Sparse sets, simplicity, and lowness

Martin Mundhenk
Universität Ulm
Abteilung für Theoretische Informatik
D-89069 Ulm, Germany

1 Introduction

Sets are encodings of information. The quantitative complexity of a set – i.e. the amount of resources needed to decide the membership problem – depends on both, the information and on its encoding. For example, let SAT be the "natural" encoding of all satisfiable Boolean formulae over the alphabet $\Sigma = \{0, 1\}$. SAT is decidable in exponential time and not assumed to be decidable in polynomial time. The set $tally(\text{SAT}) = \{0^n \mid \text{the } n\text{-th string of } \Sigma^* \text{ is in SAT}\}$ has the same information content as SAT, but because of the encoding it is decidable in polynomial time. But this easy decidability destroys "natural" structures of the information and decreases – in a qualitative meaning – the use of it. In structural complexity theory those qualitative properties – e.g. completeness, lowness, sparseness, or self-reducibility – are considered, and their effects to quantitative properties are studied. For example, SAT is known to be NP-*complete*, what means that all information which can be retrieved from *any* NP set at all can also easily be retrieved from SAT, which itself belongs to NP. The notion of NP-*lowness* means the contrary, such that the existence of a set having both these properties has some unexpected consequences. Other notions describe specific internal structures of a set. A set is *sparse*, if it contains at most polynomially many strings of each length, and it is *self-reducible* if its decision problem can be solved by some recursive program. Both these notions imply low information content in a structural way: the first one since the small information density restricts the amount of information which is easily accesible, the other one since a lot of information contained in the set describes its own structure. Nevertheless, many complete sets for several complexity classes are known to be self-reducible (for example SAT), whereas the "nature" of sparse sets implies lowness. Thus, self-reducibility and sparseness also seem to be contradictory properties in spite of they are defined by very different concepts.

In 1977, Berman and Hartmanis [BH77] conjectured that all NP-complete sets are polynomially isormorphic to SAT. This means, that the information content and encoding of all NP-complete sets has to be very similar to SAT. Since SAT has exponential density, i.e. for a constant c there are at least 2^{n^c} elements of length n contained in the set, no set having polynomial density can be NP-complete if that conjecture is true. So the question arised if independent of the conjecture the two structural properties of NP-completeness and sparseness are contradictory. To answer this question is of great interest also from a more (somehow) "practical" view. Since the P = NP question is unsolved, (and since the general believe is P $\neq$ NP,) one wants to know if NP-complete sets can be decided by algorithms with a complexity which does not differ very much from P, in other words, if NP-complete sets are decidable in "quasi-polynomial-time". Several examples for types of such algorithms were studied. Algorithms which decide almost all instances of a set in polynomial time, or which decide all instances in polynomial time but are allowed to err on few instances; polynomial-time *randomized* algorithms, or *sequences of circuits*. It turned out that the above types of algorithms have a common property, such that asking for the existence of such an algorithm for an NP-complete set is the same as asking if the set can be captured by a small amount of information and – if the answer is yes – which resources are needed to make use of this information.

In complexity theory, the restrictions of access to information encoded into a set is measured in terms of polynomial time reducibilities, the same notion underlying the concept of completeness. As a standard example for sets with low information content *sparse* sets are treated. They have the useful property that for every n there is a table of size polynomial in n which lists all the elements of the set of length at most n. Thus, the sets which are polynomial time reducible to sparse sets – called *sparse-reducible sets* from now on – have information content only differing a little from sets in P. Mahaney's theorem [Mah82] that P = NP if the NP-complete set SAT many-one reduces to a sparse set initiated a broad study of sparse-reducible sets.

This paper will give an overview on the results in this area. In Section 3 the inclusion structure of classes defined by various reducibilities to sparse sets is considered. Dependent on the computational resources of the reducibility more or less information can be extracted from a sparse set. In the next both sections we follow up the lines of different proofs showing how sparse-reducible sets collaps the Polynomial Time Hierarchy. Since every sparse-reducible set is accepted by polynomially size bounded circuits also the connection between the type of reducibility under which a set reduces to a sparse set and the resources needed to compute its circuit are considered. This leads to the notion of *advice complexity* discussed in Section 4. In a sense, it gives an upper bound of "structuredness" of sparse-reducible sets. In Section 5, these structural properties are compared with properties of internally very well structured sets, the *self-reducible sets*. Since these sets contain a lot of information about their own structure, sets which are both sparse- *and* self-reducible are known to have only low complexity. Finally in Section 6 it will be shown how the Polynomial Hierarchy collapses if the NP-complete set SAT is sparse-reducible.

2 Notations

In this section, we will define some notations used throughout this paper. The standard alphabet $\Sigma = \{0,1\}$. Σ^n denotes all words in Σ^* of length n, and $\Sigma^{\leq n} = \bigcup_{0\leq i\leq n} \Sigma^i$. For any set $A \subseteq \Sigma^*$, $A^{\leq n} = A \cap \Sigma^{\leq n}$. $|x|$ denotes the length of string x, and $||A||$ denotes the cardinality of the set A. χ_A is the characteristic function of set A. $\chi^A_{\leq n}$ denotes the *characteristic sequence* of A for all strings up to length n, i.e. it is the string $\chi_A(l_0)\cdots\chi_A(l_{2^{n+1}-1})$ where l_i denotes the i-th string in the lexicographical order of Σ^*.

A set A is called *tally*, if $A \subseteq 0^*$. A set A is called *sparse*, if for some polynomial p and every n, $||A^{\leq n}|| \leq p(n)$. SPARSE resp. TALLY denote the class of all sparse resp. tally sets. co-SPARSE is the class of all sets which have sparse complements.

We fix some Universal Turing machine U, and define the *Kolmogorov complexity* of a string x as the length of the shortest string z such that $U(z)$ outputs x. For any natural number n, $K[n]$ is the set of all strings having Kolmogorov complexity less or equal n. Observe that for a constant c, every string x is in $K[|x|+c]$. By an easy counting argument one can see that $\Sigma^{\leq n}$ contains at least one string not in $K[n-1]$ for every n. For some m, the *m-time bounded Kolmogorov complexity* (resp. the *m-space bounded Kolmogorov complexity*) of a string x is the length of the shortest string z such that $U(z)$ outputs x using not more that m steps (resp. space). $KT[m,n]$ (resp. $KS[m,n]$) is the set of strings having n-time (resp. space) bounded Kolmogorov complexity m.

