Collapsing Polynomial-Time Degrees

Klaus Ambos-Spies[1], Levke Bentzien[1], Peter A. Fejer[2], Wolfgang Merkle[1], and Frank Stephan[1]*

[1] Universität Heidelberg, Mathematisches Institut, Im Neuenheimer Feld 294, D-69120 Heidelberg, Germany
ambos|bentzien|merkle|fstephan@math.uni-heidelberg.de
[2] University of Massachusetts at Boston, Department of Mathematics and Computer Science, Boston, MA 02125-3393, USA fejer@cs.umb.edu

Abstract. For reducibilities r and r′ such that r is weaker than r′, we say that the r-degree of A, i.e., the class of sets which are r-equivalent to A, collapses to the r′-degree of A if both degrees coincide. We investigate for the polynomial-time bounded many-one, bounded truth-table, truth-table, and Turing reducibilities whether and under which conditions such collapses can occur. While we show that such collapses do not occur for sets which are hard for exponential time, we have been able to construct a recursive set such that its bounded truth-table degree collapses to its many-one degree. The question whether there is a set such that its Turing degree collapses to its many-one degree is still open; however, we show that such a set – if it exists – must be recursive.

1 Introduction and Notation

1.1 Introduction

Ladner, Lynch and Selman [12] first compared the strength of the polynomial-time reducibilities. For the most common reducibilities – namely Turing (p-T), truth-table (p-tt), bounded truth-table (p-btt), bounded truth-table of norm 1 (p-btt(1)), many-one (p-m) and one-one (p-1) – they showed that

$$\text{p-1} \prec \text{p-m} \prec \text{p-btt(1)} \prec \text{p-btt} \prec \text{p-tt} \prec \text{p-T} \ ,$$

where r $\prec r'$ indicates that r is strictly stronger than r′, i.e, r-reducibility implies r′-reducibility while there are sets A and B where A is r′-reducible to B, but is not r-reducible to B. Subsequently, these investigations have been refined, with a great deal of attention paid to the completeness notions induced by the different reducibilities for complexity classes like **NP** or **EXP**. For **EXP**, the relations among the completeness notions are fully determined. Namely, Watanabe [18] has shown that p-btt(1)-, p-btt-, p-tt-, and p-T-completeness are mutually different, whereas p-1-, p-m- and p-btt(1)-completeness coincide, as shown

* Klaus Ambos-Spies and Frank Stephan have been supported by the Deutsche Forschungsgemeinschaft under grant AM 60/9-2.

by Berman [4] and Homer, Kurtz and Royer [9]. These results can be carried over to some complexity classes containing **EXP** like e.g. **EXPSPACE** and **NEXP** (see Buhrman [6] or Buhrman and Torenvliet [7] for surveys of these results). For **NP**, the corresponding questions are still completely open, but the Berman-Hartmanis Conjecture [5], which says that – in analogy to Myhill's theorem on the (recursively) m-complete recursively enumerable sets – all p-m-complete sets are p-isomorphic, has led to a number of interesting investigations of the relations among completeness notions for reducibilities. For a survey of results about the Berman-Hartmanis-Conjecture and about similar statements for reducibilities stronger than p-m-reducibility, see [11] and [17], as well as the recent survey [1].

Since, for a complexity class **C** closed under r-reducibility, the r-complete sets for **C** form an r-degree, i.e., an r-equivalence class, the comparison of the completeness notions is closely related to the more abstract question of the possible relations among the degrees $deg_{\mathrm{r}}(A)$ of a set A for the different reducibilities r. For instance, since **EXP** is closed under p-T-reducibility, the above quoted results on completeness for **EXP** just say that, for a p-m-complete set A for **EXP**,

$$deg^{\mathrm{p}}_{1}(A) = deg^{\mathrm{p}}_{\mathrm{m}}(A) = deg^{\mathrm{p}}_{\mathrm{btt}(1)}(A) \subset deg^{\mathrm{p}}_{\mathrm{btt}}(A) \subset deg^{\mathrm{p}}_{\mathrm{tt}}(A) \subset deg^{\mathrm{p}}_{\mathrm{T}}(A) \ .$$

It is natural to ask about the possible relations between the degrees of an arbitrary set. For the reducibilities stronger than p-m, this question has been extensively studied: e.g. Kurtz, Mahaney and Royer [11] have shown that there is a set whose p-m-degree collapses to a single p-isomorphism type. For the intermediate reducibilities, however, this question has been solved only in part. It is not difficult to show that there are sets A – e.g. (appropriately defined) generic sets – such that

$$deg^{\mathrm{p}}_{\mathrm{m}}(A) \subset deg^{\mathrm{p}}_{\mathrm{btt}(1)}(A) \subset deg^{\mathrm{p}}_{\mathrm{btt}}(A) \subset deg^{\mathrm{p}}_{\mathrm{tt}}(A) \subset deg^{\mathrm{p}}_{\mathrm{T}}(A) \ .$$

The more subtle question is, what degree collapses are possible. Besides the above example of the partial collapse for the exponential time sets there is the trivial example of the polynomial-time computable sets A where

$$deg^{\mathrm{p}}_{\mathrm{m}}(A) \subset deg^{\mathrm{p}}_{\mathrm{btt}(1)}(A) = deg^{\mathrm{p}}_{\mathrm{T}}(A) \ .$$

The latter situation extends to the class of all super sparse sets, which provides nontrivial examples for the collapse of a p-T-degree to a p-btt(1)-degree (see Ambos-Spies [2]). So the strongest collapses which can be found in the literature for the polynomial-time reducibilities between p-m and p-T are that of p-btt(1) to p-m (in case of the **EXP**-complete sets) and that of p-T to p-btt(1) (in case of the super sparse sets).

