On self-reducible sets of low information content

Martin Mundhenk*
Universität Trier, Fachbereich IV - Informatik
D-54286 Trier, Germany

Abstract

Self-reducible sets have a rich internal structure. The information contained in these sets is encoded in some redundant way. Therefore a lot of the information of the set is easily accessible. In this paper it is investigated how this self-reducibility structure of a set can be used to access easily *all* information contained in the set, if its information content is small. It is shown that P can be characterized as class of self-reducible sets which are "almost" in P (i.e. sets in APT'). Self-reducible sets with low instance complexity (i.e. sets in IC[log, poly]) are shown to be in NP ∩ co-NP, and sets which disjunctively reduce to sparse sets or which belong to a certain superclass of the Boolean closure of sets which conjunctively reduce to sparse sets are shown to be in P^{NP}, if they are self-reducible in a little more restricted sense of self-reducibility.

1 Introduction

The internal properties of a set have effects on its computational complexity as on the usability of the information encoded into the set. For example, a "natural" encoding of the satisfiable Boolean formulas over an alphabet Σ of at least two letters – i.e. the set SAT – is NP-complete[1]. But also one can consider a tally encoding of SAT over a one-letter-alphabet (say *tally*(SAT)), or an encoding which does not contain any redundant information (*compress*(SAT)). Using the properties of SAT that it has exponential density, that SAT is decidable in exponential time, and that SAT contains a lot of redundant information, it follows under the assumption P $\neq$ NP, that whether *tally*(SAT) (cf. [Ber78]) nor *compress*(SAT) (cf. [BL92]) are hard for NP, which means that the information contained in these "extreme" encodings is not usable by an efficient algorithm to decide SAT. These arguments give lower and upper bounds of the information content of SAT.

The redundancy of SAT can be expressed in terms of *self-reducibility*:[2] a Boolean formula F having variables $x_1, \ldots, x_n$ is satisfiable iff $F|_{[x_1/true]}$ is satisfiable or $F|_{[x_1/false]}$ is satisfiable, where $F|_{[x_1/a]}$ describes the formula obtained from F where the variable x_1 is replaced by the value $a \in \{true, false\}$, and which therefore has variables $x_2, \ldots, x_n$. Since the satisfiability of every formula having no variables can be decided in polynomial time, one can imagine a decision procedure for $F \in$ SAT? which procedes deciding $F|_{[x_1/true]} \in$ SAT? and $F|_{[x_1/false]} \in$ SAT? using "recursive function calls", where the computation of the arguments for each call and the evaluation of the results is polynomially time bounded, and the depth of the recursion is bounded by the number of variables of F. If a set of low information content has such redundancy properties, one can derive certain upper bounds for the complexity of the set. Mahaney [Mah82] used this to show that SAT $\in$ P if SAT is many-one reducible to a sparse set, i.e. a set which contains at most polynomially many of the exponentially many possible strings of each length. Later, Ogiwara and Watanabe [OW91] found a more involved self-reducibility structure of NP-complete sets which led to the result, that P = NP if some NP-complete set many-one reduces to a set in the Boolean closure of the sets which conjunctively reduce to sparse sets [AKM92].

The general form of these results can be seen as: if a set is self-reducible and reduces in a certain way to a sparse set, then its complexity is low. Lozano and Torán [LT91] studied the lowness of self-reducible sets in the class of sets which many-one reduce to sparse sets. In this paper upper bounds of the complexity of self-reducible sets in several classes defined by various reducibilities to sparse sets are considered. At first

*This work was presented at CIAC 1994, Rome. Work done at Universität Ulm, Abt. Theoretische Informatik. Supported in part by the DAAD through Acciones Integradas 1992, 313-AI-e-es/zk. Author's e-mail address: mundhenk@uni-trier.de
[1] for example take the binary representation of Boolean formulas in a LaTeX text as encoding
[2] see Section 2 for a formal definition

a natural general definition of self-reducibility is given, where the only restriction on the self-reducibility is that the arguments of the "recursive calls" do not leave a polynomially bounded range. We will show that a restriction of this type of self-reducibility covers all of the self-reducibility types used up to now. We give characterizations of the classes which contain polynomial time "approximizable" sets as APT (the class of sets which are "almost" in P [MP79]) and P-*close* [Sch86] in terms of reduction classes to sparse sets. It will be shown that all self-reducible sets in APT are in P, and that self-reducible sets of small instance complexity (i.e. sets in IC[log, poly]) are in NP $\cap$ *co*-NP. For the restricted type of self-reducibility it is shown that self-reducible sets which disjunctively reduce to sparse sets or which are in the Boolean closure of sets which conjunctively reduce to sparse sets are low in Δ_2^p. An overview of the results is given in a figure in Section 5. It shows the relationships between the considered classes, their containments in the Extended Low Hierarchy (defined by [BBS86]), and the lowness results in case of self-reducibility as shown in this paper.

2 Preliminaries and notation

In general the notations and definitions from the standard books on structural complexity theory (for example, [BDG, Sch86]) are used.