The reducibilities are defined via oracle Turing machines having polynomial time bounds and various query mechanisms (see [LLS75, Wag90]). We will shortly review the *truth-table* reducibilities.

Definition 2.1 *Let A and B be sets.*

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

2. *A conjunctively reduces to B* ($A \le_{c}^{p} B$)*, if* $A \le_{tt}^{p} B$ *via f and g as above where g(x) is a truth-table which evaluates to true on the m-tuple consisting only of* 1*'s, and evaluates to false otherwise.*

3. *A disjunctively reduces to B* ($A \le_{d}^{p} B$)*, if* $\overline{A} \le_{c}^{p} \overline{B}$.

4. *A Hausdorff reduces to B* ($A \le_{hd}^{p} B$)*, if* $A \le_{tt}^{p} B$ *via f and g as above such that for all x, if* $f(x) = \langle x_1, \ldots, x_m \rangle$, $\chi_B(x_1) \ge \chi_B(x_2) \ge \cdots \ge \chi_B(x_m)$*, and* $x \in A$ *if and only if* $\chi_B(x_i) > \chi_B(x_{i+1})$ *for some odd i* ($1 \le i \le m-1$).

For any constant k and reduction $\le_r^p$, $A \le_{k\text{-}r}^{p} B$ denotes that $A \le_r^p B$ such that at most k queries to the oracle are made on any input x. $A \le_{br}^{p} B$ denotes that $A \le_{k\text{-}r}^{p} B$ for some k. The notation for classes of sets which reduce to a set in the class C via the reducibility $\le_r^p$ is $R_r^p(C) = \{A \mid \exists B \in C : A \le_r^p B\}$.

In the discussions of sets with small information content also classes with size-bounded circuits play an important role. They represent the concept of *non-uniform* algorithms, as defined by the non-uniform complexity classes.

Definition 2.2 *Let C be a function class. A set A is in* P/C*, if there exists a* $B \in$ P *and an* $f \in C$ *such that for all x,* $x \in A \iff \langle x, f(0^{|x|}) \rangle \in B$.

We use *poly* (resp. *log*) for the classes of functions f for which there exists a polynomial p such that for all x, $|f(x)| \le p(|x|)$ (resp. $|f(x)| \le \log p(|x|)$).

For further notations we refer to [BDG].

3 Reduction classes to sparse sets

The comparison of reduction classes to sparse sets defined by various reducibilities gives an insight in how information can be encoded into a sparse set and how weak access to an oracle restricts the usability of low information. A lot of work is done in this area, initiated by Book and Ko [BK88] who proved that (for every k) k truth-table queries to a sparse oracle are less powerful than $k+1$ queries and therefore showed that $R_{tt}^p(\text{SPARSE})$ contains an infinite hierarchy of reduction classes to sparse sets defined by restricted oracle access.

Theorem 3.1 *[BK88] For every* $k \ge 1$: $R_{k\text{-}tt}^p(\text{SPARSE}) \subset R^p(k+1)\text{-}tt(\text{SPARSE})$.

At first we will review some proof ideas which show how to encode sparse into tally sets. The next Theorem is a folklore example for this technique.

Theorem 3.2 *(see [BK88])* $R_T^p(\text{SPARSE}) = R_{tt}^p(\text{TALLY})$.

Proof Sketch Clearly $R_{tt}^p(\text{TALLY}) \subseteq R_T^p(\text{SPARSE})$ holds. For the opposite direction, let S be a sparse set. Define $T = \{0^{\langle n,m,i,b \rangle} \mid$ the i-th bit of the m-th word in $S^{=n}$ equals $b\}$. Since S is sparse, all bits of all words of $S^{\le n}$ are accessible by queries y in $Q_n = \{0^{\langle n,m,i,b \rangle} \mid m \le p(n), i \le n, b \in \{0,1\}\}$ for a polynomial p. Since $Q_n \subseteq 0^{\le q(n)}$ for a polynomial q, $T \cap Q_n$ can be computed using a truth-table reduction to T in time polynomial in n, what suffices to reconstruct $S^{\le n}$ and to simulate the Turing reduction to S. ∎

It follows immediately that $R_T^p(\text{SPARSE}) = R_{tt}^p(\text{SPARSE})$.

A very involved encoding of sparse sets into tally sets were found by Buhrman, Longpré, and Spaan.

Theorem 3.3 *[BLS93]* SPARSE $\subset R^p_c$(TALLY).

Proof Sketch Let S be a sparse set and p be a polynomial bounding the census of S. Consider S as a set of natural numbers. Every string s of $S^{=n}$ can uniquely be represented by the remainders of the ordinal number of s (in lex order) of the first $p(n)$ prime numbers. Since the size of the these prime numbers are also polynomially bounded in n, they are representable in a tally set. ∎

Theorem 3.4 $R^p_{hd}(R^p_{bd}(\text{SPARSE})) \subseteq R^p_d(\text{SPARSE})$.

Proof Sketch We sketch the proof for $R^p_{hd}(\text{SPARSE}) \subseteq R^p_d(\text{SPARSE})$. The proof of the theorem is an straightforward extension of that proof. Let $A \leq^p_{hd} S$ for a sparse set S via a reduction function f. Then for all x and $f(x) = \langle x_1, \ldots, x_{2m} \rangle$, $x \in A$ iff for some $i \leq m$: $x_{2i-1} \in S$ and $x_{2i} \notin S$. By Theorem 3.3 there exists a function g and a tally set T such that $\overline{S} \leq^p_d T$. Thus, $x \in A$ iff for some $i \leq m$: $x_{2i-1} \in S$ and $g(x_{2i}) \cap T \neq \emptyset$. Therefore A disjunctively reduces to the sparse set $\hat{S} = \{\langle x, y \rangle \mid x \in S, y \in T\}$ via the FP reduction function $f'(x) = \bigcup_{f(x)=\langle x_1,\ldots,x_{2m}\rangle} \{\langle x_{2i-1}, y \rangle \mid y \in g(x_{2i})\}$. ∎

In several cases the truth-table reducibility can be replaced by the more "structured" Hausdorff reducibility.

