Complete Sets and Structure in Subrecursive Classes

Harry Buhrman[2] & Leen Torenvliet

ABSTRACT In this expository paper, we investigate the structure of complexity classes and the structure of complete sets therein. We give an overview of recent results on both set structure and class structure induced by various notions of reductions.

1 Introduction

After the demonstration of the completeness of several problems for **NP** by Cook [Coo71] and Levin [Lev73] and for many other problems by Karp [Kar72], the interest in completeness notions in complexity classes has tremendously increased. Virtually every form of reduction known in computability theory has found its way to complexity theory. This is usually done by imposing time and/or space bounds on the computational power of the device representing the reduction.

Early on, Ladner et al. [LLS75] categorized the then known types of reductions and made a comparison between these by constructing sets that are reducible to each other via one type of reduction and not reducible via the other. They however were interested just in the relative strength of the reductions and not in comparing the different degrees of complete sets that are induced by these reducibilities. This question was picked up much later by Watanabe [Wat87] for deterministic exponential time and others following him for other classes. A recent survey of this can be found in [Buh93]

Complete sets under some type of reduction form an interesting, rich, and hence much studied subject in complexity theory. A complete set can be viewed as representing an entire complexity class. Through the translation of the reduction one can with the help of the complete set, decide all questions of membership for any set in the class. The resource bound on the reduction is (to be of interest) always much less than the resource

[1]Received October 1996; revised version February 25, 1997.
[2]Partially supported by NWO

bound that defines the complexity class. A complete set is, as such, a "most difficult object" in the complexity class.

On the other hand by that very same representation property, we observe that the complete set codes all the information for all the sets present in the complexity class. A complete set therefore necessarily has to have large parts that are computationally easily recognizable. Thus, if a complete set is "most difficult" this usually is not pertinent to all of the set, but rather to "significant parts." It then becomes an interesting question, which parts of the complete sets are difficult, and, more pressing, which are not.

Complete sets in complexity classes are also usually the only sets that arise naturally. Once a set has been identified as being of a certain complexity, it is by experience soon after exposed as being complete in the corresponding class. This phenomenon was also noted in computability theory where every naturally arising non-recursive r.e. set turned out to be complete. This initiated E. Post [Pos44] to formulate his problem and program.

The different types of reductions impose structure on both complexity class and complete sets. Different types of reductions induce different notions of completeness which may give rise to a different degree structure. Stronger forms of reduction give more restrictions on how the information is stored in the complete sets than do weaker forms. Hence stronger reduction types impose more structure on the sets that are complete.

The most interesting complexity class to study complete sets on is the class **NP**. However this is also by far the most reluctant class to reveal its secrets. Exponential time (and larger) classes have the tremendous advantage that they allow for diagonalization against polynomial time bounded reductions and therefore for the construction of sets and degrees that have the desired properties. The bulk of the results obtained (and therefore the bulk of the results reviewed in this paper) pertains to exponential time (and larger) complexity classes.

In the different sections of this paper we pay attention to the structure of complexity classes as well as the structure of complete sets in these classes under various types of reductions. We treat several subjects that have received much attention in recent years. We must perforce limit our attention to some selected topics. The selection of the topics was almost always inspired by the fact that these topics were our own subject of (a) research (paper) at some point in time. We do not wish to implicitly or explicitly valuate other related and unrelated topics not mentioned in this paper, nor do we wish to make any claims about giving a complete survey on the topics that *do* appear in this paper. Having stated this disclaimer, we can now come to a list of topics that we will treat.

Class Structure: Degrees of Complete Sets

Here we investigate the question which reduction types give rise to different degrees of complete sets (collapse or separation). We survey the work of Watanabe [Wat87] and Buhrman et al. [BHT91, BST93]

Class Structure: Isomorphism

This is actually a refinement of the degree structure question. Of importance is which sets in (the complete) many-one degrees are isomorphic. This question was also inspired by the fact that in recursive set theory the many-one degree collapses to a single isomorphism degree [Myh55].

Class Structure: Measure Theory

Until recently, measure theory was a subject that was not applicable to complexity classes (or more general to effective classes) because of their inherent countability. Lutz [Lut90] however, introduced "resource bounded measure" with which many interesting properties about "abundance" of sets in complexity classes could be derived.

Set Structure: Sparse Complete Sets

Here we discuss some new developments on the most apparent internal structural properties of a complete set—the number of elements it has per length.

Set Structure: Redundant Information

Some strings in a complete set are essential to its completeness. E.g., for many-one reductions some strings are the image of a string under this reduction and some are not. Removing the former may destroy completeness, removing the latter has no consequence. Which (subsets of) strings are crucial?

Set Structure: Instance Complexity

As we already noted much of a complete set is of trivial complexity. Which is the part that makes it difficult and how dense is this part? To measure the complexity of single instances we use the measure introduced by Orponen et al. [OKSW94].

Set Structure: Post's Program Revisited

Autoreducibility is a special form of reducibility introduced by Trakhtenbrot [Tra70]. This notion has received considerable attention recently [BFT95] because of its potential to discover answers to the fundamental questions. Autoreducibility is a structural notion that complete sets in some complexity classes do and complete sets in other complexity classes do not have and is therefore a potential separator of complexity classes. As such this can be viewed as a new instantiation of Post's Program.

Several expository papers were written on the structure of complete sets in complexity classes. The first were published in 1990 [Hom90, KMR90]. On complete sets with special structure (sparse sets) a survey was published in 1992 ([HOW92]). The present authors presented a survey in 1994 ([BT94]) in a paper that has roughly the same structure as this paper. The field is however rapidly expanding and needs surveys such as this and as [Hom97] for constant update.

2 Preliminaries

All sets (and languages) in this paper are subsets of Γ^*, where $\Gamma = \{0, 1\}$, and are denoted by capital letters A, B, C etc. Strings are elements of Γ^* and are denoted by small letters x, y etc. The characteristic function of a language A is denoted by $\chi_A : \Gamma^* \mapsto \Gamma$. The characteristic sequence of a language A is the infinite string

$$\chi_A(\lambda)\chi_A(0)\chi_A(1)\chi_A(00)\chi_A(01)\chi_A(10)\chi_A(11)\ldots$$

and is also denoted by χ_A. The complement of a set A, $\Gamma^* - A$, is denoted by $\overline{A}$. The complement of a class of languages $\mathcal{X}$, $2^{\Gamma^*} - X$, is denoted by $\overline{\mathcal{X}}$. The *class complement* of a class $\mathcal{X}$, $\{A \mid \Gamma^* - A \in \mathcal{X}\}$, is denoted by $\mathrm{Co} - \mathcal{X}$. The length of a string x is denoted by $|x|$ and the cardinality of a set A is denoted by $\|A\|$. We assume (standard) enumerations of resource bounded Turing Machines $M_1, M_2, \ldots$ where the resource bounds on the machines vary to satisfy our needs. The set accepted by a Turing machine M is called its *language* and is denoted by $L(M)$. The main complexity classes considered in this paper are **LOG**, **NLOG P**, **NP**, **PSPACE**, **E**, **NE**, **EXP** and **NEXP**, which are the classes of languages recognized by:
deterministic logarithmic space,
nondeterministic logarithmic space,
deterministic polynomial time,
nondeterministic polynomial time,
deterministic polynomial space,
deterministic linear exponential time,
nondeterministic linear exponential time,
deterministic exponential time, and
nondeterministic exponential time bounded Turing machines respectively.

2.1 Reductions

Complete sets are always complete under some form of reduction. Many different models for this concept are found in the literature. As the concept of complete sets and therefore the concept of reductions is central to this paper, we take some space here to present a uniform machine based approach in which all reductions are modeled by Oracle Turing Machines.

2.1.1 Oracle Turing Machines

An oracle Turing machine is a standard multi-tape Turing machine with two extra tapes :

1. a write only tape called the QUERY-tape.

2. a read only tape called the ANSWER-tape.

These tapes will be called the oracle tapes. Furthermore we add an extra state: the QUERY-state. We use the following convention for access to the oracle tapes: M is allowed to write on the QUERY-tape a string q, called *query*, then at some point it decides to go into the QUERY-state. Subsequently the QUERY-tape is cleared[3], and depending on the oracle, something is written on the ANSWER-tape. Now M is allowed to read the ANSWER-tape, until a next QUERY-state is reached. A Turing machine equipped with the above described extra tapes and state is called an *oracle Turing machine.* In the above discussion it was not clear what the role of the oracle is. An oracle is just a set, say A. When an oracle Turing machine M writes a string q on the QUERY-tape and enters the QUERY-state, the oracle writes down – in one step – on the ANSWER-tape the value of the characteristic function of A on q, that is $\chi_A(q)$. Informally, M asks oracle A whether y is a member of A and finds the answer on its ANSWER-tape. We note here that the role of the oracle can be more complex in the sense that it could write down not only one character but a whole string of characters. Examples of this can be found in [ABJ91, FHOS93]. Let A be a set and M be an oracle Turing machine We say that M accepts x *relative to* A if M has an accepting computation on input x, with A as oracle. We say that $L(M, A)$ is the set of strings accepted by M relative to A. As usual, we can talk about polynomial time oracle machines and computations.

2.2 Adaptive and Non-Adaptive

Essentially there are two ways an oracle machine can compute its queries:

1. adaptive: M is allowed to read the ANSWER-tape at *any* time during the computation and may compute the next query depending on the contents of the ANSWER-tape. In this case the queries are *dependent* on the oracle.

[3] With clearing a tape we mean that after clearing, the only symbols on the tape are blanks.

