Universität Ulm
Fakultät für Informatik

Bounded truth-table and conjunctive reductions to sparse and tally sets

Vikraman Arvind
Indian Institute of Technology, Delhi
Johannes Köbler
Universität Ulm
Martin Mundhenk
Universität Ulm

Nr. 92-01
Ulmer Informatik-Berichte
April 1992

Bounded truth-table and conjunctive reductions to sparse and tally sets

Vikraman Arvind*
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi
New Delhi 110016, India

Johannes Köbler and Martin Mundhenk
Abteilung für Theoretische Informatik
Universität Ulm
Oberer Eselsberg
D-W-7900 Ulm, Germany

December 19, 1994

Abstract

In this paper we study the consequences of the existence of sparse hard sets for different complexity classes under certain types of deterministic, randomized and nondeterministic reductions. We show that if an NP-complete set is bounded-truth-table reducible to a set that conjunctively reduces to a sparse set then P = NP. Relatedly, we show that if an NP-complete set is bounded-truth-table reducible to a set that co-rp reduces to some set that conjunctively reduces to a sparse set then RP = NP. We also prove similar results under the (apparently) weaker assumption that some solution of the promise problem (1SAT, SAT) reduces via the mentioned reductions to a sparse set. Finally we consider nondeterministic polynomial time many-one reductions to sparse and co-sparse sets. We prove that if a coNP-complete set reduces via a nondeterministic polynomial time many-one reduction to a co-sparse set then PH = Θ_2^p. On the other hand, we show that nondeterministic polynomial time many-one reductions to sparse sets are as powerful as nondeterministic Turing reductions to sparse sets.

1 Introduction

Sparse sets play a central role in structural complexity theory. An important aspect that has been very fruitful is the study of sparse hard sets under different kinds of reductions. This line of research opened with the question whether there can possibly exist sparse

*Work done while visiting Universität Ulm. Supported in part by an Alexander von Humboldt post-doctoral research fellowship.

complete sets for NP under polynomial time many-one reductions (it was conjectured by L. Berman and J. Hartmanis [BH77] that there are no sparse NP-complete sets).

The first results were P. Berman's proof that P = NP if some tally set is NP-complete [Ber78] and Fortune's proof that if there is a sparse coNP-complete set, then P = NP [For79]. Mahaney settled the 'sparseness' conjecture by proving that if any NP-complete set many-one reduces to a sparse set then P = NP [Mah82]. From an entirely different angle of research, 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 hierarchy collapses to Σ_2^p [KL80].

Discovering consequences of the existence of sparse complete sets for different kinds of truth-table reducibilities has remained an active research area. The next important advance was made recently by Ogiwara and Watanabe [OW91], when they proved, using a new 'left-set' technique, that if NP has sparse hard sets under polynomial time bounded truth-table reductions then P = NP. This has been, more recently, followed up by similar results for conjunctive truth-table polynomial time reductions [AHH$^+$92, RR92], and in [AHH$^+$92] even for more flexible truth-table reductions (e.g. bounded conjunctions on the 1-truth-table closure of the conjunctive closure of sparse sets). These results demonstrate the efficacy of the left-set technique introduced in [OW91] (in fact, the older result of Mahaney has now a considerably easier proof). It has also opened the possibility of a new unified explanation for the Karp/Lipton/Sipser result yielding a collapse of the polynomial hierarchy to Σ_2^p, and the stronger collapse to P under special truth-table reductions.

The main results of this paper (reported in Section 3 and in Section 4) concern the existence of sparse hard sets for NP and other complexity classes under the bounded truth-table closure of different kinds of reductions. In Section 3 we prove that if NP $\subseteq R_b^p(R_c^p(\mathrm{SPARSE}))$ then P = NP. This result subsumes and extends all previously known results on reductions of NP sets to sparse sets via various types of polynomial time truth-table reductions that yield a collapse of the polynomial hierarchy to P.

In Section 4 we consider randomized reductions to sparse sets, and show that if NP $\subseteq R_b^p(R_m^{co\text{-}rp}(R_c^p(\mathrm{SPARSE})))$ then RP = NP. (D. Ranjan and P. Rohatgi [RR92] have independently shown that if NP $\subseteq R_m^{co\text{-}rp}(\mathrm{SPARSE})$ then RP = NP.)

Relatedly, we show in Section 5 that if some solution of the promise problem $(\mathrm{1SAT}, \mathrm{SAT})$ is in $R_b^p(R_c^p(\mathrm{SPARSE}))$ then there is a solution of $(\mathrm{1SAT}, \mathrm{SAT})$ in P. We also show that the conclusion RP = NP can be derived from the assumption that some solution of the promise problem $(\mathrm{1SAT}, \mathrm{SAT})$ is in $R_b^p(R_m^{co\text{-}rp}(R_c^p(\mathrm{SPARSE})))$.

The technique used for our proofs is novel. It combines the method of left sets with a classical representation theorem due to Hausdorff on the boolean closure of set rings, i.e. classes of sets closed under union and intersection. Several theorems in this paper concern the bounded truth-table closure of reduction classes to sparse sets. Their proofs demonstrate the inevitable role played by the Hausdorff representation for these classes. The Hausdorff representation was first used in a complexity theoretic context by Wechsung and Wagner [WW85] where they relate the bounded truth-table closure of NP to the boolean hierarchy. It remains an open question whether similar results can be proved for disjunctive polynomial time reductions to sparse sets.

Pursuing the question of existence of sparse hard sets further (in Section 7), we consider nondeterministic reductions (as defined by Ladner, Lynch, and Selman [LLS75]) and show that if a coNP-complete set can be reduced via a nondeterministic reduction to a co-sparse set then the polynomial hierarchy collapses to Θ_2^p. Such a result seems unlikely for sparse sets since (as we show) nondeterministic many-one reductions to sparse sets surprisingly turn out to be as powerful as nondeterministic Turing reductions to sparse sets. We also prove that if Σ_2^p is bounded truth-table reducible to a set that can be reduced via a nondeterministic reduction to a co-sparse set then the polynomial hierarchy collapses to Δ_2^p.

2 Preliminaries and Notation

A set T is called a tally set if $T \subseteq 0^*$. The census function of a set A is $census_A(n) = |A^{\leq n}|$. A set S is called sparse if its census function is bounded above by a polynomial. We use TALLY and SPARSE to represent, respectively, the classes of tally and sparse sets. For a class $\mathcal{K}$ of sets we denote the union of all sets in $\mathcal{K}$ by $\bigcup\mathcal{K}$. Let $\langle\cdot,\cdot\rangle$ denote a standard pairing function.

The reductions discussed in this paper are the polynomial-bounded reductions defined by Ladner, Lynch, and Selman [LLS75] and by Adleman and Manders [AM77].

Notation [AHOW] For any reducibility $\leq_r^\alpha$ and any class of sets $\mathcal{C}$, let $R_r^\alpha(\mathcal{C}) = \{A \mid A \leq_r^\alpha B \text{ for } B \in \mathcal{C}\}$, where $\alpha \in \{p, np, co\text{-}np, rp, co\text{-}rp\}$ and $r \in \{m, c, d, b, T\}$.

Definition 2.1 *The join of two sets A and B, denoted $A \oplus B$ is defined as*

$$A \oplus B = \{0x \mid x \in A\} \cup \{1x \mid x \in B\}$$

Definition 2.2 *A class $\mathcal{K}$ of sets that includes $\emptyset$ and Σ^* and is closed under finite unions and finite intersections is said to be a set ring.*

Definition 2.3 [OW91] *For a set A in NP let $P_A \in$ P and q be a polynomial such that $A = \{x \mid (\exists w \in \Sigma^{q(|x|)})[\langle x,w\rangle \in P_A]\}$. For $x \in A$ let $w_{max}(x) = \max\{w \in \Sigma^{q(|x|)} \mid \langle x,w\rangle \in P_A\}$. Then Left$(A) = \{\langle x,w\rangle \mid x \in A, w \in \Sigma^{q(|x|)}$ and $w \leq w_{max}(x)\}$ is the left set of A.*

Note that the left set depends on the particular witness relation P_A.

3 Bounded truth-tables on conjunctive reductions to sparse sets

The main result of this section is that if NP $\subseteq R_b^p(R_c^p(\text{SPARSE}))$ then P = NP. This result subsumes and extends all previously known results on reductions of NP sets to sparse sets via various types of polynomial time truth-table reductions that yield a collapse to P. We also discuss similar consequences for the classes PP, $\text{C}_=\text{P}$, FewP, Few, and UP. Finally we define Hausdorff reductions (which underly the proof of the above results) and prove a trade-off result between the density of the set reduced to and the size of the reduction.

The following characterization of the boolean closure of set rings due to Hausdorff plays a key role in many results of this paper.

Theorem 3.1 [Hau14, WW85] *Let $\mathcal{K}$ be a set ring and let $BC(\mathcal{K})$ be the closure of $\mathcal{K}$ under finite union, finite intersection, and complement. Then every $A \in BC(\mathcal{K})$ can be represented as $A = \bigcup_{i=1}^{k}(A_{2i-1} - A_{2i})$, where $A_j \in K$, $1 \le j \le 2k$, and $A_1 \supseteq A_2 \supseteq \cdots \supseteq A_{2k}$.*

In order to obtain a Hausdorff representation for sets in $R_b^p(R_c^p(\mathrm{SPARSE}))$ we need to show that $R_c^p(\mathrm{SPARSE})$ is a set ring.

