SPLITTINGS, ROBUSTNESS, AND STRUCTURE OF COMPLETE SETS[*]

HARRY BUHRMAN[†], ALBRECHT HOENE[‡], AND LEEN TORENVLIET[§]

Abstract. We investigate the structure of *EXP*-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. We study for various types of reductions the question of whether the set difference $A - S$ for a hard set A and a sparse set S is still hard. We also address the question of which complete sets A can be split into sets A_1 and A_2 such that $A \equiv_r^P A_1 \equiv_r^P A_2$ for reduction type r, i.e., which complete sets are *mitotic*. We obtain both positive and negative answers to these questions depending on the reduction type and the structure of the sparse set.

Key words. reductions, mitoticity, completeness, complexity

AMS subject classifications. 03D15, 03D20, 68Q15

PII. S0097539795279724

1. Introduction. The structure of complete sets under various types of reductions is a well-studied subject in complexity theory. The question "Which sets can be complete under which type of reductions?" has been posed many times and answered for many complexity classes. (See [9] for an overview.)

A complete set represents, through the reduction under which it is complete, an entire complexity class. A membership algorithm for the complete set combined with a reduction (of the appropriate type) gives a membership algorithm for a set in the class. Viewed as such, the complete set contains the information of any particular set in the class.

In this paper we investigate the structure of the information that makes the set complete for deterministic exponential time and for various types of reductions. We take an approach pioneered by Schöning [21]. To investigate the structure of a complete set we compare this set with another set. Schöning in particular showed that for a $\leq_m^P$-complete set A in *EXP* and a polynomial time computable set D, the set $A \Delta D$ is of exponential density. Tang, Fu, and Liu took in [23] the approach of taking the difference of sets complete in exponential time with a sparse subexponential time computable (sub)set and asked the question of whether the resulting set is complete. In section 3, we extend their work by studying analogous questions for other types of completeness. We obtain similar results and show that for several types of reductions and arbitrary sparseness conditions there exists a *single* subexponential time computable set S that meets this sparseness condition and, furthermore, has the property that for any set A, complete under the reduction for which this S is constructed, the set $A - S$ is no longer complete. In addition, it seems possible to make S "almost

polynomial time computable." It follows from the constructions that, by taking a sufficiently slow enumeration of reductions, we can lower the time complexity of S to any reasonably behaved superpolynomial function. We also address the flip question in this section: "For which reductions and which sparse sets is it the case that the set $A - S$ is complete?" We study sparse sets S of simple *structure* rather than computationally simple sets. Selman [22] introduced p-selective sets as a resource bounded analog of semirecursive sets introduced by Jockusch [13]. For any tally set, a polynomial time Turing equivalent p-selective set can be found. Therefore p-selective sets can be computationally very complex. Nonetheless, 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 (or actually no less likely) to be in the set. It turns out that for most of the reductions studied in subsection 3.1 the set $A - S$ remains complete if S is a p-selective set. We prove this theorem for disjunctive, conjunctive, and 2-truth-table (2-tt) reductions. One might expect that, as is the case with many results on reductions, our 2-truth-table theorem can be extended to at least bounded truth-table reductions. From a recent result of Buhrman, Fortnow, and Torenvliet [6] it follows, however, that the 2-tt result cannot be extended even to 3-tt.

In section 4 we study sets that remain complete even when another *complete* set is removed, i.e., we study sets that can be "split" into two or more sets that are again complete. Such sets have been studied in recursion theory and are called *mitotic* sets. We follow the line of Ambos-Spies [1] and prove that $\leq_m^P$-complete sets for EXP indeed are (weakly p-m) mitotic. In contrast, we show that there exists a $\leq_{3\text{-}tt}^P$-complete set that is not weakly p-m mitotic. Finally, we show a counterpart to Ladner's splitting theorems [16], i.e., we construct a set that can be split into two parts that are strictly below the degree of complete sets and that are $\equiv_m^P$ instead of incomparable.

2. Definitions and notation. We assume that the reader is familiar with standard notions in structural complexity theory as they are defined, e.g., in [2]. 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 notation 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, $\leq_{k\text{-}tt}^P$ 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, $\leq_{k\text{-}d}^P$ for k-disjunctive truth-table reductions, and $\leq_T^P$ for Turing reductions. The symbols $\leq_m$ and $\leq_T$ also appear in the paper, without the superscript P, to indicate the version of these reductions without time bound.

Various definitions for these types of reductions can be found in the literature, e.g., in [17, 7, 5, 8]. We think of polynomial-time bounded reductions as being modeled by adaptive and nonadaptive oracle machines. We use various enumerations $\{M_i\}_i$ of (oracle) Turing machine programs with varying properties. If the type of machine is not clear from the context, we explicitly mention the machine type. An enumeration $\{M_i\}_i$ can thus stand for an enumeration of all polynomial-time bounded machines in one theorem and an enumeration of all bounded truth-table reductions in the next. In the case of enumeration of many-one reductions, i.e., where the machines are transducers, we also use $\{f_i\}_i$ to emphasize this fact. For polynomial-time bounded machines 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 of strings are denoted by capital letters and are subsets of Σ^*, where $\Sigma = \{0, 1\}$. Strings are denoted as small letters $x, y, u, v, \ldots$. The length of a string x is denoted by $|x|$.

We assume *pairing and projection functions* that are easy to compute. 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 ith coordinate. We assume all kinds of convenient properties of these pairing functions, e.g., pairing functions can be designed such that $<x, y> \; < \; <x, z>$ whenever $y < z$ or such that $|<x, z>| = |<x, z'>|$ for a large (exponential in this length) number of pairs z and z'. In fact, the specific properties of different pairing functions may depend on the context of the proof in which they are used. Integer numbers can also appear as arguments to pairing functions. If so, the integer is identified with its binary representation.