Theorem 3.5 *[AKM92b]*

1. $R^p_{btt}(R^p_{bd}(\text{SPARSE})) = R^p_{bhd}(R^p_{bd}(\text{SPARSE}))$
2. $R^p_{btt}(R^p_c(\text{SPARSE})) = R^p_{bhd}(R^p_c(\text{SPARSE}))$

Now we turn to show differences between reduction classes to sparse sets. The underlying technique is to "hide" information in a sparse set such that every reorganization which makes it usable under restricted access leads to a lost of information.

Theorem 3.6 *[Ko89]* SPARSE $\not\subseteq R^p_d$(TALLY).

Proof Sketch Let S be a set containing exactly one string of high Kolmogorov complexity of each length; i.e. $S \cap \Sigma^n \neq \emptyset$ for all n and $z \notin K[|z| - 1]$ for all $z \in S$. Clearly S is sparse. Assume for all x, $x \in S \iff f(x) \cap T \neq \emptyset$ for an FP function f and a tally set T. Let z_n be the string in S of length n. Then every z_n has the description "the string x of length n for which $f(x)$ contains 0^m" for $0^m \in f(z_n) \cap T$. Since m is polynomially bounded in n, this description has size $O(\log n)$, contradicting the high Kolmogorov complexity of all z_n. ∎

Theorem 3.7 *[Gav92a]* R^p_c(SPARSE) $\not\subseteq R^p_d$(SPARSE).

Proof Sketch There exists a non sparse set L in R^p_c(SPARSE) such that for all $c > 0$ for almost every n and for all pairs of different strings $x, y \in L^{=n}$ the Kolmogorov complexity of x relative to y is greater than $c \log n$, i.e. $K[x|y] > c \log n$. Assume for all x, $x \in L \iff f(x) \cap S \neq \emptyset$ for an *FP* function f and a sparse set S. Since L is not sparse, for infinitely many n there exist $x, y \in L^{=n}$ which produce the same oracle queries in S, i.e. $f(x) \cap f(y) \cap S \neq \emptyset$. Let i be the index of that common query in S of $f(x)$. Then "the lexicographically smallest u (different from x) of length n for which $f(u)$ contains the i-th query of $f(x)$" is a description of a string in $L^{=n}$ for infinitely many n. This description has size $O(\log n)$ relative to x, contradicting the Kolmogorov complexity of strings in L. ∎

3.1 Other complexity classes

As mentioned in the introduction, a lot of differently defined complexity classes are contained in R_T^p(SPARSE), i.e. they can be expressed using P and a small supplementary amount of information. The notion of *non-uniform* algorithms, where an algorithm is allowed to use some supplementary advice dependent on the length of the input, leads to the definition of P/*poly*. In a similar way the class IC[log, poly] of sets of low instance complexity (see [KOSW]) and its subclass P/*log* are defined. They all fit into the context of reduction classes to sparse sets.

Theorem 3.8 *1. [BH77]* R_{tt}^p(SPARSE), P/*poly*, *and the class of sets accepted by polynomial size circuits are equal.*

2. [AHH⁺92] $R_c^p(\text{SPARSE}) \cap R_d^p(\text{co-SPARSE}) = \text{IC}[\log, \text{poly}]$

3. $\text{P}/log \subseteq R_{hd}^p(\text{TALLY})$

For approximation algorithms which err only on a small number of instances, the classes *APT* (almost polynomial time [MP79]) and P-*close* [Sch86] were considered. SPARSE_P denotes the class of all sparse sets which are subsets of sparse sets in P.

Theorem 3.9 *1.* $R_m^p(\text{APT}) = R_m^p(\text{SPARSE}_P)$

2. $R_m^p(\text{P-close}) = R_{1\text{-}tt}^p(\text{SPARSE})$

The classes BPP and R of randomized algorithms are also known to be contained in P/*poly* (see [Sch86]).

3.2 Sets of high information density

Since the various polynomial time reducibilities only allow very restricted access to the oracle set, one might ask if they can extract more information from a set which has extremely high information content encoded in an unredundant way than from a sparse set, which by definition has low information density. Book and Lutz [BL92] gave the surprising answer that this is not the case. At first we will give the definition of the sets with high information density (this definition extends the definition from [BL92]).

Definition 3.10 *[AKM93a] A set B is in* HIGH, *if for every constant* $c > 0$ *there exists a polynomial q such that for almost every n,* $\chi^B_{\leq n} \notin KS[2^{n+1} - q(n), 2^{cn}]$.

Remark that the closure properties of HIGH under Boolean operations are just opposite to the closure properties of SPARSE: HIGH is closed under complementation, and it is not closed under intersection and union. Since no set in HIGH is computable with polynomial advice in exponential space, $\text{ESPACE}/poly \cap \text{HIGH} = \emptyset$. Book and Lutz showed that furthermore a HIGH set cannot provide much help to a decision algorithm for a set in ESPACE.

Theorem 3.11 *[BL92]* $\text{ESPACE} \cap R_{btt}^p(\text{HIGH}) \subseteq R_{btt}^p(\text{SPARSE})$.

In [AKM93a] several reduction classes to sparse sets are characterized by sets having characteristic sequences of relatively low Kolmogorov complexity which reduce to HIGH.

Theorem 3.12 *[AKM93a] Let* $\leq_r^p$ *be one of the reducibilities* $\leq_m^p$, $\leq_c^p$, $\leq_{bhd}^p$, $\leq_{hd}^p$, *or the composition of* $\leq_{bhd}^p$ *resp.* $\leq_{hd}^p$ *and* $\leq_c^p$. *Then* $\text{ESPACE/poly} \cap R_r^p(\text{HIGH}) = R_r^p(\text{SPARSE})$.