Lemma 3.2 *$R_c^p(\mathrm{SPARSE})$ is a set ring.*

Proof $R_c^p(\mathrm{TALLY})$ is easily seen to be closed under finite unions and intersections. A recent result of Buhrman, Longpré, and Spaan [BLS92] showing that SPARSE $\subset R_c^p(\mathrm{TALLY})$ implies $R_c^p(\mathrm{SPARSE}) = R_c^p(\mathrm{TALLY})$. Hence, $R_c^p(\mathrm{SPARSE})$ is a set ring. ■

Fact 3.3 *If a class $\mathcal{K}$ of sets contains a set different from $\emptyset$ and Σ^* and is closed under join and polynomial time many-one reductions then $R_b^p(\mathcal{K}) = BC(\mathcal{K})$.*

Remark All the reduction classes to sparse sets considered in this paper fulfil the conditions to apply Fact 3.3.

Theorem 3.4 *If $A \in \mathrm{NP}$ such that $Left(A) \in R_b^p(R_c^p(\mathrm{SPARSE}))$ then $A \in \mathrm{P}$.*

Proof Let q be a polynomial and let P_A be a polynomial-time computable set such that $A = \{x \mid (\exists w \in \Sigma^{q(|x|)})[\langle x, w\rangle \in P_A]\}$. Recall that $Left(A) = \{\langle x, w\rangle \mid x \in A \ \wedge \ w \in \Sigma^{q(|x|)} \ \wedge \ w \le w_{max}\}$, where $w_{max} = \max\{w \in \Sigma^{q(|x|)} \mid \langle x, w\rangle \in P_A\}$. In the following we describe an algorithm that on input $x \in A$ computes w_{max} (the lexicographically largest witness) by a breadth-first search on the tree of prefixes of all potential witnesses. In order to do this we use the set $prefix(Left(A)) = \{\langle x, y\rangle \mid (\exists z)[\langle x, yz\rangle \in Left(A)]\}$. Each prefix y actually represents the interval of all possible extensions of y to length $q(|x|)$. It is not hard to see that $prefix(Left(A))$ is many-one equivalent to $Left(A)$ and therefore $prefix(Left(A)) \in R_b^p(R_c^p(\mathrm{SPARSE}))$.

Using Lemma 3.2 and the representation theorem of Hausdorff stated as Theorem 3.1, it is easy to see that there exist a sparse set S and sets $C_i \in R_c^p(S)$, $1 \le i \le 2k$, such that $prefix(Left(A)) = \bigcup_{i=1}^{k}(C_{2i-1} - C_{2i})$ and $C_1 \supseteq C_2 \supseteq \ldots \supseteq C_{2k}$.

Let f_i, $1 \le i \le 2k$, be the conjunctive reduction functions witnessing $C_i \in R_c^p(S)$, i.e. $\langle x, y\rangle \in C_i \Leftrightarrow f_i(\langle x, y\rangle) \subseteq S$.

We first outline an intuitive description of the polynomial-time[1] decision procedure for A. As stated above, it performs a breadth-first search through the tree of witness prefixes for an input x. Let x be an element of A, and let $N = \{y_1, \ldots, y_t\}$ be a lexicographically ordered set of prefixes (all of the same length) that includes the prefix of w_{max} of that length. We exploit some crucial properties of the Hausdorff representation of $prefix(Left(A))$ for the design of a procedure pruning N to a polynomially size-bounded set that still includes the prefix of w_{max}.

Let y_h be the prefix of w_{max} in $\{y_1, \ldots, y_t\}$. Then, letting $d = 1$ and $l(0) = 1$, it holds that

$$\{\langle x, y_{l(d-1)}\rangle, \ldots, \langle x, y_h\rangle\} \subseteq C_{2d-1}$$

[1] It is implicit in this section that polynomial time and polynomial size always mean polynomial in $|x|$.

Let $r(d)$ be the largest index r such that $\{\langle x, y_{l(d-1)}\rangle, \ldots, \langle x, y_r\rangle\} \subseteq C_{2d-1}$ and let $l(d)$ be the least index l such that $1 \le l \le r(d)+1$ and $\{\langle x, y_l\rangle, \ldots, \langle x, y_{r(d)}\rangle\} \subseteq C_{2d}$. Observe that since $\{\langle x, y_{l(d-1)}\rangle, \ldots, \langle x, y_h\rangle\} \subseteq C_{2d-1}$ it follows that $r(d) \ge h$. Similarly, since $\{\langle x, y_{h+1}\rangle, \ldots, \langle x, y_{r(d)}\rangle\} \subseteq C_{2d}$, it holds that $l(d) \le h+1$. We consider the following two cases separately.

1. $\langle x, y_h\rangle \notin C_{2d}$.

   Then $l(d) = h+1$ since $y_h \notin \{y_{l(d)}, \ldots, y_{r(d)}\}$, i.e. $l(d) > h$.

2. $\langle x, y_h\rangle \in \mathcal{C}_{2d}$. (This case is only possible if $d < k$.)

   In this case, $y_h \in \{y_{l(d)}, \ldots, y_{r(d)}\}$. Since $\{\langle x, y_{l(d)}\rangle, \ldots, \langle x, y_h\rangle\} \subseteq \mathit{prefix}(\mathit{Left}(A))$ but $\{\langle x, y_{l(d)}\rangle, \ldots, \langle x, y_h\rangle\} \subseteq C_{2d}$, it follows that $\{\langle x, y_{l(d)}\rangle, \ldots, \langle x, y_h\rangle\} \subseteq C_{2d+1}$, and the above analysis can be repeated.

If we could compute the prefixes $y_{l(d)}$ and $y_{r(d)}$ defined above in polynomial time, we could use the above properties in order to design a recursive procedure that collects all the prefixes $y_{l(d)-1}$ found in the recursive calls. This procedure would return a small subset of N containing y_h. Starting with $N = \{\epsilon\}$, the overall algorithm can use repeatedly such a pruning step at each level of the tree of possible witness prefixes by first expanding all the prefixes y in N to $y0$ and $y1$ (thus doubling N) and then pruning N back to a small subset. In that way, the algorithm finally computes a small subset of $\Sigma^{q(|x|)}$ that contains w_{max} in case $x \in A$.

Although we cannot explicitly compute the required prefixes $y_{l(d)}$ and $y_{r(d)}$, instead we can compute, given $y_{l(d-1)}$, in polynomial time (polynomially size-bounded) sets $J_{right}(d)$ and $J_{left}(d)$ of prefixes such that $y_{r(d)} \in J_{right}(d)$ and $y_{l(d)} \in J_{left}(d)$. This suffices since for each prefix candidate $y \in J_{left}(d)$, the search for $y_{l(d+1)}$ can be done recursively. Since the depth of the recursion is the constant k, the resulting sets $J_{left}(d)$ of candidates for $y_{l(d)}$ still have polynomially bounded cardinality. We now describe the algorithm in detail. It calls a recursive pruning procedure PRUNE which in turn calls two functions SEARCH-RIGHT and SEARCH-LEFT. SEARCH-RIGHT is used to search for candidates for $y_{r(d)}$ to the right of previously found candidates for $y_{l(d-1)}$ resulting in a polynomial size-bounded set $J_{right}(d)$ containing $y_{r(d)}$. SEARCH-LEFT is used to search to the left of the prefixes in $J_{right}(d)$ to form a polynomial size-bounded set $J_{left}(d)$ containing $y_{l(d)}$. Let m be a polynomial bounding the size of the queries to the sparse set, i.e. $|z| \le m(n)$ for all $z \in \bigcup\{f_i(\langle x, y\rangle) \mid 1 \le i \le 2k, |x| = n, |y| \le r(n)\}$, and let s be a polynomial bounding the census of the sparse set S.

SEARCH-RIGHT(d, N, y_l, x)
(* if $\langle x, y_l\rangle \in C_{2d-1}$ it returns a set $J \subseteq N = \{y_1, \ldots, y_t\}$ that includes the
largest prefix $y_r \in N$ such that $\{\langle x, y_l\rangle, \ldots, \langle x, y_r\rangle\} \subseteq C_{2d-1}$ *)
begin
  $J := \{y_t\}$
  $Q := \emptyset$
  $i := l$
  repeat
    $i := i+1$
    if $f_{2d-1}(\langle x, y_i\rangle) \nsubseteq \bigcup_{j=l}^{i-1} f_{2d-1}(\langle x, y_j\rangle)$ **then**
      $J := J \cup \{y_{i-1}\}$

```
      Q := Q ∪ f_{2d−1}(⟨x, y_i⟩)
    end
  until (|Q| > s(m(|x|))) or (i = t)
  return J
end
```

Claim 1 *Function* SEARCH-RIGHT(d, N, y_l, x)*, when called with parameter* $y_l = y_{l(d-1)}$*, returns a set* J *containing* $y_{r(d)}$*.*

Proof of Claim 1 There are two cases. If $r(d) = t$ then $y_{r(d)}$ is clearly in the returned set J. Otherwise, since $\{\langle x, y_{l(d-1)}\rangle, \ldots, \langle x, y_{r(d)}\rangle\} \subseteq C_{2d-1}$ and $\langle x, y_{r(d)+1}\rangle \notin C_{2d-1}$, all the queries in the sets $f_{2d-1}(\langle x, y_{l(d-1)}\rangle), \ldots, f_{2d-1}(\langle x, y_{r(d)}\rangle)$ are in S but at least one query q in $f_{2d-1}(\langle x, y_{r(d)+1}\rangle)$ is not in S. Therefore $y_{r(d)+1}$ is the smallest prefix y in N such that $y \geq y_{l(d-1)}$ and $q \in f_d(\langle x, y\rangle)$, i.e. $y_{r(d)}$ is included in J in some step of the repeat loop since $|\bigcup_{j=l(d)}^{r(d)} f_{2d-1}(\langle x, y_j\rangle)| \leq s(m(|x|))$. □

```
SEARCH-LEFT(d, N, y_r, x)
(* returns a set J ⊆ N = {y_1, ..., y_t} that includes the smallest prefix
   y_l ∈ N such that l ≤ r + 1 and {⟨x, y_l⟩, ..., ⟨x, y_r⟩} ⊆ C_{2d} *)
begin
  J := {y_1}
  i := r
  Q := ∅
  repeat
    if f_{2d}(⟨x, y_i⟩) ⊈ ⋃_{j=i+1}^{r} f_{2d}(⟨x, y_j⟩) then
      J := J ∪ {y_{i+1}}
      Q := Q ∪ f_{2d}(⟨x, y_i⟩)
    end
    i := i − 1
  until (|Q| > s(m(|x|))) or (i = 0)
  return J
end
```