Here, we continue the investigation of the question of which collapses are possible and what sets can collapse. Our main result, which is presented in Sect. 2, shows that there is a btt-m-collapsing set. The question whether there is a tt-m- or even T-m-collapsing set remains open. We also obtain some limiting results, presented in Sect. 3. Here we show that every T-m-collapsing or tt-m-collapsing set – if there are any – has to be computable. For still stronger collapses of p-T or p-tt to p-1 or even polynomial-time length increasing one-one reducibility, we obtain exponential complexity bounds for the collapsing sets. For weaker partial collapses in the range between p-btt(1) and p-T, we obtain bounds on the relative complexity of collapsing sets. In particular we show that if a set is btt-btt(1)-collapsing or tt-btt-collapsing or T-tt-collapsing, then it is not hard for exponential time. In the proofs of these results we use genericity and Kolmogorov complexity, respectively.

Finally, in Sect. 4, we shortly discuss degree collapses for elementary recursive reducibilities. There, we can completely classify the maximum possible collapses.

1.2 Notation

We assume the reader to be familiar with the basic concepts of complexity theory, such as degrees or completeness for a class with respect to some reducibility.

All strings are over the alphabet $\Sigma = \{0, 1\}$. We identify strings with natural numbers via the isomorphism which takes the length-lexicographical ordering on $\{\lambda, 0, 1, 00, \ldots\}$ to the usual ordering on ω, the set of natural numbers. We write s_n for the string which is identified with the natural number n. For subsets A and B of ω, as usual, we let $A \oplus B$ be the unique subset of ω which does not contain the empty string and where for all binary strings x, firstly, $0x$ is in $A \oplus B$ iff x is in A and, secondly, $1x$ is in $A \oplus B$ iff x is in B.

The standard pairing function from ω^2 to ω will be denoted by $\langle ., . \rangle$. Recall that the length of $\langle x, y \rangle$ is at most twice the length of x plus the length of y and that the function $\langle ., . \rangle$ can be computed, as well as inverted, in polynomial time. For a number x, we write $(x)_0$ and $(x)_1$ for the unique numbers with $x = \langle (x)_0, (x)_1 \rangle$. All logarithms are to base 2.

A partial characteristic function is a function from some subset of the natural numbers to $\{0, 1\}$. A partial characteristic function is finite iff its domain is finite. The restriction of a partial characteristic function β to some set I is denoted by $\beta|I$ and similarly we let $P|I := \{\beta|I : \beta \text{ in } P\}$. By $B \restriction x$, we denote the restriction of a set B to $\{0, \ldots, x-1\}$, that is, from $B \restriction x$ we can read off $B(y)$ exactly for the strings y which are strictly less than x.

We write $\alpha \sqsubseteq \beta$ iff the domain of α is contained in the domain of β and α agrees there with β. For partial characteristic functions which are compatible, that is, which agree on the intersection of their domains, $\alpha \sqcup \beta$ is the partial characteristic function which agrees with α and β on their respective domains, and which is undefined otherwise. We identify (binary) strings of length n in the natural way with partial characteristic functions with domain $\{0, \ldots, n-1\}$. Given a partial characteristic function α with domain $\{z_0 < \ldots < z_{m-1}\}$, the string associated with α is the (unique) string β where $\beta(j)$ is equal to $\alpha(z_j)$ for $j = 0, \ldots, m-1$.

Next, we define the polynomial-time reducibilities we will consider and introduce some related notions. A set A is p-Turing reducible to a set B, $A \leq^{\mathrm{p}}_{\mathrm{T}} B$ in symbols, if there is a polynomial-time bounded deterministic oracle Turing machine M which computes A with oracle B. We use p-T as an abbreviation for $\leq^{\mathrm{p}}_{\mathrm{T}}$ and, for example, we will speak of a set A being p-T-reducible to a set B.

Similarly, we use p-r in order to refer to the relation denoted by $\leq_{\mathrm{r}}^{\mathrm{p}}$. Given M and B, we write $M(B)$ for the unique set which is p-T-reducible to B via M, and we write $M(B,x)$ for the value of $M(B)$ at place x. Further, for a p-T-reduction M, we let $Q_B(x,M)$ be the set of queries asked by M on input x for oracle B.

