Splittings, Robustness and Structure of Complete Sets*

Harry Buhrman[1] Albrecht Hoene[2] Leen Torenvliet[1]

[1]Department of Mathematics and Computer Science, University of Amsterdam, Plantage Muidergracht 24, 1018 TV Amsterdam, The Netherlands
[2]Department of Computer Science, Technische Universität Berlin, D-1000 Berlin 10, Germany. Partially sponsored by Deutsche Forschungsgemeinshaft, Postdoktorandenstipendium

Abstract

We investigate the structure of *EXP* and *NEXP* complete and hard sets under various kinds of reductions. In particular, we are interested in the way in which information that makes the set complete is stored in the set. To address this question for a given hard set A, we construct a sparse set S, and ask whether $A - S$ is still hard. It turns out that for most of the reductions considered and for an arbitrary given sparseness condition, there is a single subexponential time computable set S that meets this condition, such that $A - S$ is not hard for any A. Not only is this set S subexponential time computable, but a slight modification of the construction can make the complexity of S meet any reasonable superpolynomial function. On the other hand we show that for any *polynomial-time computable* sparse set S, the set $A - S$ *remains* hard. There are other properties than time complexity that make a set 'almost' polynomial-time computable. For sparse *p-selective* sets, we can show the same, $A - S$ remains hard.

In the second part of the paper we address the question whether the information that is evidently abundantly present can be used to produce two disjoint complete sets from a single complete set A, i.e. the question whether exponential-time complete sets are *mitotic*. It turns out that the many-one complete sets are, yet there is strong evidence that other complete sets may not be. In particular we show the existence of a 3-truth-table complete set that can not be split into

*Some of the results in this paper were presented at the 1993 STACS conference in Würzburg, Germany

two many-one complete sets. Finally we show a complexity theoretic counterpart to Sacks' splitting theorem, i.e. we show that any many-one complete set for EXP can be split into two incomplete sets.

1 Introduction

The structure of complete sets under various notions of the reducibility concept is a well-studied subject in structural complexity theory. In [AS84b, AS84a, BHT91, BST91, Ber76, BH77, GH89, Har78, Jon75, Kar72, LLS75, Wag86, Wat87] and *many* other papers results on this subject can be found. Questions like which reducibility notions differ and which are the same on what kind of complete sets, and what kind of subsets can be found in which type of complete sets have been extensively researched. Yet, many open problems remain. On the most interesting complexity class, NP, almost every question is open. Even in surroundings where theorems are more easily proved, like exponential time, many questions remain. For instance, the question whether all (non)deterministic exponential time $\leq_m^P$-complete set are p-isomorphic is still a very hard open problem.

In this paper we put another, albeit small, step forward on the dark path. We investigate the question whether, and if so which, harm can be done to exponential time complete sets when taking out or putting in sparse sets. This type of question was studied by Schöning [Sch86] and in [TFL93]. In [TFL93], it was shown that for an arbitrary sparseness condition, there exists a single sub-exponential time computable set S such that for any $\leq_m^P$-complete set A, the set $A - S$ is incomplete. This instable behavior seems to vanish when the set S is sparse and polynomial-time computable, i.e. for any complete set A and sparse polynomial-time computable set S, the set $A - S$ is still complete. Note here that it is not interesting to look at dense polynomial-time computable sets, because then we could take out Σ^* and certainly destroy the completeness.

We study this question for complete sets under different types of reductions. For conjunctive, disjunctive, and general bounded truth-table complete sets we can derive more or less the same theorems as for many-one complete sets. For Turing reductions the situation seems completely different. A problem here is that not much is known about the structure of $\leq_T^P$-complete sets for EXP. In particular it is not known whether $\leq_T^P$-complete sets are dense. In the—unexpected—case that there exist sparse $\leq_T^P$-complete sets, it might be possible to remove the entire set by a polynomial-time computable superset. This intuition is made precise in Section 3. Theorems of this kind suggest that the complete sets have regions where they possess information that is "crucial" for their completeness. Our theorems show that this information can be traced in subexponential (superpolynomial) time, but is evasive to polynomial-time. The time complexity of the set indeed seems to be crucial here. Other properties than just computable within a certain time bound have been defined in an attempt to characterize the 'almost feasible' notion. Selman [Sel79] introduced p-selective sets as a resource bounded analog of semi-recursive sets introduced by Jockusch [Joc68]. Although any (polynomial-time Turing) degree has a p-selective set, p-selective sets are intuitively easy to compute since for any two strings x and y it can be decided in polynomial-time which of the two is more likely to be in the set. We show that (sparse) p-selective sets behave in the same way as do polynomial-time computable sets with respect to the robustness properties we investigate here.

In the second part of the paper we study 'redundancy' of complete sets. Can we divide a complete set into two parts that are itself again complete? We show–if this is possible–that the information that makes a set complete was in a way redundantly present. These kind of questions have been studied in recursion theory, where those sets that possess this property are called *mitotic*. We follow the line of Ambos-Spies [AS84a] and prove that $\leq_m^P$-complete sets for EXP indeed are (weakly-p-m-) mitotic.

In the remaining part of the paper we show that there exists a $\leq_{3-tt}^P$-complete set that is not weakly-p-m-mitotic. Furthermore, we have a counterpart of Ladner's splitting theorems [Lad75], i.e. we construct a set that can be split into two parts that are strictly below the degree of complete sets, but are $\equiv_m^P$, instead of incomparable.

2 Definitions and Notations

We assume the reader familiar with standard notions in structural complexity theory, as are defined e.g. in [BDG88]. All kinds of *polynomial-time bounded* reductions—many-one, (disjunctive and conjunctive) truth table and Turing—are frequently used without explanation. We use the following notations for the (polynomial-time computable versions of the) different types of reductions: $\leq_m^P$, for many-one reductions, $\leq_1^P$, for one-one reductions, $\leq_{k-tt}^P$, for k-truth-table reductions, $\leq_{btt}^P$, for bounded truth-table reductions, $\leq_{tt}^P$, for truth-table reductions, $\leq_c^P$, for conjunctive truth-table reductions, $\leq_d^P$, for disjunctive truth-table reductions, $\leq_{k-d}^P$, for k-disjunctive truth-table reductions, and $\leq_T^P$, for Turing reductions. $\leq_m$ and $\leq_T$ also appear in the paper without the superscript P, to indicate the unbounded version of these reductions.

Various definitions for these types of reductions can be found in the literature, e.g. in [LLS75, BHT91, BST91]. We think of polynomial-time bounded reductions as being modeled by adaptive and non-adaptive oracle machines, and use enumerations $\{M_i\}_i$ of these reductions. In the case of many-one reductions, i.e. where the machines are transducers we also use $\{f_i\}_i$ to emphasize this fact. Without loss of generality we always assume machine M_i in such an enumeration to be time-bounded by $n^i + i$, where n is the length of the input. Usually, we denote the set of strings queried on input x by machine i with oracle A by $Q_i^A(x)$, or by $Q_i(x)$ if M_i is non-adaptive. The result of the computation (accept/reject or the value computed) of machine M_i on input x (relative to oracle A) is sometimes denoted as $M_i(x)$ ($M_i^A(x)$), where $M_i(x) = 0$ or 1 means that the computation rejects or accepts respectively.

Sets are denoted by capital letters and are subsets of Σ^*, where $\Sigma = \{0, 1\}$. The cardinality of a set A is denoted as $\|A\|$. Strings are denoted as small letters $x, y, u, v, \ldots$. The value of the *characteristic function* of a set A on a string x is denoted by $\chi_A(x)$, i.e. $\chi_A(x) = 1$ if $x \in A$ and 0 otherwise. We assume easy to compute *pairing and projection functions*. For strings, the pairing of $x_1, \ldots, x_n$ is denoted by $<x_1, \ldots, x_n>$, and $\pi_i(y)$ is the projection of y onto its i^{th} coordinate.

The length of a string x is denoted by $|x|$. For a set A:

- for $n \in \omega$, we let the notation $A^{\leq n}$ stand for the set consisting of all strings in A of length $\leq n$ and

- for a string x, we let $A^{[x]}$ stand for the x *section of* A, i.e. the set $\{<y, z> : y = x \text{ and } <y, z> \in A\}$. In this case z may also be $<z_1, \ldots, z_n>$, so that we get the set $\{<y, z_1, \ldots, z_n> : y = x$ and $<y, z_1, \ldots, z_n>$ in $A\}$.

- In order to measure the density of A, we say that A is $g(n)$-sparse for some nondecreasing function $g : \omega \to \omega$, if $\|A^{\leq n}\| < g(n)$.

The main complexity classes considered in this paper are P—polynomial time, *EXP*—exponential time, and *NEXP*—nondeterministic exponential time. For the latter classes, we allow polynomials to act as exponents in the time bounds, e.g. $EXP = \bigcup_{i\in\omega} DTIME(2^{n^i})$. For EXP, the set K is the universal complete set. $K = \{<i, x, l> : M_i \text{ accepts } x \text{ in } \leq l \text{ steps } \}$.

3 The Robustness of Completeness Notions for Exponential Time

In this section we study the question which sparse sets can be removed from exponential time complete sets of different types without disturbing their completeness. The question originates from work of Schöning [Sch86], who showed that for all complete sets A in EXP and every set D in P the set $A\Delta D$ is of exponential density. In [TFL93], Tang, Fu and Liu showed that even for subexponential time computable D, the difference $A\Delta D$ remains exponential time complete.

They further show that for an arbitrary sparseness condition, there exists a single subexponential time computable set S, such that for *any* exponential time complete set A, the set $A - S$ is no longer exponential time hard. Their proof hinges on the fact that for any exponential time computable set B and any exponential time complete set A, there exists a *length increasing* reduction from B to A. Subsequently, the subexponential time computable set is constructed by chosing a sufficiently sparse polynomial time computable subset of $\{0\}^*$, and defining S as the image of this set varying over all polynomial-time computable functions, i.e. $S = \{0^{b_i} : |f_i(0^{b_i})| > b_i\}$, where the b_i are chosen sufficiently far apart.