2. non-adaptive: M is not allowed to read the ANSWER-tape *before* it enters the last QUERY state. In this case the computation of all the queries depends solely on the input and the program, and is *independent* of the oracle.

Sometimes we want to to talk about the set of (possible) queries that M could ask on input x.

Definition 2.1 *Let M be an oracle Turing machine.*

- *$Q(M, x, A)$ is the set of all queries M wrote on its QUERY-tape with x as input and A as oracle.*
- *$Q(M, x) = \bigcup_{A \subseteq \Sigma^*} Q(M, x, A)$. This denotes the set of all possible queries M could ask.*

2.2.1 Reductions

Oracle machines are used to model (almost) all different types of reductions. To achieve this we add restrictions to the oracle machine M. More formally a *restriction* is a 4 tuple:
r = <N, COMP, ACCEPT-RESTR, QUERY-TYPE>. Where,

1. N is a function from $\mathbb{N} \times \Sigma^* \to \mathbb{N}$. This function depends on the index of M and the input. The function is the number of queries M is allowed to make during the computation. With "number of queries" we mean the number of times that M entered the QUERY-state.
2. COMP can be adaptive or non-adaptive.
3. ACCEPT-RESTR is a function from $\Sigma^* \times \Sigma^*$ to $\mathcal{P}(\{0,1\})$, where $\mathcal{P}(\{0,1\})$ denotes the *power* set of $\{0,1\}$. This function depends on the INPUT-tape and the ANSWER-tape. When this function takes on the value $\{0\}$ or $\{1\}$, the machine is forced to reject or accept respectively.
4. QUERY-TYPE is a set of additional constraints on the type of queries M is allowed to make. E.g., all the queries should start with a 0 or should be smaller in length than the input.

We say that an oracle Turing machine M_i (we assume an effective enumeration of oracle Turing machines) *obeys* restriction r, if for all input strings x:

- N $= \emptyset$ or $M_i(x)$ does not make more than N(i, x) queries, and
- COMP $= \emptyset$ or $M_i(x)$ generates it's queries in a COMP (i.e., either adaptive or non-adaptive) fashion, and

- ACCEPT-RESTR $= \emptyset$ or if $M_i(x)$ halts then $M_i(x) \in$ ACCEPT-RESTR(x, y), for y the string on the ANSWER-tape when $M_i(x)$ halts, and
- QUERY-TYPE $= \emptyset$ or $M_i(x)$ only wrote down queries q that satisfy the constraints in QUERY-TYPE.

We say that M is an r-restricted oracle machine, if r is a restriction and M is an oracle Turing machine, that obeys restriction r.

Definition 2.2 *A r reduces to B ($A \leq_r^{rec} B$) iff there exists a recursive r-restricted oracle Turing machine M, such that $A = L(M, B)$.*

In this paper, we will mainly talk about polynomial time oracle machines. Note that this does not necessarily means that the ACCEPT-RESTR function is computable in polynomial time. This notion will be called polynomial time reduction. Several forms of polynomial time reduction were first defined and compared by Ladner, Lynch and Selman [LLS75]. In the following we will not redefine the existing notions of reducibility. We will capture them in a machine based framework. We think that the most natural way to think about a reduction is as an oracle Turing machine with several restrictions on the access it has to the oracle. The most general one is the Turing reduction which has no restrictions at all. The definitions found in the literature are by no means uniform in this sense. Sometimes they define reductions as functions. Other times the machine based point of view is used.

The approach we take has also the advantage that it gives a taxonomy of the reductions in four natural groups. Several new reductions emerge from this taxonomy by varying the 4 different aspects of the reductions. Sometimes already existing reductions come out. For example *adaptive* conjunctive reductions are the same as non-adaptive conjunctive reductions, but it is probably not true that adaptive parity (or majority) reductions are the same as their non-adaptive counterparts.

Definition 2.3 *A reduces r to B in polynomial time ($A \leq_r^p B$) iff there exists an r-restricted polynomial time oracle machine M such that $A = L(M, B)$.*

We will now show that some of the standard reductions found in the literature are easily captured by our formalism. To start with the most general restriction:

1. T $= <\emptyset, \emptyset, \emptyset, \emptyset>$. ($\leq_T^p$)
   This restriction does not restrict the class of oracle machines. This reduction is called Turing reduction.
2. tt$= <\emptyset$, non-adaptive, $\emptyset, \emptyset>$. ($\leq_{tt}^p$)
   The oracle machines are restricted in the way they generate their queries. This reduction is called truth-table reduction.

3. btt $= <n_b, \text{non-adaptive}, \emptyset, \emptyset>$. ($\leq^p_{btt}$)
$n_b(i, x) = i$.
This reduction is called bounded truth-table reduction.

4. k-tt $= <n_k, \text{non-adaptive}, \emptyset, \emptyset>$. ($\leq^p_{k-tt}$)
$n_k(i, x) = k$, k a constant.
This reduction is called k-truth-table reduction. Actually this defines a whole class of reductions. One for every constant k.

5. k-T $= <n_k, \emptyset, \emptyset, \emptyset>$. ($\leq^p_{k-T}$)
$n_k(i, x) = k$, k a constant.
This reduction is called k-Turing. The bounded Turing reduction ($\leq^p_{b-T}$) is defined by replacing n_k by $n_b(i, x) = i$.

6. ctt $= <\emptyset, \text{non-adaptive}, f_c, \emptyset>$. ($\leq^p_c$)
$$f_c(x, y) = \begin{cases} \{1\} & \text{if } \forall i, y_i = 1. \quad (y = y_1 \ldots y_n) \\ \{0\} & \text{otherwise.} \end{cases}$$
This reduction is called conjunctive truth-table reduction.

7. dtt $= <\emptyset, \text{non-adaptive}, f_d, \emptyset>$. ($\leq^p_d$)
$$f_d(x, y) = \begin{cases} \{1\} & \text{if } \exists i, y_i = 1. \quad (y = y_1 \ldots y_n) \\ \{0\} & \text{otherwise.} \end{cases}$$
This reduction is called disjunctive truth-table reduction.

8. m $= <n_m, \emptyset, f_m, \emptyset>$. ($\leq^p_m$)
$n_m(i, x) = 1$.
$$f_m(x, y) = \begin{cases} \{y\} & \text{if } y = 0 \text{ or } y = 1. \\ \emptyset & \text{otherwise.} \end{cases}$$
This reduction is called a many-one reduction. Sometimes it will be more elegant to use the following equivalent definition: $A \leq^p_m B$ iff there exists a total polynomial time computable function f such that $x \in A$ iff $f(x) \in B$. Obviously f can be constructed from an oracle machine that obeys the restriction m and vice versa.

9. $\hat{\text{m}} = <n_{\hat{m}}, \emptyset, f_{\hat{m}}, \emptyset>$. ($\leq^p_{\hat{m}}$ p)
$n_{\hat{m}}(i, x) = 1$.
$$f_{\hat{m}}(x, y) = \begin{cases} \{y\} & \text{if } y = 0 \text{ or } y = 1. \\ \{0, 1\} & \text{otherwise.} \end{cases}$$
This reduction will be called extended many-one.

10. m,li $= <n_m, \emptyset, f_m, \text{LI}>$ ($\leq^p_{m,li}$ p)
n_m and f_m as above.
LI $= \forall y \in Q(M_i, x) : |y| > |x|$.
This constraint says that the queries have to be bigger (in length) than the input.

This reduction is called many-one length increasing. As in the case of the many-one reduction we sometimes use the equivalent functional definition in terms of total polynomial time computable functions that are length increasing.

11. m,1-1 = $<n_m, \emptyset, f_m, \text{ONE-ONE}>$ ($\leq^p_{m,1\text{-}1}$ p)
n_m and f_m as above.
ONE-ONE $= \forall y \in Q(M_i, x) : \forall x' < x : y \notin Q(M_i, x')$. This constraint says that each query is asked once. Clearly this implies injectivity. This reduction is called many-one and one to one reduction. We will use sometimes the existence of a total polynomial time computable function that is one-one.

12. m,1-1,li $=<n_m, \emptyset, f_m, \text{ONE-ONE-LI}>$ ($\leq^p_{m,1\text{-}1,li}$)
n_m and f_m as above.
ONE-ONE-LI = ONE-ONE and LI. This means that both the constraints (ONE-ONE and LI) have to be satisfied in order to satisfy ONE-ONE-LI. This reduction is called many-one, length increasing and one to one. Again the functional equivalent way is sometimes chosen: there exists a total polynomial time computable function that is one-one and length increasing.

13. m,1-1,eh $=<n_m, \emptyset, f_m, \text{ONE-ONE-EH}>$ ($\leq^p_{m,1\text{-}1,eh}$)
n_m and f_m as above.
EH $= \forall y \in Q(M_i, x) : 2^{|y|} > |x|$.
This constraint says that the queries do not decrease more than exponentially in length. ONE-ONE-EH = ONE-ONE and EH. This reduction is called many-one, one to one and exponentially honest. The same comment applies here: the functional variant must be exponentially honest, i.e. not decrease more than an exponential in the length of the argument.

14. pos = $<\emptyset, \emptyset, f_{pos}, \emptyset>$ ($\leq^p_{pos}$)
Let **POS** be the class of all positive boolean formulas. These are formulas, that can be represented using only disjunctions and conjunctions as connectives. For x a boolean variable, $x := 1(0)$ means $x := \top(\bot)$. $\phi = 1(0)$ if it evaluates to true (false).
$f_{pos}(x, y) = \{\phi(x_1 := y1, \ldots, x_i := y_i)\}$ $(y = y_1 \ldots, y_i)$.
For $\phi \in$ **POS**.
This reduction is called positive Turing reduction. The positive truth-table, positive bounded-truth-table and the positive k-truth-table reductions are defined as truth-table, bounded truth-table or k-truth-table reduction with f_{pos} as ACCEPT-RESTR.