Let A be a set. $A^{=n}$ ($A^{\leq n}$) denotes the set of all strings in A of length n (up to length n, respectively). The cardinality of A is denoted by $|A|$. A set S is called sparse if for some polynomial p and every n $|A^{\leq n}| \leq p(n)$. We use SPARSE and co-SPARSE to represent the classes of sparse sets and sets having a sparse complement, respectively. A set S is called $\mathcal{C}$-printable if the function which maps 0^n to an encoding of $S^{\leq n}$ can be computed in $\mathcal{C}$. A set A is said to be *low in* $\mathcal{C}$ for a complexity class $\mathcal{C}$, if $\mathcal{C}^A \subseteq \mathcal{C}$ (see [BBS86]). $\langle \cdot, \cdot \rangle$ denotes some easy computable and invertible pairing function. $L(M)$ is the set accepted by the Turing machine M, $L(M, A)$ is the set accepted accepted by the oracle Turing machine M using the set A as oracle.

The reducibilities discussed in this paper are standard polynomial time bounded reducibilities defined (see [BDG]), and the Hausdorff reducibility $\leq_{hd}^p$ by Wagner [Wag87]. At first the definitions of some of the "non-standard" reducibilities are repeated. They are generalizations of the many-one reducibility $\leq_m^p$ and restrictions of the truth-table reducibility $\leq_{tt}^p$.

Definition 2.1 *A set A is* conjunctively reducible *to a set B ($A \leq_c^p B$), if there exists a polynomial-time computable function f mapping strings to sets of strings such that for all $x \in \Sigma^*$ it holds that $x \in A \iff f(x) \subseteq B$. A set A is* disjunctively reducible *to a set B ($A \leq_d^p B$), if there exists a polynomial-time computable function f mapping strings to sets of strings such that for all $x \in \Sigma^*$ it holds that $x \in A \iff f(x) \cap B \neq \emptyset$.*

Note that $A \leq_c^p B$ if and only if $\overline{A} \leq_d^p \overline{B}$. For any reducibility $\leq_r^\alpha$ and any class $\mathcal{C}$ of sets let $R_r^\alpha(\mathcal{C}) = \{A \mid A \leq_r^\alpha B \text{ for some } B \in \mathcal{C}\}$.

The classes APT (almost polynomial time [MP79]) and P-*close* [Sch86] contain sets which only on few instances differ from sets in P and can intuitively be seen as "approximizable" in P. We will show how reduction classes to APT and P-*close* can be characterized by reduction classes to sparse sets. The following defined class APT' is easily seen to be a generalization of APT. ($A \bigtriangleup B$ denotes the symmetric difference between sets A and B.)

Definition 2.2 *A set A is in* P-close, *if $A = L(M) \bigtriangleup S$ for a deterministic polynomially time bounded Turing machine M and a sparse set S. If furthermore $S \subseteq S'$ for a sparse $S' \in P$, then A is in* APT$'$.

The notion of instance complexity and the class IC[log,poly] of sets of strings with low instance complexity were introduced by Ko, Orponen, Schöning, and Watanabe (see [OKSW]). A set A is in IC[log,poly] if there exist a constant $c > 0$, a polynomial t and a set $\Pi \subseteq \Sigma^*$ of programs such that for every $x \in \Sigma^*$ the following conditions are fulfilled. (1) There exists a $p \in \Pi^{\leq c \log(|x|)}$ that halts on input x in time $t(|x|)$, and (2) every $p \in \Pi$ that halts on input x decides the membership of x in A. IC[log, poly] can also be seen as a reduction class to sparse sets.

Theorem 2.3 [AHH$^+$92, BLS93] IC[log, poly] $= R_c^p(\text{SPARSE}) \cap R_d^p(\text{co-SPARSE})$.

An overview on the relationships between the classes defined by various reducibilities to sparse sets is given in Section 5.

A set is self-reducible, if its characteristic function is computable using "recursive function calls". If the arguments for the recursive calls are computable and the depth of the recursion is finite, clearly the self-reducible set is decidable. This leads to a very simple definition of polynomially bounded self-reducibility.

Definition 2.4 *Let $\prec$ be a partial order*[3] *on Σ^*. $\prec$ is called* polynomially well-founded, *if there exists a polynomial p such that for all $x, y \in \Sigma^*$, $x \prec y$ implies that $|x| \leq p(|y|)$. A set A is (polynomial time)* self-reducible under $\prec$, *if there is some polynomial-time oracle machine M, 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 $q \prec x$. A is* poynomially well-founded self-reducible, *if A is self-reducible under a polynomially well-founded order $\prec$.*

For a polynomially well-founded $\prec$ every "increasing chain" $x_0 \prec x_1 \prec \cdots \prec x_k$ consists of at most $2^{p(|x_k|)+1}$ elements from $\Sigma^{\leq p(|x_k|)}$ for a polynomial p. For the most "popular" types of orders for self-reducibilities – the length-decreasing and the lexicographic order (see [Ba92]) as well as the polynomially related order – this upper bound underlies a stronger restriction.

Definition 2.5 [Ko83] *A partial order $\prec$ on Σ^* is said to be* polynomially related, *if there is a polynomial p such that (1) $x \prec y$ is decidable in time polynomial in $|x| + |y|$, and (2) for all $x_1, x_2, \ldots, x_k \in \Sigma^*$: if $x_1 \prec x_2 \prec \cdots \prec x_k$, then $k \leq p(|x_k|)$.*

Clearly, every length-decreasing order is polynomially related, whereas this is not the case for the lexicographic order.

Definition 2.6 [Ko83, Ba92] *A set A is length-decreasing (resp. word-decreasing, polynomially related) self-reducible, if A is self-reducible under a length-decreasing (resp. lexicographic, polynomially related) ordering.*

One method to decide a self-reducible set $A = L(M, A)$ is to handle all oracle queries of the self-reduction machine as "recursive function calls". This leads to the following upper bounds for the complexity of self-reducible sets.