A closer look at the proof learns that, though the theorem just states that S is subexponential time computable, there are various ways of making S come arbitrarily close to polynomial-time. It therefore seems reasonable to ask, whether we can also chose S to be polynomial-time computable. The answer to this question is negative, as observed in [TFL93]. From the $\leq_1^P$-reduction of $K' = K\times\Sigma^*$ to the EXP complete set A, we can easily construct a $\leq_m^P$-reduction to $A - S$ for any polynomial-time computable sparse set S. In this section we will investigate for different types of reductions, which sparse sets can destroy the completeness of a given set, and for which sparse sets completeness is preserved.

3.1 Sparse sets that destroy completeness

The set K' defined above is, of course, $\leq_m^P$-complete for EXP. In fact it is $\leq_d^P$-complete for EXP in a special way. For a given string x either all strings $<x, y>$ are in this set, or all are out depending on $x \in K$. Therefore, as long as S is $p(n)$-sparse, the set $K' - S$ remains $\leq_d^P$-complete for EXP. The reduction from K to K' on input x just queries the set $\{<x,y>|0 \leq y \leq p(2n)+1\}$. Since all these strings have length $\leq 2|x|$, at least one of them is not in S and it is in K' iff $x \in K$. This explains why a theorem like "*There exists a sparse set S, such that for any $\leq_d^P$-complete set A for EXP, the set $A - S$ is not $\leq_d^P$-complete for EXP*" cannot exist.

The best we can hope for is a theorem for reductions that can query less strings than $\|S^{\leq n}\|$ for each length n. On the other hand, we want S to remain arbitrarily sparse, so the best bound we can hope for is some fixed but arbitrary constant. Such reductions are called *bounded truth table reductions*, and for these reductions we can obtain the theorem. In fact since the proof method of our theorem is not dependent on the reductions being non-adaptive we can get the theorem for bounded Turing reductions ($\leq_{bT}$).

Theorem 1 *Given a recursively computable non-decreasing function g with $\lim_{n\mapsto\infty} g(n) = \infty$, and a function f that is superpolynomial, subexponential, and time-constructible. There exists a $g(n)$-sparse $O(f(n))$-time computable set S, such that for any $\leq_{btt}^P$-complete set A for EXP, the set $A - S$ is no longer $\leq_{btt}^P$-complete.*

Proof. For a given set A, we demonstrate the existence of a set L_A such that $L_A \leq_{btt}^P A$ iff there exists a reduction M_i that queries at least one string $y \in A$ with $|y| > b(i)$ on input $0^{b(i)}$. Next, we let all such strings be elements of S thereby preventing the existence of such a reduction from L_{A-S} to $A - S$ and hence completeness of $A - S$. This technique was borrowed from Watanabe [Wat87].

Let $\{M_i\}_i$ be an enumeration of all $\leq_{btt}^P$-reductions. Without loss of generality we may assume that machine M_i generates $\leq i$ queries on any input. First we define a set of numbers $\{b(i)\}$ sufficiently far apart and sufficiently easy to recognize, i.e. we want that for each n the question "$\exists i : n = b(i)$?" can be answered in time $O(n)$, and furthermore we want for each i that $g(b(i)) > i^2$. This means that we can define at least i^2 strings in $S^{\leq b(i)}$ without disturbing the sparseness condition on S. In fact, we will just put $\leq i$ strings in each interval $S^{\leq b(i+1)} - S^{\leq b(i)}$. Finally, we wish to compute $Q_i(0^{b(i)})$ to define strings in S. To be able to do this in time $f(b(i))$, we need that $2^{b(i)} > f(b(i)) > (b(i))^i + i$, for all i.

We let $S = \{y : y \in Q_i(0^{b(i)}) \wedge |y| > b(i)\}$ and claim that S is the set searched for. First, $\|S^{\leq n}\| \leq g(n)$ for each n, since for each n the only strings that are in $S^{\leq n}$ are the sets in $\cup_{j\leq i} Q_j(0^{b(j)})$ for i maximal such that $b(i) \leq n$. Hence there are at most i^2 strings in $S\leq n$ and $b(i)$ was chosen such that $g(n) \geq g(b(i)) > i^2$ as required. Next, S is $O(f(n))$ time computable by the choice of $b(i)$. Finally, the set $A-S$ is not $\leq_{btt}^P$-complete. To see this, assume that $A-S$ is $\leq_{btt}^P$-complete and define the set $L_{A-S} = \{0^{b(i)} : M_i^{(A-S)^{\leq b(i)}}(0^{b(i)}) = 0\}$. L_{A-S} is exponential time computable since $2^{b(i)} > b(i)^i + i$ and since both A and S are exponential time computable, and are only queried on inputs of length $\leq b(i)$ on input $0^{b(i)}$. Therefore $L_{A-S} \leq_{btt}^P A-S$, say via reduction M_j. However $Q_j(0^{b(j)}) \cap (A-S) \subseteq \Sigma^{\leq b(j)}$ and therefore $0^{b(j)} \in L_{A-S}$ iff $M_j^{A-S}(0^{b(j)})$ rejects. A contradiction. $\square$

We observe that for *conjunctive* truth table reductions the statement of the theorem is valid without a constant bound on the number of queries. Consider the i^{th} conjunctive truth table reduction M_i. For *any* S, if $0^{b_i} \notin L_{A-S}$ then M_i can not be a $\leq_c^P$ reduction from L_{A-S} to $A-S$. If on the other hand $0^{b_i} \in L_{A-S}$ (and M_i is the reduction from L_{A-S} to $A-S$) then $Q_i(0^{b_i}) \subseteq A-S$. Hence in the construction of S it suffices to put $\min\{y : y \in Q_i(0^{b_i}) \text{ and } |y| > b_i\}$ in S. Then we get from the same construction.

Corollary 2 *Given a recursively computable non-decreasing function $g(n)$ with $\lim_{n\mapsto\infty} g(n) = \infty$. There exists a $g(n)$-sparse subexponential time computable set S such that for any $\leq_c^P$-complete set A for EXP the set $A-S$ is no longer $\leq_c^P$-complete.*

Conjunctive truth table reducibilities form an exception in yet another way. For these reductions we can even let the set A be EXP-hard instead of EXP-complete. We use the fact that for conjunctive truth table reductions (or for any truth table reduction that computes a fixed type of truth table on any input) we can get a kind of 1-1 behavior for the query sets. A similar result for $\leq_m^P$-hard sets that uses the fact that these sets are also hard under $\leq_1^P$-reductions appears in [TFL93]. We can force the query sets belonging to two distinct inputs not to be subsets of one another.

Lemma 3 *If A is $\leq_c^P$ hard for EXP, then for any set B in EXP there exists a $\leq_c^P$-reduction M_j such that $Q_j(x) \not\subset Q_j(y)$, whenever $x \neq y$.*

Proof. Let $\{M_i\}_i$ be an enumeration of $\leq_c^P$-reductions We construct a set W as follows. On input $<i,x>$ compute $Q_i(<i,x>)$. If there is a $y < x$ such that either $Q_i(<i,x>) \subset Q_i(<i,y>)$ or $Q_i(<i,y>) \subset Q_i(<i,x>)$ then let $<i,x> \in W$ iff $<i,y> \notin W$ else let $<i,x> \in W$ iff $x \in B$.

It is easy to see that W is in EXP, so there exists a $\leq_c^P$ reduction from W to A, say M_j. For this reduction it follows that if $x \neq y$ then $Q_j(<j,x>) \not\subset Q_j(<j,y>)$. Hence $M'(x) = M_j(<j,x>)$ computes a $\leq_c^P$ reduction with the required property. $\square$

From this lemma we get:

Theorem 4 *Given a recursively computable non-decreasing function $g(n)$ with $\lim_{n\mapsto\infty} g(n) = \infty$. There exists a $g(n)$-sparse set S in EXP such that for any $\leq_c^P$-hard set A for EXP the set $A - S$ is no longer $\leq_c^P$-hard.*

Proof. Again, we let the numbers b_i be sufficiently far apart to guarantee sparseness of S if we put one string in S for each b_i, and such that 0^{b_i} is again easy to recognize. Furthermore we let $2 \times (b_{i-1})^{i-1} < b_i$ to avoid confusion later on. Then we put the least string in $\bigcup_{|y|\leq b_i} Q_i(<0^{b_i}, y>)$ of length $\geq b_i/2 - 1$ in S. (If no such string exists we do nothing.)

S is exponential time computable since to decide membership of a string x in S we search for a 0^{b_i} such that $b_i \leq 2 \times |x|$ and $|x| < (b_i)^i$. (There can only be one candidate.) Now compute the query sets on the at most $2^{2|x|}$ different y in time $\leq 2^{2|x|} \times (2|x|)^i + i$, and see if x is the least string of the right length in the union of these sets.

$A-S$ is not $\leq_c^P$-complete, since if it were, then one of the $\leq_c^P$-reductions of $0^* \times \Sigma^*$ would behave as predicted by Lemma 3. However this reduction, say M_i on input $<0^{b_i}, y>$ finds one of the queried strings not in $A - S$ for some y and hence must reject, a contradiction. $\square$

Conjunctive and disjunctive truth table reducibilities are kind of each others complement on EXP. If a set is conjunctive truth table reducible to a set A then it is disjunctive truth table reducible to $\overline{A}$. So we find.