4 Advice complexity

Every sparse set has the useful property that for every n there is a table of size polynomial in n in which all the elements of the set of size up to n are listed. This is the reason for sparse-reducible sets to be computable in polynomial time if the appropriate table is given, i.e. to belong to $\mathrm{P}/poly$. In a sense, the tables (resp. the polynomial advices) are encodings of the sparse-reducible set, and the reduction function (resp. the algorithm deciding the set if the advice is given) performs a decoding of this information. Thus, every sparse-reducible set has an *en*coding which is easy to *de*code. How hard is it to compute that encoding? Since to compute an encoding of a set A one has to know the set, this question asks for the complexity of the advice function for A if A is given as oracle. In [AHH+92, AKM92b, Gav92b] was considered several reduction classes to sparse sets. Observe that for every $A \in \mathrm{P}/poly$ the set of correct advices is in co-NP^A. From this, an easy upper bound on the complexity of advice functions follows. We say that *A has a polynomial advice function in $\mathcal{C}$* for a function class $\mathcal{C} \subseteq poly$, if $A \in \mathrm{P}//f$ for some $f \in \mathcal{C}$.

Proposition 4.1 *Every set $A \in \mathrm{P}/poly$ has a polynomial advice function in* $\mathrm{NPMV}^{\mathrm{NP}^A}$.

This led to the location of sparse-reducible sets into the Extended Low Hierarchy (see [BBS86, LS91]), where the additional power given from a set to a class of the Polynomial Time Hierarchy is measured. A lower bound can be derived from the existence of a sparse oracle A for which $\mathrm{P}^A \neq \mathrm{NP}^A$. Clearly, a sparse set A for which a table containing its elements could be computed in FP^A does not separate P and NP independently from the solution of the P = NP question.

Proposition 4.2 *There exists a sparse set A having no polynomial advice functions in* FP^A.

Considering the location of sparse-reducible sets into the Extended Low Hierarchy, it finally came out that all the reduction classes to sparse sets which at least contain all sparse sets belong to the same level of the Extended Low Hierarchy, which means that this hierarchy does not give a tool to distuingish between these classes.

Theorem 4.3 *Let $\mathcal{C}$ be a class with* $\mathrm{SPARSE} \subseteq \mathcal{C} \subseteq R_T^p(\mathrm{SPARSE})$.

1. *[Köb93] $\mathcal{C}$ is contained in the third Θ-level of the Extended Low Hierarchy, i.e. for every $A \in \mathcal{C}$:* $\mathrm{P}^{\mathrm{NP}^{\mathrm{NP}^A}}[\log] \subseteq \mathrm{P}^{\mathrm{NP}^{\mathrm{SAT}\oplus A}}[\log]$.
2. *[AH92] $\mathcal{C}$ is not contained in the second Σ-level of the Extended Low Hierarchy, i.e. there exists a $B \in \mathcal{C}$ such that* $\mathrm{NP}^{\mathrm{NP}^B} \supset \mathrm{NP}^{\mathrm{SAT}\oplus B}$.

Therefore we consider here the complexity of the advice functions for sparse-reducible sets. Kadin [Kad87] showed that for sparse NP sets the complexity of the advice function – in a general sense the complexity of the tables containing the sparse NP set – has a much lower upper bound. To compute that table he introduced a census-technique. At first, a pre-computation determines the number of elements contained in the table which then allows to compute easily the whole table.

Theorem 4.4 *Every sparse* NP *set has a polynomial advice function in* $\mathrm{NPSV}/\mathrm{FP}^{\mathrm{NP}}[\log]$.

Kadin's idea of the census-technique is used to prove many of the following results. We will shortly give some idea. Let S be a sparse set. Then on input 0^n the number of elements in S up to length n is computable via a binary search in the set $\{\langle 0^n, 0^m\rangle \mid S$ contains at least m elements up to length $n\}$ for the greatest c such that $\langle 0^n, 0^c\rangle$ is in that set. This search can be done in

$\mathrm{FP}^{\mathrm{NP}^S}$[log] and computes the exact census of S. Knowing this census, an non-deterministic computation on input 0^n can guess as many strings as given by the census, lexicographically ordered and of length at most n, If they all are in S, then the computation outputs these strings.

More upper bounds for various reduction types can be found in [AHH+92, AKM92b].

Theorem 4.5 1. *[AKM92b] Every set* $A \in R^p_c(\mathrm{SPARSE}) \cap R^p_d(\text{co-SPARSE})$ *has a polynomial advice function in* $\mathrm{FP}^{\mathrm{SAT}\oplus A}$.

2. *[AKM92b] For every set* $A \in R^p_{hd}(R^p_c(\mathrm{SPARSE}))$ *there exists a function* $f \in \mathrm{FP}^{\mathrm{NP}^A}$[log] *such that* A *has a polynomial advice function in* NPSV^A/f.

3. *[AKM92b] For every set* $A \in R^p_d(\mathrm{SPARSE})$ *there exists a function* f *in* $\mathrm{FP}^{\mathrm{NP}^A}$[log] *such that* A *has a polynomial advice function in* $\mathrm{NPSV}^{A\oplus\mathrm{SAT}}/f$.

4. *[AKM92b] Every set* $A \in R^p_d(\mathrm{SPARSE})$ *has a polynomial advice function in* $\mathrm{NPMV}^A/\mathrm{FP}^{\mathrm{NP}^A}$[log].

5. *[Köb93] Every set* A *in* P/*poly* *has a polynomial advice function in* $\mathrm{NPMV}^{\mathrm{SAT}\oplus A}/\mathrm{FP}^{\mathrm{NP}^{\mathrm{SAT}\oplus A}}$[log].

These results imply directly where the considered classes are located in the Extended Low Hierarchy. $R^p_c(\mathrm{SPARSE}) \cap R^p_d(\text{co-SPARSE})$ belongs to EL^{Δ}_2, and the other considered classes belong to EL^{Θ}_3 (see [AKM92b, Köb93]).

Theorem 4.6 *[GW91] There exists a set* $A \in R^p_c(\mathrm{SPARSE})$ *having no polynomial advice functions in* NPSV^A.