An ordering on Σ^* is assumed where $x < y$ if $|x| < |y|$ and that coincides with the lexicographical ordering if $|x| = |y|$. The cardinality of a set A is denoted as $\|A\|$. 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. Following Kelly [14], we let the notation $\bigcup\{S : S \text{ meets condition }\}$ stand for the union of all sets S that meet the given condition. For sets A and B the notation $A \oplus B$ stands for the disjoint union of A and B, i.e., $\{0x : x \in A\} \cup \{1x : x \in B\}$.

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

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., $EXP = \bigcup\{DTIME(2^{n^i}) : i \in \omega\}$. 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 of which sparse sets can be removed from exponential-time complete sets of different types without disturbing the completeness of these sets. The question originates from work of Schöning [21], who showed that for every $\leq_m^P$-hard set A for EXP and every set D in P, the set $A \Delta D$ is of exponential density. In [23], Tang, Fu, and Liu showed, as a corollary to an analogous result on parity reductions, that even for subexponential time computable D, the difference $A \Delta D$ remains $\leq_m^P$-hard for exponential time.

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 choosing 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 b_i are chosen sufficiently far apart.

A closer look at the proof shows although the theorem just states that S is subexponential time computable, that there are various ways of making S come arbitrarily close to polynomial time. It therefore seems reasonable to ask whether we can also choose S to be polynomial-time computable. The answer to this question is negative, as observed in [23]. 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 queries just the set $\{<x,y> | 0 \leq y \leq p(2n) + 1\}$. Since all of 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 the theorem "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 at most $\|S^{\leq n}\|$ strings for each length n. Since we want the construction of S to meet any given sparseness condition, this implies that the number of queries cannot be a function of n that grows to infinity with n. In other words, the number of queries must be some fixed 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 nonadaptive, we can obtain the theorem for bounded Turing reductions ($\leq_{bT}$).

THEOREM 3.1. *Let g be a recursively computable nondecreasing function with* $\lim_{n \mapsto \infty} g(n) = \infty$, *and let f be a function 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 [25].

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 put just $\leq i$ 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 $S = \{y : \exists i[y \in Q_i(0^{b(i)}) \wedge |y| > b(i)]\}$ 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 in $\bigcup \{Q_j(0^{b(j)}) : j \leq i\}$ 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 $0^{b(i)}$. It follows that $L_{A-S} \leq_{btt}^{P} A - S$. Let M_j be the reduction from L_{A-S} to $A - S$. Then $Q_j(0^{b(j)}) \cap (A - S) \subseteq \Sigma^{\leq b(j)}$ and therefore $0^{b(j)} \in L_{A-S}$ iff $M_j^{A-S}(0^{b(j)})$ rejects, which is a contradiction. □

If we change S in the construction above to $\{\min\{y : y \in Q_i(0^{b_i}) \wedge |y| > b_i\} : i \in \omega\}$, we get the following corollary from the same construction.

COROLLARY 3.2. *Given a recursively computable nondecreasing 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 (c-tt) 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 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 [23]. We isolate and prove this property in the following lemma.

LEMMA 3.3. *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 from B to A such that $Q_j(x) \not\subseteq \bigcup\{Q_j(y) : y \in B \wedge y < x\}$.*

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 $Q_i(<i,x>) \subseteq \bigcup\{Q_i(<i,y>) : <i,y> < <i,x> \wedge <i,y> \in W\}$, then we let $<i,x> \notin W$; otherwise $<i,x> \in W$ 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 $Q_j(<j,x>) \not\subseteq \bigcup\{Q_j(<j,y>) : <j,y> \in W \wedge <j,y> < <j,x>\}$. By assumption on the pairing function that $<i,x> < <i,y> \leftrightarrow x < y$ the function $M'(x) = M_j(<j,x>)$ computes a $\leq_c^{P}$-reduction with the required property. □

From this lemma we get the following theorem.

THEOREM 3.4. *Let $g(n)$ be a recursively computable nondecreasing function 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 $g(n)$ 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 $\bigcup\{Q_i(<0^{b(i)}, y>) : |y| \leq b(i)\}$ 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 be only 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 $\leq_c^{P}$-hard. If it were, then one of the $\leq_c^{P}$-reductions from $D = \{<x,y> : x \in \{0\}^*, y \in \Sigma^*\}$ to A would behave as predicted by Lemma 3.3.

Let M_j be such a reduction. It follows that $\|\bigcup\{Q_j(<0^{b(j)}, y>) : |y| \leq b(j)\}\| \geq 2^{b(j)}$ from Lemma 3.3. Hence there is one string in this set of length $\geq b(j)/2 - 1$.

The least of these strings z_0 is not in $A - S$ by construction, and M_j must reject any input $<0^{b(j)}, y>$ for which z_0 is in $Q_j(<0^{b(j)}, y>)$. Yet $<0^{b(j)}, y> \in D$, so M_j cannot be a reduction from D to $A - S$. $\square$

Conjunctive and disjunctive truth-table reducibilities are kind of each other's complement. If a set is conjunctive truth-table reducible to a set A, then its complement is disjunctive truth-table reducible to $\overline{A}$. So we find the following.

COROLLARY 3.5. *Let* $g(n)$ *be a recursively computable nondecreasing function 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.*

In addition we have the following.