Corollary 5 *Given a recursively computable non-decreasing function $g(n)$ with $\lim_{n\mapsto\infty} g(n) = \infty$. There exists a $g(n)$-sparse subexponential time computable set S such that for any $\leq_d^P$-complete set A for EXP the set $A \cup S$ is no longer $\leq_d^P$-complete.*

and

Corollary 6 *Given a recursively computable non-decreasing function $g(n)$ with $\lim_{n\mapsto\infty} g(n) = \infty$. There exists a $g(n)$-sparse set S in EXP such that for any $\leq_d^P$-hard set A for EXP the set $A \cup S$ is no longer $\leq_d^P$-hard.*

Proof. If $K \leq_c^P A$ via M_i then:

$$\begin{array}{rrcl} & x \in K & \Leftrightarrow & Q_i(x) \subset A \\ \text{or:} & x \notin K & \Leftrightarrow & Q_i(x) \cap \overline{A} \neq \emptyset \end{array}$$

Now $\overline{K} \in EXP$ so $\overline{K} \leq_m^P K$ say via f.

$$\begin{array}{lrcccl} \text{so:} & x \in \overline{K} & \Leftrightarrow & f(x) \in K & \Leftrightarrow & Q_i(f(x)) \subset A \\ \text{or:} & x \notin \overline{K} & \Leftrightarrow & f(x) \notin K & \Leftrightarrow & Q_i(f(x)) \cap \overline{A} \neq \emptyset \\ \text{or:} & x \in K & \Leftrightarrow & f(x) \notin K & \Leftrightarrow & Q_i(f(x)) \cap \overline{A} \neq \emptyset \end{array}$$

So $\overline{A}$ is $\leq_d^P$-hard via $M_i(f(x))$. Along the same lines: If A is $\leq_d^P$-hard then $\overline{A}$ is $\leq_c^P$-hard. So A is $\leq_d^P$-complete (hard) iff $\overline{A}$ is $\leq_c^P$-complete (hard) for *EXP*. But if $\overline{A}$ is $\leq_c^P$-complete (hard) then there exists a $g(n)$ sparse subexponential (exponential) time computable set S such that $\overline{A} - S$ is no longer $\leq_c^P$ complete (hard), and then $A \cup S$ is no longer $\leq_d^P$-complete (hard). □

3.2 Polynomial time is the cut-off point

As in the case of $\leq_m^P$-reductions in the case of complete sets, we can let the time complexity of the set S in the previous subsection come arbitrarily close to polynomial-time, and as in the case of $\leq_m^P$-reductions, polynomial-time computability is exactly the cut-off point.

Theorem 7 *For any set A that is $\leq_c^P$-hard for EXP and any $p(n)$-sparse polynomial-time computable set S, the set $A - S$ remains $\leq_c^P$-hard for EXP.*

Proof. We construct an *EXP* computable set W consisting of pairs $<i, x>$ such that there exists an i for which K reduces $\leq_m^P$ to $W^{[i]}$, and $W^{[i]}$ reduces $\leq_c^P$ to $A - S$. Then $A - S$ is $\leq_c^P$-hard. As before, $\{M_i\}_i$ is an enumeration of polynomial-time $\leq_c^P$-reductions. Assume that for some fixed s, membership in S is computable in time $n^s + s$. W is constructed according to the following rules:
If $z > p(|<i, x, z>|^i + i) + 1$ or $|x|^i + i > 2^{|x|}$ or $|x|^s + s > 2^{|x|}$ then $<i, x, z> \notin W$. For $<i, x, z>$ with $0 \leq z \leq p(|<i, x, z>|^i + i) + 1$, compute the union of the query sets for smaller inputs: $U(i, x, z) = \bigcup_{z' < z} (Q_i(<i, x, z'>) \cap S)$.
If $Q_i(<i, x, z>) \cap S \subseteq U(i, x, z)$ then $<i, x, z> \in W$.
If $Q_i(<i, x, z>) \cap S \subseteq U(i, x, z)$ then $<i, x, z> \in W$ iff $x \in K$.

First, W is exponential time computable since membership in a queryset or in S needs only be computed if sufficient time is available. Hence one of the machines, say M_w is a polynomial-time $\leq_c^P$-reduction from W to A. Next there is a $\leq_c^P$-reduction from $W^{[w]} = \{<y,x,z>|y = w \text{ and } <y,x,z> \in W\}$ to $A - S$.

Proof. The reduction works as follows. On input $<w,x,z>$ reject if $z > p(|<i,x,z>|^w + w) + 1$ or if $|x|^i + i > 2^{|x|}$ or if $|x|^i + i > 2^{|x|}$. Otherwise compute $U(w,x,z)$ and $Q_w(<w,x,z>) \cap S$. If $(Q_w(<w,x,z>) \cap S) \not\subseteq U(w,x,z)$ then accept, otherwise accept only if $Q_w(<w,x,z>) - S \subseteq A - S$. □

Note that the construction of W allows for $Q_w(<w,x,z>) - S$ to be empty, which means that $Q_w(<x,w,z>) \subseteq U(w,x,z) \subseteq A$. Since M_w is the reduction from W to A this implies $<w,x,z> \in W$ (and $x \in K$). It is debatable whether a reduction that produces an empty set of queries may be called a conjunctive truth-table reduction, but since emptyness of the query set is easily checked and $A - S$ is non-empty, this is easily fixed. Finally, we show that there is a many-one reduction from K to $W^{[w]}$.

Proof. On input x if $|x|^w + w > 2^{|x|}$ or if $|x|^s + s > 2^{|x|}$, we use a table to see if $x \in K$ and produce a fixed string in $W^{[w]}$ or $\overline{W^{[w]}}$ accordingly. Otherwise, compute the least z such that $Q_w(<w,x,z>) \cap S \subseteq U(w,x,z)$ and let $f(x) = <i,x,z>$. □

□

In the proof of this theorem we actually showed the existence of a special set W and a reduction of W to the set $A-S$ that avoids queries about strings in S. By the definition of W we force that queries about strings in S must get a positive answer, and hence if the same queries are asked on larger inputs then the answer to those queries is already known and we can let membership in W of such a larger input depend only on queries outside S. It is not very difficult now to define a set W that forces a reduction from W to A to consider queries about S as *negatively* answered and so the same strategy yields:

Theorem 8 *For any set A that is $\leq_d^P$-hard for EXP and any $p(n)$-sparse polynomial-time computable set S, the set $A-S$ remains $\leq_d^P$-hard for EXP.*

Proof. We now build the set W according to the rules:
On input $<i,x,z>$ with x and z within the same boundaries as in the previous proof, we again compute the union of the query sets $U(i,x,z) =$

$\bigcup_{z'<z}(Q_i(<i,x,z'>)\cap S)$. But now, if $(Q_i(<i,x,z>)\cap S)\subseteq U(i,x,z)$ then $<i,x,z>\in W$ iff $x\in K$ else $<i,x,z>\notin W$. We thus obtain a $\leq_d^P$-reduction from K to $A-S$ along the same lines. □

The next logical step would be to prove that the same statement holds for $\leq_{btt}^P$-reductions. This however seems to require more involved techniques. As things stand, we only have a proof for $\leq_{2-tt}^P$-reductions and it is rather lengthy.

Theorem 9 *Let A be $\leq_{2-tt}^P$-hard for EXP and S a polynomial time computable $p(n)$-sparse set. The set $A-S$ is still $\leq_{2-tt}^P$-hard for EXP.*

Proof. Again, we construct an exponential time computable set W. Let $\{M_i\}_i$ be an enumeration of all $\leq_{2-tt}^P$-reductions. This time, we have the following goal in mind for the reduction M_w from W to A. For some fixed polynomial q depending only on p we want for each x that there exists a pair of strings $(<w,x,z>,<w,x,z'>)$ for $0\leq z\leq q(n)+1$ such that $\|Q_w(w,x,z)\cap S\|+\|Q_w(w,x,z')\cap S\|<2$. That is, *either* for one of the sets both queried strings are outside S, or they have at most one (common) string in S. Then, we will construct W such that one of the two strings $<w,x,z>$, $<w,x,z'>$ is in W iff $x\in K$ and its membership in W can be decided on the basis of two strings in $(Q_w(<w,x,z>)\cup Q_w(<w,x,z'>))-S$. So $A-S$ remains $\leq_{2-tt}^P$-hard.

Consider the i^{th} reduction M_i, and a string x. For z an integer less than or equal to $2\times\binom{p(|<i,x,z>|^i+i)}{2}+p(|<i,x,z>|^i+i)+1$, we will decide whether to put strings of the form $<i,x,z>$ in or out of W. *All* other strings are not element of W. Without loss of generality we assume that the strings $<i,x,z>$ are of the same length for all z considered. To make W exponential time computable we shall let strings $<i,x,z>$ not be member of W if either $|x|^i+i>2^{|x|}$ (i.e. exponential time is not sufficient to recompute the query set(s)) or $|x|^s+2>2^{|x|}$ (i.e. exponential time is not sufficient to compute membership in S). We will silently assume that in these—for fixed i and s finitely many—cases, table lookup is used to finalize the definition of the reduction(s).

We treat three different cases according to $\|Q_i(<i,x,z>)\cap S\|$.

- First, for any of the z considered: If $Q_i(<i,x,z>)\cap S=\emptyset$ then $<i,x,z>\in W$ iff $x\in K$.
- Next, we make sure that the number of strings for which both queried strings are in S, i.e. $Q_w(<w,x,z>)\subseteq S$, is limited. There are at

most $\binom{p(|<i,x,z>|^i+i)}{2}$ different pairs of strings in S. Hence if we assume $Q_i(<i,x,z>) \subseteq S$ for $0 \le z \le 2\binom{p(|<i,x,z>|^i+i)}{2} + 1$. Then we must find $z_1 < z_2 < z_2$ such that $Q_i(<i,x,z_1>) = Q_i(<i,x,z_2>) = Q_i(<i,x,z_2>)$. Let $z_1 < z_2 < \ldots$ be such that $Q_i(<i,x,z_1>) = Q_i(<i,x,z_2>) = \ldots$ and $Q_i(<i,x,z_1>) \subseteq S$.