Claim 2 *Function* SEARCH-LEFT(d, N, y_r, x)*, when called with parameter* $y_r = y_{r(d)}$*, returns a set* J *containing* $y_{l(d)}$*.*

Proof of Claim 2 Again, there are two cases. If $l(d) = 1$ then $y_{l(d)}$ is clearly in the returned set J. Otherwise, since $\{\langle x, y_{l(d)}\rangle, \ldots, \langle x, y_{r(d)}\rangle\} \subseteq C_{2d}$ and $\langle x, y_{l(d)-1}\rangle \notin C_{2d}$, all the queries in the sets $f_{2d}(\langle x, y_{l(d)}\rangle), \ldots, f_{2d}(\langle x, y_{r(d)}\rangle)$ are in S but at least one query q in $f_{2d}(\langle x, y_{l(d)-1}\rangle)$ is not in S. Therefore $y_{l(d)-1}$ is the largest prefix y in N such that $y \leq y_{r(d)}$ and $q \in f_{2d}(\langle x, y\rangle)$, i.e. $y_{l(d)}$ is included in J in some step of the repeat loop since $|\bigcup_{j=l(d)}^{r(d)} f_{2d}(\langle x, y_j\rangle)| \leq s(m(|x|))$. □

```
PRUNE(N, J'_{left}, d, x)
(* returns a subset of N = {y_1, ..., y_t} that contains the prefix y_h of w_{max} if
   y_h ∈ N, ⟨x, y_h⟩ ∈ C_{2d−1} and {⟨x, y_l⟩, ..., ⟨x, y_h⟩} ⊆ C_{2d−1} for a y_l ∈ J'_{left}
   with l ≤ h *)
```

```
begin
  if d = k + 1 then return ∅ end
  J_right := ∅
  for each z ∈ J'_left do
    J_right := J_right ∪ SEARCH-RIGHT(d, N, z, x)
  end
  J_left := ∅
  for each z ∈ J_right do
    J_left := J_left ∪ SEARCH-LEFT(d, N, z, x)
  end
  return {y_{l-1} | y_l ∈ J_left} ∪ PRUNE(N, J_left, d + 1, x)
end
```

Claim 3 *If $y_h \in N$, $\langle x, y_h\rangle \in C_{2d-1}$, and $y_{l(d-1)} \in J'_{left}$ then* PRUNE(N, J'_{left}, d, x) *returns a set containing y_h.*

Proof of Claim 3 If $y_h \in N$ and $\langle x, y_h\rangle \in C_{2d-1}$ then $\langle x, y_h\rangle$ is also in the sets $C_{2d-2}, \ldots, C_1$. By the above analysis (since always case 2 happens up to $d-1$) it follows that $\{\langle x, y_{l(d-1)}\rangle, \ldots, \langle x, y_h\rangle\} \subseteq C_{2d-1}$. Since $y_{l(d-1)} \in J'_{left}$, using Claim 1, $y_{r(d)}$ is included in J_{right} by the call of SEARCH-RIGHT$(d, N, y_{l(d-1)}, x)$. Then, using Claim 2, $y_{l(d)}$ is included in J_{left} by the call of SEARCH-LEFT$(d, N, y_{r(d)}, x)$. Now we can prove by induction that y_h is included in the set returned by PRUNE. If $\langle x, y_h\rangle \notin C_{2d}$ (which must be true in the base case $d = k$) then $y_h = y_{l(d)-1}$ and y_h is included in the set returned by PRUNE. If $\langle x, y_h\rangle \in C_{2d}$ then $\langle x, y_h\rangle$ is in C_{2d+1} and we can use the induction hypothesis. □

We complete the algorithm with a description of the main program.

```
input x
N := {ε}
for i := 1 to q(|x|) do
  N := {y0 | y ∈ N} ∪ {y1 | y ∈ N}   (* expand the prefixes to length i *)
  N := PRUNE(N, {y_1}, 1, x)
end
(* N now includes w_max if x ∈ A *)
if there is a witness for x in N then accept else reject
```

In order to prove the correctness of the algorithm it suffices to observe that it follows from Claim 3 that the prefix y_h of w_{max} is included in the pruned set returned by PRUNE$(N, \{y_1\}, 1, x)$ provided that y_h is in N. Also, since the sets returned by SEARCH-RIGHT and SEARCH-LEFT are bounded in size by $s(m(|x|)) + 2$, it follows inductively that the set J_{left} computed by PRUNE at level d is bounded in size by $(s(m(|x|)) + 2)^{2d}$. Therefore, since the depth of recursion of function PRUNE is bounded by a constant, the finally returned set being the union of all the J_{left}'s is polynomially bounded in size, and it is easy to see that the algorithm runs in polynomial time. ■

We now discuss the application of the above results to the classes UP, FewP, Few, PP, and $C_=P$.

Theorem 3.5 *If* UP *is contained in* $R_b^p(R_c^p(\mathrm{SPARSE}))$ *then* UP = P.

Proof We first note that for every set $A \in$ UP it holds that *Left*(A) is in UP. Hence, if UP $\subseteq R_b^p(R_c^p(\mathrm{SPARSE}))$ then *Left*(A) is in $R_b^p(R_c^p(\mathrm{SPARSE}))$, and by Theorem 3.4 it follows that A is in P. ■

Theorem 3.6 *If* FewP *is contained in* $R_b^p(R_c^p(\mathrm{SPARSE}))$ *then* P = Few.

Proof By a similar proof as above it can be inferred that P = FewP. Since Few $\subseteq \mathrm{P}^{\mathrm{FewP}}$ [KSTT] it follows that P = Few. ■

Theorem 3.7 *If* PP *is contained in* $R_b^p(R_c^p(\mathrm{SPARSE}))$ *then* P = PP.

Proof Consider the PP-complete set $\{\langle x, m\rangle \mid$ there are at least m satisfying assignments for $x\}$ which has exactly the required properties of left sets. Under the assumption that this set is in $R_b^p(R_c^p(\mathrm{SPARSE}))$ we can use the algorithm described in the proof of Theorem 3.4 to compute in polynomial time a set of numbers that includes the number $\#\mathrm{SAT}(x)$ of satisfying assignments of the formula x. Now we can use the result of Cai and Hemachandra [CH91] and Toda (see [ABG90]) that P = PP if there is an FP function that computes on input x a set of numbers that includes $\#\mathrm{SAT}(x)$. ■

Theorem 3.8 *If* $\mathrm{C_{=}P}$ *is contained in* $R_b^p(R_c^p(\mathrm{SPARSE}))$ *then* P = $\mathrm{C_{=}P}$.

Proof There exist complete sets in $\mathrm{C_{=}P}$ that are one word decreasing self-reducible [OL91]. Balcázar has shown that every one word decreasing self-reducible set in $R_T^p(\mathrm{SPARSE})$ is in Σ_2^p [Bal90]. Therefore it follows from the assumption of the theorem that $\mathrm{C_{=}P} \subseteq \Sigma_2^p$. Furthermore, since coNP $\subseteq \mathrm{C_{=}P}$, if $\mathrm{C_{=}P} \subseteq R_b^p(R_c^p(\mathrm{SPARSE}))$ then also NP $\subseteq R_b^p(R_c^p(\mathrm{SPARSE}))$, and it follows from Theorem 3.4 that P = Σ_2^p. ■

Theorem 3.7 could also be proved in the same way as Theorem 3.8.

4 Bounded truth-tables on randomized reductions to sparse sets

In this section we consider randomized reductions to sparse sets. Randomized reductions were introduced by Adleman and Manders [AM77] and have played an important role in complexity theory. We show that NP cannot have sparse hard sets under certain randomized reductions unless NP = RP. Namely, we show that if NP reduces via a bounded truth-table reduction to a set that reduces via a co-rp reduction to a sparse set then NP = RP. We also prove that if any solution to the promise problem (1SAT, SAT) is reducible to a sparse set via the same type of reduction we have NP = RP.

Definition 4.1 [AM77] $A \leq_m^{rp} B$ *if there exist a polynomial time function f and a polynomial q such that*

$$x \in A \Rightarrow \mathrm{Prob}_{w \in \Sigma^{q(|x|)}}[f(\langle x, w\rangle) \in B] \geq 3/4$$

$$x \notin A \Rightarrow \mathrm{Prob}_{w \in \Sigma^{q(|x|)}}[f(\langle x, w\rangle) \notin B] = 1$$

Similarly, $A \leq_m^{co-rp} B$ *if there exist a polynomial time function* f *and a polynomial* q *such that*

$$x \in A \Rightarrow \mathrm{Prob}_{w \in \Sigma^{q(|x|)}}[f(\langle x, w\rangle) \in B] = 1$$

$$x \notin A \Rightarrow \mathrm{Prob}_{w \in \Sigma^{q(|x|)}}[f(\langle x, w\rangle) \notin B] \geq 3/4$$

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

We first show that if NP $\subseteq R_m^{co-rp}(R_c^p(\mathrm{SPARSE}))$ then NP = RP (this result is independently due to D. Ranjan and P. Rohatgi [RR92]). Then we extend this to the result that NP $\subseteq R_b^p(R_m^{co-rp}(R_c^p(\mathrm{SPARSE}))$ implies NP = RP. We need the following folklore result on amplification for randomized reductions.