COROLLARY 3.6. *Let* $g(n)$ *be a recursively computable nondecreasing function 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

$$x \in K \quad \Leftrightarrow \quad Q_i(x) \subset A,$$
$$\text{or} \quad x \notin K \quad \Leftrightarrow \quad Q_i(x) \cap \overline{A} \neq \emptyset.$$

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

$$\text{So} \quad x \in \overline{K} \quad \Leftrightarrow \quad f(x) \in K \quad \Leftrightarrow \quad Q_i(f(x)) \subset A,$$
$$\text{or} \quad x \notin \overline{K} \quad \Leftrightarrow \quad f(x) \notin K \quad \Leftrightarrow \quad Q_i(f(x)) \cap \overline{A} \neq \emptyset,$$
$$\text{or} \quad x \in K \quad \Leftrightarrow \quad f(x) \notin K \quad \Leftrightarrow \quad Q_i(f(x)) \cap \overline{A} \neq \emptyset,$$

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). $\square$

3.2. Easy sparse sets. As in the case of $\leq_m^P$-complete sets, we can let the time complexity of the set S in the previous subsection come arbitrarily close to polynomial time. For some—but surprisingly not all—reductions, a polynomial-time computable sparse set that destroys completeness does not exist. To show this we take a slightly more general view of the complexity of the set S. Instead of taking S polynomial-time computable, we let S be p-selective. p-selective sets were introduced by Selman [22] as a resource bounded analogue of semirecursive sets, which were introduced by Jockusch [13].

DEFINITION 3.7. *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. $\chi_A(f(x, y)) = \max\{\chi_A(x), \chi_A(y)\}$.

The following "ordering lemma" is used in the proofs of the theorems in this section and can be found (in various forms) in, e.g., [24, 10, 11].

LEMMA 3.8. *Let* $V = \{v_1, \ldots, v_n\}$ *be a finite set of strings, and let* A *be a* p-selective *set with* p-selector f. *The strings in* V *can be ordered "according to* f*" as* $v_{i_1}, \ldots, v_{i_n}$ *such that* $v_{i_j} \in A \to v_{i_{j+1}} \in A$ *in time polynomial in* $|v_1| + \cdots + |v_n|$.

As only polynomial time is involved, we sometimes assume a finite set "ordered according to a p-selector," without loss of generality.

For any (nonfinite) sparseness condition a p-selective set (which may be infinite) that meets this condition can be constructed. For polynomially sparse p-selective

sets we can show counterparts to the theorems of subsection 3.1. For conjunctive, disjunctive, and 2-tt reductions, for a set A that is complete under this reduction, and for a set S that is polynomially sparse and p-selective, the set $A - S$ remains complete under the same reduction. Since all sets in P are p-selective, the same results follow for sparse polynomial-time computable sets.

It might be instructive here to compare our results to those of Tang, Fu, and Liu [23, Theorems 3.6 and 3.7]. They show that sets complete under conjunctive and disjunctive reductions stay complete when a sparse subexponential-time computable *subset* is removed. Our sparse set may be *any* set.

THEOREM 3.9. *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. We construct an exponential time-computable set W such that, for some index w of a conjunctive truth-table reduction M_w from W to A, it holds that the universal set K is many-one reducible to $W^{[w]}$ and $W^{[w]}$ is in turn conjunctive truth-table reducible to $A - S$. This then establishes completeness of $A - S$.

The set W consists of strings $<i, x, z>$ for $i \in \omega$, $x \in \Sigma^*$, and z ranging from $0^{|x|}$ through $1^{|x|}$. Fix i and x and let $n = |<i, x, x>|$. By appropriate assumptions on the pairing function, all of the pairs $<i, x, z>$ are of the same length. So, since S is $p(n)$ sparse, there exists some polynomial q such that at most $q(n)$ of the different strings queried on such an input can be in S. Let Z be the set of the lexicographically first $q(n) + 1$ strings in the interval $0^{|x|}$ through $1^{|x|}$. We describe membership of $<i, x, z>$ in W. If $z \notin Z$, then $<i, x, z> \notin W$. Let $U = \bigcup \{Q_i(<i, x, z>) : z \in Z\}$. Assume that $U = \{u_1, u_2, \ldots, u_m\}$ such that $u_i \in S \to u_{i+1} \in S$ and let $U' = \{u_{m-q(n)+1}, \ldots, u_m\}$. If $U = \emptyset$ or time 2^n is insufficient to compute U and sort U according to the p-selector, then no string $<i, x, z>$ is in W. It will follow from the construction that the first case cannot occur if i is a program that computes a c-tt reduction from W to A. The latter case can only occur for finitely many different x.

Now there are two cases.

Case 1. There is a $z \in Z$ such that $Q_i(<i, x, z>) \subseteq \bigcup \{Q_i(<i, x, z'>) : z' \in Z \wedge z' \neq z\}$. Let z_0 be the least such z. We let $<i, x, z> \in W \iff z \neq z_0$.

Case 2. There is no such z. Then $\|Q_i(<i, x, z>) - \bigcup \{Q_i(<i, x, z'>) : z' \in Z \wedge z' < z\}\| \geq 1$ for all $z \in Z$ and hence, there is a z such that $Q_i(<i, x, z>) - U' \neq \emptyset$ and $Q_i(<i, x, z>) \cap U' \subseteq \bigcup \{Q_i(<i, x, z'>) : z' \in Z \wedge z' < z\}$. Let z_0 be the least such z. We let $<i, x, z_0> \in W \leftrightarrow x \in K$. We let $<i, x, z> \in W$ for all $z < z_0$ in Z and let $<i, x, z> \notin W$ for all $z > z_0$ in Z.

This ends the construction of W.