Proposition 2.7 *1. Every polynomially well-founded self-reducible set is in EXPSPACE.*

2. **[BBS86]** *Every polynomially related or length-decreasing self-reducible set is in PSPACE.*

3. **[Ba92]** *Every word-decreasing self-reducible set is in ETIME.*[4]

One also can reverse the "top down" decision method for a self-reducible set A into a "bottom up" method, if one knows the ordering $\prec$ which restricts the queries of the self-reduction machine M. To decide an instance x, at first compute all $z \prec x$ which can be decided by M without oracle queries; let's call this set Z_0. Z_0 contains all strings having the "minimality" property, that there exists no y in Σ^* which is smaller than z under $\prec$. Since M correctly decides every instance from Z_0 without any query to the oracle, the set $A_0 = Z_0 \cap A$ can be constructed. Now – inductively – the set Z_{i+1} of all $z' \prec x$ which can be decided by M with oracle queries to $\bigcup_{j \leq i} Z_j$ can be computed, and the set $A_{i+1} = Z_{i+1} \cap A$ can be constructed, since M with oracle $\bigcup_{j \leq i} A_j$ decides all instances from Z_{i+1} correctly. This process has to be repeated until all oracle queries of M on input x are covered by $\bigcup_{j \leq k} Z_j$ and x therefore can be decided correctly by M with oracle $\bigcup_{j \leq k} A_j$. Observe that Z_{i+1} contains all strings $z' \prec x$ in $\Sigma^* - Z_i$ having the minimality property that there exists no $y \in \Sigma^* - Z_i$ which is smaller than z' under $\prec$. We will give a formal description for that minimality property.

Let B and C be sets and $\prec$ be an ordering. We say that a string x *is minimal in B for C w.r.t. $\prec$*, if (1) $x \in B$, and (2) $\forall z: \ z \prec x \Rightarrow z \notin B$, and (3) $\exists y \in C: \ x \prec y$.

Lozano and Torán [LT91] showed how the above "bottom up" decision process for a polynomially related or word-decreasing self-reducible set can be shortened if the set itself is sparse or many-one reducible to a sparse set. In their proofs the complexity of the computation of the minimal strings played an important role. Therefore we now define a property of orderings dependent on that complexity.

[3] i.e. an irreflexive, transitive, and antisymmetrical relation

[4] ETIME $= \bigcup_{i \geq 0} \mathrm{DTIME}(2^{cn})$

Definition 2.8 *We say that a polynomially well-founded ordering is* $\mathrm{FP}^{\mathrm{NP}}$-minimizable, *if 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* $\mathrm{FP}^{\mathrm{NP}}$-minimizable self-reducible, *if* A *is polynomial time self-reducible under an* $\mathrm{FP}^{\mathrm{NP}}$*-minimizable ordering* $\prec$.

Using a prefix search technique and Kadin's census technique [Kad87] we can prove

Proposition 2.9 *The lexicographic order and every polynomially related order are* $\mathrm{FP}^{\mathrm{NP}}$*-minimizable.*

So all of the standard types of self-reducibility can be covered by the new defined type.

Proposition 2.10 *If a set* A *is length-decreasing, word-decreasing, or polynomially related self-reducible, then* A *is* $\mathrm{FP}^{\mathrm{NP}}$*-minimizable self-reducible.*

Observe that the so called left-sets introduced in [OW91] are $\mathrm{FP}^{\mathrm{NP}}$-minimizable self-reducible and not polynomially related or word-decreasing self-reducible.

3 Reduction classes to sparse sets

We will show now how APT' and P-*close* fit into the context of reduction classes to sparse sets. Let SPARSE_P be the class of all subsets of sparse sets in P.

Theorem 3.1 $R^p_m(\mathrm{APT}') = R^p_m(\mathrm{SPARSE}_P)$.

Proof At first it is shown that $APT' \subseteq R^p_m(\mathrm{SPARSE}_P)$. Let $A \in APT'$ via M, S and S' where $S \subseteq S'$, $S' \in \mathrm{P}$, and $A = L(M) \triangle S$. Let $\hat{M}$ be a polynomially time bounded Turing machine with $L(\hat{M}) = L(M) \cup S'$, and let $\hat{S} = (S' - (L(M) \cup S)) \cup (L(M) \cap S)$. Clearly, $\hat{S} \subseteq S'$, and $A = L(\hat{M}) - \hat{S}$. The following function f is easily seen to reduce $A \le^p_m \hat{S}$:

$$f(x) = \begin{cases} y_{out}, & \text{if } x \notin S' \text{ and } M(x) \text{ rejects} \\ y_{in}, & \text{if } x \notin S' \text{ and } M(x) \text{ accepts} \\ x, & \text{if } x \in S' \end{cases}$$

for fixed $y_{out} \notin \hat{S}$ and $y_{in} \in \hat{S}$. Therefore we have $APT' \subseteq R^p_m(\mathrm{SPARSE}_P)$. By the transitivity of the $\le^p_m$-reducibility we get $R^p_m(APT') \subseteq R^p_m(\mathrm{SPARSE}_P)$. From $\mathrm{SPARSE}_P \subseteq APT'$, the inclusion $R^p_m(APT') \supseteq R^p_m(\mathrm{SPARSE}_P)$ follows and completes the proof. ∎

Using Theorem 3.1 and Theorem 2.3 we immediately get

Theorem 3.2 $R^p_m(\mathrm{APT}') \subseteq \mathrm{IC}[\log, \mathrm{poly}]$.