- We evaluate the truth table generated by M_i on input $<i,x,z_1>$ on all 4 possible settings $(\{0,0\},\{0,1\},\{1,0\},\{1,1\})$, and define $<i,x,z_1> \in W$ iff the number of settings on which M_i rejects is *greater* than the number of settings on which M_i accepts. This leaves at most two possible settings for the strings in $Q_i(<i,x,z_1>)$.
- We evaluate the truth table generated by M_i on input $<i,x,z_3>$ on the remaining two possible settings and let $<i,x,z_2> \in W$ iff M_i rejects on both settings. Now there is only one possible setting left.
- We evaluate the truth table generated by M_i on input $<i,x,z_3>$ on the last remaining setting, and let $<i,x,z> \in W$ iff M_i rejects with this setting.
- if $j > 3$ then $<i,x,z_j> \notin W$

We can now safely assume that, if M_w is the reduction from W to A, then there are at most $2 \times \binom{p(|<w,x,z>|^w+w)}{2}$ different z for which $Q_w(<w,x,z>) \subseteq S$.

- The final and most elaborate task is to define membership of $<i,x,z>$ for z such that $\|Q_i(<i,x,z>) \cap S\| = 1$. We observe that if $z_1 < z_2 < z_3 < \ldots < z_k$ is a sequence such that $\|Q_i(<i,x,z_j>) \cap S\| = 1$ and $Q_i(<i,x,z_j>) \cap S \neq Q_i(<i,x,z_{j'}>) \bigcap S$ for all $j \neq j'$ then $k \le p(|<i,x,z>|^i + i)$.

  So let $<i,x,z>$ be such that $\|Q_i(<i,x,z> \cap S\| = 1$. There are two cases:
  CASE A:$(\forall z' < z)[Q_i(<i,x,z>) \cap S \neq Q_i(<i,x,z'>) \cap S]$

  Let $Q_i(<i,x,z>) = \{y_1, y_2\}$ and let $\alpha = \alpha(\chi_A(y_1), \chi_A(y_2))$ or $\alpha = \alpha(\chi(y_1))$ be the truth table generated.

  We evaluate α for all (≤ 4) different pairs of values $(\chi_A(y_1), \chi_A(y_2))$ and act differently depending on the outcome of these evaluations.

There are at most sixteen possible truth tables corresponding to the following four cases.

1. If $\alpha \equiv 1$ we let $<i, x, z> \notin W$ and if $\alpha \equiv 0$ we let $<i, x, z> \in W$ respectively. (Hence for the reduction of W to A this case can not occur. This covers 2 of the 16 cases)
2. If there is only *one* combination of $a, b \in \{0, 1\}$ for which $\alpha(a, b) = 1$ then $<i, x, z> \in W$, and if there is only *one* combination of $a, b \in \{0, 1\}$ for which $\alpha(a, b) = 0$ then $<i, x, z> \notin W$ (Then we fix *both* $\chi_A(y_1)$ and $\chi_A(y_2)$. This covers 8 of the 16 cases.)
3. If α is in fact a $\leq^P_{1-tt}$ truth table then let y_1 be the string on which the value of α depends. There are two subcases:
   (a) If y_1 in S then we choose $<i, x, z>$ $(\in W, \notin W)$ such that $y_1 \notin A$.
   (b) If $y_1 \notin S$ then we let $<i, x, z> \in W$ iff $x \in K$.
   (this covers 4 of the 16 cases plus the case where $\|Q_i(<i, x, z>\| = 1$.)
4. In the remaining two cases α is either $\chi_A(y_1) = \chi_A(y_2)$ or $\chi_A(y_1) \otimes \chi_A(y_2)$. In case $=$ we let $<i, x, z> \in W$ and in case $\otimes$ we let $<i, x, z> \notin W$. (Then in both cases $y_1 \in A \Leftrightarrow y_2 \in A$.)

If CASE A does not hold then we are in
CASE B: $\exists z' < z$ such that $Q_i(<i, x, z'>) \cap S = Q_i(<i, x, z>) \cap S$. Let z_0 be the least such z'. We perform the case analysis above for $<i, x, z_0>$. If this analysis falls into case 3b then we let $<i, x, z> \notin W$. Otherwise we let $<i, x, z> \in W$ iff $x \in K$. This concludes the definition of W. Now we show that W helps to define a $\leq^P_{2-tt}$-reduction from K to $A - S$. First, if M_w is the $\leq^P_{2-tt}$-reduction of W to A then the set $W^{[w]}$ reduces $\leq^P_{2-tt}$ to $A - S$.
Proof. The reduction works as follows. On input $<w, x, z>$ reject if $z > 2 \times \binom{p(|<w,x,z>|^w + w)}{2} + p(|<w, x, z>|^w + w) + 1$. Otherwise compute $Q_w(<w, x, z>)$.

There are three cases.

1. $Q_w(<w, x, z>) \cap S = \emptyset$. In this case we can query both strings in $Q_w(<w, x, z>)$ and accept iff M_w accepts with the setting found.
2. $Q_w(<w, x, z>) \subseteq S$. Now by construction of W there can be at most one $z' < z$ such that $Q_w(<w, x, z>) = Q_w(<w, x, z'>)$. Find out if such a z exists. If it does then find the at most two possible settings

of $Q_w(<w,x,z>) = Q_w(w,x,z')$, evaluate the truth table on both settings. Accept iff M_w rejects on both settings. If such a z' does not exist then accept iff M_w rejects on at least half of the possible settings of it's truth-table.

3. $\|Q_w(<w,x,z>) \cap S\| = 1$. Again there are two possibilities

   (a) If there is no $z' < z$ with $Q_w(<w,x,z'>) \cap S = Q_w(<w,x,z>) \cap S$. Then compute the truth table and accept or reject according to the type of truth table and the rules for W above.

   (b) If $(\exists z' < z)[Q_w(<w,x,z'>) \cap S = Q_w(<w,x,z>) \cap S = \{y\}$, then find the least such z'. Perform the case analysis on the pair $<w,x,z'>$. If this falls into case 3b then reject. Otherwise compute $\chi_W(<w,x,z'>)$ with the rules for W above and the truth table generated on $<w,x,z'>$. Query $(Q_w(<w,x,z'> \cup Q_w(<w,x,z>)) - S$.
   Now either $\chi_W(<w,x,z'>)$ fixes $\chi_A(y)$ or $\chi_A(y) = \chi_A(u)$, where $\{u\} = Q_w(<w,x,z'>) - S$. Both cases give enough information to compute $\chi_W(<w,x,z>)$.

$\square$

$\square$

Finally, we show that K is $\leq_m^P$-reducible to $W^{[w]}$
Proof. On input x we compute the query sets $Q_w(<w,x,z>)$ for $z \leq 2 \times \binom{p(|<w,x,z>|^w + w)}{2} + p(|<w,x,z>|^w + w) + 1$ until we either find a string $<w,x,z>$ for which $Q_w(<w,x,z>) \cap S = \emptyset$ or $\exists z' < z : Q_w(w,x,z') \cap S = Q_w(w,x,z) \cap S = \{y\}$ and then let $f(x) = <w,x,z>$
Since $K \leq_m^P W^{[w]}$ and $W^{[w]} \leq_{2-tt}^P A - S$ we have that $A - S$ is $\leq_{2-tt}^P$-complete. $\square$

For $\leq_{3-tt}^P$-reductions the case analysis quickly blows up. We would have to find a more general property behind the reductions before we could attack the problem for $\leq_{btt}^P$-reductions. For general truth table reductions, such an attack would not even work. The proof method above relativizes, and we can show that there exists an oracle set A relative to which EXP has a $\leq_{tt}^P$-complete tally set T. Then $T - \{0\}^* = \emptyset$, which cannot be complete.

Lemma 10 *If $EXP \subseteq P/poly$ then there exists a tally set T that is $\leq_{tt}^P$-complete for EXP.*

Proof. It is well known that A is in $P/poly$ iff $A \leq_{tt}^{P} T$ for some tally set T. So from our hypothesis this gives us a tally set T, that is truth table *hard* for EXP. We will show how to construct a tally set T' (from T) that is complete for EXP.

From the fact that there exists a tally set T that is hard for EXP we get a truth-table reduction, say by machine M_i from K to T. Now fix n and consider all strings of length n. W.l.o.g. we may assume that M_i queries on input x of length n, always the same strings to T, namely $y_1, \ldots, y_{n^i}$, where $y_i = 0^i$. The idea is to find the minimal (in some way) setting of the y_j's in T such that $x \in K$ iff $M_i(x)$ accepts with this setting. Let x_j indicate the j^{th} string of length n in lexicographic order. Let $P_j = \{<a_1, \ldots, a_{n^i}> : (M_i(x_j)$ accepts iff $x_j \in K)$ **and** $\chi_T(y_i) = a_i\}$. (the answer to query $y_i = a_i)\}$. P_j codes exactly those tally sets T' that, when used as oracle for $M_i(x_j)$ let M_i compute the correct answer for $x_j \in K$. Note here that $\|P_j\| \leq 2^{n^i}$ for $0 \leq j < 2^n$. Set $P' = \bigcap_{i=1}^{2^n} P_i$ and let p_i be the i^{th} projection of the tuple y, where y is the minimal $y \in P'$. Put $0^{<n,i>}$ in T' iff $p_i = 1$. Obviously T' is tally and from the construction it is clear that T' is computable in exponential time. From the fact that T exists we get that T' exists and that $K \leq_{tt}^{P} T'$ □

Theorem 11 *There exists an oracle A, s.t. there exists a $\leq_{tt}^{P^A}$-complete set B for EXP^A and a polynomial-time computable, sparse set S, such that $B - S$ is not $\leq_{tt}^{P^A}$-hard for EXP^A.*

Proof. Wilson [Wil85] showed the existence of an oracle A where $EXP^A \subseteq P^A/poly$. Using this oracle together with Lemma 10 we get that there exists a tally set T that is complete for EXP^A. Setting $B = T$ and $S = \{0\}^*$, we get that $B - S = \emptyset$ and $\emptyset$ is not $\leq_{tt}^{P^A}$-complete for EXP^A. □

3.3 Close to polynomial-time computability

In the previous two subsections we found that for arbitrary sparseness conditions and arbitrary superpolynomial-time functions sparse sets can be constructed that destroy the exponential time hardness of a set. However if membership in the sparse set is polynomial-time computable then, in most cases, a polynomial-time reduction that avoids queries in the sparse set can be constructed from the original reduction to the hard set. Hence hardness of the set is preserved. In this section we show that there exists at least one other property that allows for this 'reconstruction' of the reduction.