Lemma 4.2 *If* $A \leq_m^{co-rp} B$ *then for every polynomial* p *there exist a co-rp reduction function* f *from* A *to* $AND_\omega(B) = \{\langle x_1, ..., x_i\rangle \mid x_j \in B$ *for each* j, $1 \leq j \leq i\}$ *and a polynomial* q *such that*

$$x \in A \Rightarrow \mathrm{Prob}_{w \in \Sigma^{q(|x|)}}[f(\langle x, w\rangle) \in AND_\omega(B)] = 1$$

$$x \notin A \Rightarrow \mathrm{Prob}_{w \in \Sigma^{q(|x|)}}[f(\langle x, w\rangle) \notin AND_\omega(B)] \geq 1 - 2^{-p(|x|)}.$$

Fact 4.3 *For every set* B, $AND_\omega(B) \in R_c^p(B)$*.*

Fact 4.3 shows that if a set B conjunctively reduces in polynomial time to a sparse set S then $\mathrm{AND}_\omega(B)$ also conjunctively reduces to S. The following lemma is an easy consequence of Lemma 4.2 and Fact 4.3.

Lemma 4.4 *If* $A \in R_m^{co-rp}(R_c^p(\mathrm{SPARSE}))$ *then for every polynomial* p *there exist a sparse set* S*, an* FP *function* f *and a polynomial* q *such that*

$$x \in A \Rightarrow \mathrm{Prob}_{w \in \Sigma^{q(|x|)}}[f(\langle x, w\rangle) \subseteq S] = 1$$

$$x \notin A \Rightarrow \mathrm{Prob}_{w \in \Sigma^{q(|x|)}}[f(\langle x, w\rangle) \not\subseteq S] \geq (1 - 2^{-p(|x|)})$$

The following result has been independently obtained by D. Ranjan and P. Rohatgi [RR92]).

Theorem 4.5 *If* NP $\subseteq R_m^{co-rp}(R_c^p(\mathrm{SPARSE}))$ *then* NP = RP*.*

Proof Let A be an NP-complete set such that $A \in R_m^{co-rp}(R_c^p(\mathrm{SPARSE}))$. As in the proof of Theorem 3.4 let r be a polynomial and let P_A be a polynomial-time set such that $A = \{x \mid \exists w \in \Sigma^{r(|x|)} : \langle x, w\rangle \in P_A\}$ and $Left(A) = \{\langle x, w\rangle \mid x \in A \ \wedge \ w \in \Sigma^{r(|x|)} \ \wedge \ w \leq w_{max}\}$, where $w_{max} = \max\{w \in \Sigma^{r(|x|)} \mid \langle x, w\rangle \in P_A\}$.

We describe a randomized polynomial time algorithm that computes on input x in A with high probability the lexicographically largest witness w_{max} by a breadth-first search on the tree of possible witness prefixes. In order to do this we again use the set $prefix(Left(A)) = \{\langle x, y\rangle \mid \exists z : \langle x, yz\rangle \in Left(A)\}$ which is in $R_m^{co-rp}(R_c^p(\mathrm{SPARSE}))$ since it is many-one equivalent to $Left(A)$.

Let p be a polynomial such that for all n, $(1 - 2^{-p(n)})^{r(n)} \geq 3/4$. By Lemma 4.4 there exist a sparse set S, an FP function f and a polynomial q such that

$$\langle x, y\rangle \in \textit{prefix}(\textit{Left}(A)) \Rightarrow \mathrm{Prob}_{w\in\Sigma^{q(|x|)}}[f(\langle x,y,w\rangle) \subseteq S] = 1$$

$$\langle x, y\rangle \notin \textit{prefix}(\textit{Left}(A)) \Rightarrow \mathrm{Prob}_{w\in\Sigma^{q(|x|)}}[f(\langle x,y,w\rangle) \not\subseteq S] \geq 1 - 2^{-p(|x|)}$$

Let m be a polynomial bounding the size of the queries to the sparse set, i.e. $|z| \leq m(n)$ for all $z \in \bigcup\{f(\langle x,y,w\rangle) \mid |x| = n, |y| \leq r(n), |w| = q(n)\}$, and let s be a polynomial bounding the census of the sparse set S.

We first describe the randomized algorithm for testing membership in A and then prove its correctness.

input x
$N := \{\epsilon\}$
for $l := 1$ **to** $r(|x|)$ **do**
  $N := \{y0 \mid y \in N\} \cup \{y1 \mid y \in N\}$; (* expand prefixes to length l *)
  (* let $y_1, \ldots, y_t$ be the prefixes in N in lexicographical order *)
  $M := \emptyset$
  $i := 1$
  repeat
    $i := i + 1$
    compute a set $Q(y_i) = f(\langle x, y_i, w\rangle)$ of conjunctive queries
    where w is chosen uniformly at random from $\Sigma^{q(|x|)}$
    if $Q(y_i) \not\subseteq \bigcup_{1\leq j<i} Q(y_j)$ **then** $M := M \cup \{y_i\}$
  until $(|\bigcup_{1\leq j\leq i} Q(y_j)| > s(m(|x|)))$ or $(i = t)$
  $N := \{y_{i-1} \mid y_i \in M\} \cup \{y_t\}$
end
if there is a witness in N **then** *accept* **else** *reject*

It is clear that the above algorithm runs in polynomial time since the set N contains at most $2(s(m(|x|)) + 2)$ prefixes at any stage of the loop. In order to prove that it is an RP algorithm for A we need to show that if $x \in A$ then the algorithm accepts x with high probability, and if $x \notin A$ then the algorithm always rejects. The latter is obvious from the fact that the algorithm accepts only if it finds a witness. It remains to show that if $x \in A$ then the algorithm finds w_{max} with probability at least 3/4.

We show that if $N = \{y_1, \ldots, y_t\}$ contains a prefix of w_{max} (call it y_h; we assume that $h < t$ since y_t is always included in the pruned set) then with probability at least $1 - 2^{-p(|x|)}$ the prefix y_h is included in N after the repeat loop. In order to see this we observe the following.

1. For every $w \in \Sigma^{q(|x|)}$ and i, $1 \leq i \leq h$, it holds that $f(\langle x, y_i, w\rangle) \subseteq S$. This follows from the fact that $\langle x, y_i\rangle \in \textit{prefix}(\textit{Left}(A))$ for $1 \leq i \leq h$.

2. Since $\langle x, y_{h+1}\rangle \notin \textit{prefix}(\textit{Left}(A))$, it holds that $\mathrm{Prob}_{w\in\Sigma^{q(|x|)}}[f(\langle x, y_{h+1}, w\rangle) \not\subseteq S] \geq (1 - 2^{-p(|x|)})$.

It follows that $|\bigcup_{1\leq j\leq h} Q(y_j)| \leq s(m(|x|))$ and with probability at least $1 - 2^{-p(|x|)}$ it holds that $Q(y_{h+1}) \not\subseteq \bigcup_{1\leq j\leq h} Q(y_j)$. Hence N includes y_h with probability at least $1 - 2^{-p(|x|)}$ at the end of the repeat loop.

Since the outer for-loop has $r(|x|)$ iterations, and since at the beginning $N = \{\epsilon\}$ contains a prefix of w_{max}, it follows that with probability at least $(1 - 2^{-p(|x|)})^{r(|x|)}$ the algorithm finds w_{max}. By choice of p this probability is more than 3/4. ∎

We state the next theorem without proof as it can be proved exactly as Theorem 4.5.

Theorem 4.6 *If* $\mathrm{NP} \subseteq R_m^{rp}(R_d^p(\text{co-SPARSE}))$ *then* $\mathrm{NP} = \mathrm{RP}$.

We now extend the above results to prove that if $\mathrm{NP} \subseteq R_b^p(R_m^{co-rp}(R_c^p(\mathrm{SPARSE})))$ then $\mathrm{NP} = \mathrm{RP}$. We first show that the class $R_m^{co-rp}(R_c^p(\mathrm{SPARSE}))$ is a set ring so that we can assume the existence of a Hausdorff representation for any set in $R_b^p(R_m^{co-rp}(R_c^p(\mathrm{SPARSE})))$.