Since W is computable in exponential time, there is a conjunctive truth-table reduction from W to A. Let M_w be such a reduction. We show a reduction from $W^{[w]}$ to $A - S$. Let $<w, x, z>$ be some input.

1. As mentioned above, there is a finite number of cases for which the string $<w, x, z>$ is not in W because of insufficient computation time. In these cases, or in the case where z is not among the first $q(|<w, x, x>|) + 1$ strings, our reduction rejects. (Or, equivalently, produces some fixed string not in $A - S$.)

2. Otherwise, compute the set $U = \{u_1, \ldots, u_m\}$ as described above. (Since w is a fixed constant, this can be done in polynomial time in $|<w, x, z>|$.) Since M_w is a reduction from W to A it follows that the construction falls in Case 2. If z is the least string in Z satisfying Case 2, then produce $Q_w(<w, x, z>) - U'$. Otherwise accept or reject according to Case 2.

This reduction accepts $<w, x, z>$ iff $<w, x, z> \in W^{[w]}$ and queries only strings in $A - S$.

The reduction from K to $W^{[w]}$ is the following. If x belongs to the finite number of exceptional cases, then we find out by table-lookup if $x \in K$ and accept or reject accordingly. Otherwise, we compute U for this w and x and the least z satisfying Case 2 and let $f(x) = <w, x, z>$. $\square$

In the proof of this theorem we defined a set W such that the reduction of W to A must query strings that are in A on certain inputs. We did this by defining certain strings in W (hence a conjunctive reduction must get the answer yes on queries produced on this input). On the other hand, the reduction of W to A must query some strings outside S on some inputs. We did this by diagonalizing against reductions for which the cardinality of the union of the query sets on the inputs $<i, x, z>$ was small enough to "fit" inside S. Then a reduction from any exponential time-computable set to $A - S$ can be constructed, using W, by assuming that queries that may be in S are answered yes, and querying the remaining ones.

For $\leq_d^P$-hard sets, we can change the strategy of the proof to construct a set W such that we know that queries that may be in S are answered *negatively* (by leaving enough strings *out* of W). A completely analogous proof then yields the following.

THEOREM 3.10. *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.*

By complementation Theorems 3.9 and 3.10 yield the following corollaries, respectively.

COROLLARY 3.11. *For any set A that is $\leq_d^P$-hard for EXP and any $p(n)$-sparse p-selective set S, the set $A \cup S$ remains $\leq_d^P$-hard for EXP.*

COROLLARY 3.12. *For any set A that is $\leq_c^P$-hard for EXP and any $p(n)$-sparse p-selective set S, the set $A \cup S$ remains $\leq_c^P$-hard for EXP.*

We can also prove the same theorem for $\leq_{2\text{-}tt}^P$-hard sets.

THEOREM 3.13. *Let A be $\leq_{2\text{-}tt}^P$-hard for EXP and let S a p-selective $p(n)$-sparse set. The set $A - S$ is still $\leq_{2\text{-}tt}^P$-hard for EXP.*

Proof. We will construct an exponential time-computable set W that is used in defining a $\leq_{2\text{-}tt}^P$-reduction from K to $A - S$. Let f be the p-selector for S. W consists of strings $<i, x, z>$, where $|z| = |x|$.

Fix some i and x. Let $n = |<i, x, x>|$. We will define membership of $<i, x, z>$ in W for all z. For this proof we will assume that $\|Q_i(x)\| = 2$ for all i and all x.

There exists some polynomial q' such that S can contain at most $q'(n)$ of the strings in $\bigcup\{Q_i(<i, x, z>) : |z| = |x|\}$. There exists some polynomial q'' such that S can contain at most $q''(n)$ different *pairs* of strings in $\bigcup\{Q_i(<i, x, z>) : |z| = |x|\}$. We let $q(n) = q'(n) + q''(n) + 1$. It holds for all but finitely many x that $\|\{z : |z| = |x|\}\| > q(n)$ and $q''(|x|) > 2q'(|x|) + 1$. (Note that q'' is quadratic in q'.)

In the remainder of this proof we will consider only strings for which both inequalities hold. (We will use tabular lookup in the reduction for other strings.) Let Z consist of the lexicographically first $3 \times q(n)$ strings of length $|x|$. Only strings $<i, x, z>$ with $z \in Z$ may enter W. Let $U = \bigcup\{Q_i(<i, x, z>) : z \in Z\}$. Assume $U = \{u_1, \ldots, u_m\}$, where $u_i \in S \rightarrow u_{i+1} \in S$, and let $U' = \{u_{m-q'(n)+1}, \ldots, u_m\}$. Note that $U \cap S \subseteq U'$.

There are two cases.

Case 1. $\|U\| < q''(n)$.

As there are more than $3 \times q''(n)$ strings in Z it must hold that there are $z_1 \neq z_2 \neq z_3$ in Z such that $Q_i(<i, x, z_1>) = Q_i(<i, x, z_2>) = Q_i(<i, x, z_3>)$. Consider the truth table(s) produced on this input. By setting $<i, x, z>$ in
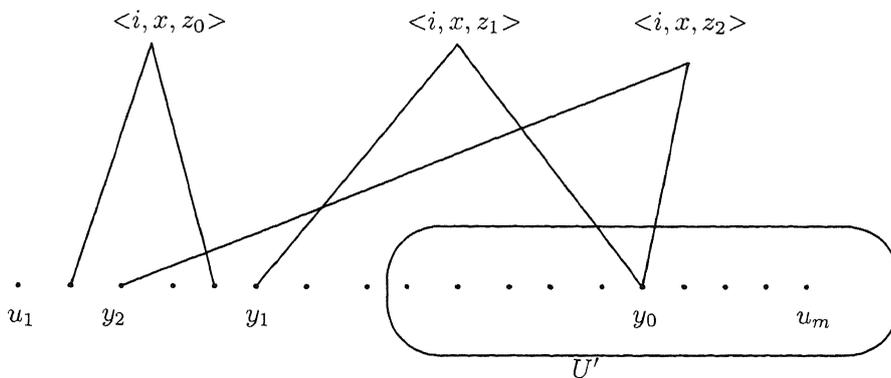
or out of W we can eliminate at least half of the possible settings (element of A or not) of the two queries in $Q_i(<i,x,z_1>)$. As there are only four possible settings for these two strings, repeating this procedure for z_2 and z_3, we eliminate all possible settings, thereby forcing the fact that M_i is not a reduction from W to A.