Selman [Sel79] defined p-selectivity as a resource bounded analog of semi-recursiveness introduced by Jockusch [Joc68]. P-selective sets are defined as follows:

Definition 12 *A set A is called* p-selective *iff there exists a polynomial-time computable function $f : \Sigma^* \times \Sigma^* \mapsto \Sigma^*$, called a* p-selector*, such that for any $x, y \in \Sigma^*$*

1. *$f(x, y) \in \{x, y\}$ and*
2. *$\chi_A(f(x, y)) = \max\{\chi_A(x), \chi_A(y)\}$*

Although [Sel79] every (polynomial-time) Turing degree has a p-selective set, p-selective sets are in some sense close to P. Given a pair of strings, the p-selector of the set can, in polynomial-time, decide which of the two is 'more likely' to be in the set. In this section we investigate whether p-selectivity can replace polynomial-time computability in the theorems of the previous subsections. As it turns out, p-selectivity can indeed be substituted for polynomial-time computability in all cases that appear. Before we start matching polynomial-time computability and p-selectivity, let us start with quoting a useful property of p-selective sets that is either implicitly or explicitly used in many papers on p-selective sets.

Lemma 13 *Let B be a p-selective set and $V \subseteq \Sigma^*$ a finite set. The p-selector for B induces a total quasi order $\preceq_f$ such that $V = \{z_1, z_2, \ldots, z_n\}$, where $z_{i+1} \preceq z_i$ and such that $z_i \in B \rightarrow z_{i+1} \in B$. Moreover, this sequence can be found in time polynomial in $\|V\| \times \max\{|x| : x \in V\}$*

Proof. We prove this by induction. The base case is trivial. For any two strings x_i and x_j in V, either $x_i \preceq_f x_j$ or $x_j \preceq_f x_i$ or both, and in time polynomial in $\|V\| \times \max\{|x_i|, |x_j|\}$ we can decide which of the three cases hold. Now we play a *knock-out tournament* among the n strings, where we say that x beats y if $x \preceq_f y$. (If $x \preceq_f y$ is established, then $y \preceq_f x$ is not examined, so a draw may end in an arbitrary winner, which is o.k., since in that case $x \in B$ iff $y \in B$.) Let x_0 be the winner of the tournament. If $x_0 \in B$, then $x_i \in B$ for all i, and if there is an i such that $x_i \notin B$, then $x_0 \notin B$. By induction hypothesis, the set of $n - 1$ strings $V - \{x_0\}$ can be ordered into a chain $\{x_1, \ldots, x_{n-1}\}$, such that $x_i \preceq_f x_{i+1}$. Then $x_0 \preceq_f x_1 \preceq_f \ldots \preceq_f x_{n-1}$ is the chain searched for. $\Box$

Let *PSEL* stand for the class of all p-selective sets. Though *PSEL* has a nonempty intersection with every (polynomial-time) Turing degree, and therefore contains highly non-computable sets, one can easily see that *EXP* has non-p-selective sets. Maybe less trivial, the class *EXP* has *PSEL*-immune sets as well as P-immune sets, as we show first.

Theorem 14 *There is a PSEL immune set in EXP*

Proof. Let $\{f_i\}_i$ be an enumeration of all polynomial time computable selector functions. We construct a set A in exponential time that is infinite, yet has no infinite subset in *PSEL*. The set A is constructed in stages. At stage n we define the strings in A of length n. We start at stage 0 with $A = \emptyset$.

stage n:

Let m be the largest integer $\leq n$ such that $n^m + m < 2^n$. Without loss of generality we assume that $f_i(x, y)$ can be computed in time $n^m + m$ for any pair of strings x, y of length n and any $i \leq m$. Sort the 2^n strings $x_1, \ldots, x_{2^n}$ according to the first m selector functions. Obtaining an ordered set $x_{i_1}, \ldots, x_{i_{2^n}}$ for selector function f_i such that for any set B with f_i as it's selector it holds that $x_{i_{j+1}} \in B \rightarrow x_{i_j} \in B$. Now put in A all strings of length n except x_{i_1} for $i = 1, \ldots m$

end of stage n

As $2^n > n \geq m$, at least one string is put into A at each stage, hence $\|A\| = \infty$. On the other hand if B is a p-selective set, that has f_j as its selector function, and strings of length $\geq j$ then for such a string x, there is a string y in B with $|x| = |y|$ and $y \notin A$ by construction. □

P-selective sets of arbitrary sparseness can be constructed. In fact, it is much easier to construct a sparse p-selective set than to construct a dense p-selective set. For sparse p-selective sets instead of sparse polynomial-time computable sets, our theorems all go through. The reason for this is that although we cannot compute which strings are in the p-selective set, we can compute which strings are certainly not. Suppose we know that S contains k strings of a certain length. If we are given $k + 1$ strings then we know by Lemma 13 that one of these strings has the property that if it is in S then all of the other strings are also in S. We conclude that this string cannot be in S. We use this property to prove the theorems for p-selective sets. The proofs of the following theorems run along the same lines as the proofs of the corresponding theorems in the previous sections. In each of the cases we construct a set W from which the reduction from K to $A - S$ is reconstructed. We diagonalize (if necessary) against reductions that produce the same query set on too many different inputs of the form $<i, x, z>$. So for enough different inputs of the form $<i, x, z>$ we produce a set of queried strings that is more dense than S. By chain-ordering these queries, we find which strings must certainly lie outside S. These strings are then used to code K into W.

Theorem 15 *For any set A that is $\leq_c^P$-hard for EXP and any $p(n)$-sparse p-selective set S, the set $A - S$ remains $\leq_c^P$-hard for EXP.*

Proof. Let f be a p-selector for S. Let $\ell(i, x, z) = p(|<i, x, z>|^i + i)$ be the maximal number of strings that can be in S up to the maximal length of strings that can be queried by machine M_i on input $<i, x, z>$. We construct set an exponential time computable set W consisting of strings of the form $<i, x, z>$.

As before, we take care that membership of $<i, x, z>$ in W is computable in time $O(2^{|<i,x,z>|})$ by deciding a priori that $<i, x, z>$ can only enter W if time $2^{|<i,x,z>|}$ is sufficient to compute $Q_i(<i, x, z>)$ for $z = 1, \ldots, \ell(i, x, z)$, to compute $f(x, y)$ for all the pairs of queries in the union of these querysets, and to do a polynomial (depending on i, x and z) amount of processing on the side. For fixed i, this may mean for a finite number of x that $<i, x, z> \notin W$ for any z.

Let $U(i, x) = \bigcup_{z \leq \ell(i,x,z)+1} Q_i(<i, x, z>) = \{u_1, u_2, \ldots, u_{m+1}\}$. Without loss of generality, assume that $U(i, x) \neq \emptyset$. Otherwise, $Q_i(<i, x, 1>) \subseteq A$ and so $M_i^A(<i, x, 1>)$ accepts if M_i is a conjunctive truth-table reduction and we let $<i, x, 1> \notin W$, diagonalizing against the possibility that M_i is the conjunctive reduction from W to A. Without loss of generality, we also assume that $u_{i+1} \in S \rightarrow u_i \in S$, for $i = 1, \ldots, m$. Now let z_0 be the least z such that $Q_i(<i, x, z>) \cap \{u_1, \ldots, u_{\ell(i,x,z)}\} \subseteq \bigcup_{z' \neq z \wedge z \leq \ell(i,x,z)} Q(i, x, z)$ let $<i, x, z_0>$ in W iff $x \in K$ and $<i, x, z> \in W$ for $z \neq z_0$ and $z \leq \ell(i, x, z)$. This ends the construction of W.
Since W is computable in exponential time, there is a conjunctive truth-table reduction, say M_w that witnesses this reduction. We claim that $W^{[w]}$ reduces to $A - S$.

1. As mentioned above there is a finite number of cases for which the string $<w, x, z>$ is not in W for reasons of insufficient computation time. In these cases, or in the case where $z > \ell(w, x, z) + 1$, our reduction produces a fixed string not in $A - S$ (It cannot reject without

querying the oracle, since it is supposed to be a conjunctive truth-table reduction.)

2. Otherwise, compute $\bigcup_{z \leq \ell(i,x,z)+1} Q_w(<w,x,z>) = \{u_1, u_2, \ldots, u_{m+1}\}$, and

   (a) find out if z is the least z' s.t. $Q_w(<w,x,z'>) \cap \{u_1, \ldots, u_{\ell(i,x,z)}\} \subseteq \bigcup_{z' \neq z \wedge z \leq \ell(i,x,z)} Q(w,x,z)$. If this is the case then the set of queries of the new reduction consists of the strings that are not in S, namely, $Q_i(<w,x,z>) \cap \{u_{\ell(i,x,z)+1}, \ldots, u_{m+1}\}$.

   (b) Otherwise produce the empty set as queryset (i.e. accept).