Lemma 4.7 *1.* $R_m^{co-rp}(R_c^p(\mathrm{SPARSE})) = R_m^{co-rp}(R_c^p(\mathit{TALLY}))$

2. $R_m^{rp}(R_d^p(\text{co-SPARSE})) = R_m^{rp}(R_d^p(\mathit{TALLY}))$

Proof Follows from the facts that $R_d(\text{co-SPARSE}) = R_d(\mathrm{TALLY})$ and $R_c^p(\mathrm{SPARSE}) = R_c^p(\mathrm{TALLY})$. ■

Lemma 4.8 $R_m^{co-rp}(R_c^p(\mathrm{SPARSE}))$ *is a set ring.*

Proof Since $R_m^{co-rp}(R_c^p(\mathrm{SPARSE})) = R_m^{co-rp}(R_c^p(\mathrm{TALLY}))$, it is enough to show that $R_m^{co-rp}(R_c^p(\mathrm{TALLY}))$ is a set ring. Assume that $A \leq_m^{co-rp} A_1$ and $B \leq_m^{co-rp} B_1$, where A_1 and B_1 are in $R_c^p(\mathrm{TALLY})$. We define two sets C and D which are readily seen to be in $R_c^p(\mathrm{TALLY})$ since A_1 and B_1 are tally sets and since $R_{bd}^p(R_c^p(\mathrm{TALLY})) = R_c^p(\mathrm{TALLY})$ and $R_{bc}^p(R_c^p(\mathrm{TALLY})) = R_c^p(\mathrm{TALLY})$.

$$C = \{\langle a, b\rangle \mid a \in A_1 \text{ or } b \in B_1\}$$

$$D = \{\langle a, b\rangle \mid a \in A_1 \text{ and } b \in B_1\}.$$

Let f and g be co-rp reduction functions witnessing $A \leq_m^{co-rp} A_1$ and $B \leq_m^{co-rp} B_1$, respectively. We can assume that there is a uniform polynomial q corresponding to both reduction functions such that

$$x \in A \Rightarrow \mathrm{Prob}_{w \in \Sigma^{q(|x|)}}[f(\langle x, w\rangle) \in A_1] = 1$$

$$x \notin A \Rightarrow \mathrm{Prob}_{w \in \Sigma^{q(|x|)}}[f(\langle x, w\rangle) \notin A_1] \geq 3/4$$

and

$$x \in B \Rightarrow \mathrm{Prob}_{w \in \Sigma^{q(|x|)}}[g(\langle x, w\rangle) \in B_1] = 1$$

$$x \notin B \Rightarrow \mathrm{Prob}_{w \in \Sigma^{q(|x|)}}[g(\langle x, w\rangle) \notin B_1] \geq 3/4$$

We define a reduction function h combining f and g as follows. For $w_1, w_2 \in \Sigma^{q(|x|)}$, $h(\langle x, w_1 w_2\rangle) = \langle f(\langle x, w_1\rangle), g(\langle x, w_2\rangle)\rangle$. Then we have

$$x \in A \cup B \Rightarrow \mathrm{Prob}_{w \in \Sigma^{2q(|x|)}}[h(\langle x, w\rangle) \in C] = 1$$

$$x \notin A \cup B \Rightarrow \mathrm{Prob}_{w \in \Sigma^{2q(|x|)}}[h(\langle x, w\rangle) \notin C] \geq (3/4)^2$$

Note that the probability $(3/4)^2$ can be amplified using Lemma 4.4 to $3/4$ as required. Similarly we have

$$x \in A \cap B \Rightarrow \mathrm{Prob}_{w \in \Sigma^{2q(|x|)}}[h(\langle x, w\rangle) \in D] = 1$$

$$x \notin A \cap B \Rightarrow \mathrm{Prob}_{w \in \Sigma^{2q(|x|)}}[h(\langle x, w\rangle) \notin D] \geq 1 - (1/4)^2.$$

Hence $A \cap B$ and $A \cup B$ co-rp reduce to sets in $R_c^p(\mathrm{TALLY})$. Therefore $R_m^{co\text{-}rp}(R_c^p(\mathrm{SPARSE}))$ is closed under finite union and intersection. ∎

We are now ready to prove the main result of this section.

Theorem 4.9 *If A is in* NP *such that $\mathit{Left}(A) \in R_b^p(R_m^{co\text{-}rp}(R_c^p(\mathrm{SPARSE})))$ then $A \in \mathrm{RP}$.*

Proof The proof is quite similar to that of Theorem 3.4. The essential difference is that the procedures corresponding to SEARCH-LEFT and SEARCH-RIGHT in the proof of Theorem 3.4 will now be randomized algorithms (similar to the pruning part used in the algorithm in the proof of Theorem 4.5).

Let q be a polynomial and let P_A be a polynomial-time set such that $A = \{x \mid \exists w \in \Sigma^{r(|x|)} : \langle x, w\rangle \in P_A\}$ and $\mathit{Left}(A) = \{\langle x, w\rangle \mid x \in A \;\wedge\; w \in \Sigma^{r(|x|)} \;\wedge\; w \leq w_{max}\}$, where $w_{max} = \max\{w \in \Sigma^{r(|x|)} \mid \langle x, w\rangle \in P_A\}$. We describe an RP-algorithm that on input $x \in A$ computes with high probability w_{max} by a breadth-first search. By the hypothesis of the theorem we can assume that the set $\mathit{prefix}(\mathit{Left}(A)) = \{\langle x, y\rangle \mid \exists z : \langle x, yz\rangle \in \mathit{Left}(A)\}$ is in $R_b^p(R_m^{co\text{-}rp}(R_c^p(\mathrm{SPARSE})))$. Since $R_m^{co\text{-}rp}(R_c^p(\mathrm{SPARSE}))$ is closed under join and polynomial time many-one reductions, it follows from Fact 3.3 that $BC(R_m^{co\text{-}rp}(R_c^p(\mathrm{SPARSE})) = R_b^p(R_m^{co\text{-}rp}(R_c^p(\mathrm{SPARSE})))$. From Lemma 4.8 and Theorem 3.1, it follows that there exist a sparse set S and sets $C_i \in R_m^{co\text{-}rp}(R_c^p(S))$ such that $\mathit{prefix}(\mathit{Left}(A)) = \bigcup_{i=1}^k (C_{2i-1} - C_{2i})$ and $C_1 \supseteq C_2 \supseteq \ldots \supseteq C_{2k}$.

Let p be a polynomial such that for all n, $(1-2^{-p(n)})^{r(n)\cdot 2k} \geq 3/4$. Then by Lemma 4.4 we can assume that there exist co-rp reduction functions f_i witnessing $C_i \in R_m^{co\text{-}rp}(R_c^p(S))$, $1 \leq i \leq 2k$, and a polynomial q such that the following holds

$$\langle x, y\rangle \in C_i \Rightarrow \mathrm{Prob}_{w \in \Sigma^{q(|x|)}}[f_i(\langle x, y, w\rangle) \subseteq S] = 1$$

$$\langle x, y\rangle \notin C_i \Rightarrow \mathrm{Prob}_{w \in \Sigma^{q(|x|)}}[f_i(\langle x, y, w\rangle) \not\subseteq S] \geq (1 - 2^{-p(|x|)})$$

The indices $l(d)$ and $r(d)$, $1 \leq d \leq k$, are defined in the same way as in Theorem 3.4. Let y_h be the prefix of w_{max} in $\{y_1, \ldots, y_t\}$. As in Theorem 3.4, we cannot compute the required prefixes $y_{l(d)}$ and $y_{r(d)}$. Instead we design a randomized algorithm that, given $y_{l(d-1)}$, computes in polynomial time (polynomially size-bounded) sets $J_{right}(d)$ and $J_{left}(d)$ of prefixes such that with high probability $y_{r(d)} \in J_{right}(d)$ and $y_{l(d)} \in J_{left}(d)$. We now describe the algorithm in detail. It calls a recursive pruning procedure PRUNE which in turn calls two functions RANDOM-SEARCH-RIGHT and RANDOM-SEARCH-LEFT. RANDOM-SEARCH-RIGHT is used to search for candidates for $y_{r(d)}$ to the right of previously found candidates for $y_{l(d-1)}$ resulting in a polynomially bounded set $J_{right}(d)$ containing $y_{r(d)}$. RANDOM-SEARCH-LEFT is used to search to the left of the prefixes in $J_{right}(d)$ to form a polynomial size-bounded set $J_{left}(d)$ containing $y_{l(d)}$.

Let m be a polynomial bounding the size of the queries to the sparse set, i.e. $|z| \leq m(n)$ for all $z \in \bigcup\{f_i(\langle x, y, w\rangle) \mid 1 \leq i \leq 2k, |x| = n, |y| \leq r(|x|), |w| = q(|x|)\}$, and let s be a polynomial bounding the census of the sparse set S.

RANDOM-SEARCH-RIGHT(d, N, y_l, x)
(* if $\langle x, y_l\rangle \in C_{2d-1}$ it returns a set $J \subseteq N = \{y_1, \ldots, y_t\}$ that includes with high probability the largest prefix $y_r \in N$ such that $\{\langle x, y_l\rangle, \ldots, \langle x, y_r\rangle\} \subseteq C_{2d-1}$ *)

```
begin
  J := {y_t}
  Q := ∅
  i := l
  repeat
    i := i + 1
    compute a set Q(y_i) = f_{2d-1}(⟨x, y_i, w⟩) of conjunctive queries
     where w is chosen uniformly at random from Σ^{q(|x|)}
    if Q(y_i) ⊈ ⋃_{j=l}^{i-1} Q(y_j) then
      J := J ∪ {y_{i-1}}
      Q := Q ∪ Q(y_i)
    end
  until (|Q| > s(m(|x|))) or (i = t)
  return J
end
```

Claim 4 *Function* RANDOM-SEARCH-RIGHT(d, N, y_l, x), *when called with parameter* $y_l = y_{l(d-1)}$, *returns a set* J *that with probability at least* $1 - 2^{-p(|x|)}$ *contains* $y_{r(d)}$.

Proof of Claim 4 We first note that if $r(d) = t$ then $y_{r(d)}$ is always included in J. Otherwise observe the following.

1. For every $w \in \Sigma^{q(|x|)}$ and i, $l(d-1) \le i \le r(d)$, it holds that $f_{2d-1}(\langle x, y_i, w\rangle) \subseteq S$. This follows from the fact that $\langle x, y_i\rangle \in C_{2d-1}$ for all i, $1 \le i \le r(d)$.
2. Since $\langle x, y_{r(d)+1}\rangle \notin C_{2d-1}$ it holds that $\mathrm{Prob}_{w\in\Sigma^{q(|x|)}}[f_{2d-1}(\langle x, y_{r(d)+1}, w\rangle) \not\subseteq S] \ge 1 - 2^{-p(|x|)}$.