Another way of looking at this reduction is as follows: M_i is a positive Turing reduction if for all oracles A and B it holds that if $A \subseteq B$ then $L(M, A) \subseteq L(M, B)$.

Reductions stronger than $\leq_m^p$-reductions can be modeled by circuits that have oracle gates, or circuits that compute functions. In particular so-called $\mathbf{AC}^0$, $\mathbf{NC}^0$ and $\mathbf{NC}^1$ reductions have recently received some attention. Here $\mathbf{AC}^k$ and $\mathbf{NC}^k$ are the classes of languages recognized families of circuits of polynomial size and depth $\log^k$ of unbounded and bounded fan in respectively.

2.3 Resource Bounded Measure

Classical Lebesque measure is an unusable tool in complexity classes. As these classes are all countable, everything we define in such a class has measure 0. Yet, we might wish to have a notion of "abundance" and "randomness" in complexity classes. Lutz [Lut87, Lut90] introduced the notion of *resource bounded measure*, and gave a tool to talk about these notions inside complexity classes.

Definition 2.4 *A* martingale d *is a function from* Γ^* *to* $\mathcal{R}^+$ *with the property that* $d(w0) + d(w1) \leq 2d(w)$ *for every* $w \in \Gamma^*$*.*

Definition 2.5 *A p-martingale is a martingale* $d : \Gamma^* \mapsto \mathcal{Q}^+$ *that is polynomial time computable.*

Definition 2.6 *A martingale* d succeeds *on a language* A *if*

$$\limsup_{n\mapsto\infty} d(\chi_A[0\ldots n-1]) = +\infty$$

We write $S^\infty[d] = \{A \mid d \text{ succeeds on } A\}$

Definition 2.7 *Let* $\mathcal{X}$ *be a class of languages.*

- $\mathcal{X}$ *has p-measure* 0 $(\mu_p(\mathcal{X}) = 0)$ *iff there exists a p-martingale* d *such that* $\mathcal{X} \subseteq S^\infty[d]$.
- $\mathcal{X}$ *has p-measure* 1 $(\mu_p(\mathcal{X}) = 1)$ *iff* $\mu_p(\overline{\mathcal{X}}) = 0$
- $\mathcal{X}$ *has p-measure* 0 *in* $\mathbf{E}$ $(\mu_p(\mathcal{X}|\mathbf{E}) = 0)$ *iff* $\mu_p(\mathcal{X} \cap \mathbf{E}) = 0$
- $\mathcal{X}$ *has p-measure* 1 *in* $\mathbf{E}$ $(\mu_p(\mathcal{X}|\mathbf{E}) = 1)$ *iff* $\mu_p(\overline{\mathcal{X}} \cap \mathbf{E}) = 0$

2.4 Completeness and Degrees

Definition 2.8 *Let* $\leq_r$ *be a reduction and* $\mathcal{C}$ *be a complexity class. A set* C *is* r-hard *for* $\mathcal{C}$ *iff* $\forall A \in \mathcal{C}, A \leq_r C$*. If, moreover,* $C \in \mathcal{C}$ *then* C *is* r-complete *for* $\mathcal{C}$*.*

Definition 2.9 *Let* $\leq_r$ *be a reduction and* A *be a set. The* r*-degree of* A*, denoted* $\mathbf{a}$ *is the class* $\{B \mid A \leq_r B \wedge B \leq_r A\}$

Definition 2.10 *A set* C *is* weakly-r-hard *for* $\mathcal{C}$ *if* $\mu_p\{A \mid A \in \mathcal{C} \wedge A \leq_r C\} \neq 0$*. If, moreover* $C \in \mathcal{C}$ *then* C *is weakly-*r-complete *for* $\mathcal{C}$*.*

2.5 Complexity of Instances

The computational complexity of a single string is always a constant. Indeed, for each string we can define a Turing machine that recognizes just that string and nothing else. This machine works in constant time (where the constant depends on the size of the single string recognized). Yet, we think that some instances of a set may be computationally harder than others. Kolmogorov complexity and generalized Kolmogorov complexity provide means to talk about (descriptional) complexity of individual strings. Instance complexity is a notion closely related to Kolmogorov complexity but of a more computational nature.

Consider the class of Turing machines that on each input always output 1 (accept) or 0 (reject) or ? (don't know). Such a Turing machine is said to be *consistent* with a set A iff, for all inputs x such that $M(x) \neq ?$ it holds that $M(x) = \chi_A(x)$. The t-bounded instance complexity of an instance x with respect to a set A, $\mathrm{ic}^t(x : A)$ is the size of the smallest–in length–Turing machine M such that

1. M is consistent with A
2. M runs in time bound t for all inputs
3. $M(x) \neq ?$

For a precise definition see [OKSW94].

3 Degrees of Complete Sets

Complete sets under some kind of reduction in a complexity class have (by definition) the property that they all reduce to each other under that reduction. I.e., they form a degree. The different reductions that are available thus form different complete degrees. Most of these degrees are ordered by inclusion by the comparative strength of the reductions. One very natural question to ask is whether two consecutive complete degrees in this ordering differ. It is trivially true that all the degrees of complete sets under polynomial time reductions on **P** are the same (namely **P** itself). On **P** all polynomial time degrees are known to *collapse* to the many-one degree (but known *not* to collapse to the even smaller isomorphism degree.) Therefore showing two degrees of complete sets under polynomial time different on any complexity class $\mathcal{C}$, implies that this class is not equal to **P**. Unfortunately, until now the only complexity classes on which separation of degrees has been a successful undertaking are classes which encompass exponential time.

Theorem 1 ([Wat87, BHT91, HKR93, BST93])
For $\mathcal{C} \in \{\mathbf{E}, \mathbf{EXP}, \mathbf{NE}, \mathbf{NEXP}\}$

1. *The degree of the* $\leq^p_{1-tt}$*-complete sets collapses to the degree of the* $\leq^p_m$*-complete sets.*

2. *The degrees of the* $\leq^p_m$-, $\leq^p_{btt}$-, $\leq^p_c$-, $\leq^p_d$-, $\leq^p_{tt}$- *and* $\leq^p_T$*-complete sets are all different and, moreover, when not obviously ordered by inclusion, incomparable.*

In [BST93] also the degrees of query-bounded reductions are compared and it turns out that

Theorem 2 ([BST93]) *For* $\mathcal{C} \in \{\mathbf{E}, \mathbf{EXP}, \mathbf{NE}, \mathbf{NEXP}\}$

1. *For any* $k \geq 2$, $\leq^p_{k-c}$-, *and* $\leq^p_{k-d}$*-completeness are incomparable on* $\mathcal{C}$

2. *For any* k *and* l, *with* $k < l \leq 2^k - 2$, $\leq^p_{k-T}$- *and* $\leq^p_{l-tt}$*-completeness are incomparable on* $\mathcal{C}$.

These two theorems reveal all relations between degrees of complete sets on exponential time and beyond under all reductions known from [LLS75]. Recently, Lutz [Lut94] introduced a new notion of complete set on exponential time, the *weakly* complete set. A set A is weakly hard under reduction $\leq_r$, if a non-significant part (i.e., a class with non-zero p-measure) of $\mathbf{E}$ reduces to A. It was not known until [ASMZ96] whether the degrees of weakly complete sets under various reduction types are different. As it turns out, these degrees behave the same as the classical complete degrees. All are different except for the many-one and 1-tt, which coincide. (See Theorem 10)

In connection with the isomorphism problem (See Theorem 6), degrees defined by reducibilities even stronger than $\leq^p_m$ have been studied. In particular in [AAR96] the $\mathbf{AC}^0$ degree is shown to collapse to the $\mathbf{NC}^0$ degree for every complexity class $\mathcal{C}$ that is closed under $\mathbf{NC}^1$ reductions.

On the class $\mathbf{NP}$ all relations of polynomial time complete degrees necessarily remain open questions (without a proof that $\mathbf{P} \neq \mathbf{NP}$). Even from the assumption $\mathbf{P} \neq \mathbf{NP}$ it is still open whether any two reductions induce different degrees on $\mathbf{NP}$. From the stronger assumption that $\mathbf{E} \neq \mathbf{NE} \cap \mathrm{Co} - \mathbf{NE}$, Selman [Sel82] showed that the reductions many-one and Turing differ, but not the completeness notions. A first result in the direction of separation of completeness notions on $\mathbf{NP}$ was achieved by Longpré and Young [LY90] who showed that for each k there exist many-one $\mathbf{NP}$-complete sets that A and B such that $A \leq^p_{2-d} B$, where the reduction needs linear time, but A does not reduce many-one to B using less than n^k time. The most recent development on $\mathbf{NP}$ is a theorem by Lutz and Mayordomo [LM94] who prove that complete degrees differ on $\mathbf{NP}$ from the very strong assumption that $\mathbf{NP}$ has nonzero measure in $\mathbf{EXP}$.