5 Self-reducibility

A set A is self-reducible if A is accepted by a polynomial time bounded machine M using oracle A. Thus, each query can be interpreted as a recursive call to M. The computation of M on an instance can be represented as a "tree" of recursive calls: the root is the instance itself, its sons are all queries computed by M, the sons of the sons are the queries computed by M on the sons, and so on. The queries underly some restrictions such that no path in the tree is infinite and there is a polynomial which bounds the length of every internal node in the length of the root. Therefore, the tree is finite and the leafs of are decidable in polynomial time without further queries to the oracle. Every instance of A is related to such a tree which has a certain depth. The "inductive" structure of self-reducible sets can now be used as follows. Consider a self-reducible set A in P/*poly*, and let M be the machine performing the self-reducibility. Advices for instances of depth 0 are $\mathrm{NPMV}^{\mathrm{SAT}}$ computable by guessing an advice a_0 and checking its correctness with a query to a co-NP oracle: does it hold that for all instances (up to a certain length) of depth 0 the decision of M is the same as that of the "advice interpreter" with a_0? If an advice a_i for all instances up to depth i is computed, then an advice a_{i+1} for all instances up to depth $i+1$ has the property that for all instances up to depth $i+1$ M where the oracle queries are decided using the advice interpreter with a_i decides in the same way as the advice interpreter with a_{i+1}. Therefore, we observe that every self-reducible set in P/*poly* has polynomial advices in $\mathrm{NPMV}^{\mathrm{SAT}}$. In contrast to the general advice complexity of sets $A \in$ P/*poly* we see that the oracle A is omitted. So we can expect that the advice complexity also decreases for self-reducible sets in several reduction classes to sparse sets. Lozano and Torán [LT91] broadly studied self-reducible sets which are sparse-reducible. They showed a (relativized) lower bound for the advice complexity of sparse self-reducible sets.

Theorem 5.1 *[LT91] There is an oracle B and a sparse word-decreasing self-reducible set S such that S has no polynomial advice functions in* $\mathrm{NPMV}/\mathrm{FP}^{\mathrm{SAT}}[\log]$, *relative to B.*

Thus, the best possible upper bound for the advice complexity of sparse- and self-reducible sets seems to be $\mathrm{FP}^{\mathrm{NP}}$. Indeed, Lozano and Torán [LT91] proved that upper bound for sets which many-one reduce to sparse sets and are length-decreasing or word-decreasing self-reducible. In the following we define a type of self-reducibility which captures both self-reducibility types above and also the "left-set" self-reducibility of [OW91], and allows to get the optimal upper bound for several classes of sparse-reducible sets.

Definition 5.2 *[AKM93b] Let $\prec$ be a partial order on Σ^*. We say that $\prec$ is polynomially well-founded, if there exists a polynomial p such that for all x, y, $x \prec y$ implies that $|x| \leq p(|y|)$. $\prec$ is called* $\mathrm{FP}^{\mathrm{NP}}$-minimizable, *if it is polynomially well-founded and there exists an $\mathrm{FP}^{\mathrm{NP}}$ transducer M using an oracle, such that for all B and i it holds that $M^B(0^i)$ outputs an x which is minimal in B for Σ^i w.r.t. $\prec$. A set A is (polynomial time)* $\mathrm{FP}^{\mathrm{NP}}$-minimizable self-reducible, *if there is some polynomial-time oracle machine M and an $\mathrm{FP}^{\mathrm{NP}}$-minimizable ordering $\prec$ such that $A = L(M, A)$ (i.e. M with oracle A accepts A) and, on every input x, M queries the oracle only about words $q \prec x$.*

Consider a sparse self-reducible set $S = L(M, S)$. We show how to construct an advice for S, i.e. the elements of S up to a certain length. In the beginning, let $T = \emptyset$ and let $all(T)$ be the predicate that M accepts exactly the elements of T. For a given sparse T this predicate is in co-NP. While this predicate is not true, find the instance having smallest depth which is accepted by M using T as oracle set and add it to T. This step is performed using the $\mathrm{FP}^{\mathrm{NP}}$-minimizing property of the self-reducibility order. Since using an inductive argument as above every element added to T is indeed in S, after a polynomial number of repetitions all elements of S up to a certain length are contained in T, and therefore the advice is computed. This proves that sparse self-reducible sets have polynomial advices in $\mathrm{FP}^{\mathrm{NP}}$. The following theorem shows that also larger reduction classes have the same advice complexity.

Theorem 5.3
1. *[LT91] All length-decreasing or word-decreasing self-reducible sets which $\leq_m^p$-reduce to sparse sets have polynomial advice functions in* $\mathrm{FP}^{\mathrm{NP}}$.
2. *[AKM93b] All* $\mathrm{FP}^{\mathrm{NP}}$*-minimizable self-reducible sets which reduce via $\leq_c^p$, $\leq_d^p$, $\leq_{hd}^p$, or the composition of $\leq_{hd}^p$ and $\leq_c^p$ to sparse sets have polynomial advice functions in* $\mathrm{FP}^{\mathrm{NP}}$.

Again, an immediate consequence is the lowness of the considered classes for P^{NP}. For the class $R_c^p(\mathrm{SPARSE}) \cap R_d^p(\text{co-SPARSE})$ finally the minimization property is not needed. The class max-P is the class of functions where the function value is the maximum fo an NPMV function. mim-P is defined respectively.

Theorem 5.4 *[AKM93b] Every polynomially well-founded self-reducible set $A \in R_c^p(\mathrm{SPARSE}) \cap R_d^p(\text{co-SPARSE})$ has polynomial advice functions in max-P$\cap$min-P.*

In [KOSW] it was shown that sets in the latter class which have computation "trees" of polynomial depth – i.e. polynomially-related self-reducible sets – are in P. Polynomially well-founded self-reducibility allows exponential depth of recursion, which allows only an NP$\cap$co-NP upper bound.

6 Collapses

Research on the existence of sparse-reducible NP-complete sets yielded a lot of important results for structural complexity. Self-reducibility properties of complete sets and properties of reduction types were found, and also algorithmic techniques to "speed-up" computations which underly specific structures were developed, which had influence also on other areas of complexity theory. At first, a short overview on the development of collaps results is given.

Theorem 6.1 *1. [For79] If* $\overline{\text{SAT}}$ $\leq_m^p$*-reduces to a sparse set, then* P = NP.

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

3. [KL80] If SAT $\leq_T^p$*-reduces to a sparse set, then* PH = Σ_2^p.

After the above results of Mahaney, and of Karp an Lipton, the question arised for which reductions having strength between many-one and Turing reducibility sparse NP-complete sets imply P = NP. Several results were found for positive reducibilities. In the proofs mostly Fortunes "tree-pruning" technique is used [For79].