Therefore, since $|\bigcup_{j=l(d-1)}^{r(d)} Q(y_j)| \le s(m(|x|))$, the repeat loop includes $y_{r(d)}$ in J with probability at least $1 - 2^{-p(|x|)}$. □

```
RANDOM-SEARCH-LEFT(d, N, y_r, x)
(* returns a set J ⊆ N = {y_1, ..., y_t} that includes with high probability
   the smallest prefix y_l ∈ N such that {⟨x, y_l⟩, ..., ⟨x, y_r⟩} ⊆ C_{2d} *)
begin
  J := {y_1}
  i := r
  Q := ∅
  repeat
    compute a set Q(y_i) = f_{2d}(⟨x, y_i, w⟩) of conjunctive queries
    where w is chosen uniformly at random from Σ^{q(|x|)}
    if Q(y_i) ⊈ ⋃_{j=i+1}^{r} Q(y_j) then
      J := J ∪ {y_{i+1}}
      Q := Q ∪ Q(y_i)
    end
    i := i - 1
  until (|Q| > s(m(|x|))) or (i = 1)
  return J
end
```

Claim 5 *Function* RANDOM-SEARCH-LEFT(d, N, y_r, x)*, when called with parameter* $y_r = y_{r(d)}$*, returns a set* J *containing* $y_{l(d)}$ *with probability at least* $1 - 2^{-p(|x|)}$.

Proof of Claim 5 There are two cases again. If $l(d) = 1$ then $y_{l(d)}$ is clearly in the returned set J. Otherwise we have

1. Since $\{\langle x, y_{l(d)}\rangle, \ldots, \langle x, y_{r(d)}\rangle\} \subseteq C_{2d}$ and $\langle x, y_{l(d)-1}\rangle \notin C_{2d}$, it holds for every $w \in \Sigma^{q(|x|)}$ and for every i, $l(d) \leq i \leq r(d)$, that $f_{2d}(\langle x, y_i, w\rangle) \subseteq S$.
2. $\mathrm{Prob}_{w\in\Sigma^{q(|x|)}}[f_{2d}(\langle x, y_{l(d)-1}, w\rangle) \not\subseteq S] \geq (1 - 2^{-p(|x|)})$.

Therefore, since $|\bigcup_{j=l(d)}^{r(d)} Q(y_j)| \leq s(m(|x|))$, $y_{l(d)}$ is included in J with probability at least $1 - 2^{-p(|x|)}$ in some step of the repeat loop. The claim follows. $\square$

PRUNE(N, J'_{left}, d, x)
(* returns a subset of $N = \{y_1, \ldots, y_t\}$ that with high probability contains the prefix y_h of w_{max} if $y_h \in N$, $\langle x, y_h\rangle \in C_{2d-1}$, and $\{\langle x, y_l\rangle, \ldots, \langle x, y_h\rangle\} \subseteq C_{2d-1}$ for a $y_l \in J'_{left}$ with $l \leq h$ *)
begin
  if $d = k + 1$ **then return** $\emptyset$ **end**
  $J_{right} := \emptyset$
  for each $z \in J'_{left}$ **do**
    $J_{right} := J_{right} \cup$ RANDOM-SEARCH-RIGHT(d, N, z, x)
  end
  $J_{left} := \emptyset$
  for each $z \in J_{right}$ **do**
    $J_{left} := J_{left} \cup$ RANDOM-SEARCH-LEFT(d, N, z, x)
  end
  return $\{y_{l-1} \mid y_l \in J_{left}\} \cup$ PRUNE$(N, J_{left}, d + 1, x)$
end

The next claim follows from Claim 4 and Claim 5 and is similarly proved as Claim 3 in Theorem 3.4.

Claim 6 *If* $y_h \in N$*,* $\langle x, y_h\rangle \in C_{2d-1}$*, and* $y_{l(d-1)} \in J'_{left}$ *then the function call* PRUNE(N, J'_{left}, d, x) *returns a set containing* y_h *with probability at least* $(1 - 2^{-p(|x|)})^{2k}$.

We complete the algorithm with a description of the main program.

input x
$N := \{\epsilon\}$
for $i := 1$ **to** $r(|x|)$ **do**
  $N := \{y0 \mid y \in N\} \cup \{y1 \mid y \in N\}$ (* expand the prefixes to length i *)
  $N :=$ PRUNE$(N, \{y_1\}, 1, x)$
end
(* N now includes w_{max} if $x \in A$ with probability at least 3/4 *)
if there is a witness for x in N **then** *accept* **else** *reject* **end**

We first note that an input $x \notin A$ is rejected with probability 1 since no witness can be found. In order to prove the correctness of the algorithm it suffices to observe that Claim 6 implies that with probability at least $(1 - 2^{-p(|x|)})^{2k}$ the prefix y_h of w_{max} is included in the pruned set returned by PRUNE$(N, \{y_1\}, 1, x)$ provided that y_h is in N. Hence, after exiting the for-loop in the main program, N includes w_{max} with probability at least $((1-2^{-p(|x|)})^{2k})^{r(|x|)}$ (which is more than 3/4 by choice of p). It is easy to see that the algorithm runs in polynomial time. ∎

The proof of the following theorem is analogous to the proof of Theorem 3.5.

Theorem 4.10 *If* UP *is contained in* $R_b^p(R_m^{co-rp}(R_c^p(\text{SPARSE})))$ *then* UP $\subseteq$ RP.

Proof Since for every set $A \in$ UP it holds that *Left*(A) is in UP and since UP $\subseteq R_b^p(R_m^{co-rp}(R_c^p(\text{SPARSE})))$ it follows that *Left*(A) is in $R_b^p(R_m^{co-rp}(R_c^p(\text{SPARSE})))$, and therefore by Theorem 4.9 A is in RP. ∎

5 Promise problems and randomized reductions to sparse sets

We show in this section that it is enough to assume that some solution to the promise problem (1SAT, SAT) is reducible to a sparse set via the randomized reduction considered in Theorem 4.9 to get the conclusion NP = RP. We first give the definition of promise problems and state its relation to randomized reductions.

Definition 5.1 [ESY84] *A promise problem is a pair of sets (Q,R). A set L is called a solution to the promise problem (Q,R) if* $(\forall x)[x \in Q \Rightarrow (x \in L \Leftrightarrow x \in R)]$.

Let 1SAT denote the set of formulas with at most one satisfying assignment. Observe that a solution for the promise problem (1SAT, SAT) has to agree with SAT in the formulas having a unique satisfying assignment as well as in the unsatisfiable formulas. The well known result of Valiant and Vazirani proving the NP-hardness of USAT under (a weaker version of) randomized reductions [VV86] has the following implication for the promise problem (1SAT, SAT).

Theorem 5.2 [VV86] *If there is a solution of the promise problem* (1SAT, SAT) *in* RP *then* NP = RP.

We now prove the generalization of Theorem 4.9.

Theorem 5.3 *If there is a solution in* $R_b^p(R_m^{co-rp}(R_c^p(\text{SPARSE})))$ *for the promise problem* (1SAT, SAT) *then* NP = RP.

Proof Let $L \in R_b^p(R_m^{co-rp}(R_c^p(\text{SPARSE})))$ be a solution of the promise problem (1SAT, SAT). Then, by definition, $(\forall x)[x \in \text{1SAT} \Rightarrow (x \in L \Leftrightarrow x \in \text{SAT})]$. The natural left set associated with SAT is *Left*(SAT) $= \{\langle x, w\rangle \mid x \in \text{SAT},\ w \in \Sigma^{l(x)}$ and $w \leq w_{max}\}$ where w_{max} is the maximum satisfying assignment for x and $l(x)$ is the number of variables in x. The set *prefix*(*Left*(SAT)) $= \{\langle x, y\rangle \mid (\exists z)[\langle x, yz\rangle \in$ *Left*(SAT)$]\}$

is easily seen to be accepted by an NP-machine that on input $\langle x, y\rangle$ guesses a truth assignment $w \geq y0^{l(x)-|y|}$ and verfies that w satisfies x. It is clear that $x \in$ 1SAT implies that for all $y \in \Sigma^{\leq l(|x|)}$, the above mentioned NP-machine has at most one accepting path on input $\langle x, y\rangle$. Let g be a parsimonious many-one reduction from *prefix*(*Left*(SAT)) to SAT. Then it is clear from the discussion that $x \in$ 1SAT implies $g(\langle x, y\rangle) \in$ 1SAT for all $y \in \Sigma^{\leq l(|x|)}$.

Let $Q = \{\langle x, y\rangle \mid x \in \text{1SAT}\}$ and let $L' = \{\langle x, y\rangle \mid g(\langle x, y\rangle) \in L\}$. Clearly g many-one reduces L' to L.