Theorem 3 *If* $\mu(\mathbf{NP} \mid \mathbf{EXP}) \neq 0$ *then*

1. *The $\leq_m^p$ -complete degree is different from the $\leq_{2-T}^p$ -complete degree for* **NP** *[LM94].*

2. *The $\leq_{2-tt}^p$ -complete degree is different from the $\leq_{2-T}^p$ -complete degree for* **NP** *[May94b].*

A last result we would like to mention in this section is about the many-one and Turing degrees on **PSPACE**. Here also, it is of course not known whether this class differs from **P** and therefore an assumption is necessary. Assuming that either randomized completeness notions differ or that **PSPACE** has a set with a dense subset of high generalized Kolmogorov complexity, Watanabe and Tang [WT89] show that the many-one and Turing complete degrees differ on **PSPACE**. From recent work of Buhrman and Fortnow [BF96] it follows that there exists a relativized world in which the $\leq_m^p$ and the $\leq_{1-tt}^p$ complete set on **PSPACE** differ.

On **NP** most problems remain open. With respect to $\leq_m^p$- and $\leq_{1-tt}^p$-complete degree on **NP**, Fortnow [For96] informed us of the existence of an oracle where these degrees are different.

Even under assumptions that imply $\mathbf{P} \neq \mathbf{NP}$ these problems remain hard. In particular the following seem most urgent

- Can the complete degrees for **NP**-under various reduction types be separated under the assumption $\mathbf{P} \neq \mathbf{NP}$?

- Assume $\mu(\mathbf{NP}|\mathbf{EXP}) \neq 0$. Can the $\leq_{btt}^p$-complete degree for **NP** be separated from the $\leq_{tt}^p$-complete degree, and can the $\leq_{tt}^p$-complete degree be separated from the $\leq_T^p$-complete degree?

- Assume $\mu(\mathbf{NP}|\mathbf{EXP}) \neq 0$. Does the $\leq_{1-tt}^p$-complete degree on **NP**-coincide with the $\leq_m^p$-complete degree?

4 Isomorphism

Special among the complete degrees are the degrees of isomorphism. Berman and Hartmanis [BH77] proved that all natural **NP**-complete sets are inter-reducible via length-increasing 1-1 reductions and could therefore, via a polynomial time analog of the Cantor-Bernstein-Myhill theorem show that these sets are all isomorphic under polynomial time computable isomorphisms.

After a long period of unsuccessful work in trying to prove isomorphism of all $\leq_m^p$-complete sets in **NP**, and oracle proof that the conjecture might not hold (most prominent, the conjecture fails relative to a random oracle [KMR89]) opinion shifted against the idea that the isomorphism conjecture might be true. It was not until 1992 that Fenner, Fortnow and Kurtz [FFK92] proved the existence of an oracle relative to which the isomorphism conjecture holds. In 1985, Joseph and Young [JY85] constructed unnatural sets, the so-called k-creative sets for which every 1-1 polynomial time computable honest function is a productive function. Hence these sets are very unlikely to be isomorphic to SAT. Also, there is a strong belief in the existence of so-called one-way functions. These are functions that are honest and polynomial time computable, but not polynomial time invertible. The image of SAT under such a function may be a complete set that is not isomorphic to SAT. This conjecture (in some sense opposing the isomorphism conjecture) became known as the "Encrypted Complete Set Conjecture" (ECC). It has also met with relativized counter examples [Rog95]. Modern versions of the isomorphism conjecture are more in the direction of stronger reductions. Most of the sets known to be $\leq_m^p$-complete in **NP** are also complete under much stronger reductions. Allender [All88] was the first to show that sets in **PSPACE**, complete under 1-L reductions (which is a function computable by a logspace bounded Turing machine that has a one-way input head) are polynomial time isomorphic. Burtschick and Hoene [BH92b] showed on the other hand that these sets are *not* necessarily isomorphic under 1-L computable isomorphisms.

The following theorem by Agrawal and Biswas, is the most general theorem known for the 1-L reductions.

Theorem 4 ([AB93]) *Let C be a complexity class that is closed under lin-log reductions, e.g.,* **P**, **NP**, **PSPACE**. *The sets complete for C under 1-L reductions are all p-isomorphic.*

The first order projection (defined by Valiant in [Val82]) is another example of a very strong form of reduction. Allender, Balcázar and Immerman showed in [ABI93] that for the first order projections an isomorphism theorem holds

Theorem 5 ([ABI93]) *Let C be a nice complexity class, e.g.* **P**, **NP**, **PSPACE**. *All sets complete for C under first order projections are isomorphic under first order isomorphisms.*

It is natural to try to weaken the reduction types in order to get stronger and stronger versions of isomorphism theorems for **NP**. One way to weaken the first order projections is to consider reductions that can be defined by circuits. A circuit can be equipped with oracle gates that can compute answers to queries. The class $\mathbf{NC}^0$ is the class of circuits of polynomial size and constant depth with bounded fan-in. The class $\mathbf{AC}^0$ is the class of circuits of polynomial size and constant depth with unbounded fan-in. There has recently be some good progress with respect to $\mathbf{AC}^0$ reductions by Agrawal, Allender and Rudich.

Theorem 6 ([AAR96]) *All* $\mathbf{AC}^0$ *complete sets for* **NP** *are* $\mathbf{AC}^0$ *isomorphic*

The proof of the above theorem relies heavily on the nonuniformity of the reduction. It is open whether an isomorphism theorem is true for **NP** when uniform reductions are used. On the other hand it is true that the existence of one-way polynomial time functions implies the existence of one-way $\mathbf{AC}^0$ functions [AAR96]. Hence the above theorem heavily rocks the belief in the ECC.

5 Measure Theory

Having established a difference between the degrees of complete sets under various reductions, one might continue to investigate the part(s) of the complexity class that consist(s) of complete sets. It is easily seen that a single complete set, by padding, gives a countable class of complete sets. But "countable class in countable class" gives no structural information about the class. A more informative concept is the resource bounded measure of a degree. For instance if a complete degree would have measure 1 in a complexity class, then every non-negligible part of that complexity class would have a complete set. That is, complete sets are all over the place. Even if one can just show that a complete degree has non-zero measure, (note that this can either mean that it has measure one or that it is not measurable) then the complete degree has a non-empty intersection with any measure one class in the complexity class. A complete set can then be found in any rich enough substructure. Until now results obtained seem to point in another direction. It seems that indeed chaos (as classes of random sets do not have measure 0) is more abundant than structure, even in the small universe of complexity classes.

Theorem 7

1. *The $\leq_m^p$ -complete degree for* $\mathbf{E}$ *has measure 0 in* $\mathbf{E}$ *[JL93, May94a].*
2. *For all k, the $\leq_{k-tt}^p$ -complete degree for* $\mathbf{E}$ *has measure 0 in* $\mathbf{E}$ *[BM95].*
3. *The $\leq_{btt}^p$ -complete degree for* $\mathbf{E}$ *has measure 0 in* $\mathbf{E}$ *[ASNT94].*

The question remains open for $\leq_T^p$-complete sets for **E**. Some progress has been made however. Allender and Straus showed the following.

Theorem 8 ([AS94]) *For almost every set A in* **EXP**, $\mathbf{BPP}^A = \mathbf{P}^A$.

This theorem shows that if **BPP** would be equal to **EXP** then the Turing complete sets for **EXP** would not have measure 0. Hence a proof that the Turing complete sets for **EXP** have measure 0 would separate **BPP** from **EXP**. It is well known that there exist relativized worlds where **EXP** = **BPP**, and since Theorem 8 relativizes it follows that there exist relativized worlds where the Turing complete sets for **EXP** do not have measure 0, but in fact have measure 1. On the other hand Ambos-Spies [AS96] has informed us of the construction of an oracle where the Turing complete degree for **EXP** has measure 0.

The class of weakly complete sets, a generalization of the classical notion of completeness, would not even have been studied without the introduction of resource bounded measure. A comparison between weakly complete sets and other complete sets is of course the first goal. It turns out that weakly complete sets differ from classical complete sets, both in the sense that their degrees are different and in the sense that they behave differently.

Theorem 9 ([JL94])

1. *Every language that is weakly $\leq_m^p$-complete for* **E** *is weakly $\leq_m^p$-complete for* **EXP**.

2. *There is a language in* **E** *that is weakly $\leq_m^p$-complete for* **EXP**, *but not for* **E**.

Which is rather surprising if one takes into account that, by padding, a set is $\leq_m^p$-complete for **E** if and only if it is $\leq_m^p$-complete for **EXP**.

As was shown recently by Ambos-Spies, Mayordomo and Zheng [ASMZ96], the weakly complete degrees and other complete degrees, both in **E** and in **EXP** form an interleaved structure. In particular.

Theorem 10 *[ASMZ96]*

1. *For any set A, A is weakly-$\leq_m^p$-complete for* **E** *if and only if A is weakly-$\leq_{1-tt}^p$-complete for* **E**.

2. *Let $k \geq 1$ there is a set A such that A is $\leq_{k+1-d}^p$-complete for* **E** *(and hence weakly-complete for both* **E** *and* **EXP***), but neither weakly-$\leq_{k-T}^p$-complete for* **E** *nor $\leq_{k-T}^p$-complete for* **EXP**

3. *There is a set A such that A is $\leq_d^p$-complete for* **E** *(and hence weakly dttp-complete for* **E** *and* **EXP***),but A is neither weakly $\leq_{btt}^p$-complete for* **E** *nor weakly-$\leq_{btt}^p$-complete for* **EXP**.

4. *There is a set A such that A is $\leq_T^p$-complete for* **E** *(hence weakly-$\leq_T^p$-complete for* **E** *and* **EXP***), but A is neither weakly-$\leq_{tt}^p$-complete for* **E** *nor weakly $\leq_{tt}^p$-complete for* **EXP**.