Theorem 6.2 *1. [Yes83] If* $\overline{\text{SAT}}$ $\leq_{pos\text{-}btt}^p$*-reduces to a sparse set, then* P = NP.

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

3. [Ukk83, Yap83] If $\overline{\text{SAT}}$ $\leq_c^p$ *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.

A real break-through were the exploration of left-sets – a special type of self-reducibility of NP-complete sets – by Ogiwara and Watanabe [OW91].

Theorem 6.3 *1. [OW91] If* SAT $\leq_{btt}^p$*-reduces to a sparse set, then* P = NP.

2. [AHH+92] If SAT $\leq_c^p$ *reduces to a sparse set, then* P = NP.

Finally, the following result subsumes all of the above results.

Theorem 6.4 *[AKM92a] If* SAT $\in R_{bhd}^p(R_c^p(\text{SPARSE}))$*, then* P = NP.

We will sketch now the method as used in the proof of the above theorem which gives an upper bound on the complexity of sets having a special self-reducibility property which Hausdorff reduce to sets of bounded densities. Ogiwara and Watanabe [OW91] found the notion of *Left-sets.* Let $\langle\cdot,\cdot\rangle$ be a polynomial time computable and invertibel function $\Sigma^* \times \Sigma^* \to \Sigma^*$ such that $\langle x, y\rangle < \langle x, y'\rangle$ if $y < y'$ in lexicographic order. For a predicate R and a function r, let *Left*(R, r) be the set of all pairs $\langle x, y\rangle$ where $|y| = r(|x|)$, and y is lexicographically smaller than some y' of the same length for which $\{x, y'\} \in R$. We define $L(R, r)$ as the prefix set of the left-set, i.e.

$$L(R,r) \;=\; \{\langle x, u\rangle \mid \exists y \in \Sigma^{r(|x|)},\ v \in \Sigma^* : |uv| = |y| \wedge uv \leq y \wedge \langle x, y\rangle \in R\}.$$

$L(R, r)$ is self-reducible in a very strong sense: For $y \in \Sigma^{<r(|x|)}$ it holds that $\langle x, y\rangle \in L(R, r) \iff (\langle x, y0\rangle \in L(R, r) \vee \langle x, y1\rangle \in L(R, r))$, and for $y \in \Sigma^{r(|x|)}$ it holds that $\langle x, y\rangle \in L(R, r) \iff (\langle x, y\rangle \in R \vee \langle x, y'\rangle \in L(R, r))$, where y' is the successor of y in the lexicographical order. If $y = 1^{r(|x|)}$, then $\langle x, y\rangle \in L(R, r) \iff \langle x, y\rangle \in R$. Let $w_{max}(R, r, x) = \max\{w \in \Sigma^{r(|x|)} \mid \langle x, w\rangle \in R\}$. Then $\langle x, u\rangle \in L(R, r)$ if u is a prefix of $w_{max}(R, r, x)$, or if for the longest common prefix v of u and $w_{max}(P_A, q, x)$ it holds that $v0$ is a prefix of u and $v1$ is a prefix of $w_{max}(R, r, x)$.

Assume for some R, r, that $L(R,r)$ Hausdorff reduces to a sparse set. The reduction function f and the sparse set S can be choosen in a way such that f on all inputs $\langle x,y\rangle$ for $y \in \Sigma^{\leq r(|x|)}$ produces a list of $k(|x|)$ oracle queries, each of length $l(|x|)$, for polynomials k and l. Let $|S^{=l(|x|)}| \leq d(|x|)$ for a polynomial d.

We sketch an algorithm which on input x computes a set $W \subseteq \Sigma^{r(|x|)}$ which contains $w_{max}(R,r,x)$ – for simplicity called w_{max} from now on – if it exists. It can be seen as a pruning algorithms on the tree of all strings in $\Sigma^{\leq r(|x|)}$. Let's consider some fixed x. Let $f(i,y)$ denote the i-th query of the list of queries $f(\langle x,y\rangle)$ produced by the reduction function. And also, for sake of easiness, we will write y_l instead of $\langle x,y_l\rangle$ (it is always clear from the context what is meant). Let $m = d(|x|)^{k(|x|)}+1$. For some $i \leq r(|x|)$, let $W = \{y_1,\ldots,y_t\}$ be a lexicographically ordered subset of Σ^i, such that W contains a prefix y_h of w_{max} and $t > m$.

We now give a description of a pruning step, which prunes W to a set of size $\leq m$ which contains y_h. It uses a property of the Hausdorff reducibility. If $z \in L(R,r)$, then $f(1,z) \in S$, since because of the monotonicity of the Hausdorff reduction function otherwise $\{f(1,z),\ldots,f(k(|x|),z)\} \subseteq \overline{S}$ which would contradict the correctness of the reduction function. In general, for even j it holds that $z \in L(R,r) \Rightarrow (f(j,z) \in S \Rightarrow f(j+1,z) \in S)$, and for odd j it follows that $z \in L(R,r) \Rightarrow (f(j,z) \in S \Rightarrow f(j+1,z) \in S)$. But now, back to the algorithm. Observe that $\{y_1,\ldots,y_h\} \subseteq L(R,r)$ and $\{y_{h+1},\ldots,y_t\} \cap L(R,r) = \emptyset$. By the properties of the Hausdorff reducibility it follows that $\{f(1,y_1),\ldots,f(1,y_h)\} \subseteq S$. Let i_1 be the largest index i for which $\{f(1,y_1),\ldots,f(1,y_i)\} \subseteq S$. Clearly, $i_1 \geq h$. Since

$$\{f(1,y_{h+1}),\ldots,f(1,y_{i_1})\} \subseteq S$$

and since

$$\{y_{h+1},\ldots,y_{i_1}\} \cap L(R,r) = \emptyset$$

it follows, again by the properties of the Hausdorff reducibility, that

$$\{f(2,y_{h+1}),\ldots,f(2,y_{i_1})\} \subseteq S.$$

Therefore, let i_2 be the smallest index for which

$$\{f(2,y_{i_2+1}),\ldots,f(2,y_{i_1})\} \subseteq S.$$

Clearly, $i_2 < h$. Inductively, for odd j let i_j the greates index such that

$$\{f(j,y_{i_{j-1}}),\ldots,f(j,y_{i_j})\} \subseteq S.$$

Then $i_j \geq h$ or $h \in \{i_1,i_2,\ldots,y_{j-1}\}$. For even j let i_j be the smallest index such that

$$\{f(j,y_{i_{j+1}}),\ldots,f(j,y_{i_{j-1}})\} \subseteq S.$$

Also, $i_j < h$ or $h \in \{i_1,i_2,\ldots,y_{j-1}\}$. Finally we get that $h \in \{y_1,y_2,\ldots,y_{k(|x|)}\}$.