Claim 7 *L' is a solution of the promise problem $(Q, \textit{prefix}(\textit{Left}(\text{SAT})))$.*

Proof of Claim
We have to show that for every pair $\langle x, y\rangle \in Q$ it holds that $\langle x, y\rangle \in L' \Leftrightarrow \langle x, y\rangle \in \textit{prefix}(\textit{Left}(\text{SAT}))$. Since L is a solution of (1SAT, SAT) and since $\langle x, y\rangle \in Q$ implies $g(\langle x, y\rangle) \in$ 1SAT, it follows that $g(\langle x, y\rangle) \in L$ if and only if $g(\langle x, y\rangle) \in$ SAT. Since g many-one reduces both *prefix*(*Left*(SAT)) to SAT and L' to L, it follows that $\langle x, y\rangle \in L' \Leftrightarrow \langle x, y\rangle \in \textit{prefix}(\textit{Left}(\text{SAT}))$. □

Since $L' \leq_m^p L$ it follows that $L' \in R_b^p(R_m^{co\text{-}rp}(R_c^p(\text{SPARSE})))$. Also, L' can be written as $\bigcup_{i=1}^k (C_{2i-1} - C_{2i})$ for sets $C_1 \supseteq C_2 \supseteq \ldots \supseteq C_{2k}$ in $R_m^{co\text{-}rp}(R_c^p(\text{SPARSE}))$, since $R_m^{co\text{-}rp}(R_c^p(\text{SPARSE}))$ is a set ring. Let f_i, $1 \leq i \leq 2k$ be co-rp reduction functions witnessing $C_i \in R_m^{co\text{-}rp}(R_c^p(\text{SPARSE}))$. Consider the algorithm described in the proof of Theorem 4.9 in which we use the reduction functions f_i defined above. We claim that on input $x \in \text{1SAT} \cap \text{SAT}$ this algorithm computes with high probability the unique satisfying assignment for x. In order to see this, note that the algorithm on input $x \in \text{1SAT} \cap \text{SAT}$ always computes query sets $F(\langle x, y, w\rangle)$ for pairs $\langle x, y\rangle \in Q$. Since for such pairs $\langle x, y\rangle \in L' \Leftrightarrow \langle x, y\rangle \in \textit{prefix}(\textit{Left}(\text{SAT}))$, the arguments in the proof of Theorem 4.9 apply. Hence there is an RP solution for the promise problem (1SAT, SAT) and by Theorem 5.2 it follows that NP = RP. ■

The following theorem concerning deterministic reductions can be similarly proved.

Theorem 5.4 *If there is a solution of* (1SAT, SAT) *in* $R_b^p(R_c^p(\text{SPARSE}))$ *then there is a solution of* (1SAT, SAT) *in* P.

We need the following lemma for the next corollary.

Lemma 5.5 *Let L be a solution of* (1SAT, SAT) *then* Few $\subseteq \text{P}^L$.

Proof Since Few $\subseteq \text{P}^{\text{FewP}}$ [KSTT] it suffices to show that FewP is contained in P^L. Let A be a set in FewP via some nondeterministic machine M. Let p be the polynomial bounding the number of accepting paths of M. Consider the following NP set B.

$$B = \{\langle x, i\rangle \mid M(x) \text{ has at least } i \text{ accepting paths }\}$$

Let $acc_M(x)$ denote the number of accepting paths of M on input x. Clearly, there is an NP machine M' accepting B in such a way that M' on input $\langle x, acc_M(x)\rangle$ has exactly one accepting path. Then it holds that $f(\langle x, j\rangle)$ is in 1SAT for all $j \geq acc_M(x)$, where f is a parsimonious reduction from $L(M')$ to SAT. Therefore x is in A if and only if $f(\langle x, i\rangle) \in L$ for some $i, 1 \leq i \leq p(|x|)$. ■

Corollary 5.6 *If there is a solution of* (1SAT, SAT) *in* $R_b^p(R_c^p(\text{SPARSE}))$ *then* Few = P.

6 A trade-off analysis

It is interesting to note that the proof of Theorem 3.4 is constructive in the following sense: given a polynomial time truth-table condition generator g witnessing $\mathit{prefix}(\mathit{Left}(A))$ in $R_b^p(R_c^p(S))$ for a sparse set S, the conjunctive query sets $f_i(\langle x,y\rangle)$ can be computed from the truth-table condition $g(\langle x,y\rangle)$ by a polynomial time algorithm (as can be derived from a general result in [WW85] that applies to different set rings). Therefore, given an FP transducer computing g and a polynomial bound on the census of S, we get a polynomial time decision procedure for A. The question arises how the running time of that algorithm is influenced if the number k of the conjunctive queries produced by g is a function $k(n)$ depending on the length n of x rather than a constant, and if the census of S is allowed to be a superpolynomial function.

In the next theorem we precisely analyze the running time of the algorithm in terms of the functions k and $census_S$, assuming that g directly generates truth-table conditions suitable for our algorithm.

More precisely, we assume that g is a truth-table reduction of the following type which we call Hausdorff reduction and which is a variation of the reduction defined by K.W. Wagner in [Wag87].

Definition 6.1 *Let $h(x_1, x_2, \ldots, x_k)$ be the boolean formula $\oplus_{i=1}^{k}(\wedge_{j=1}^{i} x_i)$, where $\oplus$ denotes the parity operator. We say that a set A is $k(n)$-Hausdorff reducible to B if A is truth-table reducible to B via the boolean function $h(x_1, x_2, \ldots, x_{k(n)})$, i.e. there is a polynomial time computable query generator g such that for all x, $x \in A \Leftrightarrow h(\chi_B(y_1), \cdots, \chi_B(y_{k(|x|)})) = 1$ where $g(x) = \langle y_1, \cdots, y_{k(|x|)}\rangle$.*

Observe that $A \in R_b^p(R_c^p(\mathrm{SPARSE}))$ if and only if A is (bounded) Hausdorff reducible to some set in $R_c^p(\mathrm{SPARSE})$. We now state the trade-off result.

Theorem 6.2 *If B is a set of density bounded by an* FP *function c_B and if some* NP *complete set is polynomial time reducible to a set in $R_c^p(B)$ by a $k(n)$-Hausdorff reduction, then* $\mathrm{NP} \subseteq \bigcup_{j\geq 0} \mathrm{DTIME}(n^j \cdot c_B(n^j)^{O(k(n^j))})$.

Proof Suppose that a set B as in the statement exists. Then for $A \in \mathrm{NP}$ the set $\mathit{prefix}(\mathit{Left}(A))$ is reducible via a $k(n^{O(1)})$-Hausdorff reduction to a set in $R_c^p(B)$. It is not hard to see that there is an FP function f such that for all x,y

$$\langle x,y\rangle \in \mathit{prefix}(\mathit{Left}(A)) \Leftrightarrow \max\{i \mid 1 \leq i \leq k(|x|), f(\langle i,x,y\rangle) \subseteq B\} \text{ is odd}$$

Observe that the function f can be used instead of the conjunctive reduction functions f_i in the proof of Theorem 3.4. Therefore with a minor modification we can use the algorithm described in the proof of Theorem 3.4 to compute w_{max} on input $x \in A$. It only remains to accurately analyze the running time of the algorithm taking into account the growth rate of the function k and of the density of B.

First note that the depth of recursive calls made by procedure PRUNE is $k(n^{O(1)})$. Next we examine the calls to SEARCH-LEFT and SEARCH-RIGHT. The size of the set returned by each such call is bounded by $c_B(n^{O(1)})$. Since the pruned subset of N returned by PRUNE is constructed by taking prefixes corresponding to all subsets returned by SEARCH-LEFT in all the recursive calls by PRUNE, the size of this pruned subset is bounded by $c_B(n^{O(1)})^{O(k(n^{O(1)}))}$. The running time of each call is bounded by $n^{O(1)} \cdot |N|$.

Furthermore, the total number of calls made by PRUNE to both SEARCH-RIGHT and SEARCH-LEFT (including the recursive calls) is bounded by $2k(n^{O(1)})\cdot|N|$. This implies that the running time of one call to prune is bounded by $k(n^{O(1)})\cdot n^{O(1)}\cdot c_B(n^{O(1)})^{O(k(n^{O(1)}))}$. The overall running time of the algorithm has only an additional polynomial factor. This completes the proof. ■

An interesting point in the above result is that the actual number of queries in the conjunctive reduction plays no real role in the trade-off. The next corollary to the above theorem is similar to a result in [HL91] concerning $f(n)$-tt hard sets of certain densities for NP.

Corollary 6.3 *If B is a set of density $log(n)$ such that an* NP *complete set is reducible to a set in $R_c^p(B)$ by a $log(n)/log(log(n))$-Hausdorff reduction then* P = NP.

Corollary 6.4 *If an* NP *complete set is reducible to a set in R_c^p(SPARSE) by a $log(n)$-Hausdorff reduction then* NP $\subseteq$ DTIME$(2^{O(\log^2 n)})$.

We now give a trade-off analysis for the algorithm in Theorem 4.9.

Theorem 6.5 *If B is a set of density $c_B(n)$ such that an* NP *complete set is polynomial-time reducible to a set in $R_m^{co\text{-}rp}(R_c^p(B))$ by a $k(n)$-Hausdorff reduction then* NP $\subseteq \bigcup_{j\geq 0}$ RTIME$(n^j\cdot c_B(n^j)^{O(k(n^j))})$.

Proof Follows easily (as done in Theorem 6.2) from a simple analysis of the corresponding algorithm in the proof of Theorem 4.9. One thing to be noted is that, for amplification, the polynomial p in the proof of Theorem 4.9 can be again chosen so that $(1-2^{-p(|x|)})^{r(|x|)\cdot 2k(n)}\geq 3/4$, since $k(n)$ is anyway bounded by some polynomial. ■

Corollary 6.6 *If an* NP *complete set is reducible to a set in $R_m^{co\text{-}rp}(R_c^p(B))$ by a* $\log(n)/\log(\log(n))$*-Hausdorff reduction, where B is a set of density* $\log(n)$ *then* NP = RP.

Corollary 6.7 *If an* NP *complete set is reducible to a set in $R_m^{co\text{-}rp}(R_c^p(B))$ by a* $\log(n)$*-Hausdorff reduction, where B is a set of density* $\log(n)$ *then* NP $\subseteq$ RTIME$(2^{\log^2 n})$.