6 Sparse Complete Sets

One of the first structural questions to ask of a *set* is how many of the 2^n strings of each length are in or out of the set. I.e., one of the first apparent questions about the structure of a set is the question of its density. Also other notions of structure seem very closely related to this basic question. For instance it is known that the class of sets (Turing) reducible to sparse sets coincides with the class of sets reducible to P-selective sets and coincides with the class of sets that are recognizable by families of small (polynomial size) circuits and coincides with the class of sets that are recognizable in polynomial time given a polynomial amount of advice. In other words, the amount of information that can be stored in a set seems closely related to the number of strings that a set has per length. It therefore comes as no big surprise that a sparse set cannot be $\leq_m^p$-complete for **EXP**. This follows from an old theorem by Berman [Ber77]. For **NP** this question is a lot harder (if $\mathbf{P} = \mathbf{NP}$ then any set in **P** is $\leq_m^p$-hard for **NP** so also the sparse ones). It was answered by Mahaney [Mah82], building on earlier work of Fortune [For79], who showed that sparse sets could not be $\leq_m^p$-complete unless $\mathbf{NP} = \mathbf{P}$. For **P** the question needs reformulation. Under $\leq_m^p$-reductions any set is complete in **P**, but under $\leq_m^{logspace}$-reductions this is only true if $\mathbf{P} = \mathbf{LOG}$. An analogous question about the density of $\leq_m^{logspace}$-complete sets in **P** can thus be posed. Indeed, it was conjectured by Hartmanis in [Har78] that no sparse set can be complete for **P** under logspace reductions unless $\mathbf{P} = \mathbf{LOG}$. It was not until recently that this conjecture was proven true.

Theorem 11 ([Ogi95, CS95]) *If a sparse set S is hard for* **P** *under many-one reductions computable in logspace, then* $\mathbf{LOG} = \mathbf{P}$.

This Theorem was extended by Van Melkebeek to bounded truth-table reductions.

Theorem 12 ([vM96]) *If a sparse set S is hard for* **P** *under bounded truth-table reductions computable in logspace, then* $\mathbf{LOG} = \mathbf{P}$.

An extensive survey of the (im)possibilities of sparse complete sets, which for chronological reasons lacks the above results, but not many others was given by Hemachandra, Ogiwara and Watanabe [HOW92]. For exponential time classes it seems that the most pressing open question is the existence of Turing hard sparse sets. As mentioned above the question of a sparse Turing hard set coincides with the existence of polynomial sized circuits for the class and this question seems particularly interesting for **EXP**. (Until now the smallest class known not to be computable by polynomial size circuits is **MA(exp)**.) This result is due to Buhrman and Thierauf. (See [KW95].)

There are some steps set along this path however.

Theorem 13 ([Fu93])

1. *For $\alpha < 1$, all $\leq_{n^\alpha-T}^p$ -hard sets for* **EXP** *are exponentially dense.*

2. *For $\alpha < \frac{1}{4}$, all $\leq_{n^\alpha-T}^p$ -hard sets for* **E** *are exponentially dense.*

An incomparable theorem dealing with the part of **E** that may have small circuits (is in **P/poly**) is the following.

Theorem 14 **([LM93])** *For $\alpha < 1$, all $\leq^p_{n^\alpha-tt}$ -weakly-hard sets for* **E** *and* **EXP** *are exponentially dense.*

Finally, Homer and Mocas studied the possibility of exponential time computable sets being decidable with a fixed polynomial amount of advice. (Recall that if **EXP** has sparse complete sets then **EXP** $\subseteq$ **P/poly**.) They prove the following.

Theorem 15 **([HM93])** *for every k there exists a set A in* **EXP** *such that A is not in* $\mathbf{DTIME}(2^{n^k})/\mathbf{ADVICE}(n^k)$.

Improving upon these results seems very interesting, but also seems very hard since it would require non relativizing techniques. Wilson [Wil85] shows the existence of an oracle relative to which **EXP** has polynomial size circuits.

Specifically we note the following questions.

- For any k: are the $\leq^p_{n^k-T}$ complete sets for **EXP** exponentially dense?
- For any k: are the $\leq^p_{n^k-T}$ weakly complete sets for **EXP** exponentially dense?
- Related to [BH92a], are the $\leq^p_T$-complete sets for **NEXP** not sparse unless **NEXP** $= \Sigma^P_2$?

7 Redundant Information

As noted in the previous section, some complexity classes do not allow for sparse complete sets under some reductions. For such complexity classes in particular the question can be asked: "How dense must these sets be?" By the result of Berman [Ber77] the many-one complete sets for **EXP** for instance cannot be sparse. Schöning [Sch86], building upon the work of Yesha [Yes83], showed that for complete sets A in **EXP** and every set D in **P**, the set $A\Delta D$ is of exponential density. If sets must be really dense to be complete, we can also ask the question: "Can a small set perhaps be taken out of the set so that the remaining set is still complete?"

This question was first taken up by Tang, Fu and Liu [TFL93], who showed, inspired by Schönings theorem, that the set D in this theorem can be taken subexponential time computable. They go on to show that for arbitrary sparseness condition, there exists a single subexponential time computable sets S, such that for *any* exponential time complete sets A, the set $A - S$ is no longer exponential time complete.

The natural question to ask next after the result is obtained for many-one reductions is: "How do complete sets under weaker types of reductions behave?" The answer to this question was given in [BHT93]. They show that the observation on this structural aspect of complete sets is not limited to many one completeness *or* to deterministic exponential time.

Theorem 16
Given a recursive non-decreasing function $g(n)$ with $\lim_{n\mapsto\infty} g(n) = \infty$. There exists $g(n)$-sparse subexponential time computable sets S_1, S_2, S_3 and S_4 such that: [TFL93] For any $\leq^p_m$-hard set A, the set $A - S_1$ is not $\leq^p_m$-hard, and [BHT93]

1. *For any $\leq^p_{btt}$-complete set A for* **EXP** *the set $A - S_2$ is not $\leq^p_{btt}$-hard.*
2. *For any $\leq^p_c$-hard set A for* **EXP** *the set $A - S_3$ is not $\leq^p_c$-hard.*
3. *For any $\leq^p_d$-hard set A for* **EXP** *the set $A \cup S_4$ is not $\leq^p_d$-hard.*

Not only the sparseness of the sets S_i is controllable, but also their "subexponentiality." The construction can be slowed down to bring the set arbitrarily close to being polynomial time computable, but not quite. Tang, Fu and Liu [TFL93] already noted that for many-one reductions it holds that a $\leq^p_m$-complete set remains $\leq^p_m$-complete if an arbitrary sparse, polynomial time computable set is taken out. This however seems to have less to do with its polynomial time computability than with its structural simplicity. In [BHT93] this question was re-addressed for P-selective sets i.s.o. **P**-sets (which is of course stronger). The following was shown.

Theorem 17 ([BHT93]) *For any set A that is $\leq^p_m$-, $\leq^p_c$-, $\leq^p_d$-, or $\leq^p_{2-tt}$-hard for* **EXP** *and any $p(n)$ sparse P-selective set S, the set $A - S$ remains hard w.r.t. the same reduction.*

It may seem strange that this theorem can only be proven for $\leq^p_{2-tt}$-reductions. It follows however directly from a recent result of Buhrman, Fortnow and Torenvliet [BFT95] that this result is optimal. They show the existence of a $\leq^p_{3-tt}$-complete set in **EXP** that is not $\leq^p_{btt}$-autoreducible. (We will meet this result again in Section 9). Inspection of the proof learns 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 btt-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 then follows immediately.

Corollary 18 ([BFT95]) *There exists a 3-tt complete set A in* **EXP** *and a sparse set S in* **P** *such that $A - S$ is not btt-hard for* **EXP**.

As a $\leq^p_{tt}$-complete set may not be $\leq^p_{btt}$-complete this last corollary does not answer all remaining questions. The following questions still remain open. (Wilson's [Wil85] oracle forces non-relativizable proofs on answers.)

- Let A be $\leq_{tt}^{p}$ or $\leq_{T}^{p}$-complete for **EXP**. Does there exist a sparse subexponential time computable set S such that $A - S$ is not complete?
- Let A be $\leq_{btt}^{p}$-complete for **EXP**. Is for every sparse $S \in \mathbf{P}$, $A - S$ still $\leq_{btt}^{p}$-complete.

Related to the question whether removal of sparse polynomial time computable subsets destroys completenes is the question whether exponential time complete sets *have* polynomial time computable subsets. It follows easily from the theorem of Berman [Ber77] already cited in Section 6 that all $\leq_{m}^{p}$-complete sets in **EXP** have infinite polynomial time computable subsets. This question has however remained open for **NEXP** for a long time. It was Tran [Tra95] who proved the following.

Theorem 19 ([Tra95]) *All $\leq_{m}^{p}$-complete sets for* **NEXP** *are not* **P***-immune.*

For weaker reductions, **P**-immune complete sets can be shown to exist by straightforward diagonalization.

8 Instance Complexity

The hardness of individual instances of a complete set may not seem to be a structural question about complete sets and therefore beyond the scope of this paper. One can however ask questions about the distribution of hard and easy instances, which is certainly a question about the structure of the set. Questions like: "Are there an infinite number of hard instances in the set?" and: "Is the subset of hard instances dense or sparse?" certainly fall within the category of structural questions about sets and therefore these questions will be surveyed here with respect to complete sets. Complexity of individual strings was first studied by Kolmogorov and Chaitin (See [Har83, LV93]). Instance complexity is closely related to Kolmogorov complexity. Recall that the t-bounded Kolmogorov complexity of a string x is the size of the smallest Turing machine M that on input λ outputs x and takes no more than $t(|x|)$ steps. (Here also "size" is defined relative to a fixed universal machine.)