The other reducibilities are obtained by imposing some restrictions on the machine M : A is p-truth-table reducible to B, $A \leq_{\mathrm{tt}}^{\mathrm{p}} B$, if $A \leq_{\mathrm{T}}^{\mathrm{p}} B$ via a machine M with nonadaptive queries, i.e., $Q_X(x,M) = Q_{\emptyset}(x,M)$ for all inputs x and oracles X. Here, we call the function q which maps n to the maximum number of queries on an input of length n, i.e., $q(n) = max\{|Q_{\emptyset}(x,M)| : |x| = n\}$, the norm of the reduction M. Then, A is p-bounded truth-table reducible to B, $A \leq_{\mathrm{btt}}^{\mathrm{p}} B$, if A is truth-table reducible to B via a reduction of constant norm. In particular, A is p-bounded truth-table of norm 1 reducible to B, $A \leq_{\mathrm{btt}(1)}^{\mathrm{p}} B$, if on any input x only one query is asked in the reduction. A p-many-one reduction or p-m-reduction, for short, from A to B is a p-btt(1)-reduction in which the evaluation of the query is positive, i.e. x is in $M(B)$ iff y is in B (for $Q_B(x,M) = \{y\}$). If, moreover, the queries are different for different inputs, then we call a p-m-reduction a p-one-one reduction or p-1-reduction, for short.

For a p-tt-reduction G and a string x, we write $Q(x,G)$ for $Q_{\emptyset}(x,G)$ and, for a set I of strings, we let $Q(I,G)$ be the union of all sets $Q(x,G)$ with x in I. As usual, we describe a p-m or p-1-reduction by giving the polynomial-time function f mapping the input to the query, which in case of a p-1-reduction is one-to-one. Similarly, a p-btt-reduction M of norm k is described by the tuple $(h, g_1, \ldots, g_k)$ consisting of the k selector functions $g_i : \Sigma^* \to \Sigma^*$ selecting the queries of the reduction ($Q_B(x,M) = \{g_1(x), \ldots, g_k(x)\}$) and the evaluator function $h : \Sigma^* \times \Sigma^k \to \Sigma^*$, where $h_x(i_1, \ldots, i_k) = h(x, i_1, \ldots, i_k)$ is the Boolean function used for evaluating the queries on input x, i.e., for all sets B and all x,

$$M(B,x) \;=\; h_x(B(g_1(x)), \ldots, B(g_k(x))) \;\; .$$

Note that all these functions are polynomial-time computable.

A reduction is length-increasing if the input is always shorter than the corresponding queries. For a reducibility r, r-li denotes the restriction of this concept to length-increasing reductions.

We call a reducibility r at least as strong as a reducibility r′, $\mathrm{r} \preceq \mathrm{r}'$, if for all sets A and B, the fact $A \leq_{\mathrm{r}} B$ implies $A \leq_{\mathrm{r}'} B$. If this relation is strict, we say r is stronger than r′ and we write $\mathrm{r} \prec \mathrm{r}'$. (Obviously, p-1 $\preceq$ p-m $\preceq$ p-btt(1) $\preceq$ p-btt $\preceq$ p-tt $\preceq$ p-T and Ladner, Lynch, and Selman [12] have shown that all these relations are strict.)

Note that all reductions r introduced above are reflexive and transitive, whence r-equivalence defined by

$$A =_{\mathrm{r}} B \quad \text{iff} \quad A \leq_{\mathrm{r}} B \text{ and } B \leq_{\mathrm{r}} A$$

is an equivalence relation. The r-equivalence class of A is called the r-degree of A and is denoted by $deg_{\mathrm{r}}(A)$. Then, $\mathrm{r} \preceq \mathrm{r}'$ if and only if $deg_{\mathrm{r}}(A) \subseteq deg_{\mathrm{r}'}(A)$ for *all* sets A. For $\mathrm{r} \prec \mathrm{r}'$, say that a set A is r′-r-collapsing if $deg_{\mathrm{r}}(A) = deg_{\mathrm{r}'}(A)$. We also describe the latter situation by saying that the r′-degree of A collapses to an r-degree. In connection with collapses of polynomial time bounded reducibilities, we usually suppress the mentioning of polynomial time; for example, we say a set is btt-m-collapsing instead of p-btt-p-m-collapsing.

2 Collapsing p-btt and p-m

We will show in this section that there is a set which is btt-m-collapsing. Before we state this result in Theorem 5, we discuss notions and facts to be used in its proof. We represent a Boolean function τ in n variables by the concatenation of its 2^n function values $\tau(0,\dots,0)$ through $\tau(1,\dots,1)$. For fixed n, this representation is a bijection between Boolean functions in n variables and strings of length 2^n and thus, in particular, there are 2^{2^n} such functions. By this representation Boolean functions become linearly ordered by the usual length-lexicographical ordering on strings and accordingly it makes sense to speak of the jth Boolean function in n variables for $j = 1,\dots,2^{2^n}$.

For a string α of length n, we denote by α^* the string of length 2^{2^n} where $\alpha^*(j)$ is the result of applying the jth Boolean function in n variables to (the assignment of its variables given by) α. We say a finite partial characteristic function β with domain of size m has the *single query property* if its associated string has the form $\alpha\alpha^*0^k$ where the length n of α is maximal with $n + 2^{2^n} \leq m$. In this situation, informally, we refer to α and to α^*0^k as the independent and the dependent part of β, respectively.