The next Theorem solves an open problem from [Ko83]. $A \le^p_{1\text{-}tt} B$ holds, if there exist a polynomial time bounded Turing machine N computing a truth-table for one variable, and a polynomial time computable function g, such that for all x, $x \in A$ if and only if the truth-table $N(x)$ evaluated with the value of "$g(x) \in B$" equals 1.

Theorem 3.3 $R^p_m(\text{P-close}) = R^p_{1\text{-}tt}(\mathrm{SPARSE})$.

Proof It is straightforward to see that P-*close* $\subseteq R^p_{1\text{-}tt}(\mathrm{SPARSE})$ and therefore $R^p_m(\text{P-}close) \subseteq R^p_{1\text{-}tt}(\mathrm{SPARSE})$. For the other inclusion direction let $A \le^p_{1\text{-}tt} S$ via N for a sparse set S. Then for all x, N on input x computes a 1-truth-table α_x and a query $g(x)$ to the sparse set S such that $x \in A \iff \alpha_x(g(x) \in S) = 1$ for all x. W.l.o.g. we can assume that α_x is one of the both 1-truth-tables $t_{id}(x) = [x = 1]$ and $t_{neg}(x) = [x = 0]$. Define $S' = \{0z \mid z \notin S\} \cup \{1z \mid z \in S\}$. Then the following function f reduces $A \le^p_m S'$:

$$f(x) = \begin{cases} 1g(x), & \text{falls } \alpha_x = t_{id} \\ 0g(x), & \text{falls } \alpha_x = t_{neg} \end{cases}$$

Let $S'' = \{0z \mid z \in S\} \cup \{1z \mid z \in S\}$. S'' is sparse. Since $S' = 0\Sigma^* \triangle S''$, we get that $S' \in$ P-*close*. ∎

Corollary 3.4 $R^p_{btt}(\text{P-close}) = R^p_{btt}(\mathrm{SPARSE})$.

4 Complexity of self-reducible sets

For self-reducible sets which reduce to sparse sets the first upper bound for their complexity follows from Karp and Lipton [KL80]. Lozano and Torán [LT91] considered the effect of different self-reducibilities for sets in more restricted reduction classes to sparse sets.

Theorem 4.1 *1.* **[BBS86, Ba92]** *Every length-decreasing or word-decreasing self-reducible set in* $R^p_{tt}(\mathrm{SPARSE})$ *is low in* Σ^p_2*.*

2. **[AKM93]** *Every length-decreasing self-reducible set in* $R^p_{hd}(R^p_c(\mathrm{SPARSE}))$ *is low in* Δ^p_2*.*

3. **[LT91]** *Every word-decreasing or polynomially related self-reducible set in* $R^p_m(\mathrm{SPARSE})$ *is low in* Δ^p_2*.*

4. **[OKSW]** *Every length-decreasing self-reducible set in* IC[log, poly] *is in* P.

We now state the new results about self-reducible sets.

Proposition 4.2 *Every polynomially well-founded self-reducible set in* APT' *is in* P.

Proof Sketch Let $A \in APT'$ be witnessed by M, S and $S' \in \mathrm{P}$, i.e. $A = L(M) \triangle S$ and $S \subseteq S'$, and let $A = L(N, A)$ for a self-reduction machine N which respects the polynomially well-founded order $\prec$. Let N' simulate N in a "top down" manner, i.e. N' simulates N and handles all oracle queries in S as "recursive function calls". Furthermore N stores all arguments of the recursive calls in a set R and all arguments which finally are accepted in R^+. Then N needs to compute the status of every element of S at most once, since every time a recursive call to an element of R is made it can be decided using R^+. Since $\prec$ is transitive and irreflexive it follows that the number of recursive calls of N' is bounded by a polynomial, and therefore N' is a polynomially time bounded machine deciding A. ■

Since every set in P is self-reducible and $\mathrm{P} \subseteq APT'$ we get

Proposition 4.3 P *equals the class of all polynomially well-founded self-reducible sets in* APT'*.*

We now show that all polynomially well-founded self-reducible sets in IC[log,poly] are in NP ∩ co-NP. Compared with the result from [OKSW] as cited in Theorem 4.1 the complexity of sets in IC[log, poly] which are decidable by a procedure with exponential recursion-depth is not much greater than that of sets in IC[log, poly] decidable with polynomially deep recursion.

Theorem 4.4 *Every polynomially well-founded self-reducible set in* $R^p_c(\mathrm{SPARSE}) \cap R^p_d(\text{co-SPARSE})$ *is in* NP ∩ co-NP.

Proof Let $A \in R^p_c(\mathrm{SPARSE}) \cap R^p_d(\text{co-SPARSE})$ be a polynomially well-founded self-reducible set. Since SPARSE is closed under marked union, it follows that $R^p_c(\mathrm{SPARSE}) \cap R^p_d(\text{co-SPARSE}) = \bigcup_{S \in \mathrm{SPARSE}} R^p_c(S) \cap R^p_d(\overline{S})$. Therefore, there exist a sparse set S and FP functions f and g such that $A \leq^p_c S$ via f, and $\overline{A} \leq^p_c S$ via g. Remark that for all x, $f(x) \subseteq S \iff g(x) \not\subseteq S$. Since A is polynomially well-founded self-reducible there exists a polynomially time bounded oracle Turing machine M which asks queries to the oracle with respect to a polynomially well-founded order $\prec$, such that $A = L(M, A)$. Let p be a polynomial which bounds $\prec$ and the running time of f, g, and M. Consider the following Turing machine M' which on an input consisting of a string x and a (finite) set T accepts, rejects, or halts without decision.

```
input ⟨x, T⟩
simulate M on input x, but
  whenever M queries q to the oracle then
    if f(q) ⊆ T ⇎ g(q) ⊈ T
      then halt without decision
      else if f(q) ⊆ T
        then continue with answer "yes"
        else continue with answer "no" (* g(q) ⊆ T *)
      end
    end (* of the simulation in case of a query of M to the oracle *)
if M accepts then accept else reject end
```