The following simple relationship holds between Kolmogorov complexity and instance complexity.

Proposition 20 ([OKSW94]) *For any time constructible function t, there exists a constant c such that for any set A and string x,*

$$\mathrm{ic}^{t'}(x : A) \leq K^{t}(x) + c,$$

where $t'(n) = ct(n)\log(t(n)) + c$.

The proposition states that the Kolmogorov complexity always is an upper bound on the instance complexity. On the other hand, we are interested in saying that a set A has instances that are hard or difficult. A natural way of expressing this is as follows:

Definition 8.1 *A set A has t-hard instances iff there exists a constant c such that for infinitely many instances x, $\mathrm{ic}^t(x : A) \geq K^t(x) - c$.*

In this definition we did not consider the $\log(n)$ factor that comes out of Proposition 20. Only considering polynomial instance complexity and admitting this ($\log(n)$) factor we also have the following notion of hard instances:

Definition 8.2 *A set A has $\mathbf{P}$-hard instances if for any polynomial t there exist a polynomial t' and a constant c such that for infinitely many x, $\mathrm{ic}^t(x : A) \geq K^{t'}(x) - c$.*

In [OKSW94] it was conjectured that any set not recognizable in a certain time bound t will have t-hard instances.

Conjecture 21 ([OKSW94]) *Let A be a set not in $\mathbf{DTIME(t)}$. Then there exist infinitely many x and a constant c such that $\mathrm{ic}^t(x : A) \geq K^t(x) - c$.*

As evidence for their conjecture it is proved for any set A in $\mathbf{E} \setminus \mathbf{P}$:

Theorem 22 ([OKSW94]) *Let A be a set in $\mathbf{E} \setminus \mathbf{P}$. There exists a constant c such that for any polynomial t there exists a constant c and infinitely many x, such that, $\mathrm{ic}^t(x : A) \geq K^{2^{cn}}(x) - c$.*

More recently, Fortnow and Kummer gave more evidence for this conjecture.

Theorem 23 ([FK95]) *Let A be a set not in $\mathbf{E}$. Then for any c there exists a c' and d such that for infinitely many x, $\mathrm{ic}^{2^{cn}}(x : A) \geq K^{2^{c'n}}(x) - d$.*

We will now shift our attention to complete sets.

8.1 *Complete sets and instance complexity*

The natural question addressed here is: "Do complete sets have hard instances?" Of course for classes like $\mathbf{NP}$ it is important to fix the time bound of the instance complexity to polynomial. In this section we will therefore only consider $p(n)$-bounded instance complexity for $p(n)$ some polynomial.

The first partial results along these lines were obtained by Orponen in [Orp90]. In this paper it was shown that it cannot be the case that the instance complexity of $\leq^p_{btt}$-hard sets for $\mathbf{NP}$ is low for all the instances unless $\mathbf{P} = \mathbf{NP}$.

Theorem 24 (**[Orp90]**) *If A is self-reducible and there exist a constant c and a polynomial t such that for all $x, \mathrm{ic}^t(x : A) \leq c\log(|x|) + c$ then A is in* **P**.

It now follows, using the fact that the instance complexity can not decrease much through a $\leq^p_{btt}$-reduction, that all classes that possess self-reducible Turing complete sets—such as **PSPACE** and **NP**—cannot have $\leq^p_{btt}$-hard sets with low instance complexity everywhere unless they are equal to **P**. The above theorem yields as an easy corollary an absolute result about **EXP**. Later on we will see that this can be improved.

Corollary 25 *There does not exists a $\leq^p_{btt}$-hard set A for* **EXP** *such that for all x, $\mathrm{ic}^t(x : A) \leq c\log(|x|) + c$ for some constant c and polynomial t.*

Proof. Note that sets with low instance complexity everywhere, as in the corollary, are in **P/poly**. Assume for a contradiction that the corollary is not true, it now follows by a well-known theorem of Karp and Lipton [KL80] that if **EXP** has a $\leq^p_T$ -hard set that is sparse, that then $\mathbf{EXP} = \mathbf{\Sigma}^{\mathbf{P}}_2$. Furthermore, $\mathbf{\Sigma}^{\mathbf{P}}_2$ contains self-reducible Turing complete sets and hence by Theorem 24 above it follows that $\mathbf{\Sigma}^{\mathbf{P}}_2 = \mathbf{P}$, but now we have a contradiction: $\mathbf{EXP} = \mathbf{\Sigma}^{\mathbf{P}}_2 = \mathbf{P}$. □

The above theorem and corollary are still far away from the instance complexity conjecture. Recently Fortnow and Kummer [FK95] showed the instance complexity conjecture in a special case

Theorem 26 (**[FK95]**) *Every tally set not in* **P** *has* **P**-*hard instances.*

Shifting to complete sets it is shown in [BO94] that the conjecture is correct for $\leq^p_m$-complete sets for **E** and polynomial time bounds t.

Theorem 27 (**[BO94]**) *Let A be a $\leq^p_m$-complete set for* **E**. *Then there exists an exponentially dense set $C \subseteq A$, such that for all $x \in C : \mathrm{ic}^t(x : A) \geq K^t(x) - c$, for c some constant and $t \in \omega(n\log(n))$.*

Note that the theorem is also true for **EXP** and $\leq^p_{1-tt}$-reductions. The theorem states that there is an exponentially dense subset of any complete set for **E** that has hard instances. This is in contrast with the fact that there also exists a subset of A that is in **P** [Ber77] [4] and hence has constant instance complexity for all x. Note also that Theorem 27 implies that any set B that is $\leq^p_{btt}$-hard for **E** contains infinitely many instances x such that $\mathrm{ic}^t(x : B) \geq cK^t(x) - d$ for some constants c and d. It remains open however whether Theorem 27 can be proven for complete sets with respect to other reductions. It even remains open if we drop the density requirement.

[4] See [HW94] for an extensive study on immunity and $\leq^p_m$-complete sets for **EXP** and **NEXP**.

It is not known whether Theorem 27 is true for $\leq_m^p$-complete sets for **NP**. However, Orponen showed in [Orp90] that under the assumption that $\mathbf{E} \neq \mathbf{NE}$ the **NP**-hard sets have **P**-hard instances.

Theorem 28 ([Orp90]) *If* $\mathbf{NE} \neq \mathbf{E}$ *then all* $\leq_{1-tt}^p$*-hard sets for* **NP** *have* **P***-hard instances.*

A special type of complete sets in **EXP** does have p-hard instances as is shown in [FK95]

Theorem 29 ([FK95]) *If A is complete in* **EXP** *under honest* $\leq_T^p$*-reductions, then A has* **P***-hard instances.*

Specific open problems remain:

1. If A is complete for **EXP** under $\leq_{btt}^p$, $\leq_{tt}^p$ or $\leq_T^p$, does A have t-hard instances, for t a polynomial?
2. If A is complete for **EXP** under $\leq_{tt}^p$ or $\leq_T^p$, does A have **P**-hard instances?
3. Is the set of hard instances in 1 or 2 dense?
4. Do many-one complete sets for **NP** have t-hard instances?

9 Post's Program Revisited

The final structural property that we wish to address in this paper is that of auto-reducibility and the closely related notion of mitoticity.

Trakhtenbrot [Tra70] introduced the notion of autoreducibility on recursive sets. Informally a set is auto-reducible, if can be recognized by an oracle machine that on input x never queries x and uses the set itself as an oracle. As such, a set that is auto-reducible can be viewed as having a redundancy of information. The information that a string x is a member of A is also present in other strings in A and this is true *for every member of A*. The notion of autoreducibility is closely related to the notion of mitoticity introduced by Ladner in [Lad73]. Informally a set is mitotic if it can be split into two disjoint subsets, such that both parts are reducible to each other and moreover the original set is reducible to both parts (and vice versa.) As such, mitoticity is a form of ordered auto-reducibility. The parts of the set that contain information about the other parts are neatly ordered in disjoint subsets.

The term mitotic stems from biology, where the mitosis indicates the splitting of a cell into two cells that both contain the same information stored in the DNA of the original cell. As such this term is very appropriately chosen for the subsets that contain precisely the information of the original set. Ladner showed that the two seemingly different, but apparently related notions of auto-reducibility and mitoticity coincide for r.e. sets.

Ambos-Spies [AS84] was the first to carry over the notions of auto-reducibility and mitoticity to the realm of complexity theory. The auto-reducibility notion translates into:

Definition 9.1 *A set A is polynomially autoreducible (autoreducible for short) if there exists a polynomial time oracle Turing machine M such that:*

1. $A = L(M, A)$.
2. *for all x: $x \notin Q(M, x, A)$.*

Where $Q(M, x, A)$ is the set of queries M generates on input x with A as an oracle.

There also exists a randomized version of auto-reducibility called *coherence* [Yao90, BF92].

Translating mitoticity [AS84] to the polynomial time setting gives a less clean transition.

Definition 9.2

1. *A recursive set A is polynomial time m(T)-mitotic (m(T)-mitotic for short) if there exists a set $B \in \mathbf{P}$ such that:*
   $A \equiv^p_{m(T)} A \bigcap B \equiv^p_{m(T)} A \bigcap \overline{B}$.
2. *A recursive set A is polynomial time weakly m(T)-mitotic (weakly m(T)-mitotic for short) if there exists disjoint sets A_0 and A_1 such that:*
   $A = A_0 \bigcup A_1$ *and* $A \equiv^p_{m(T)} A_0 \equiv^p_{m(T)} A_1$.