Lemma 1. *Let B be the set of partial characteristic functions with domain $\{z_1 < \ldots < z_m\}$ which have the single query property. Then, for every given Boolean function τ in m variables, we can compute an index i_τ such that for all β in B, the value $\beta(z_{i_\tau})$ is just the result of applying τ to (the string associated with) β.*

Proof. By definition of the concept single query property, there is some n such that for each member of B, its associated string is of the form $\alpha\alpha^*0^k$ for some α of length n. Thus, for every z_i, there is a Boolean function τ_i in n variables such that for all β in B, the value of β at z_i is equal to the result of applying τ_i to $\beta(z_1)$ through $\beta(z_n)$. For $i = 1,\dots,m$, we substitute τ_i for the ith variable in τ and obtain a Boolean function in variables z_1 through z_n. But then, again by definition of single query property, there is some j such that for all β in B, the result of applying τ to β is given by $\beta(z_j)$. □

In connection with Definition 2, recall from the introduction that $B|I$ denotes the set $\{\alpha|I : \alpha \text{ in } B\}$ of restrictions of elements in B to domain I.

Definition 2. *A set B of partial characteristic functions is* free *over a set I of natural numbers iff the set $B|I$ contains all partial characteristic functions with domain I.*

Lemma 3 (Sauer). *Let B be a set of partial characteristic functions which all have identical domain I of finite cardinality h. Then, for every k in $\{1,\dots,h\}$,* *if there is no set of size k over which B is free we have*

$$|B| \ \leq \ \binom{h}{0} + \binom{h}{1} + \ldots + \binom{h}{k-2} + \binom{h}{k-1} \ . \tag{1}$$

Here, as usual, $\binom{h}{j}$ denotes the binomial coefficient $\frac{h!}{(h-j)! \cdot j!}$.

For a proof of the lemma, see Sauer [14].

Corollary 4. *Let B be a nonempty set of partial characteristic functions which all have identical domain I of size $h \geq 2$. Then, B is free over a set of cardinality $\lfloor \frac{\log |B|}{\log h} \rfloor$.*

Proof. We let $k = \lfloor \frac{\log |B|}{\log h} \rfloor$. In case $k = 0$, we are done because the nonempty set B is free over the empty set by definition. In case $k = 1$, we are done because if B were not free over a set of cardinality 1, then B would be a singleton, which would contradict $k = 1$. In case $k \geq 2$, we show that (1) is false for k, from which it is then immediate by Sauer's lemma that there is a free set of size k. By definition of k, we have $k \log h \leq \log |B|$, and raising both sides to a power of two yields $h^k \leq |B|$. But in the right-hand side of (1), each of the $k-1$ terms $\binom{h}{1}$, $\binom{h}{2}$, $\ldots$, $\binom{h}{k-1}$ is at most h^k/k, and the first term 1 is strictly less than h^k/k, whence the right-hand side of (1) evaluates to strictly less than h^k, and thus to strictly less than $|B|$. □

Theorem 5. *Let* $\mathbf{C}$ *be a recursively presentable class. Then, there is a recursive set A not in* $\mathbf{C}$ *such that the* p-btt-*degree of A collapses to its* p-m-*degree.*

Proof. In constructing the set A, we will consider disjoint intervals $I_1, I_2, \ldots$, where I_s contains the first $l_s = s + 2^{2^s}$ strings of length n_s. During stage s of the construction, we will specify the restriction of A to I_s, whereas outside the intervals I_s, the set A will be empty. The sequence $n_0, n_1, \ldots$ increases fast enough to satisfy several technical conditions. In particular, firstly, specifying A on some interval does not interfere with any diagonalizing action taken during previous intervals, and, secondly, l_s is so small compared to n_s that in time polynomial in n_s we can perform all kinds of exhaustive searches related to the interval I_s. We will choose the n_s such that (2) through (4) hold. Here, g is a fast-growing recursive function, which will be defined below, and iv is the function where $iv(x) = s$, in case x is in interval I_s, and $iv(x) = 0$, if x is in none of the I_s.

For all s, we have $g(n_s) < n_{s+1}$. (2)

There is an algorithm for deciding A which, for all s, uses at most n_{s+1} computation steps on every input of length at most n_s. (3)

The function iv is computable in polynomial time. (4)

Next, and before looking at the details of the construction, we show that by applying the recursion theorem we can indeed find a sequence $n_0, n_1, \ldots$ as required. For all e, let T_e be the e-th Turing machine, let φ_e be the partial recursive function computed by T_e, and let $\Phi_e(x)$ be the running time of T_e on input x. Now we assume that an index e is given as an input to the construction and we define a partial function $\widetilde{\rho}$ by

$$\widetilde{\rho}(e,n) \quad = \quad \max\left[\{\Phi_e(x) : |x| \leq n\} \cup \{g(n), n+1\}\right] \quad . \tag{5}$$

Here $\widetilde{\rho}(e,n)$ is defined iff $\Phi_e(x)$ is defined for all x of length at most n.