This reduction accepts $<w,x,z>$ iff $<w,x,z> \in W^{[w]}$.
In Case 1, $<w,x,z> \notin W^{[w]}$ by definition. Likewise, in Case 2b, $<w,x,z> \in W^{[w]}$ by definition. The only remaining case is Case 2a, but here we know that $<w,x,z> \in W^{[w]}$ iff $M_w^A(<w,x,z>)$ accepts, iff $Q_w(<w,x,z>) \subseteq A$. But $(Q(w,x,z) \cap \{u_1, \ldots, u_{\ell(i,x,z)}\}) \subseteq A$ since all these strings are members of a queryset that is a subset of A, and the remaining set of strings, $Q(w,x,z) - \{u_1, \ldots, u_{\ell(i,x,z)}\}$ is queried. Hence, our reduction accepts iff $M_w^A(<w,x,z>)$ accepts iff $<w,x,z> \in W^{[w]}$. Furthermore, the reduction queries no strings in S since if one of the queried strings were in S then so would $u_1, \ldots, u_{\ell(i,x,z)}$ and this would violate the sparseness condition on S.

Finally, the reduction from K to $W^{[w]}$ is the following. If x belongs to the finite number of exceptional cases then we find out if $x \in K$ by table-lookup and produce a fixed string in $W^{[w]}$ or outside of $W^{[w]}$ accordingly. Otherwise, we compute the least z with $Q_w(<w,x,z'>) \cap \{u_1, \ldots, u_{\ell(i,x,z)}\} \subseteq \bigcup_{z' \neq z \wedge z \leq \ell(i,x,z)} Q(w,x,z)$, and let $f(x) = <w,x,z>$. □

For more or less the same reasons as in the previous subsection the following theorem also holds

Theorem 16 *For any set A that is $\leq_d^P$-hard for EXP and any $p(n)$-sparse p-selective set S, the set $A - S$ remains $\leq_d^P$-hard for EXP.*

Finally the construction for two-truth-table complete sets can be carried through for p-selective sets.

Theorem 17 *Let A be $\leq_{2-tt}^P$-hard for EXP and S a p-selective $p(n)$-sparse set. The set $A - S$ is still $\leq_{2-tt}^P$-hard for EXP.*

Proof. (Sketch) We construct the set W analogously to the proof of Theorem 9. We will not repeat the entire proof, but only highlight the important parts. Let f be the p-selector for S. W consists of strings $<i,x,z>$. Where $<i,x,z> \notin W$ for those x where time $2^{|<i,x,z>|}$ is insufficient to compute $Q(<i,x,z>)$ for $z = 1,\ldots, 2 \times \binom{p(|<i,x,z>|^i+i)}{2} + p(|<i,x,z>|^i + i) + 1$ plus the time needed to sort these strings with the selector f.

For other strings $<i,x,z>$, we compute the sets $Q(<i,x,z>)$ if $z \in \{1,\ldots, 2\times \binom{p(|<i,x,z>|^i+i)}{2} + p(|<i,x,z>|^i+i)+1\}$, and we define $<i,x,z> \notin W$, for all other z. Let $\{u_1,\ldots,u_{m+1}\}$ be the union of these query-sets, again such that $u_{i+1} \in S \rightarrow u_i \in S$ for $i = 1,\ldots m$. We assume that this union is non-empty. Otherwise, we diagonalize against machine M_i. Suppose that $Q_i(<i,x,z>) = \{u_{k(z)}, u_{\ell(z)}\}$. Now there are three possible cases:

1. There is a z s.t. $k(z) > p(|<i,x,z>|^i+i))$ and $\ell(z) > p(|<i,x,z>|^i+i)$. In this case, we let $\{i,x,z_0\} \in W$ iff $x \in K$ for z_0 the least such z. For all other z we define $\{i,x,z\} \notin W$.

2. There are three strings $<i,x,w>$ that produce the same set of queries $\subseteq \{u_1,\ldots,u_{p(|<i,x,z>|^i+i)}\}$. In this case, we diagonalize against M_i in exactly the same way as in the proof of Theorem 9.

3. There are two strings $<i,x,z>$ and $<i,x,z'>$ with $z \neq z'$ such that there is a string $u_{k(z)}$ with $k(z) \leq p(|<i,x,z>|^i+i)$ and $Q_i(<i,x,z>) \cap Q_i(<i,x,z'>) = \{u_{k(z)}\}$ Furthermore, both $\ell(z) > p(|<i,x,z>|^i + i)$ and $\ell(z') > p(|<i,x,z>|^i + i)$. In this case (assume that z' is the smaller one of the two), we follow the steps of the proof of Theorem 9 either fixing membership in A for $u_{k(z')}$ or making membership in A of $u_{k(z')}$ and $u_{\ell(z')}$ equivalent, or diagonalizing against M_i if this is possible. We let $<i,x,z> \in W$ iff $x \in K$.

Since W is exponential-time computable there is some $\leq^P_{2-tt}$ reduction of W to A and from this reduction and the construction rules for W we construct a reduction that queries only strings outside S, i.e. strings in $\{u_{p(|<i,x,z>|^i+i)+1},\ldots,u_{m+1}\}$.

The reduction from K to $W^{[w]}$ on input x recomputes a polynomial, depending only on w and p, number of strings of $W^{[w]}$. and finds the first z such that $<w,x,z> \in W \leftrightarrow x \in K$. $f(x) = <w,x,z>$ for this z. $\square$

4 Splittings of *EXP* Complete Sets

In this section we want to investigate to what extent one can *split EXP*-complete sets. A splitting of an r.e. (*EXP*) set is the construction of two r.e. (*EXP*) sets $A_0, A_1 \subseteq A$, such that $A_0 \bigcap A_1 = \emptyset$ and $A_0 \bigcup A_1 = A$. One of the things to look at is, can this splitting be done so that the subsets both have the same information as A? For complete sets this would mean that the complete set can be split into subsets, that are itself again complete. This type of question has been studied in a recursion theoretical setting by Ladner [Lad73]. He observed that there exist sets that are non-splittable, or non-*mitotic* as he called them. The recursion theoretical definition is as follows:

Definition 18 *An r.e. set A is called (m-)mitotic iff there exist r.e. sets A_1 and A_0 such that*

1. $A_1 \subseteq A,\ A_0 \subseteq A,\ A_1 \bigcap A_0 = \emptyset,\ A_1 \bigcup A_0 = A$
2. $A \equiv_{(m)T} A_1 \equiv_{(m)T} A_0$

Note here that Point 2 in the definition can be weakened, in the case of $\leq_T$-reductions, to $A_1 \equiv_T A_0$. To see this note that $A \leq_T A_1 \oplus A_0$ and $A_1 \oplus A_0 \leq_T A_1$. To reduce A_1 to A, the reduction queries on input x whether x is in A. If this is not the case it rejects straight out. Otherwise, it starts enumerating A_1 and A_0, since x must be in one of them.

Later, Ambos-Spies [AS84a] studied the complexity theoretical variant of mitotic sets and introduced the term *p-mitotic* sets. It is not clear how to define mitoticity in the complexity theoretical setting. Ambos-Spies introduced four definitions; two for the Turing reductions and two for the many-one reductions. One option is to change Point 2 in Definition 18 into demanding $A \equiv_T^P A_1 \equiv_T^P A_0$. Since we are interested in complete sets for complexity classes, we could demand that A_0 and A_1 should be in the complexity class under consideration. A problem is, that this definition can not be weakened to $A_1 \equiv_T^P A_0$. (Take for example $\Sigma^* = A \bigcup \overline{A}$ for some *EXP* Turing complete set A. Now A as well as $\overline{A}$ are both in *EXP* and split Σ^* and are Turing equivalent but obviously do not Turing reduce to Σ^*.)

Ambos-Spies chose a Breidbart/Owings [Bre78, Owi75] type of splitting (*by* another set):

Definition 19 *A recursive set A is p-m(T)-mitotic if there is a set $B \in P$ such that $A \equiv_{m(T)}^P A \bigcap B \equiv_{m(T)}^P A \bigcap \overline{B}$.*

When using this definition, the problem of reducing A_1 to A is settled for the Turing case. Namely, x is in A_1 iff x is in B and x is in A. A disadvantage of this definition however is that the requirement that the splitting has to be polynomial-time computable seems too strong. In order to capture this feeling, we also want to look at the definition discussed above. Note here also that since our main interest is in complete sets, we will not have the trouble that A_0 (or A_1) does not reduce to A. (This is because A is complete.)

Definition 20 *An r.e. set A is called weakly-p-m(T) mitotic iff there exist r.e. sets A_1 and A_0, such that*

1. $A_1 \subseteq A,\ A_0 \subseteq A,\ A_1 \bigcap A_0 = \emptyset,\ A_1 \bigcup A_0 = A$
2. $A \equiv^P_{m(T)} A_1 \equiv^P_{m(T)} A_0$

One of the questions that arise is, are $\leq^P_m$-complete sets for *EXP* (weakly) p-m mitotic? In order to answer this question, we first take a look at the r.e. complete sets. There it is known, due to Myhill, that the $\leq_m$-complete sets are all isomorphic. Now using the fact that K, the standard r.e. $\leq_m$-complete set, is m-mitotic and that this property is preserved under isomorphisms, it follows that all $\leq_m$-complete sets are m-mitotic. Unfortunately it is not known whether the $\leq^P_m$-complete sets for *EXP* are p-isomorphic, but it is known that they are all 1-1, length increasing equivalent [Ber77, GH89, Wat87]. This now is sufficient to prove that they are weakly-p-m-mitotic.