Case 2. $\|U\| \geq q''(n)$. (It then follows that $\|U\| \geq 2 \times q'(n) + 1$.)

Either of the following holds (see also Figure 3.1):

Case 2a. There is a z in Z such that $Q_i(<i,x,z>) \cap U' = \emptyset$. Let the least such z be z_0. We let $<i,x,z_0>$ be in W iff $x \in K$ and let $<i,x,z> \notin W$ for all other z.

Case 2b: There is no such z. Then for each of the $\geq q'(n) + 1$ strings u_j in $U - U'$ there is a $z(u_j)$ in Z such that $\|Q_i(<i,x,z(u_j)>) \cap U'\| = \|Q_i(<i,x,z(u_j)>) \cap U - U'\| = 1$. It follows that there is a pair z_j, z'_j such that $\|Q_i(<i,x,z_j>) \cap Q_i(<i,x,z'_j>) \cap U'\| = 1$. Without loss of generality, we assume that z_1 and z_2 are minimal with this property and $z_1 < z_2$. Let

$$\{y_0\} = Q_i(<i,x,z_1>) \cap Q_i(<i,x,z_2>) \cap U',$$
$$\{y_1\} = Q_i(<i,x,z_1>) \cap U - U',$$
$$\{y_2\} = Q_i(<i,x,z_2>) \cap U - U'.$$

Then we know that $\{y_1,y_2\} \cap S = \emptyset$. In the following we will set, depending on the truth table computed by M_i on input $<i,x,z_1>$, the string $<i,x,z_1>$ in W such that $<i,x,z_2> \in W$ can be computed from the queries y_1 and y_2 and hence from queries to $A - S$.

We let $<i,x,z_2> \in W \leftrightarrow x \in K$. In the case we are considering here, only the strings $<i,x,z_1>$ and $<i,x,z_2>$ may enter W. So $<i,x,z> \notin W$ for all $z \in Z - \{z_1,z_2\}$.

Let α be the truth table computed by M_i on input $<i,x,z_1>$. There are sixteen possible truth tables that fall into four cases.

   i. If $\alpha \equiv 1$, we let $<i,x,z_1> \notin W$ and if $\alpha \equiv 0$, we let $<i,x,z_1> \in W$. This prohibits M_i from being a 2-truth-table reduction of W to A. This covers two of the sixteen cases.

   ii. If there is only *one* combination of $a,b \in \{0,1\}$ for which $\alpha(a,b) = 1$, then $<i,x,z_1> \in W$, and if there is only *one* such combination where $\alpha(a,b) = 0$, then $<i,x,z_1> \notin W$. If this M_i is a 2-truth-

table reduction from W to A, then for this particular input we can compute $\chi_A(y_0)$ and $\chi_A(y_1)$ in polynomial time. This covers eight of the sixteen cases.

iii. If α is in fact a 1 truth table then the value of α depends on either y_0 or y_1. In the first case we put $<i, x, z_1>$ in W if and only if α takes on *true* if the entry for y_0 is fixed to *false*. Otherwise we let $<i, x, z_1> \in W$ iff $x \in K$.

iv. In the remaining two cases, α is either $\chi_A(y_0) = \chi_A(y_1)$ (equality) or $\chi_A(y_0) \oplus \chi_A(y_1)$ (parity). In the first case we let $<i, x, z_1> \in W$ and in the second case we let $<i, x, z_1> \notin W$ (then in both cases $y_0 \in A \Leftrightarrow y_1 \in A$.)

This completes the construction of W.

We will now show that we can use W to compute a $\leq_{2\text{-}tt}^P$-reduction for any set in EXP to $A - S$. Since W is exponential-time computable there is some $\leq_{2\text{-}tt}^P$-reduction of W to A. Let M_w be the machine that computes this reduction. For any set B in EXP let f_B be the many-one reduction of B to K. The $\leq_{2\text{-}tt}^P$-reduction from B to $A - S$ works as follows.

On input y, first compute $f_B(y)$. Let $x = f_B(y)$. Now use w and x to compute the set Z as above. Compute U, sort it according to the p-selector, find U', and find out which of the two cases (Case 2a or Case 2b) holds. (We have already argued that Case 1 cannot occur if w is the reduction from W to A.) In Case 2a, find the least z such that $Q_w(w, x, z) \cap U' = \emptyset$ and produce $Q_w(w, x, z)$ as a query set. In Case 2b find z_1 and z_2. We are in one of the subcases ii–iv. In subcases ii and iii we have either fixed the value of $\chi_A(y_0)$ or this value doesn't influence the value of the truth table, so we need only query y_1. Then we can compute membership of $x \in K$ from either the truth table produced by M_w on input $<i, x, z_1>$ or $<i, x, z_2>$, respectively. In subcase iv we fixed things such that $\chi_A(y_0) = \chi_A(y_1)$. We can query y_1 and y_2 and use these values as $\chi_A(y_0)$ and $\chi_A(y_2)$ in the truth table computed by M_w on input $<w, x, z_2>$. □

As every set in P is p-selective, we note the following corollary.