Although mitoticity implies weak mitoticity, the two notions are quite different in the polynomial time setting. From [AS84] we learn the following about their relation.

Theorem 30 ([AS84])

1. *For any A, if A is m(T)-mitotic then A is m(T)-autoreducible.*
2. *On recursive tally sets m-mitoticity and m-autoreducibility are the same.*
3. *There exists a weakly m-mitotic set that is not m-autoreducible.*
4. *There exists a tally set which is T-autoreducible but not T-mitotic.*
5. *There exists a weakly T-mitotic set that is not T-autoreducible.*

The questions whether

- m-autoreducibility implies m-mitoticity, or
- m-autoreducibility implies weakly m-mitoticity, or
- T-autoreducibility implies weakly T-mitoticity,

remain open. In the following we will turn back to the complete sets.

9.1 Completeness

In this subsection we will address the question of mitoticity and autoreducibility for complete sets. We will treat these questions separately.

9.1.1 Mitoticity

On the class **NP** we note that all natural **NP** complete sets are m-mitotic. These sets are all p-isomorphic to SAT [BH77] and SAT can easily be shown to be mitotic. Moreover all the notions of mitoticity and autoreducibility are invariant under p-isomorphisms. Thus it follows that if the isomorphism conjecture is true, then all $\leq_m^p$ -complete sets for **NP** are m-mitotic (and hence weakly m-mitotic). Similar observations are valid for all levels of the Polynomial Hierarchy and **PSPACE**. However it is open whether any complete degree (under some reduction) for these classes is completely (weakly) m(T)-mitotic. For **EXP** a little bit more can be said:

Theorem 31 ([BHT93]) *All $\leq_m^p$ -complete sets for* **EXP** *are weakly m-mitotic.*

This theorem parallels nicely the situation for r.e. sets. On the other hand it is shown also that not all $\leq_{tt}^p$ -complete sets are weakly m-mitotic.

Theorem 32 ([BHT93]) *There exists a $\leq_{3-tt}^p$ -complete set for* **EXP** *that is not weakly m-mitotic.*

For **NEXP** the situation is even less clear. It can be shown that every $\leq_m^p$ -complete set can be split into *infinitely* many disjoint subsets, such that each of these subsets is $\equiv_m^p$ to A [BHT93], but nothing is known for the mitotic case. Specific open problems are:

- Are all $\leq_m^p$-complete sets for **EXP** m-mitotic?
- Are all $\leq_m^p$-complete sets for **NP**, **PH**, **PSPACE**, or **NEXP** (weakly) m(T)-mitotic?
- Are all $\leq_T^p$-complete sets for **NP**, **PH**, **PSPACE**, **EXP** or **NEXP** (weakly) $m(T)$-mitotic?

9.1.2 Autoreducibility

The situation with respect to autoreducibility is somewhat better understood. Surprisingly, the parallel with recursion theory disappears with respect to complete sets. Of course the same remarks about $\leq_m^p$-complete sets that were made for mitoticity are true for autoreducibility but the $\leq_T^p$-complete sets seem to behave differently.

Theorem 33 (**[BF92, BT96]**) *Every $\leq_T^p$-complete set for* **NP** *is autoreducible.*

In fact in [BF92] it is shown that all $\leq_T^p$-degrees that contain a self-reducible set are completely autoreducible hence:

Theorem 34 (**[BF92]**) *All $\leq_T^p$ -complete sets for all levels of the Polynomial Hierarchy and* **PSPACE** *are autoreducible.*

Unfortunately the techniques in [BF92] only apply to sets within **PSPACE**. Extending the techniques in [BT96] it can be shown that also the complete sets for **EXP** are autoreducible.

Theorem 35 (**[BFT95]**) *All $\leq_T^p$ -complete sets for* **PSPACE** *and* **EXP** *are autoreducible.*

It can also be shown that all $\leq_{2-tt}^p$ -complete sets for **EXP** are 2-tt-autoreducible. Recently Buhrman, Fortnow and Torenvliet [BFT95] established that this is not true for $\leq_{3-tt}^p$-reductions. They show the existence of a set that is 3-tt complete for **EXP** but not $\leq_{btt}^p$-auto-reducible. Translation of this theorem to a larger complexity class gives a 3-tt complete set in **EEXPSPACE** that is not (Turing) auto-reducible. It follows that it is most urgent to determine the autoreducibility of sets complete in the class **EXPSPACE**. If all these sets are auto-reducible then **NL** $\neq$ **NP**, and if there is one that is not auto-reducible then **P** $\neq$ **PSPACE** (See also [BFvMT98].)

Other questions that remain:

- Are all $\leq_m^p$ -complete sets for **NP** or **PSPACE** m-autoreducible?
- Are all $\leq_{tt}^p$ -complete sets for **NP**, **PSPACE** or **EXP** *tt*-autoreducible?
- Are all $\leq_T^p$ -complete sets for **NEXP** autoreducible?

Acknowledgements

We would like to thank Dieter van Melkebeek, Lance Fortnow, Steve Homer and Peter van Emde Boas for their help and comments on earlier versions of this paper.

10 References

[AAR96] M. Agrawal, E. Allender, and S. Rudich. Reductions in circuit complexity: An isomorphism theorem and a gap theorem. Submitted for publication to JCSS Special Issue on the Eleventh Annual IEEE Conference on Computational Complexity. A preliminary version appeared in the proceedings of that conference, 1996, pp. 2-11., 1996.

[AB93] M. Agrawal and S. Biswas. Polynomial isomorphism of 1-L complete sets. In *Proc. Structure in Complexity Theory 7th annual conference*, pages 75–80, San Diego, California, 1993. IEEE Computer Society Press.

[ABI93] E. Allender, J.L. Balcázar, and N. Immerman. A first-order isomorphism theorem. In *STACS 93, Lecture Notes in Computer Science 665*, pages 163–174, 1993.

[ABJ91] C. Àlvarez, J. Balcázar, and B. Jenner. Functional oracle queries as a measure of parallel time. In C. Choffrut and M.Jantzen, editors, *STACS 91, Lecture Notes in Computer Science 480*, pages 422–433. Springer-Verlag, 1991.

[All88] E. Allender. Isomorphisms and 1-L reductions. *J. Computer and System Sciences*, 36(6):336–350, 1988.

[AS84] 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.

[AS94] E. Allender and M. Strauss. Measure on small complexity classes, with applications for BPP. In *Proc. 35th IEEE Symposium Foundations of Computer Science*, pages 817–818, 1994.

[AS96] K. Ambos-Spies. Personal communication. 1996.

[ASMZ96] K. Ambos-Spies, E. Mayordomo, and X Zheng. A comparison of weak completeness notions. In *Proceedings 11th annual IEEE conference on Computational Complexity*, pages 171–178, Philadelphia, 1996. IEEE Computer Society Press.

[ASNT94] K. Ambos-Spies, H-C. Neis, and S.A. Terwijn. Genericity and measure for exponential time. In *Proceedings MFCS94, Lecture Notes in Computer Science*, volume 841, pages 221–232. Springer Verlag, 1994.

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

[BF92] R. Beigel and J. Feigenbaum. On being incoherent without being very hard. *Computational Complexity*, 2(1):1–17, October 1992.

[BF96] H. Buhrman and L. Fortnow. Two queries. Technical Report CS 96-20, University of Chicago Department of Computer Science, 1996.

[BFT95] H. Buhrman, L. Fortnow, and L. Torenvliet. Using autoreducibility in separating complexity classes. In *Proc. 36d IEEE Symposium on Foundations of Computer Science*, pages 520–527, Milwaukee, Wis, 1995.

[BFvMT98] H. Buhrman, L. Fortnow, D. van Melkebeek, and L. Torenvliet. Separating complexity classes using autoreducibilty. Technical Report FI-XCT1998-002, Fields Institute, 1998.

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

[BH92a] H. Buhrman and S. Homer. Superpolynomial circuits, almost sparse oracles and the exponential hierarchy. In R. Shyamasundar, editor, *Proc. 12th Conference on the Foundations of Software Technology & Theoretical Computerscience, Lecture Notes in Computer Science*, pages 116–127. Springer Verlag, 1992.

[BH92b] H. Burtschick and A. Hoene. The degree structure of 1-L reductions. In *STACS 92, Lecture Notes in Computer Science 629*, pages 175–184, 1992.

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

[BHT93] H. Buhrman, A. Hoene, and L. Torenvliet. Splittings, robustness and structure of complete sets. In *STACS 93, Lecture Notes in Computer Science 665*, pages 175–184, 1993.

[BM95] H. Buhrman and E. Mayordomo. An excursion to the Kolmogorov random strings. In *Proc. Structure in Complexity Theory 10th annual conference*, pages 197–203, Minneapolis, Minnesota, 1995. IEEE computer society press.

[BO94] H. Buhrman and P. Orponen. Random strings make hard instances. In *Proc. Structure in Complexity Theory 9th annual conference*, Amsterdam, Holland, 1994. IEEE computer society press.

[BST93] H. Buhrman, E. Spaan, and L. Torenvliet. Bounded reductions. In K. Ambos-Spies, S. Homer, and U. Schöning, editors, *Complexity Theory*, pages 83–99. Cambridge University Press, December 1993.

[BT94] H. Buhrman and L. Torenvliet. On the structure of complete sets. In *Proc. Structure in Complexity Theory 9th annual conference*, pages 118–133, Amsterdam, Holland, 1994. IEEE computer society press.