Theorem 21 *All $\leq^P_m$-complete sets for EXP are weakly-p-m-mitotic.*

Proof. Let A be a $\leq^P_m$-complete set for *EXP*. Let $A \oplus A \leq^P_m A$ via f that is 1-1 and length increasing. Set $A_0 = \{y : \exists 0x[x \in A \wedge y = f(0x)]\}$. Since f is 1-1 and length increasing A_0 is in *EXP*. It is also $\leq^P_m$-complete, because $x \in A$ iff $f(0x) \in A_0$. Now set $A_1 = A - A_0$. Then $A_0 \bigcup A_1 = A$ and $A_0 \bigcap A_1 = \emptyset$. It remains to show that A_1 is also $\leq^P_m$-complete. Let $A^1 = \{1x : x \in A\}$. Note that $A^1 \subseteq A \oplus A$, and is $\leq^P_m$-complete. Because f is 1-1 : $1x \in A^1 \Rightarrow f(1x) \in A - A_0$, and $1x \notin A^1 \Rightarrow 1x \notin A \oplus A \Rightarrow f(1x) \notin A \Rightarrow f(1x) \notin A - A_0$. □

For *EXP*, the 1-1 length increasing property seems to be enough to get weak p-m-mitoticity. For *NEXP* the situation is somewhat different, because we do not know whether we have the length increasing property for complete sets. We do however have the 1-1 property and the fact that the reductions are not more than exponential length decreasing, i.e. $2^{|f(x)|} > |x|$. (The precise term here is 'exponentially honest' [GH89].) The main problem however is that when applying the same proof as above, the set difference used to define $A_1 = A - A_0$ is not known to be in *NEXP*, because it is not known whether *NEXP* is closed under complementation (and thus under set difference). We *can* prove something that at a first glance looks hopeful in order to prove weakly-p-m-mitoticity for *NEXP*-complete sets.

Theorem 22 *Every $\leq_m^P$-complete set A for NEXP can be split into infinitely many disjoint subsets $A_1, A_2 \ldots$ such that $\bigcup_{i=0}^{\infty} A_i = A$, such that for all i, $A_i \subseteq A$ and A_i is complete for NEXP.*

Proof. We start the same way as in the *EXP* case. Let $A \oplus A \leq_m^P A$ via f that is 1-1 and exponentially honest. Set $A_0 = \{y : \exists\, 0x[y = f(0x) \wedge y \in A]\}$. Note that it is equivalent, in the definition of A_0, to say that $x \in A$ or $y \in A$, because f is a many-one reduction. Now A_0 is in *NEXP* : on input y guess the $0x$ such that $f(0x) = y$. This can be done in nondeterministic exponential time because $|x| < 2^{|y|}$, by the exponential honesty of f. Now accept y iff $y \in A$. We define A_1 in a similar way. $A_1 = \{y : \exists\, 1x[y = f(1x) \wedge y \in A]\}$. We now have two complete sets A_0 and A_1 and some leftover of A, namely $T_0 = A - (A_0 \bigcup A_1)$. At this point we repeat this procedure with A_0 resulting in A_{00} and A_{01} and again some leftover T_1. Repeating this we get an infinite sequence of sets A_{0^l} and a set $T = \bigcup_{i=0}^{\infty} T_i$ so that $(\bigcup_{i=0}^{\infty} A_{0^l}) \bigcup T = A$. Since T is countable (it is a subset of ω), we can add the i^th element of T to A_{0^l} resulting in a sequence $A'_{0^i 1}$ satisfying the properties of the theorem. $\square$

Although this looks hopeful, the following example shows that the infinite version of mitoticity can be independent of mitoticity. Ladner [Lad73] showed the existence of *non-mitotic* sets. Together with the following observation this yields the somewhat bizarre existence of a set that cannot be split into two parts, but can be split in infinitely many parts of the same complexity.

Observation 23 *every r.e. set A can be split into infinitely many disjoint r.e. subsets $A_1, A_2, \ldots$ of A such that they remain in the same Turing degree as A.*

Proof. It is well known that every r.e. set A has an infinite subset B that is recursive. Let B be such an infinite recursive subset of A. Since

B is recursive and infinite, it is (recursively) isomorphic to Σ^*. So we can code A into B. I.e., let f be the isomorphism between Σ^* and B, define $A' = \{f(x) \mid x \in A\}$. Obviously A' is r.e and $A' \equiv_T (A - B)$. Furthermore there exists a T_1 such that $A = (A - B) \bigcup A' \bigcup T_1$. Now using the same "divide and split" trick as in the previous theorem we get the desired sequence of subsets. □

We try to follow the same line now as Ladner [Lad73], and try to prove that there exist non (weakly-p-m) mitotic sets in *EXP*. We succeed in this and can also proof that those sets can be $\leq^P_{3-tt}$-complete. (Note that this also proves that the same result is true for p-m mitoticity.)

Theorem 24 *There exists a set A in EXP that is not weakly-p-m-mitotic and $\leq^P_{3-tt}$-complete.*

Proof. In order to prove this, we prove the following: There exists a set A, so that for all sets $A_0, A_1 \in EXP$ that split A, $A_0 \not\equiv^P_m A_1$. It is enough to prove this for *EXP* computable splittings (i.e. A_i is in *EXP*) to avoid weak p-m-mitoticity. This is because for a weakly-p-m-mitotic set B in *EXP*, the parts B_0 and B_1 are in *EXP*, because they are $\equiv^P_m$ B. Let $\{M_i\}_i$ be an enumeration of exponential time machines that run in time $2^{n^i} + i$, and let $\{f_j\}_j$ be an enumeration of polynomial-time many one reductions.

To construct A, we have requirements for all $n = <i,j,k,\ell>$: If $L(M_i)$ and $L(M_j)$ split A, then either $L(M_j) \not\leq^P_m L(M_i)$ via f_ℓ or $L(M_i) \not\leq^P_m L(M_j)$ via f_k.

We introduce a function b to have a set of strings to diagonalize over. Let $b(0) = 1$ and $b(i) = b(i-1)^{i-1} + 1$. The idea is then, to put three copies of K into A. and make sure that at least one of the pairs $<i,x>$ $(i = 1,2,3)$ is in A. A is then $\leq^P_{3-tt}$(in fact $\leq^P_{3-d}$)-complete by the following reduction from K to A: $x \in k$ iff $<i,x>(i = 1,2,3)$ in A. On the other hand we can leave out at most 2 of the pairs $<i,x>$ to prevent A from becoming a mitotic set.

Construction:

stage 0: $A = \emptyset$

stage n:

Let $n = <i,j,k,\ell>$ and $b = b(n-1)^{n-1} + 1$. We will make sure that only strings of length $> b$ are defined in A at this stage. All strings of length $\leq b$ in A will not be changed by this stage and because $b(i)$ is length increasing, not by later stages. Consider $f_k(0^{b(n)}) = y_0$ and $f_\ell(0^{b(n)}) = y_1$. Set A to be $A \bigcup \{<i,x> \mid i = 1,2,3 \text{ and } x \in K\}$. Without loss of generality, we assume for this proof that the pairing function does not output strings that start with a 0. Then, we can define strings of the form $0^{b(i)}$ outside A without disturbing the completeness. We have several cases to consider.

1. $y_0 = 0^{b(n)}$ or $y_1 = 0^{b(n)}$ in this case we put $0^{b(n)}$ into A.

2. $y_0 = y_1$, and not Case 1. We have two subcases:

   (a) $|y_0| > b$. Leave y_0 out of A and put $0^{b(n)}$ in A.

   (b) $|y_0| \leq b$. Put $0^{b(n)}$ in A iff $y_0 \notin A$

3. $y_0 \neq y_1$, and not Case 1. We have 3 subcases:

   (a) $|y_0| > b$ and $|y_1| > b$. Leave both y_0 and y_1 out of A and put $0^{b(n)}$ in A.

   (b) $|y_0| > b$ and $|y_1| \leq b$. ($|y_0| \leq b$ and $|y_1| > b$ is treated analogously) We leave y_0 out of A and have two subcases:

      i. $y_1 \notin A$. Put $0^{b(n)}$ in A.

      ii. $y_1 \in A$. $0^{b(n)} \in A$ iff $M_j(y_1)$ accepts.

   (c) $|y_0| \leq b$ and $|y_1| \leq b$. We have 3 subcases.

      i. $y_1 \notin A$ and $y_0 \notin A$. Put $0^{b(n)}$ in A.

      ii. $y_1 \in A$ and $y_0 \notin A$. (The other way around is symmetric) $y_1 \in A$ iff $M_j(y_1)$ accepts.

      iii. $y_1 \in A$ and $y_0 \in A$. $0^{b(n)} \in A$ iff $M_i(y_0)$ accepts *and* $M_j(y_1)$ accepts.

end of stage n
End of Construction

This ends the construction of A. The proof of correctness of the construction is an analysis of the cases in the construction. First A is exponential-time computable. A consists of strings $<i,x>$ where $x \in K$. Possible membership of these strings in A can be decided because K is exponential-time computable. To decide membership of other strings y in A we need to compute f_i on input $0^{b(n)}$ for $i \leq n$, and $b(n) \leq |y|$ This can be done in exponential time since $b(n)^i + i < 2^{b(n)}$ for $n \geq 3$.

Next, if A were m-mitotic, then there exists a pair i, j such that A is split into A_0 and A_1 and $A_0 \leq_{m1}^{A}$ via f_i and $A_1 \leq_{m0}^{A}$ via f_j. Moreover $A_0 \in EXP$

as is witnessed by machine M_k and $A_1 \in EXP$ as is witnessed by machine M_ℓ.

The cases of the construction can be split in 2 types: namely the type where y_0 and y_1 are free i.e. $> b$. In those cases we can diagonalize by putting $0^{b(n)}$ in A and leaving both y_1 and y_0 out of A. Since $0^{b(n)} \in A$ it has to be either in A_0 or in A_1, but if it is in either one then the corresponding reduction fails since it's image is not in A (and therefore certainly not in the other part of A). In these cases we diagonalize directly against the many-one reductions. In the other cases we are forced to leave y_1 and y_0 in A in order not to destroy the the work done at previous stages. But in these cases we are able to compute in exponential time the splittings we want to diagonalize against. We will show the correctness of Case 3(c)iii in the construction. The other cases have a similar proof. In this case, both y_0 and y_1 are fixed and in A. By putting $0^{b(n)}$ in A, we force y_0 and y_1 for any *possible correct* splitting both to be in either in A_0 or A_1. Yet, the machines M_k and M_ℓ witness that y_0 and y_1 are in A_0 and A_1 respectively. On the other hand by leaving $0^{b(n)}$ out of A, y_1 has to be in A_i and and $y1$ in A_{1-i} ($i = 0, 1$). Since one of the machines rejects in this case, the other machine has to witness that y_0 and y_1 are both in A_0 or both in A_1. If, for instance M_k witnesses that both strings are in A_0, then, since $0^{b(n)} \notin A_1$ reduction f_j fails to be a reduction from A_1 to A_0. $\square$

The next logical step would be to prove this result for Turing complete sets. We are not yet able to do this but suspect that there exist Turing complete sets that are not weakly-p-T-mitotic.