COROLLARY 3.14. *Theorems 3.9, 3.10, and 3.13 also hold when "p-selective" is replaced by "polynomial-time computable."*

We notice an interesting phenomenon here. In a recent paper, Buhrman, Fortnow, and Torenvliet [6] proved the existence of a 3-tt complete set in EXP that is not b-tt autoreducible. (Recall that a set A is r autoreducible if $A \leq_r^P A$ via a reduction that does not query its input.) Inspection of the proof shows that the set is constructed by diagonalizing against autoreductions on inputs in the set $\{0^{b(n)} : n \in \omega\}$, where $b(n)$ is some suitably chosen gap function. They prove that a 3-tt complete set A can be constructed such that every b-tt reduction (from A to A) that does not query its input must, for some n, incorrectly compute membership of $0^{b(n)}$ in A. Without essentially changing the proof, $b(n)$ can be chosen such that $\{0^{b(n)} : n \in \omega\}$ is a polynomial-time computable sparse set. The following corollary follows immediately from their proof.

COROLLARY 3.15 (see [6]). *There exists a 3-tt complete set A in EXP and a sparse set S in P such that $A - S$ is not b-tt hard for EXP.*

So Theorem 3.13 states an optimal result. A set that is not b-tt hard may of course still be Turing hard. Hence the corollary above hence does not rule out that every Turing-complete set may remain Turing complete when a sparse set is removed. However, we can show that such a result can only be obtained by nonrelativizing methods. We 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. We suspect the following to be a folk theorem. We prove it here just for completeness.

LEMMA 3.16. *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.

The fact that there exists a tally set T that is hard for EXP gives us a truth-table reduction, say, by machine M_i from K to T. Now fix n and consider all strings of length n. Without loss of generality, 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 jth string of length n in lexicographic order. Let $P_j = \{<a_1, \ldots, a_{n^i}> : $ when a_i used as answer to query y_i then $M_i(x_j)$ accepts iff $x_j \in K\}$.

P_j codes exactly those tally sets T' that, when used as an 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 ith 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 $K \leq_{tt}^{P} T'$. $\square$

THEOREM 3.17. *There exists an oracle A, such that 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 [26] showed the existence of an oracle A where $EXP^A \subseteq P^A/poly$. Using this oracle, together with Lemma 3.16, 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. $\square$

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 $A_0 \cap A_1 = \emptyset$ and $A_0 \cup A_1 = A$. One of the questions to look at is: "Can this splitting be done so that both subsets have the same information as A?" For complete sets this would mean that the complete set can be split into subsets that are themselves again complete. This type of question has been studied in a recursion theoretical setting by Ladner [15]. He observed that there exist sets that are nonsplittable, or non-*mitotic*, as he called them. The recursion theoretical definition is as follows.

DEFINITION 4.1. *An r.e. set A is called mitotic iff there exist r.e. sets A_1 and A_0 such that*

1. $A_1 \subseteq A$, $A_0 \subseteq A$, $A_1 \cap A_0 = \emptyset$, $A_1 \cup A_0 = A$.
2. $A \equiv_T A_1 \equiv_T A_0$.

If additionally $A \equiv_m A_1 \equiv_m A_0$, then A is called m-mitotic.

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 or not 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.

Ambos-Spies [1] 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 4.1 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 cannot 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 both A and $\overline{A}$ are in EXP, split Σ^*, and are Turing equivalent but obviously do not Turing reduce to Σ^*.)

Ambos-Spies chose a Breidbart–Owings [4, 19] type of splitting (*by* another set).

DEFINITION 4.2 (see [1]). *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 4.3. *An r.e. set A is called weakly p-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. $A \equiv_T^P A_1 \equiv_T^P A_0$.

If additionally $A \equiv_m^P A_1 \equiv_m^P A_0$, then A is called weakly p-m mitotic.

One of the questions that arise is: "Are $\leq_m^P$-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 [18], that all of the $\leq_m$-complete sets are 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 $\leq_m^P$-complete sets for EXP are p isomorphic, but it is known that they are all 1-1 length increasing equivalent [3, 12, 25]. This is sufficient to prove that they are weakly p-m mitotic.

THEOREM 4.4. *All $\leq_m^P$-complete sets for EXP are weakly p-m mitotic.*

Proof. Let A be a $\leq_m^P$-complete set for EXP. Let $A \oplus A \leq_m^P A$ 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 $\leq_m^P$-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 $\leq_m^P$-complete. Let $A^1 = \{1x : x \in A\}$. Note that $A^1 \subseteq A \oplus A$ and is $\leq_m^P$-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 is 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" [12].) 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 first glance looks hopeful in order to prove weakly p-m mitoticity for $NEXP$-complete sets.

THEOREM 4.5. *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, $A_i \subsetneq A$ 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 $0x$ 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 have some leftover T_1. The process of repeatedly splitting the set A_{0^ℓ} thus results in an infinite sequence of sets $(A_{0^\ell 1})_{\ell \in \omega}$ and a set $T = \bigcup_{i=0}^{\infty} T_i$ so that $(\bigcup_{\ell=0}^{\infty} A_{0^\ell 1}) \bigcup T = A$. Since T is countable (it is a subset of ω), we can add the ith element of T to $A_{0^i 1}$ resulting in a sequence $A_{0^i 1}'$ satisfying the properties of the theorem.   □

Although this looks hopeful, the following example shows that the infinite version of mitoticity can be independent of mitoticity. Ladner [15] showed the existence of *nonmitotic* 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 into infinitely many parts of the same complexity.