The machine runs in polynomial time, and it is "robust" in the sense that for every $T \subseteq S$ the decision of M' on input $\langle x, T\rangle$ answers $x \in A$, if it decides.

Claim 1 *Let $T \subseteq S$. Then for all x, if $M'(\langle x, T\rangle)$ decides its input, then $M'(\langle x, T\rangle)$ accepts iff $x \in A$.*

Proof Let $T \subseteq S$. If $f(q) \subseteq T$, then $f(q) \subseteq S$ and therefore $q \in A$. If $g(q) \subseteq T$, then also $g(q) \subseteq S$ and $q \notin A$. Since $M'(\langle x, T\rangle)$ halts with a decision only if for all queries q it holds that $f(q) \subseteq T \iff g(q) \not\subseteq T$, and since - by the properties of the reduction functions – *not* both $f(q') \subseteq T$ and $g(q') \subseteq T$ can hold, the simulation is correct. □

Define the function *query* as

$$query(x) = \begin{cases} f(x), & x \in A \\ g(x), & x \notin A \end{cases},$$

and let $Queries(x_1, \ldots, x_n) = \bigcup_{1 \le i \le n} query(x_i)$. By definition, $query(x) \subseteq S$ for all x. We prove the existence of a short sequence of strings whose *Queries*-set can be used as oracle to decide all instances of A up to a certain length.

Claim 2 *For every n, there exist an $m \le p(p(n))$ and a sequence $x_1, \ldots, x_m$ of strings in $\Sigma^{\le p(n)}$, such that for all $x \in \Sigma^{\le n}$ it holds that $f(x) \subseteq Queries(x_1, \ldots, x_m) \iff g(x) \not\subseteq Queries(x_1, \ldots, x_m)$ and $M'(\langle x, Queries(x_1, \ldots, x_m)\rangle)$ decides.*

Proof Let $x_1, \ldots, x_m$ be a longest sequence of strings in $\Sigma^{\le p(n)}$ with the two properties that (1) $query(x_i) \not\subseteq Queries(x_1, \ldots, x_{i-1})$ and (2) for all $y \prec x_i$ it holds that $f(y) \subseteq Queries(x_1, \ldots, x_{i-1}) \iff g(y) \not\subseteq Queries(x_1, \ldots, x_{i-1})$; i.e. x_i produces a new query and $M'(\langle x_i, Queries(x_1, \ldots, x_{i-1})\rangle)$ decides its input. Clearly, such a sequence exists. Since S is a sparse set, it follows that $m \le p(p(n))$, because otherwise condition (1) cannot hold for all strings in that sequence. Since $x_1, \ldots, x_m$ is the longest sequence having the above properties, it follows that for all $y \in \Sigma^{\le n}$ it holds that $f(y) \subseteq Queries(x_1, \ldots, x_m) \iff g(y) \not\subseteq Queries(x_1, \ldots, x_m)$. Since $Queries(x_1, \ldots, x_m) \subseteq S$ this implies that $M'(\langle y, Queries(x_1, \ldots, x_m)\rangle)$ decides all $y \in \Sigma^{\le n}$. □

The following Turing machine N decides A in the sense of strong non-determinism.

```
input y
T := ∅
guess m ≤ p(p(|y|)), x_1, ..., x_m ∈ Σ^{≤p(|y|)}
for i := 1 to m do
  if M'(⟨x_i, T⟩) halts without decision
    then halt without decision
    else if M'(⟨x_i, T⟩) accepts
      then T := T ∪ f(x_i)
      else T := T ∪ g(x_i)
    end
  end
end (* of the for-loop *)
if M'(⟨y, T⟩) accepts then accept else if M'(⟨y, T⟩) rejects then reject end
```

Every "path" of a computation of N decides correct, if it decides, since using Claim 1, $T \subseteq S$ at every time of a computation of N. The "completeness" of the algorithm follows from Claim 2, since for every input y there exists a computation path which accepts or rejects. It is easy to see, that N performs its computations in polynomial time. Therefore, N is a strong non-deterministic polynomial time bounded Turing machine deciding A, which completes the proof. ■

Corollary 4.5

1. *Every word-decreasing self-reducible set in* IC[log,poly] *is in* NP ∩ co-NP.
2. *Every polynomially well-founded self-reducible set in* R^p_m(APT) *is in* NP ∩ co-NP.
3. *Every polynomially well-founded self-reducible set in* $R^{co\text{-}np}_m$(SPARSE) ∩ R^{np}_m(co-SPARSE) *is in* P^{NP}.

Since every set in NP ∩ co-NP is low in NP [Sch83] it follows that

Theorem 4.6 *Every polynomially well-founded self-reducible set in* IC[log, poly] *is low in* NP.