Further, we let ρ be a partial recursive function such that, firstly, $\rho(e,n)$ is always larger than $\widetilde{\rho}(e,n)$ and, secondly, ρ is time-constructible in the sense that it is computed by a Turing machine such that whenever $\rho(e,n)$ is defined, the computation terminates in at most $c \cdot \rho(e,n)$ steps for some constant c. In order to obtain such a function ρ, we fix a Turing machine T which computes $\widetilde{\rho}$. Then we construct a new Turing machine which basically works like T, but writes the output of T to a special tape while appending a 1 to its output tape for each step of T.

The construction of the set A is then done in stages. In an initialization step, we let $n_0 = 0$ and $A(\lambda) = 0$, and during each stage $s = 1, 2, \ldots$, we first try to compute $n_s = \rho(e, n_{s-1})$ and on success, we fix the restriction of A to strings of length at most n_s according to the effective procedure described below. In general, the construction might get stuck because for the given e, the value $\rho(e, n_{s-1})$ might be undefined for some stage s. This cannot happen, however, if we assume that e is a fixed point of the construction, that is, if φ_e agrees with A on all arguments. More precisely, if we enter stage s at all, then the values $n_1, \ldots, n_{s-1}$ and the restriction of A to strings of length at most n_{s-1} have already been fixed. But then, in case that e is a fixed point of the construction, φ_e and Φ_e are defined for all strings of length at most n_{s-1} and consequently $\rho(e, n_{s-1})$ is also defined.

By the recursion theorem, we can assume that the input e to the construction is indeed a fixed point. Accordingly, we write r for the recursive function $\rho(e, .)$ and at the beginning of the construction we define a sequence $n_0, \ldots$ by $n_0 = 0$ and $n_{s+1} = r(n_s)$. By the definition of r, it is immediate that this sequence satisfies (2) and (3). So it remains to show that the function iv is indeed computable in polynomial time. Here, it suffices to show that for an input x of length n we can compute in time polynomial in n the maximum s_0 among all s with $n_s \leq n$, together with the value n_{s_0} itself because if $n_{s_0} < n$, then $iv(x)$ must be 0 and if $n_{s_0} = n$, then we have simply to decide whether x is among the first l_{s_0} strings of length n. Now the sequence $n_0, n_1, \ldots$ is strictly monotonic and consequently we have $s_0 \leq n$. Further, the function r is time-constructible and thus $n_{s+1} = r(n_s)$ can be computed from n_s in time $c \cdot n_{s+1}$ for some constant c and the whole sequence $n_1 = r(0), n_2 = r(r(0)), \ldots, n_{s_0}$ can be computed in at most $c \cdot n^2$ steps. So, in order to compute s_0, we simply have to compute successively $n_0, n_1, \ldots$ for a total of $c \cdot n^2$ steps, and on timeout we know that n_{s_0} is the maximal value $n_s \leq n$ which has been computed.

We proceed with describing the stages of the construction. Outside the intervals I_s, the set A will always be empty and so during stage s we just have to specify A on the interval I_s. On entering stage s, we let $P(s,1)$ be the set of all partial characteristic functions β with domain I_s such that β has the single query property. Then, during stage s, the restriction α_s of A to I_s will be chosen in $P(s, 1)$. By the latter condition and by the choice of the intervals I_s, every p-T-reduction to A can be replaced by a p-m-reduction. Here in addition, we exploit the fact that the function g will be chosen to grow so fast that for all $s \geq 1$ we have $2^{l_s} \leq n_s$; see (11) below.

Claim 1. Every set which is p-T-reducible to A is also p-m-reducible to A.

Proof. We show first that every p-T-reduction M to A can be replaced by a p-tt-reduction. The set A does not contain elements outside the intervals I_s and each interval I_s contains exactly the first l_s strings of length n_s, where l_s is less than n_s. Consequently, for each length m, there are at most m nontrivial queries to A of length m. Assuming that the running time of M is bounded by $q(n)$ for some polynomial q, we infer that M on an input of length n can only query strings of length at most $q(n)$. By the preceding discussion we can replace M by an equivalent machine which on an input of length n first queries nonadaptively the at most $q(n)^2$ nontrivial queries of length at most $q(n)$ and then simulates M while using the answers to these queries instead of the oracle.

Next, given an arbitrary but fixed truth-table reduction G, with running time bounded by a polynomial p, we construct a polynomial time computable function h such that for all x, we have $G(A, x) = A(h(x))$. Recall that $Q(x, G)$ is the set of places queried by G on input x. According to (4), checking whether a query z in $Q(x, G)$ is an element of some interval I_s can be done in time polynomial in $|z|$, and hence in time polynomial in $|x|$. In case $Q(x, G)$ intersects none of the intervals I_s, we evaluate $G(\emptyset, x)$ and we map x to some fixed member of A in case the result is 1, and to some fixed string not in A, otherwise. In case $Q(x, G)$ intersects I_s for some s, we let s_0 be maximal with this property. For each query z in $Q(x, G)$ which is not in I_{s_0} we can compute $A(z)$ in time polynomial in n_{s_0} due to (3) and because A is empty outside the intervals I_s. As a consequence all such queries can be answered in time polynomial in n_{s_0}, and hence in time polynomial in $|x|$. We hard-wire these answers into the truth-table computed by G on input x and obtain a Boolean function τ which depends only on α_{s_0}. Now, α_{s_0} has the single query property and, as in Lemma 1, we obtain a y in I_{s_0} such that $\alpha_{s_0}(y)$ agrees with the result of applying τ to α_{s_0}. So we can simply let $h(x) = y$. We leave it to the reader to show that y in fact can be computed in time polynomial in $|x|$ by evaluating τ at all the 2^{s_0} many possible assignments for the independent part of α_{s_0}. □