OBSERVATION 4.6. *Every infinite 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 infinite 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 and define $A' = \{f(x) \mid x \in A\}$. Obviously A' is an infinite r.e. set 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 follow the same line as Ladner [15] and try to prove that there exist non-(weakly p-m) mitotic sets in *EXP*. We succeed in this and can also prove that those sets can be $\leq_{3-tt}^P$-complete. (Note that this also proves that the same result is true for p-m mitoticity.)

THEOREM 4.7. *There exists a set A in EXP that is not weakly p-m mitotic and $\leq_{3-tt}^P$-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, $A_0 \neq_m^P A_1$. First note that if $A_i \notin EXP$, then $A_i \neq_m^P A$ so, in the construction, we may assume that both A_0 and A_1 are recognized by exponential-time bounded machines. Or, equivalently, only pairs of exponential-time bounded machines represent candidate splittings of A against which we have to diagonalize.

Let $\{M_i\}_i$ be an enumeration of exponential-time machines that run in time 2^{n^i}, and let $\{f_k\}_k$ be an enumeration of polynomial-time many-one reductions computable in time n^k.

To construct A, we have requirements for all $n = \langle i, j, k, \ell \rangle$: if $L(M_i)$ and $L(M_j)$ split A, then either $L(M_j) \not\leq_m^P L(M_i)$ via f_ℓ or $L(M_i) \not\leq_m^P L(M_j)$ via f_k.

We introduce a function b having a set of strings to diagonalize on. Let $b(0) = 1$ and $b(i) = b(i-1)^{i-1} + 1$. At each stage n we will also define a number d_{n+1} such that if $n + 1 = \langle i', j', k', \ell' \rangle$, i.e., the *next* stage will diagonalize against ma-

chines $M_{i'}$ and $M_{j'}$ and functions $f_{k'}$ and $f_{\ell'}$, then d_{n+1} is some number satisfying $b(n+1)^{1/\max\{i',j'\}} > d_{n+1} > \max\{k', l'\} \times \log b(n+1)$. We will assume appropriate properties on the pairing function such that the interval from d_n to d_{n+1} is suitable for the construction. All strings in A with length between d_n and d_{n+1} will be defined at stage n.

The idea is 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 $\leq^P_{3\text{-}tt}$(in fact $\leq^P_{3\text{-}d}$)-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 two of the pairs $<i,x>$ to prevent A from becoming a mitotic set.

Construction:

stage 0: $A = \{<i,x> \mid i = 1,2,3$ and $x \in K\}$. This is the "base set A." At any subsequent stage n of the construction, at most two strings y_0 and y_1, respectively, with $d_n \leq |y_0|, |y_1| < d_{n+1}$ and the string $0^{b(n)}$ will be added to or removed from A. A will be decided for all strings of length $\leq d_{n+1}$ after stage n.

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 $0^{b(i)}$ in or outside A without disturbing the completeness.

stage n:

Let $n = <i,j,k,\ell>$. This stage will diagonalize against the possibility that $A = L(M_i) \cup L(M_j)$ and/or that $L(M_i) \equiv^P_m L(M_j)$ via the functions f_k and f_ℓ Consider $f_k(0^{b(n)}) = y_0$ and $f_\ell(0^{b(n)}) = y_1$.

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) $|y_0| > d_n$. Leave y_0 out of A and put $0^{b(n)}$ in A.
   (b) $|y_0| \leq d_n$. Put $0^{b(n)}$ in A iff $y_0 \notin A$

3. $y_0 \neq y_1$, and not case 1. We have three subcases:
   (a) $|y_0| > d_n$ and $|y_1| > d_n$. Leave both y_0 and y_1 out of A and put $0^{b(n)}$ in A.
   (b) $|y_0| > d_n$ and $|y_1| \leq d_n$. ($|y_0| \leq d_n$ and $|y_1| > d_n$ 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) $|y_0| \leq d_n$ and $|y_1| \leq d_n$. We have three 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) $0^{b(n)} \in A$ 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 $f_\ell(0^{b(n)})$ where $d_n \leq |y|$. This can be done in (linear) exponential time since $|y| \geq \max\{\ell,k\} \times \log b(n)$. Finally, to decide membership of $0^{b(n)}$ in A we sometimes need to compute $M_j(y)$ and/or $M_k(y)$ on some input y of length $\leq d_n$. This can be done in linear exponential time since $b(n) \geq d_n^{\max\{i,j\}}$.

Next, if A were m mitotic, then there exists a pair k,ℓ such that A is split into A_0 and A_1 and $A_0 \leq_m^P A_1$ via f_k and $A_1 \leq_m^P A_0$ via f_ℓ. Moreover, $A_0 \in EXP$ as is witnessed by machine M_i and $A_1 \in EXP$ as is witnessed by machine M_j.

The cases of the construction can be split into two types, namely, the type where both y_0 and y_1 are free, i.e., $> d_n$, and the type where either y_0 or y_1 is forced ($\leq d_n$). In the "free" cases we 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 in either A_0 or A_1, but if it is in either one then the corresponding reduction fails since its 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 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 3c(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 A_0 or A_1. Yet, the machines M_i and M_j 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_p and and y_0 in A_{1-p} ($p = 0, 1$). 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 A_1. If, for instance, M_i witnesses that both strings are in A_0, then, since $0^{b(n)} \notin A_1$, reduction f_ℓ 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 and T mitoticity. This question remains open for further research.