[BT96] H. Buhrman and L. Torenvliet. P-selective self-reducible sets: A new characterization of *P*. *J. Computer and System Sciences*, 53(2):210–217, 1996.

[Buh93] H.M. Buhrman. *Resource Bounded Reductions.* PhD thesis, University of Amsterdam, June 1993.

[Coo71] S. Cook. The complexity of theorem-proving procedures. In *Proc. 3rd ACM Symposium Theory of Computing*, pages 151–158, Shaker Heights, Ohio, 1971.

[CS95] J.-Y. Cai and D. Sivakumar. The resolution of a Hartmanis conjecture. In *Proceedings 36th annual Symposium on Foundations of Computer Science*, pages 362–373, Milwaukee, Wis, 1995. IEEE Computer Societey Press.

[FFK92] S. Fenner, L. Fortnow, and S.A. Kurtz. The isomorphism conjecture holds relative to an oracle. In *Proc. 33rd IEEE Symposium Foundations of Computer Science*, pages 30–39, 1992.

[FHOS93] S. Fenner, S. Homer, M. Ogiwara, and A. Selman. On using oracles that compute values. In *STACS 1993, Lecture Notes in Computer Science 665*, 1993.

[FK95] L. Fortnow and M. Kummer. Resource bounded instance complexity. In *Proceedings STACS, Springer Lecture Notes in Computer Science*, volume 900, pages 597–608, 1995.

[For79] S. Fortune. A note on sparse complete sets. *SIAM J. Comput.*, 8:431–433, 1979.

[For96] L. Fortnow. Personal communication. 1996.

[Fu93] B. Fu. With quasi-linear queries, *EXP* is not polynomial time Turing reducible to sparse sets. In *Proc. Structure in Complexity Theory 8th annual conference*, pages 185–193, San Diego, California, 1993. IEEE computer society press.

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

[Har83] J. Hartmanis. On sparse sets in NP-P. *Info. Proc. Let.*, 16:53–60, 1983.

[HKR93] S. Homer, S. Kurtz, and J. Royer. A note on many-one and 1-truth table complete sets. *Theoretical Computer Science*, 115(2):383–389, July 1993.

[HM93] S. Homer and S. Mocas. Nonuniform lower bounds for exponential time classes. manuscript, 1993.

[Hom90] S. Homer. Structural properties of nondeterministic complete sets. In *Proc. Structure in Complexity Theory 5th annual conference*, pages 3–10, Barcelona, Spain, 1990. IEEE Computer Society Press.

[Hom97] S. Homer. Structural properties of complete problems for exponential time. In L.A. Hemaspaandra and A.L. Selman, editors, *Complexity Theory Retrospective*, volume II, chapter 6, pages 135–150. Springer-Verlag, New York, 1997.

[HOW92] L. Hemachandra, M. Ogiwara, and O. Watanabe. How hard are sparse sets? In *Proc. Structure in Complexity Theory 7th annual conference*, pages 222–238, Boston, Mass., 1992. IEEE Computer Society Press.

[HW94] S. Homer and J. Wang. Immunity of complete problems. *Theoretical Computer Science*, 110(1):119–130, April 1994.

[JL93] D.W. Juedes and J.H. Lutz. The complexity and distribution of hard problems. In *Proc. 34th IEEE Symposium on Foundations of Computer Science*, pages 177–185, 1993.

[JL94] D.W. Juedes and J.H. Lutz. Weak completeness in E and E_2. manuscript, 1994.

[JY85] D. Joseph and P. Young. Some remarks on witness functions for non-polynomial and non-complete sets in NP. *Theoretical Computer Science*, 39:225–237, 1985.

[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.

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

[KMR89] S. Kurtz, S. Mahaney, and J. Royer. The isomorphism conjecture fails relative to a random oracle. In *Proc. 21nd Annual ACM Symposium on Theory of Computing*, pages 157–166, 1989.

[KMR90] S. Kurtz, S. Mahaney, and J. Royer. The structure of complete degrees. In A.L. Selman, editor, *Complexity Theory Retrospective*, pages 108–146. Springer-Verlag, 1990.

[KW95] J. Koebler and O. Watanabe. New collapse consequences of np having small circuits. In *Proceedings ICALP, Springer Lecture Notes in Computer Science*, volume 944, pages 196–207, 1995.

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

[Lev73] L. Levin. Universal sorting problems. *Problemy Peredaci Informacii*, 9:115–116, 1973. in Russian.

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

[LM93] J.H. Lutz and E. Mayordomo. Measure, stochasticy, and the density of hard languages. In *STACS 93, Lecture Notes in Computer Science 665*, pages 38–47, 1993.

[LM94] J.H. Lutz and E. Mayordomo. Cook versus Karp-Levin: Separating reducibilities if NP is not small. In *STACS 1994, Lecture Notes in Computer Science.* Springer-Verlag, 1994.

[Lut87] J.H. Lutz. *Resource-Bounded Category and Measure in Exponential Complexity Classes.* PhD thesis, Department of Mathematics, California Institute of Technology, 1987.

[Lut90] J. Lutz. Category and measure in complexity classes. *SIAM J. Comput.*, 19(6):1100–1131, December 1990.

[Lut94] J.H. Lutz. Weakly hard problems. In *Proc. Structure in Complexity Theory 9th annual conference*, Amsterdam, Holland, 1994. IEEE computer society press.

[LV93] Ming Li and P.M.B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications.* Springer-Verlag, 1993.

[LY90] L. Longpré and P. Young. Cook reducibility is faster than Karp reducibility in NP. *J. Computer and System Sciences*, 41:389–401, 1990.

[Mah82] S. Mahaney. Sparse complete sets for NP: solution of a conjecture of Berman and Hartmanis. *J. Comput. System Sci.*, 25:130–143, 1982.

[May94a] E. Mayordomo. Almost every set in exponential time is p-bi-immune. *Theoretical Computer Science*, 1994. to appear.

[May94b] E. Mayordomo. *Contributions to the study of resource-bounded measure.* PhD thesis, Universitat Politècnica de Catalunya, 1994.

[Myh55] J. Myhill. Creative sets. *Zeit. Math. Log. Grund. Math.*, 1:97–108, 1955.

[Ogi95] M. Ogihara. Sparse P-hard sets yield space-efficient algorithms. In *Proceedings 36th annual Symposium on Foundations of Computer Science*, pages 354–361, Milwaukee, Wis, 1995. IEEE Computer Societey Press.

[OKSW94] P. Orponen, K-I Ko, U. Schöning, and O. Watanabe. Instance complexity. *J. Assoc. Comput. Mach*, 41(1):96–121, 1994.

[Orp90] P. Orponen. On the instance complexity of NP-hard problems. In *Proc. Structure in Complexity Theory 5th annual conference*, pages 20–27, Barcelona, Spain, 1990. IEEE Computer Society Press.

[Pos44] E. Post. Recursively enumerable sets of integers and their decision problems. *Bull. Amer. Math. Soc.*, 50:284–316, 1944.

[Rog95] J. Rogers. The isomorphism conjecture holds and one-way functions exist relative to an oracle. In *Proc. Structure in Complexity Theory 10th annual conference*, pages 90–101, Minneapolis, Minnesota, 1995. IEEE computer society press.

[Sch86] U. Schöning. Complete sets and closeness to complexity classes. *Math. Systems Theory*, 19:29–41, 1986.

[Sel82] A. L. Selman. Analogues of semirecursive sets and effective reducibilities to the study of NP complexity. *Information and Control*, 52(1):36–51, January 1982.

[TFL93] S. Tang, B. Fu, and T. Liu. Exponential time and subexponential time sets. *Theoretical Computer Science*, 115(2):371–381, July 1993.

[Tra70] B. Trakhtenbrot. On autoreducibility. *Doklady Akad. Nauk. SSSR*, 192:1224–1227, 1970. In Russian, translation in Soviet Math. Dokl. 11 (1970), 814–817.

[Tra95] N. Tran. On P-immunitiy of nondeterministic complete sets. In *Proc. Structure in Complexity Theory 10th annual conference*, pages 262–263, Minneapolis, Minnesota, 1995. IEEE computer society press.

[Val82] L. Valiant. Reducibility by algebraic projections. *L'Enseignement mathématique*, 28:3–4, 1982.

[vM96] D. van Melkebeek. Reducing P to a sparse set using a constant number of queries collapses P to L. In *Proc. Structure in Complexity Theory 11th annual conference*, pages 88–96, Philadelphia, Pennsylvania, 1996. IEEE computer society press.

[Wat87] O. Watanabe. A comparison of polynomial time completeness notions. *Theoretical Computer Science*, 54:249–265, 1987.

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

[WT89] O. Watanabe and S. Tang. On polynomial time Turing and many-one completenes in PSPACE. In *Proceedings of the 4th Structure in Complexity Theory Conference*, pages 15–23, Eugene, Oregon, June 1989. IEEE Computer Society Press.

[Yao90] A.C. Yao. Coherent functions and program checkers. In *Proc 22nd ACM Symp. Theory of Computing*, pages 84–94, 1990.

[Yes83] Y. Yesha. On certain polynomial-time truth table reducibilities of complete sets to sparse sets. *SIAM J. Comput.*, 12:411–425, 1983.

Authors addresses

Harry Buhrman
Centrum voor Wiskunde en Informatica,
Kruislaan 413, 1098 SJ Amsterdam
email: `buhrman@cwi.nl`

Leen Torenvliet
University of Amsterdam, Department of Computer Science
Plantage Muidergracht 24, 1018 TV Amsterdam
email: `leen@wins.uva.nl`