Each stage s of the construction consists of s substages $e = 1, \ldots, s$. During the substages we cancel elements in $P(s, 1)$ and eventually α_s is set equal to the least element left. We let A_s be the set which agrees on $I_1, \ldots, I_s$ with $\alpha_1, \ldots, \alpha_s$, respectively, and which is empty, otherwise. For a set X and a partial characteristic function β, we denote by $\langle X, \beta \rangle$ the set which agrees with β on the domain of β, and agrees with X, otherwise. Then, given a partial characteristic function β with domain I_s, the set $\langle A_{s-1}, \beta \rangle$ agrees with A up to strings of length $n_{s+1} - 1$ iff we decide during stage s that α_s shall be β. For all s and

$e = 1, \ldots, s$, we denote by $P(s,e)$ the subset of $P(s,1)$ which is left at the beginning of substage e of stage s.

In order to make the set A differ from all sets in $\mathbf{C}$, we choose a sequence $C_1, C_2, \ldots$ which witnesses that $\mathbf{C}$ is recursively presentable. Then, for all e, we will satisfy the diagonalization requirement

$$D^e: \quad A(x) \neq C_e(x) \text{ for some } x \ .$$

In order to do so, during substage e of stage s, we check whether, firstly, D^e has not already been met by action at an earlier stage and, secondly, we have a chance to satisfy D^e at stage s, that is, some element β of $P(s,e)$ differs from C_e on I_s. If so, we let $P(s,e+1) = \{\beta\}$.

In order to collapse the p-btt-degree of A to its p-m-degree, whenever A and a set B are mutually p-btt-reducible, then A and B have to be mutually p-m-reducible. By Claim 1 this condition is equivalent to

$$B \leq^{\mathrm{p}}_{\mathrm{m}} A \text{ and } A \leq^{\mathrm{p}}_{\mathrm{btt}} B \text{ together imply } A \leq^{\mathrm{p}}_{\mathrm{m}} B \ . \tag{6}$$

We let $F_0, F_1, \ldots$ and $G_0, G_1, \ldots$ be an appropriate effective enumeration of all p-m-reductions and all p-btt-reductions, respectively. In order to satisfy (6), for every encoded pair $e = \langle i,j \rangle$, $e > 0$, we try to ensure that A is not reducible to $B := F_i(A)$ via G_j, or, equivalently, we try to make A different from $H_e(A)$, where H_e is the composition of G_j with F_i, i.e., $H_e(X) = G_j(F_i(X))$ for all sets X. During the construction, for all such e, we try to satisfy the requirement

$$R^e: \quad A(x) \neq H_e(A,x) \text{ for some } x \ .$$

Similar to the case of the requirements D^e, during substage e of stage s, we check whether, firstly, R^e has not already been met by action at an earlier stage and, secondly, we have a chance to satisfy R^e at stage s, that is, some β in $P(s,e)$ differs from the restriction of $H_e(\langle A_{s-1}, \beta \rangle)$ to I_s. If so, we let $P(s,e+1) = \{\beta\}$. In case we cannot find such β, that is, in case we have

$$\beta \quad = \quad [H_e(\langle A_{s-1}, \beta \rangle)] \,|\, I_s \qquad \text{for all } \beta \text{ in } P(s,e) \ , \tag{7}$$

we try to maintain a p-m-reduction from A to the set $F_i(A)$ via cancelling elements of $P(s,e)$. Formally, in the latter case we let $P(s,e+1)$ be equal to a subset $P''(s,e)$ of $P(s,e)$.

A single stage of the construction together with its substages is summarized in Fig. 1. Here, each iteration of the for-loop corresponds to a single substage and during a substage the action taken is determined by the first applicable case. We postpone the description of the way in which the sets $P''(s,e)$ are chosen until the end of the verification of the construction; however, even before then, we will use the fact that, for all e and k, there is a k_0 such that for all stages s where $P(s,e+1)$ is set equal to $P''(s,e)$ according to Case E of substage e, we have

$$|P(s,e)| \quad \geq \quad k_0 \qquad \text{implies} \qquad |P''(s,e)| \quad \geq \quad k \ . \tag{8}$$

Stage s of the construction

$*$ *We assume that A_{s-1}, $P(s,1)$ and I_s have been fixed before.*

for $e = 1, \ldots, s$

Case A: $P(s,e)$ is a singleton;
$P(s,e+1) := P(s,e)$;

Case B: Case B has not applied at substage e of any stage $k < s$;
choose β in $P(s,e)$ where $\beta \neq C_e|I_s$;
$P(s,e+1) := \{\beta\}$;

Case C: Case D applied at substage e of some stage $k < s$;
$P(s,e+1) := P(s,e)$;

Case D: $\beta \neq H_e(\langle A_{s-1}, \beta\rangle)|I_s$ for some β in $P(s,e)$;
choose such β;
$P(s,e+1) := \{\beta\}$;

Case E: otherwise;
$P(s,e+1) := P''(s,e)$;

let α_s be the least element of $P(s,s+1)$.