Since S cannot be assumed to be known, the correct indices i_j are in general not computable. But, since S has bounded density, we know that i_1 is contained in the set I_1 of all indices i for which

$$f(1,y_i) \notin \bigcup_{1 \leq l < i} \{f(1,y_l)\},$$

and furthermore we know that I_1 needs to contain only the m smallest indices having the above property. Inductively defined, for odd j, I_j contains for every $i' \in I_{j-1}$ the m smallest indices with

$$f(j,y_i) \notin \bigcup_{i' \leq l < i} \{f(j,y_l)\}$$

and for even j it contains the m greatest indices with

$$f(j, y_i) \notin \bigcup_{i<l\leq i'} \{f(j, y_l)\}.$$

It then follows that $h \in \bigcup_{1\leq j\leq k(|x|)} I_j$. The size of I_j is $(d(|x|)+1)^{k(|x|)}$. Keeping only those y_i in W for which $i \in \bigcup_{1\leq j\leq k(|x|)} I_j$ guarantees that $|W| \leq (d(|x|)+1)^{k(|x|)}$ and that $y_h \in W$. This completes the description of the pruning step.

The overall algorithm on input x starts with $W = \{\varepsilon\}$, and repeats $r(|x|)$ an *expansion* of W, i.e. W is replaced by the set $W' = \{w0, w1 \mid w \in W\}$. Clearly, if W contains a prefix of w_{max} then W' also contains such a prefix. Observe that $|W'| \leq 2m$. If $|W| > m$ now, then W is pruned by the above procedure to a set of size at most m.

We now consider the complexity of the algorithm. Since the reduction function f is computable in polynomial time, the time to compute a set I_j is bounded by $|I_{j-1}| \cdot 2m \cdot d(|x|) \cdot p(|x|)$, for a polynomial p. Since the size of the union of all I_j is bounded by $m \cdot d(|x|)$, we get that every pruning step needs time at most $p(|x|)d(|x|)^{k(|x|)+c}$, for a constant c and a polynomial p. Since the expansion and pruning is repeated $r(|x|)$ times we finally get that the overall algorithm is performed in time $r(|x|)d(|x|)^{k(|x|)+c}$ for a constant c and a polynomial p. We here remark only that replacing the Hausdorff reduction function by a composition of a Hausdorff an a conjunctive reduction function increases the computation time by a polynomial factor. This completes the sketch of the method.

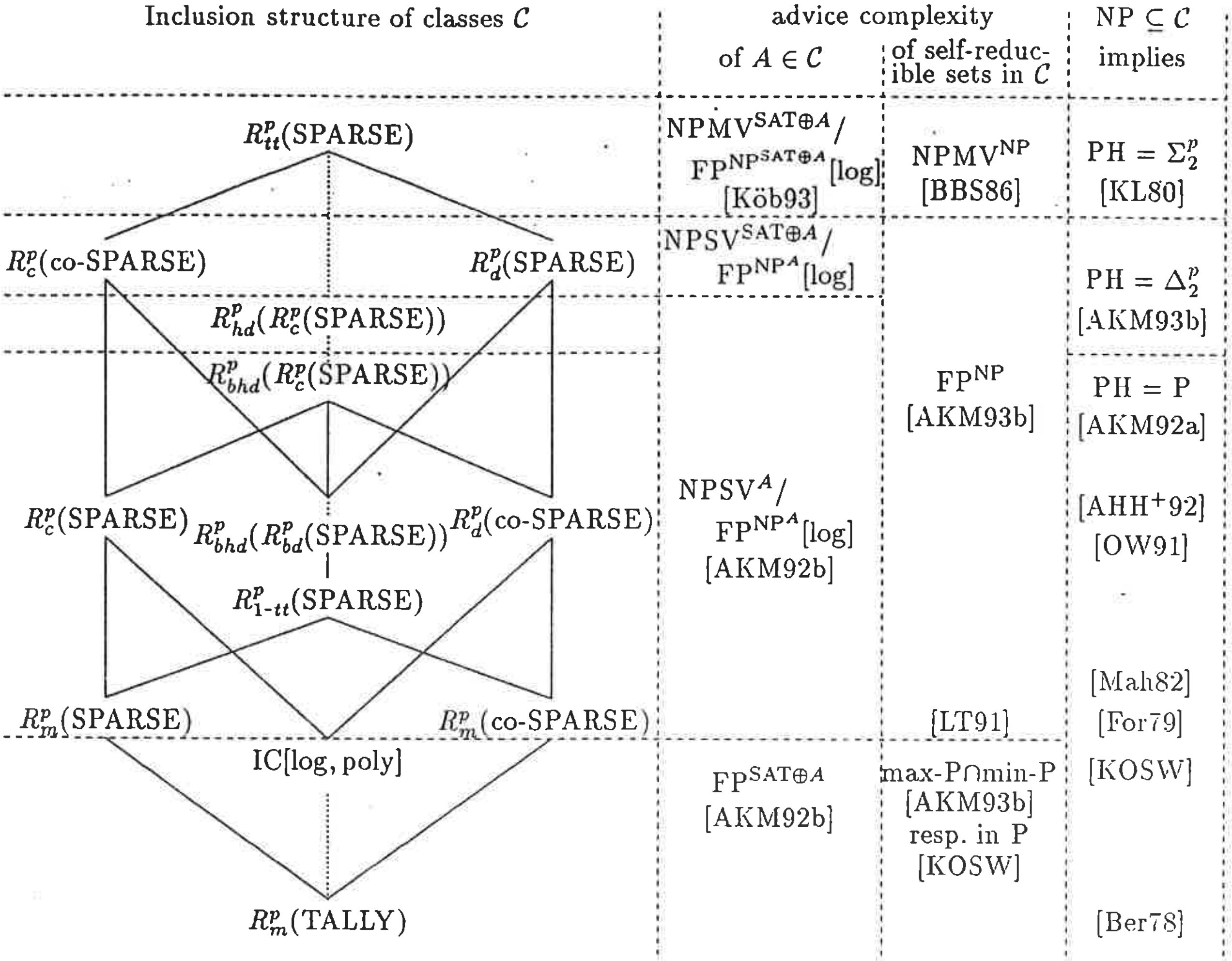
Every set $A \in$ NP can be described using a P predicate R_A and a polynomial r as $A = \{x \mid \exists y \in \Sigma^{r(|x|)} : \langle x, y\rangle \in R_A\}$. Clearly, $A \leq_m^p L(R_A, r)$ and $L(R_A, r) \in$ NP. If A is NP-complete then $L(R_A, r) \leq_m^p A$ follows. Therefore, the assumption SAT $\in R_{bhd}^p(R_c^p(\text{SPARSE}))$ implies that also $L(R_{\text{SAT}}, r_{\text{SAT}}) \in R_{bhd}^p(R_c^p(\text{SPARSE}))$. Since for a bounded Hausdorff reduction the function k from the above sketch is a constant, and the functions r and d from above are polynomials in this case, it follows that $L(R_{\text{SAT}}, r_{\text{SAT}}) \in$ P and therefore SAT $\in$ P. So Theorem 6.4 follows.