Now we consider self-reducible sets which disjunctively reduce to sparse sets. Lozano and Torán [LT91] showed that – in some relativized world – there exists a self-reducible sparse set which is not low for Θ_2^p. Therefore, to prove lowness for Δ_2^p of self-reducible sets which reduce to sparse sets is the "best" we can hope for.

Theorem 4.7 *For every* $\mathrm{FP}^{\mathrm{NP}}$*-minimizable self-reducible set A in $R_d^p(\mathrm{SPARSE})$ there exists an* $\mathrm{FP}^{\mathrm{NP}}$*-printable sparse set S' such that $A \le_d^p S'$.*

Proof Let $A \in R_d^p(\mathrm{SPARSE})$. From an observation in [AKM92b] it follows that $A \le_d^p S$ via a function f, such that for a strictly increasing polynomial r and a polynomial t, $f(x) \subseteq \Sigma^{r(|x|)}$ and $|f(x)| \le t(|x|)$ for all $x \in \Sigma^*$. Let A be an $\mathrm{FP}^{\mathrm{NP}}$-minimizable self-reducible set, witnessed by the self-reduction machine M and the $\mathrm{FP}^{\mathrm{NP}}$-minimizable order $\prec$. Define the following polynomially time bounded machine M' which gets as input a string x and a set O and simulates M using the disjunctive query mechanism to O to answer the oracle queries.

> **input** $\langle x, O\rangle$
> simulate M on input x, but
>   whenever M queries q to the oracle then
>   **if** $f(q) \cap O \neq \emptyset$
>   **then** continue with answer "yes"
>   **else** continue with answer "no"
>   **end**
> *accept* if and only if M accepts

We can use the inductive structure of $\prec$ to derive that M' on input $\langle x, O\rangle$ decides the membership of x w.r.t. A correctly if x and O have certain properties.

Claim 3 *For all $x \in \Sigma^*$ and $Q \subseteq \Sigma^*$: if for all y with $y \prec x$ it holds that $\langle y, Q\rangle \in L(M') \iff f(y) \cap Q \neq \emptyset$, then for all y with $y \prec x$ or $y = x$ it follows that $\langle y, Q\rangle \in L(M') \iff y \in A$.*

Proof By definition there exists a polynomial p such that $|x| \le p(|y|)$ for all x, y with $x \prec y$. For every x each increasing chain $x_0 \prec x_1 \prec \cdots \prec x_k$ ending with $x = x_k$ has length bounded by $2^{p(|x|)}$. The proof of the Claim is by induction on the maximal length of increasing chains ending with x. At first, consider x for which there exists no increasing chain ending with x. This means $\forall y$: $y \not\prec x$. Then the oracle machine M on input x does not ask any query to the oracle and decides "$x \in A$?" correctly independent from the oracle, therefore $\langle x, Q\rangle \in L(M') \iff x \in A$ for every set Q. Now, assume as induction hypothesis that the claim holds for all y, for which the longest increasing chain $x_0 \prec x_1 \prec \cdots \prec x_k$ ending with $y = x_k$ has length k. For the induction step, take an x where the longest increasing chain ending with x has length $k+1$, and let Q be a set such that for all y with $y \prec x$ it holds that $\langle y, Q\rangle \in L(M') \iff f(y) \cap Q \neq \emptyset$. M on input x only queries the oracle for strings $q \prec x$. By the induction hypothesis it holds for all $q \prec x$ that $\langle q, Q\rangle \in L(M') \iff q \in A$ and therefore $q \in A \iff f(y) \cap Q \neq \emptyset$. From this we get that M' on input $\langle x, Q\rangle$ simulates M with oracle A on input x correctly, i.e. $\langle x, Q\rangle \in L(M') \iff x \in A$. □

Let $err(\langle x, Q\rangle)$ = "$\langle x, Q\rangle \in L(M') \not\Leftrightarrow f(x) \cap Q \neq \emptyset$". Clearly, err is in P. Since $\prec$ is $\mathrm{FP}^{\mathrm{NP}}$-minimizable, there exists an $\mathrm{FP}^{\mathrm{NP}}$ transducer T such that for all B, $T^B(0^i)$ computes a minimal string in B for Σ^i. Define a function h such that $h(\langle 0^i, Q\rangle) = T^{\{x \mid err(\langle x, Q\rangle)\}}(0^i)$. Clearly, h can be computed by simulating T and answering all oracle queries q using the P predicate $err(\langle x, Q\rangle)$. Therefore h is in $\mathrm{FP}^{\mathrm{NP}}$.

The following function CONSTRUCT computes on input 0^i a sparse set S_i, for which it will be shown that $A \le_d^p S_i$ via f for all instances of length i. The union over all those S_i is a sparse subset of $\bigcup_{x \in A} f(x) - \bigcup_{x \notin A} f(x)$.

> CONSTRUCT(0^i)
> $Q := \emptyset$
>   **while** $\exists y \in \Sigma^i$: $err(\langle y, Q\rangle)$ **do**
> $x := h(\langle 0^i, Q\rangle)$
>   **if** $\langle x, Q\rangle \in L(M')$
>   **then** $Q := Q \cup f(x)$
>   **else** $Q := Q - f(x)$
>   **end**
>   **end** (* while *)
>   **return** Q

Consider the computation of CONSTRUCT(0^i) for some i. Let Q_n be the contents of the set variable Q in this computation after the n-th repetition of the while-loop, and let $x_n = h(\langle 0^i, Q_{n-1}\rangle)$. By the properties of the function h and Claim 3 it follows that

Claim 4 $x_n \in A \iff \langle x_n, Q_{n-1}\rangle \in L(M')$.