Fig. 1. The construction and its stages.

Before we verify the construction, we have to fix the function g for which we have required $g(n_s) \leq n_{s+1}$ in the inductive definition of the n_s. For all n, we let $g(n)$ be the least number such that

$$\max\{|y| : y \text{ in } Q(x, H_e) \text{ where } |x| = n, e \leq n\} \quad < \quad g(n) \ , \tag{9}$$

$$2^{2^{2^{n+2}}} \quad \leq \quad g(n) \ , \tag{10}$$

and in addition, if $n > 0$, then $g(n) > g(n-1)$. By definition of g and the n_s we have $n_1 \geq 2^{16}$ and $n_0 < n_1 < \ldots$, whence $s \leq n_{s-1}$ for all $s \geq 2$. We obtain $n_1 > 2^{l_1}$ and for all $s \geq 2$, we infer

$$2^{l_s} = 2^{\left(s+2^{2^s}\right)} \leq 2^{\left(2^{2^{s+1}}\right)} \leq 2^{\left(2^{2^{n_{s-1}+1}}\right)} < g(n_{s-1}) \leq n_s \ , \tag{11}$$

where the relations hold, from left to right, by definition of l_s, because the exponent is increased, by $s \leq n_{s-1}$, by (10), and finally by the choice of n_s.

Claim 2. For all e and s, if Case D applies during substage e of stage s, then R^e is met.

Proof. Fix e and s, and assume that Case D applies during substage e of stage s. Then, we have $e \leq s$ and $P(s, e+1)$ will just contain the witness β found while evaluating the condition of Case D. Further, for k equal to $e+1$ through s, Case A applies during substage k and $P(s, k+1)$ will also be set equal to $\{\beta\}$, whence, by choice of β, the requirement R^e will be satisfied for some x in I_s, but with A_s in place of A. Now, the set A will agree with A_s on all strings of length strictly less than n_{s+1}. Thus, since x is in I_s, $A(x) = A_s(x)$. Also, by (9) and due to $e \leq s \leq n_s$, the computation of H_e on inputs of length n_s depends only on strings of length strictly less than $g(n_s) < n_{s+1}$, whence $H_e(A, x) = H_e(A_s, x)$, so R^e is met. □

Claim 3 can be shown similarly to Claim 2, where we use the fact that if Case B is reached, then $P(s, e)$ has at least two distinct elements, which disagree at some x in I_s, and thus one of them differs from C_e at x.

Claim 3. For all e and s, if Case B applies during substage e of stage s, then D^e is met.

Claim 4. For all e and for each of the Cases B and D, there is at most one stage s such that this case applies during substage e of stage s.

Proof. This is immediate from the construction. □

Claim 5. For all e and all k, the cardinality of $P(s, e)$ is greater than or equal to k for almost all s.

Proof. We show the assertion by induction on e. For $e = 1$, the assertion holds by choice of $P(s, 1)$. In the induction step, we observe that by Claim 4 for almost all stages s, $P(s, e+1)$ is set equal to $P(s, e)$ or is set equal to $P''(s, e)$. Then for given k, we choose $k_0 \geq k$ as in (8). Now we are done, because, by the induction hypothesis, for almost all stages s, we have $|P(s, e)| \geq k_0$ and for all these stages, we have $|P''(s, e)| \geq k$, by choice of k_0. □

Claim 6. For all e, D^e is met.

Proof. Given e, by Claim 5, there is a stage $s \geq e$ where $P(s, e)$ is not a singleton. Then, either Case B applies at substage e of some stage $k < s$ or else Case B applies at substage e of stage s. By Claim 3, D^e is met. □

Claim 7. For all e, either for almost all stages Case C applies during substage e or for almost all stages Case E applies during substage e.

Proof. By the previous claims, each of the Cases A, B, and D can apply only finitely often during a substage e and so for some s_0 and all stages $s > s_0$, either Case C or Case E applies. Now we are done because either Case C applies never during substage e of a stage $s > s_0$ or Case C applies once and then also for all later stages. □

Claim 7 shows that for all e, either we satisfy requirement R^e, that is, we diagonalize against the pair of reductions given by $e = \langle i, j \rangle$, or, for almost all stages s, α_s is chosen from $P''(s, e)$. We will now choose the sets $P''(s, e)$ such that in the latter case A is indeed p-m-reducible to $B = F_i(A)$. Together with Claim 1, this shows that for all encoded pairs $e = \langle i, j \rangle$ either we successfully diagonalize against A and B being in the same p-btt-degree via the reductions F_i and G_j or we are able to put them in the same p-m-degree.