Considering larger classes containing more than $R_{bhd}^p(R_c^p(\text{SPARSE}))$ only weaker collapses of the Polynomial Time Hierarchy can be concluded by the self-reducibility structure of the NP complete set SAT.

Theorem 6.5 *[AKM93a, AKM93b] If* NP $\subseteq R_{hd}^p(R_c^p(\text{SPARSE}))$ *or* NP $\subseteq R_d^p(\text{SPARSE})$, *then* PH $=$ P^{NP}.

7 Summary

The following pictures summarizes the results about the inclusions of classes of sets reducible to sparse sets, their simplicity properties, the simplicity of self-reducible sets in these classes, and the consequences of the assumption that NP is contained in one of these classes.



References

[AH92] E. Allender and L. Hemachandra. Lower bounds for the low hierarchy. *Journal of the ACM*, 39(1):234-250, 1992.

[AHOW91] E. Allender, L. Hemachandra, M. Ogiwara and O. Watanabe. Relating equivalence and reducibility to sparse sets. *Proceedings of the 6th Structure in Complexity Theory Conference*,220–229. IEEE Computer Society Press, 1991.

[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. *Proceedings of the 19th International Colloquium on Automata, Languages, and Programming*, Lecture Notes in Computer Science, #623:162-173, Springer Verlag, 1992.

[AKM92a] 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.

[AKM92b] V. Arvind, J. Köbler, and M. Mundhenk. Lowness and the complexity of sparse and tally descriptions. *Proceedings Third International Symposium on Algorithm and Computation*, Lecture Notes in Computer Science, #650:249–258, Springer Verlag, 1992.

[AKM93a] V. Arvind, J. Köbler, and M. Mundhenk. Hausdorff reductions to sparse sets and to sets of high information content. *Proc. 18th MFCS*, Lecture Notes in Computer Science, #711:232–241, Springer Verlag, 1993.

[AKM93b] V. Arvind, J. Köbler, and M. Mundhenk. Upper bounds on the complexity of sparse and tally descriptions. Submitted. 1993.

[Ba92] J. Balcazár. Self-reducibility. *Journal of Computer and System Sciences*, 41:367-388, 1990.

[BBS86] J.L. Balcázar, R. Book, and U. Schöning. Sparse sets, lowness and highness. *SIAM Journal on Computing*, 23:679-688, 1986.

[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*, S. 63–71. *Lecture Notes in Computer Science #62*, 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.

[BL92] R. Book and J. Lutz. On languages with very high information content. *Proc. 7th Structure in Complexity Theory Conference*, 255-259, IEEE Computer Society Press, 1992.

[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.

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

[Gav92a] R. Gavaldà. On conjunctive and disjunctive reductions to sparse sets. Manuscript. 1992.

[Gav92b] R. Gavaldà. Bounding the complexity of advice functions. *Proc. 7th Structure in Complexity Theory Conference*, 249–254. IEEE Computer Society Press, 1992.

[GW91] R. Gavaldà and O. Watanabe. On the computational complexity of small descriptions. *Proc. 6th Structure in Complexity Theory Conference*, 89–101. IEEE Computer Society Press, 1991.

[Kad87] J. Kadin. $P^{NP[\log n]}$ 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.

[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.

[KOSW] K. Ko, P. Orponen, U. Schöning, and O. Watanabe. Instance complexity. *Journal of the ACM*, to appear.

[Köb93] J. Köbler. Locating P/poly optimally in the extended low hierarchy. *Proc. 10th Annual Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science, #665:28–37, Springer Verlag, 1993.

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

[LS91] T.J. Long and M.-J. Sheu. A refinement of the low and high hierarchies. Technical Report OSU-CISRC-2/91-TR6, The Ohio State University, 1991.

[LT91] A. Lozano and J. Torán. Self-reducible sets of small density. *Mathematical Systems Theory*, 24:83-100, 1991.

[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.

[MP79] A. Meyer, M. Paterson. With what frequency are apparently intractable problems difficult? Tech. Report MIT/LCS/TM-126, Lab. for Computer Science, MIT, Cambridge, 1979.

[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.

[Sch83] U. Schöning. A low and a high hierarchy within NP. *Journal of Computer and System Sciences*, 27:14-28, 1983.

[Sch86] U. Schöning. *Complexity and Structure*, Lecture Notes in Computer Science, #211, Springer Verlag, 1985

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

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

[Wag90] K.W. Wagner. Bounded query classes. *SIAM Journal on Computing*, 19(5):833–846, 1990.

[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.

[You92] P. Young. How reductions to sparse sets collapse the polynomial-time hierarchy. *Technical Report 92-03-07*, University of Washington, March 1992.