For every $x_n \in A$ at least one new element of S is put into Q_{n-1}, whereas for no $x_n \notin A$ an element of S becomes removed from Q_{n-1}.

Claim 5 *If $x_n \in A$, then $Q_{n-1} \cap S \subsetneq Q_n \cap S$.*

Proof Assume $x_n \in A$. Then $f(x_n) \cap S \neq \emptyset$. Since $\langle x_n, Q_{n-1}\rangle \in L(M')$ (by Claim 4) it follows that $Q_n = Q_{n-1} \cup f(x_n)$. Since $err(\langle x_n, Q_{n-1}\rangle)$, we get that $f(x_n) \cap Q_{n-1} = \emptyset$, and therefore $Q_{n-1} \cap S \subsetneq (Q_{n-1} \cup f(x_n)) \cap S$. □

Claim 6 *If $x_n \notin A$, then $Q_{n-1} \cap S = Q_n \cap S$ and $Q_n \subsetneq Q_{n-1}$.*

Proof Assume $x_n \notin A$. Then $f(x_n) \cap S = \emptyset$. By Claim 4 it holds that $x_n \notin L(M', Q_{n-1})$, and therefore $Q_n = Q_{n-1} - f(x_n)$. Since $err(\langle x_n, Q_{n-1}\rangle)$, it follows that $f(x_n) \cap Q_{n-1} \neq \emptyset$, and therefore $Q_{n-1} \cap S = (Q_{n-1} - f(x_n)) \cap S$. So the claim follows. □

The next Claims are used to show that the number of the repetitions of the while-loop of the function CONSTRUCT is polynomially bounded.

Claim 7 *For all n:* $|\{x_1, \ldots, x_n\} \cap A| \leq |S^{\leq r(i)}|$.

Proof Since by Claim 5 every x_i in A puts at least one new element of $S^{\leq r(i)}$ into Q which – by Claim 6 – will not become removed from Q, the Claim follows. □

Claim 8 *For all n:* $\{x_{n+1}, \ldots, x_{n+|Q_n - S|+1}\} \cap A \neq \emptyset$.

Proof Assume the contrary. If $\{x_{n+1}, \ldots, x_{n+|Q_n-S|+1}\} \subseteq \overline{A}$, then by Claim 6 it follows that $|Q_{n+|Q_n-S|+1}| \leq |Q_n| - (|Q_n - S| + 1) < |Q_n \cap S|$. But then $Q_{n+|Q_n-S|+1} \cap S \subsetneq Q_n \cap S$, yielding a contradiction to Claim 6. □

From Claim 7 it follows for all n that $|Q_n| \leq |S^{\leq r(i)}| \cdot t(i)$, where t is the polynomial bounding the number of queries produced by f on inputs of length at most i. Putting Claims 7 and 8 together, it follows that the number of repetitions of the while-loop in the computation of CONSTRUCT(0^i) is at most $|S^{\leq r(i)}| \cdot t(i)$, which is polynomial in i. Therefore the set S_i is sparse. Define $S' = \bigcup_{i \geq 0}(S_i \cap \Sigma^{r(i)})$. Since for every S_i the condition of the while-loop is not fulfilled, we get $A \leq_d^p (S_i \cap \Sigma^{r(i)})$ via f for all i and all instances in Σ^i. Therefore $A \leq_d^p S'$ via f. The function CONSTRUCT can easily be modified to output $S_i \cap \Sigma^{r(i)}$ on input 0^i. Therefore it is straightforward to see that S' is $\mathrm{FP}^{\mathrm{NP}}$-printable. ■

The same result holds if A is in R_c^p(co-SPARSE). Let A be a $\mathrm{FP}^{\mathrm{NP}}$-minimizable self-reducible set in R_d^p(SPARSE). Since Δ_2^p is closed under disjunctive reductions it follows from the above Theorem that $A \in \Delta_2^p$. Furthermore, since the use of A as oracle can be replaced by a precomputation in Δ_2^p which computes a sparse set to which A disjuntively reduces, and every oracle query to A can now become replaced by a polynomial time computation which disjunctively queries the precomputed sparse set, the next Theorem follows.

Theorem 4.8 *Every $\mathrm{FP}^{\mathrm{NP}}$-minimizable self-reducible set in R_d^p(SPARSE) or R_c^p(co-SPARSE) is low in Δ_2^p.*

Finally we consider the class $R_{hd}^p(R_c^p(\mathrm{SPARSE}))$ which contains the Boolean closure of R_c^p(SPARSE).

Theorem 4.9 *Every $\mathrm{FP}^{\mathrm{NP}}$-minimizable self-reducible set in $R_{hd}^p(R_c^p(\mathrm{SPARSE}))$ is low in Δ_2^p.*

The proof is a straightforward modification of a proof from [AKM93] where it is shown that every length-decreasing self-reducible set A in $R_{hd}^p(R_c^p(\mathrm{SPARSE}))$ is in $R_{hd}^p(R_c^p(S))$ for a sparse Δ_2^p-printable set S.