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 [20] and the time bounded versions by Ladner [16]. 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 $\leq_{2-d}^P$-complete sets for EXP that are not $\leq_m^P$-complete [25, 12].

THEOREM 4.8. *If A is $\leq_m^P$-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, length increasing reduction from A to K, and let g be 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 cannot be $\leq_m^P$-complete either. Again let $\{f_i\}_i$ be an enumeration of $\leq_m^P$-reductions such that f_i runs in time $n^i + i$. We also need a function $b(n)$ to denote the set of strings to diagonalize on. 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 $\leq b(n-1)^{n-1}$. 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. $\exists \, y' \leq b(n-1)^{n-1}$ such that $h^i(y') = y$. Put $0^{b(n)}$ in W iff $y' \notin A_{i \bmod 2}$.
> 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.
>   - $\exists \, i$ such that $h^i(x) = y$
>   - 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, $x_r \in A_j$, and $h^i(x_r) = x$. Put x in
> $A_{(i+j) \bmod 2}$.
> 2. $x = y$. Put x in A_0 iff $0^{b(n)} \notin W$.

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 is 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 $y \notin A_0$. $\square$

REFERENCES

[1] K. AMBOS-SPIES, *On the structure of polynomial time degrees*, in Proc. Symposium on Theorectical Aspects of Computer Science, Lecture Notes in Comput. Sci. 166, M. Fontet and K. Mehlhorn, eds., Springer-Verlag, Berlin, 1984, pp. 198–208.

[2] J. BALCÁZAR, J. DÍAZ, AND J. GABARRÓ, *Structural Complexity* I, Springer-Verlag, Berlin, 1988.

[3] L. BERMAN, *Polynomial Reducibilities and Complete Sets*, Ph.D. thesis, Cornell University, Ithaca, NY, 1977.

[4] S. BREIDBART, *On splitting recursive sets*, J. Comput. System Sci., 17 (1978), pp. 56–64.

[5] H. BUHRMAN, *Resource Bounded Reductions*, Ph.D. thesis, University of Amsterdam, 1993.

[6] H. BUHRMAN, L. FORTNOW, AND L. TORENVLIET, *Using autoreducibility in separating complexity classes*, in Proc. 36th Annual IEEE Symposium on Foundations of Computer Science, Milwaukee, WI, IEEE Computer Society Press, Los Alamitos, CA, 1995, pp. 520–527.

[7] H. BUHRMAN, S. HOMER, AND L. TORENVLIET, *On complete sets for nondeterministic classes*, Math. Systems Theory, 24 (1991), pp. 179–200.

[8] H. BUHRMAN, E. SPAAN, AND L. TORENVLIET, *The relative power of logspace and polynomial time reductions*, Comput. Complexity, 3 (1993), pp. 231–244.

[9] H. BUHRMAN AND L. TORENVLIET, *On the structure of complete sets*, in Proc. 9th Annual IEEE Conference on Structure in Complexity Theory, Amsterdam, Holland, IEEE Computer Society Press, Los Alamitos, CA, 1994, pp. 118–133.

[10] H. BUHRMAN, L. TORENVLIET, AND P. V. EMDE BOAS, *Twenty questions to a p-selector*, Inform. Process. Lett., 48 (1993), pp. 201–204.

[11] H. BUHRMAN, AND L. TORENVLIET, *P-selective self-reducible sets: A new characterization of P*, J. Comput. System Sci., 53 (1996), pp. 210–217.

[12] K. GANESAN AND S. HOMER, *Complete problems and strong polynomial reducibilities*, SIAM J. Comput., 21 (1992), pp. 733–742.

[13] C. JOCKUSCH, *Semirecursive sets and positive reducibility*, Trans. Amer. Math. Soc., 131 (1968), pp. 420–436.

[14] J. KELLY, *General Topology*, D. van Nostrand Company Inc., New York, 1955.

[15] R. LADNER, *Mitotic recursively enumerable sets*, J. Symbolic Logic, 38 (1973), pp. 199–211.

[16] R. LADNER, *On the structure of polynomial time reducibility*, J. Assoc. Comput. Mach., 22 (1975), pp. 155–171.

[17] R. LADNER, N. LYNCH, AND A. SELMAN, *A comparison of polynomial time reducibilities*, Theoret. Comput. Sci., 1 (1975), pp. 103–123.

[18] J. Myhill, *Creative sets*, Zeit. Math. Log. Grund. Math., 1 (1955), pp. 97–108.

[19] J. OWINGS, *Splitting a context-sensitive set*, J. Comput. System Sci., 10 (1975), pp. 83–87.

[20] G. SACKS, *On degrees less than 0'*, Ann. of Math., 2 (1963), pp. 211–231.

[21] U. SCHÖNING, *Complete sets and closeness to complexity classes*, Math. Systems Theory, 19 (1986), pp. 29–41.

[22] A. SELMAN, *P-selective sets, tally languages, and the behavior of polynomial time reducibilities on NP*, Math. Systems Theory, 13 (1979), pp. 55–65.

[23] S. TANG, B. FU, AND T. LIU, *Exponential time and subexponential time sets*, Theoret. Comput. Sci., 115 (1993), pp. 371–381.

[24] S. TODA, *On polynomial-time truth-table reducibilities of intractable sets to P-selective sets*, Math. Systems Theory, 24 (1991), pp. 69–82.

[25] O. WATANABE, *A comparison of polynomial time completeness notions*, Theoret. Comput. Sci., 54 (1987), pp. 249–265.

[26] C. WILSON, *Relativized circuit complexity*, J. Comput. System Sci., 31 (1985), pp. 169–181.