Another line of splittings in recursion theory is the existence of a splitting of an r.e. set A in A_0 and A_1 that are incomparable. Examples of this are the splitting theorem of Sacks [Sac63] and the time bounded versions by Ladner [Lad75]. The next theorem is in a way a counterpart to this.

In the original splittings one gets the following structure: A_0 and A_1 are Turing (or many one) incomparable but do reduce to A, thus achieving that A does not reduce to A_0 or A_1. I.e. A_0 and A_1 are strictly below A. In the next theorem the sets A_0 and A_1 are strictly below A, but are in the same many one degree. Seen in another light, this theorem can be seen as a generalization of the fact that there exists $\leq^P_{2-d}$-complete sets for EXP that are not $\leq^P_m$-complete [Wat87, GH89].

Theorem 25 *If A is $\leq^P_m$-complete for EXP, then A can be split into A_0 and A_1, such that:*

- $A_0 \equiv_m^P A_1$
- *A_0 and A_1 are $\leq_{2-d}^P$-complete for EXP but not $\leq_m^P$-complete.*

Proof. Let A be $\leq_m^P$-complete and K be the standard $\leq_m^P$-complete set. Since the $\leq_m^P$-complete sets for EXP are 1-1 length increasing equivalent, we can construct the following length increasing 1-1 function h from A to A. Let f be the 1-1, l.i. reduction from A to K and g the one from K to A. Let $h(x) = f(g(x))$. We say that x is a *root* if $h^{-1}(x)$ is undefined and x is *on a chain* if $h^{-1}(x)$ is defined.

One possible way to construct A_0 and A_1 is as follows (the real construction follows later): $A_0 = \{x \mid x \in A \text{ and } x \text{ is a root }\} \bigcup \{x \mid x \in A \text{ and } x \text{ is on a chain and } h^i(x_r) = x \text{ and } x_r \text{ is a root and } i \text{ is even }\}$ and $A_1 = \{x \mid x \in A \text{ and } x \text{ is on a chain and } h^i(x_r) = x \text{ and } x_r \text{ is a root and } i \text{ is odd }\}$

Clearly A_0 and A_1 split A, are in EXP and $A_0 \equiv_m^P A_1$ via h. A_0 and A_1 are $\leq_{2-d}^P$-complete: $x \in A \iff (x \in A_0 \text{ or } x \in A_1) \iff (x \in A_0 \text{ or } h(x) \in A_0)$. The only thing to do now is to show that A_0 and A_1 are not $\leq_m^P$-complete. Note here that in order to get the above properties (i.e. splitting of A, the $\equiv_m^P$ and the $\leq_{2-d}^P$-completeness) it doesn't matter if the roots are in A_0 or A_1 as long as it holds that: $x \in A_i$ then $h(x) \in A_{1-i}$. This gives enough freedom to diagonalize against $\leq_m^P$-reductions. We are going to construct a set $W \in EXP$ so that W not $\leq_m^P A_0$. Note that then A_1 can not be $\leq_m^P$-complete either. Let $\{f_i\}_i$ be an enumeration of $\leq_m^P$-reductions such that f_i runs in time n^i. We also need a function $b(n)$ to denote the set of strings to diagonalize against f_n. Let $b(0) = 1$ and $b(i+1) = b(i)^i + 1$. W is going to be a subset of 0^*.

We construct W, A_0, and A_1 in stages, such that elements of A_0 and A_1 are either roots or successive elements on a chain, and W is exponential time computable, but reducible to neither A_0 nor A_1.
At stage 0, $W = A_0 = A_1 = \emptyset$
stage n:

We have constructed W, A_0 and A_1 up to strings of length $b(n-1)^{n-1} + 1$. We simulate $f_n(0^{b(n)}) = y$. We now have three cases for the construction of W:

1. $|y| \leq b(n-1)^{n-1}$. Put $0^{b(n)}$ in W iff $y \notin A_0$.
2. $\exists\, y' \leq b(n-1)^{n-1}$ s.t. $h^i(y') = y$ and i is even. Put $0^{b(n)}$ in W iff $y' \notin A_0$.
3. Otherwise put $0^{b(n)}$ in W.

This ends the construction of W.
Construction of A_0 and A_1.
Let $b(n-1)^{n-1} < |x| < b(n)^n + 1$ and $x \in A$.

1. $x \neq y$
   (a) x is a root.
      - $\exists\ i$ s.t. $h^i(x) = y$
         - put x in A_0 iff i is odd.
         - put x in A_1 iff i is even.
      - put x in A_0

   (b) x is on a chain, x_r is the root of x and $h^i(x_r) = x$. Put x in A_0 if i is even and $x_r \in A_0$ otherwise put x in A_1.

2. $x = y$. Put x in A_1.

end of stage n

It is not hard to see that W, A_0 and A_1 are all in EXP. Furthermore A_0 and A_1 split A and $A_0 \equiv_m^P A_1$. It remains to be shown that $W \not\leq_m^P A_0$. Suppose it does via reduction f_j, then $f_j(0^{b(j)}) = y$. For Cases 1 and 2 in the above construction, $0^{b(j)} \in W$ iff $y \notin A_0$ and from the construction of A_0 and Case 3, $0^{b(j)} \in W$ and $y \notin A_0$. □

Acknowledgements. We would like to thank Peter van Emde Boas for discussions about the results. Furthermore we would like to thank Steven Homer, Dick de Jongh and Jan van Neerven for discussions and ideas. Finally we would like to thank Stephen Wilcox for pointing out an error that we thought was grammatical, but that turned out to be mathematical.

References

[AS84a] K. Ambos-Spies. On the structure of polynomial time degrees. In M. Fontet and K. Mehlhorn, editors, *STACS 84, Lecture Notes in Computer Science 166*, pages 198–208. Springer-Verlag, 1984.

[AS84b] K. Ambos-Spies. p-mitotic sets. In E. Börger, G. Hasenjäger, and D. Roding, editors, *Logic and Machines, Lecture Notes in Computer Science 177*, pages 1–23. Springer-Verlag, 1984.

[BDG88] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* Springer-Verlag, 1988.

[Ber76] L. Berman. On the structure of complete sets: Almost everywhere complexity and infinitely often speedup. *Proc. 17th IEEE Symposium on Foundations of Computing*, pages 76–80, 1976.

[Ber77] L. Berman. *Polynomial Reducibilities and Complete Sets.* PhD thesis, Cornell University, 1977.

[BH77] L. Berman and H. Hartmanis. On isomorphisms and density of NP and other complete sets. *SIAM J. Comput.*, 6:305–322, 1977.

[BHT91] H. Buhrman, S. Homer, and L. Torenvliet. On complete sets for nondeterministic classes. *Math. Systems Theory*, 24:179–200, 1991.

[Bre78] S. Breidbart. On splitting recursive sets. *J. Comput. System Sci.*, 17:56–64, 1978.

[BST91] H. Buhrman, E. Spaan, and L. Torenvliet. Bounded reductions. In *STACS 1991, Lecture Notes in Computer Science 480*, pages 410–421. Springer Verlag, 1991. Will appear in Complexity Theory, Cambridge University press 1993.

[GH89] K. Ganesan and S. Homer. Complete problems and strong polynomial reducibilities. In B. Monien and R. Cori, editors, *STACS 89, Lecture Notes in Computer Science 349*, pages 240–250, 1989.

[Har78] J. Hartmanis. On the logtape isomorphism of complete sets. *Theoretical Computer Science*, 7:273–286, 1978.

[Joc68] C.G. Jockusch. Semirecursive sets and positive reducibility. *Trans. of the Amer. Math. Soc.*, 131:420–436, 1968.

[Jon75] N. Jones. Space bounded reducibilities among combinatorial problems. *J. Comput. System Sci.*, 11:68–85, 1975.

[Kar72] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.

[Lad73] R. Ladner. Mitotic recursively enumerable sets. *J. Symbolic Logic*, 38(2):199–211, 1973.

[Lad75] R. Ladner. On the structure of polynomial time reducibility. *J. Assoc. Comput. Mach.*, 22:155–171, 1975.

[LLS75] R. Ladner, N. Lynch, and A. Selman. A comparison of polynomial time reducibilities. *Theoret. Comput. Sci.*, 1:103–123, 1975.

[Owi75] J.C. Owings. Splitting a context-sensitive set. *J. Computer and System Sciences*, 10:83–87, 1975.

[Sac63] G.E. Sacks. On degrees less than $\mathbf{0}'$. *Ann. of Math.*, 2(77):211–231, 1963.

[Sch86] U. Schöning. Complete sets and closeness to complexity classes. *Math. Systems Theory*, 19:24–41, 1986.

[Sel79] A. Selman. P-selective sets, tally languages, and the behavior of polynomial time reducibilities on NP. *Math. Systems Theory*, 13:55–65, 1979.

[TFL93] S. Tang, B. Fu, and T. Liu. Exponential time and subexponential time sets. *Theoretical Computer Science*, 115(2), july 1993.

[Wag86] K. Wagner. More complicated questions about maxima and minima and some closures of NP. In *13th ICALP, Lecture Notes in Computer Science 226*, pages 434–443, 1986.

[Wat87] O. Watanabe. A comparison of polynomial time completeness notions. *Theoret. Comput. Sci.*, 54:249–265, 1987.

[Wil85] C.B. Wilson. Relativized circuit complexity. *J. Comput. System Sci.*, 31:169–181, 1985.