Corollary 4.10 *Every word-decreasing or length-decreasing self-reducible set in $R_{hd}^p(R_c^p(\mathrm{SPARSE}))$ is low in Δ_2^p.*

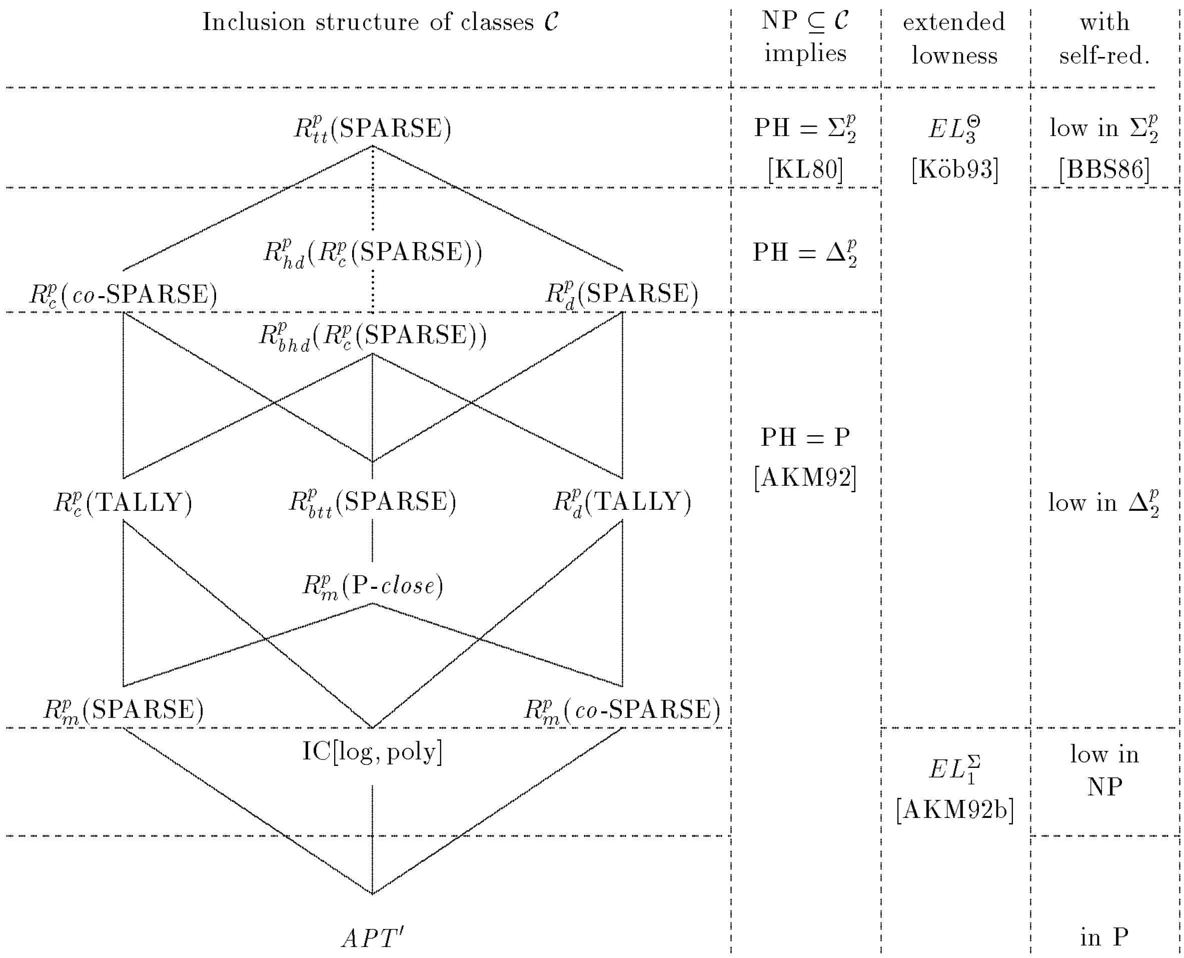
Since the NP-complete set SAT and and the PSPACE-complete set QBF are length-decreasing self-reducible we get

Proposition 4.11 *1. If* SAT $\in R^p_d(\text{SPARSE}) \cup R^p_{hd}(R^p_c(\text{SPARSE}))$*, then the Polynomial Time Hierarchy collapses to* Δ^p_2.

2. If QBF $\in R^p_d(\text{SPARSE}) \cup R^p_{hd}(R^p_c(\text{SPARSE}))$*, then* PSPACE $= \Delta^p_2$.

5 Summary and overview

The following picture summarizes the results of this paper and shows their relations to previously known results. It gives an overview on the relationships of the reduction classes to sparse sets, the collapses of the Polynomial Time Hierarchy PH under the assumption that NP is contained in one of these classes, the extended lowness (as defined in [BBS86]) of the reduction classes, and on the lowness of self-reducible sets in these classes. Dotted lines in the inclusion diagram stand for $\subseteq$, the other inclusions are proper ($\subsetneq$) (cf. [Ko89, AHH+92, AKM92b, BLS93]).



Acknowledgements

I like to thank Ricard Gavaldà and Johannes Köbler for helpful comments and discussions.

References

[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.

[AKM92] V. Arvind, J. Köbler, and M. Mundhenk. On bounded truth-table, conjunctive, and randomized reductions to sparse sets. *Proceedings of the 12th Conference on the Foundations of Software Technology & Theoretical Computer Science,* 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. *Proc. Third International Symposium on Algorithm and Computation*, Lecture Notes in Computer Science, #650:249–258, Springer Verlag, 1992.

[AKM93] 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.

[Ba92] J. Balcázar. 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.* EATCS Monographs on Theoretical Computer Science, Springer Verlag, 1988.

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

[BL92] R. Book and J. Lutz. On languages with very high information content. *Proc. of the 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.

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

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

[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.

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

[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.

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

[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

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