7 Nondeterministic reductions to Sparse and Tally sets

In this section we consider classes of sets reducible to sparse and tally sets via polynomial time nondeterministic reductions. We show that nondeterministic polynomial time many-one reductions to sparse sets are as powerful as nondeterministic Turing reductions to sparse sets. On the other hand, nondeterministic polynomial time many-one reductions to co-sparse sets are much weaker. We substantiate this claim by proving, applying essentially Kadin's census technique [Kad88], that if coNP is nondeterministic polynomial time many-one reducible to a co-sparse set then PH = Θ_2^p. A similar result for sparse sets is unlikely since it would imply that the Karp/Lipton/Sipser result [KL80] that NP $\subseteq$ P/*poly* implies PH = Σ_2^p could be improved (known to be impossible in relativized worlds [IM89]).

The following definitions of nondeterministic polynomial time reductions are due to Ladner, Lynch and Selman [LLS75].

Definition 7.1 [LLS75]

1. *A set A is polynomial time nondeterministic many-one reducible to a set B (denoted $A \leq_m^{np} B$) if there exists a polynomial time nondeterministic Turing machine M such that for every $x \in \Sigma^*$, $M(x)$ outputs a string along each computation path, and, $x \in A$ iff 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 $A \leq_T^{np} B$) if there exists a polynomial time nondeterministic oracle Turing machine M such that for every $x \in \Sigma^*$, $x \in A$ iff there exists an accepting computation path of M relative to B.*

Our first result in this section is the equality $R_m^{np}(\text{SPARSE}) = R_T^{np}(\text{SPARSE})$. Indeed, we show that every set in $R_T^{np}(\text{SPARSE})$ nondeterministically many-one reduces to very sparse sets (in the sense that they contain at most one string of each length). Note that in [AHOW] it is shown, using the inclusion $R_b^p(\text{SPARSE}) \subseteq R_d^p(\text{SPARSE})$ that $R_c^{np}(\text{SPARSE}) = R_T^{np}(\text{SPARSE})$ and that $R_m^{np}(\text{SPARSE}) = R_d^{np}(\text{SPARSE})$.

Theorem 7.2 $R_m^{np}(\text{SPARSE}) = R_T^{np}(\text{SPARSE})$

Proof Let $A = L(M, S) \in R_T^{np}(S)$ for a sparse set S and a nondeterministic Turing machine M. Let p be a polynomial bounding the running time of M, and let q be a polynomial bounding the census of S. Clearly, each query generated by M along any computation path on an input x is in $\Sigma^{\leq p(|x|)}$. We define a sparse set S' to which A can be nondeterminstically many-one reduced in polynomial time.

$$S' = \{\langle 0^n, y_1, y_2, \cdots, y_r\rangle \mid y_1 < \cdots < y_r \text{ and } \{y_1, \cdots, y_r\} = S^{\leq n}\}$$

Note that S' has for each length at most one element. Consider the following NP-machine M'.

> **input** x
> guess $r \in \{0, 1, 2, \ldots, q(p(|x|))\}$
> guess strings $y_1 < y_2 < \cdots < y_r$ in $\Sigma^{\leq p(|x|)}$
> guess a path ρ of M on input x with oracle $\{y_1, y_2, \cdots, y_r\}$
> **if** ρ is accepting **then**
>   **output** $\langle y_1, y_2, \cdots, y_r\rangle$
> **else** **output** y_0 (* a fixed string not in S' *)
> **end**

It is not hard to see that M' witnesses $A \in R_m^{np}(S')$. This completes the proof.

∎

Related to the broad question discussed in Sections 3 and 4 whether NP can have hard sparse sets with respect to deterministic or randomized polynomial time reductions (of different kinds) one can ask similar questions with respect to nondeterministic polynomial time reductions.

1. Can coNP have sparse hard sets under nondeterministic polynomial time many-one reductions?

2. Can coNP have co-sparse hard sets under nondeterministic polynomial time many-one reductions?

Such questions have been implicitly considered in the literature with regard to nonuniform classes. Balcázar and Schöning [BaSchö88] show that coNP $\subseteq$ NP/*log* implies PH $= \Theta_2^p$. Similarly it is known that coNP $\subseteq$ NP/*poly* implies PH $= \Sigma_3^p$ [BBS86].

Theorem 7.3 *If* coNP $\subseteq R_m^{np}$(co-SPARSE) *then* PH $= \Theta_2^p$.

Proof Let A be a complete set for coNP that is many-one reducible via a nondeterministic polynomial time Turing machine M to the complement $\overline{S}$ of a sparse set S. We define a sparse set $S' \in$ (SPARSE $\cap$ NP) such that $A \in \mathrm{NP}^{S'}$. This proves the theorem since by Kadin's result [Kad88] it follows that PH $= \Theta_2^p$. S' contains all strings that are provably not in $\overline{S}$.

$$S' = \{y \mid \exists x \in \overline{A} : M \text{ on input } x \text{ outputs } y \ \}$$

It is clear that $S' \subseteq S$ and it remains to show that $A \in \mathrm{NP}^{S'}$. Consider the following nondeterminstic oracle machine M'.

> On input x, M' simulates M on x. If M outputs y at the end of the simulated computation path then M' queries oracle S' for y and accepts if and only if $y \notin S'$.

If $x \in A$ then some output y of M on input x is in $\overline{S}$. Since $S' \subseteq S$, it holds that $y \notin S'$ and hence $x \in L(M', S')$. If $x \notin A$ then every output of M on input x is in S' and therefore $x \notin L(M', S')$. Therefore $L(M', S') = A$. ∎

From Theorem 7.2 we know that R_m^{np}(SPARSE) $=$ NP/*poly*, hence it appears difficult to get a comparable collapse of PH as in Theorem 7.3 under the assumption that coNP $\subseteq R_m^{np}$(SPARSE). It is known [BBS86] that coNP $\subseteq$ NP/*poly* implies PH $= \Sigma_3^p$.

Finally, we have similar to Theorem 3.4 a result on the collapse of the polynomial hierarchy for nondeterministic reductions.

Theorem 7.4 *If* $\Sigma_2^p \subseteq R_b^p(R_m^{np}(\text{co-SPARSE}))$ *then* PH $= \Delta_2^p$.

References

[AM77] L. Adleman and K. Manders. Reducibility, randomness and intractability, In *Proceedings of the 9th ACM Sympos. Theory of Computing* 151–163.

[AHH+92] V. Arvind, Y. Han, L. Hemachandra, J. Köbler, A. Lozano, M. Mundhenk, M. Ogiwara, U. Schöning, R. Silvestri, and T. Thierauf. Reductions to sets of low information content. *In Proceedings of the 19th International Colloquium on Automata, Languages, and Programming.* To appear.

[AHOW] E. Allender, L. Hemachandra, M. Ogiwara, and O. Watanabe. Relating equivalence and reducibility to sparse sets. *SIAM Journal on Computing.* To appear. Preliminary version appears as [AHOW91].

[AHOW91] E. Allender, L. Hemachandra, M. Ogiwara, and O. Watanabe. Relating equivalence and reducibility to sparse sets. *Proceeding 6th Structure in Complexity Theory Conference*, 1991.

[ABG90] A. Amir, R. Beigel, and W. Gasarch. Some connections between bounded query classes and non-uniform complexity. In *Proceedings of the 5th Structure in Complexity Theory Conference*, pages 232–243. IEEE Computer Society Press, July 1990.

[Bal90] J.L. Balcázar. Self-reducibility. *Journal of Computer and System Sciences*, 41(3):367–388, 1990.

[BBS86] J. Balcázar, R. Book, and U. Schöning. Sparse sets, lowness and highness. *SIAM J. Comput.*, 15:739–745, 1986.

[BaSchö88] J.L. Balcázar and U. Schöning. Logarithmic advice classes. Bericht 9/88, Informatik, EWH Koblenz, to appear in Theoretical Computer Science.

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

[BLS92] H. Buhrman, L. Longpré, and E. Spaan. personal communication, 1992.

[CH91] J. Cai and L. Hemachandra. A note on enumerative counting. *Information Processing Letters*, 38(4):215–219, 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.

[Hau14] F. Hausdorff. *Grundzüge der Mengenlehre*. Leipzig, 1914.

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

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

[Kad88] J. Kadin. *Restricted Turing reducibilities and the structure of the polynomial time hierarchy*. PhD thesis, Cornell University, 1988.

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

[KSTT89] J. Köbler, U. Schöning, S. Toda, J. Torán. Turing machines with few accepting computations and low sets for PP. In *Proceedings of the 4th Structure in Complexity Theory Conference*, pages 208–215. IEEE Computer Society Press, June/July 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*. To appear. Preliminary version appears as [KSTT89].

[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.

[OL91] M. Ogiwara and A. Lozano. On one query self-reducible sets. In *Proceedings of the 6th Structure in Complexity Theory Conference*, pages 139–151. IEEE Computer Society Press, June/July 1991.

[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, June 1991.

[RR92] D. Ranjan and P. Rohatgi. Randomized reductions to sparse sets. In *Proceedings 7th Structure in Complexity Theory Conference*. To appear.

[VV86] L.G. Valiant and V.V Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47(1):85-93, 1986.

[Wag87] K.W. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science*, 51(1):53-80, 1987.

[WW85] G. Wechsung and K.W. Wagner. On the boolean closure of NP. Manuscript. (Extended abstract by: G. Wechsung, On the boolean closure of NP, in *Proc. 1985 International Conference on Fundamentals of Computation Theory*, pages 485–493. Lecture Notes in Computer Science, Springer-Verlag, 1985.)