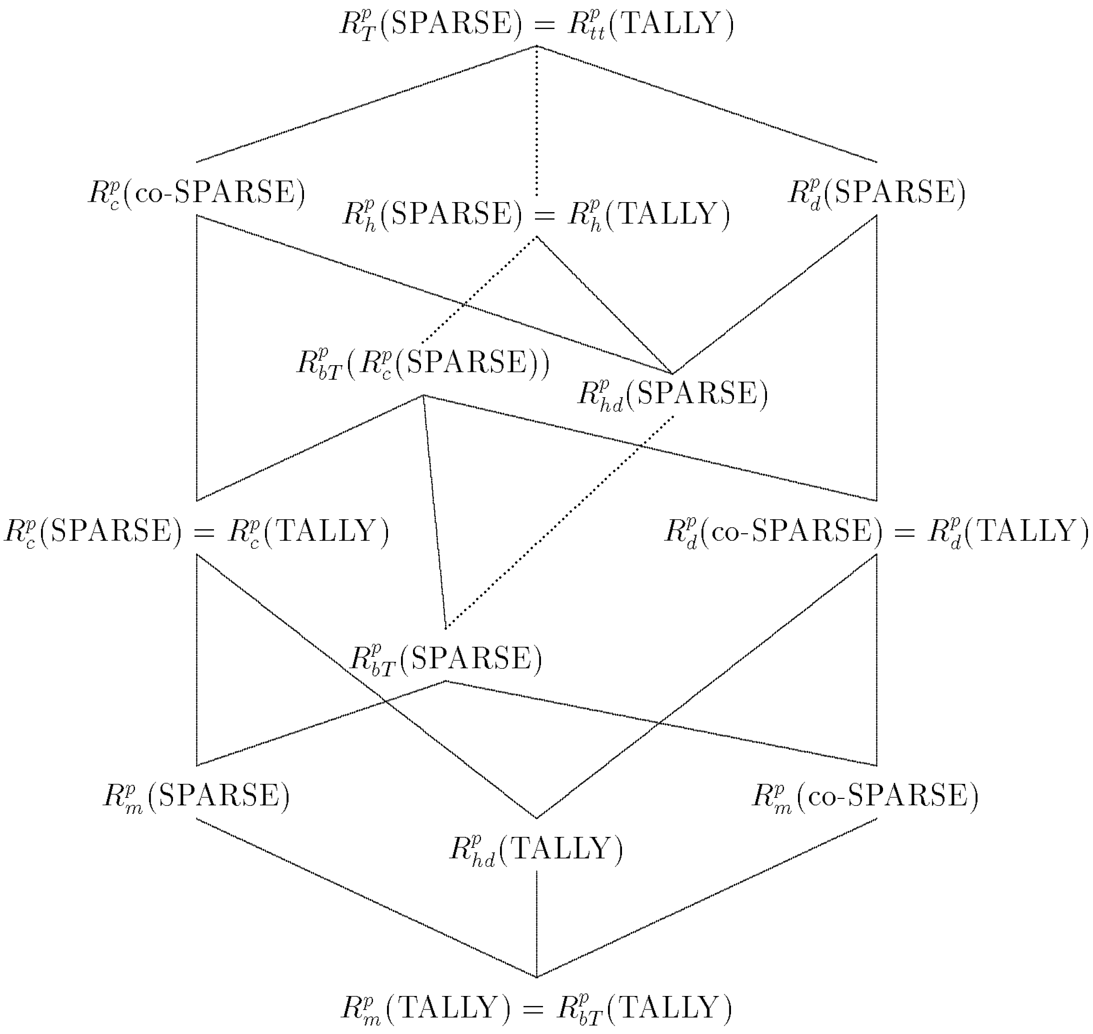

Figure 1: Structure of inclusions. (Dotted lines indicate that it is not known whether the inclusions are proper.)