We will formulate the choice of the sets $P''(s, e)$ in terms of functions Free and Unique. First, given a set B of partial characteristic functions with common finite domain I, among all sets over which B is free, we consider the sets with maximal size and order them. We write Free(B) for the least set in this ordering and we call this set the designated free set of B. Here, while ordering the maximal size sets, we identify a set $\{x_1 < \ldots < x_l\}$ with the length l string $x_1 \ldots x_l$ over the alphabet ω and we apply the lexicographic ordering on such strings. Second, given a set B of partial characteristic functions with common finite domain I and a subset J of I, in general for each element γ of $B|J$ there will be several elements of B which agree on J with γ. We let Unique(B, J) be the subset of B which contains for each such γ exactly the maximal such β, that is, the partial characteristic function which is maximal among all such β with respect to the lexicographic ordering on the associated strings.

Recall that for $e = \langle i, j \rangle$ and for a set I of strings, the set $Q(I, H_e)$ contains all queries made by the reduction H_e on arguments in I or, equivalently, contains all images under the p-m-reduction F_i of the strings queried by the p-btt-reduction G_j on inputs in I. Observe that in case G_j has norm c, then for all i and $e = \langle i, j \rangle$, we have

$$|Q(I, H_e)| \;\leq\; c \;\cdot\; |I| \;\; . \tag{12}$$

For all stages s and all corresponding substages k, we let $J(s, k)$ be the designated free set of $P(s, k)$. Note that during the construction the cardinality of the sets $J(s, k)$ will decrease in k; however, in general $J(s, k + 1)$ will not be contained in $J(s, k)$. We leave it to the interested reader to show that $J(s, 1)$ corresponds to the independent part of the elements in $P(s, 1)$. For all stages s and all corresponding substages e, we let

$$Q(s, e) \;=\; Q(J(s, e), H_e) \;, \qquad\qquad J'(s, e) \;=\; \mathrm{Free}(P(s, e)|Q(s, e)) \;\; .$$

Intuitively speaking, $J'(s, e)$ is the designated free set of $P(s, e)$ if we count only free sets which are subsets of $Q(s, e)$. Further, we let

$$P'(s, e) \;=\; \mathrm{Unique}(P(s, e), J'(s, e)) \;\; .$$

The set $P'(s, e)$ is free over $J'(s, e)$ and has cardinality $2^{|J'(s,e)|}$, whence $J'(s, e)$ is a maximal-size free set of $P'(s, e)$. In general, however, $J'(s, e)$ need not be the designated free set of $P'(s, e)$ because there might be lesser free sets of the same cardinality. Finally, in case $|J'(s, e)| \geq 2$, we let

$$P''(s, e) \;=\; \{\beta \text{ in } P'(s, e) : \beta|J'(s, e) \text{ has the single query property } \}$$

and let $J''(s,e)$ be the set which corresponds to the independent part of the elements in $P''(s,e)|J'(s,e)$ and, otherwise, we let $P''(s,e)$ just contain the least element in $P'(s,e)$ and let $J''(s,e)$ be equal to the empty set. The set $J''(s,e)$ is a maximal-size free set, but not necessarily the designated free set, of $P''(s,e)$, as follows by the same argument as in the case of $J'(s,e)$.

Claim 8. For all s and for $e = 1, \dots, s+1$, the cardinality of $P(s,e)$ is $2^{|J(s,e)|}$.

Proof. For every given s, we show the claim by induction on e. For $e = 1$, the assertion follows from the choice of $P(s,1)$. Now assume $e \geq 2$ and consider the definition of $P(s,e)$ according to the cases of substage $e-1$ of stage s. If Case A, B, or D applies, then $P(s,e)$ is a singleton and we are done. If Case C applies, $P(s,e)$ is set equal to $P(s,e-1)$ and the assertion follows by the induction hypothesis. Finally, if Case E applies, the assertion follows by definition of $P''(s,e-1)$. □

We still have to show the assertion of Claim 9 which has already been used in the verification of the construction via referring to (8).

Claim 9. For all e and k, there is k_0 such that for all stages s where Case E applies during substage e, the fact $|P(s,e)| \geq k_0$ implies $|P''(s,e)| \geq k$.

Proof. The set $P''(s,e)$ is free over $J''(s,e)$ and thus in order to ensure that $P''(s,e)$ contains at least k elements, it suffices to ensure $|J''(s,e)| \geq \log k$. Now $J''(s,e)$ corresponds to the independent part of partial characteristic functions with domain $J'(s,e)$, and thus in turn $|J''(s,e)| \geq \log k$ holds if $|J'(s,e)| \geq k_1$ for some appropriate constant k_1.

By assumption we only have to consider stages s where Case E applies, whence in particular Case D does not apply and for all β in $P(s,e)$, β agrees on I_s, and hence also on $J(s,e)$, with $H_e(\langle A_{s-1}, \beta\rangle)$. But, for inputs in $J(s,e)$, the reduction H_e queries only strings in $Q(s,e)$ and consequently for all β_0 and β_1 in $P(s,e)$ which differ on $J(s,e)$, the sets $\langle A_{s-1}, \beta_0\rangle$ and $\langle A_{s-1}, \beta_1\rangle$ must differ on $Q(s,e)$. Consequently, the set

$$E \;=\; \{(\langle A_{s-1}, \beta\rangle | Q(s,e)) : \beta \text{ in } P(s,e)\}$$

must have at least as many elements as $P(s,e)|J(s,e)$. But $P(s,e)$ is free on $J(s,e)$ and so the cardinality of E is at least $2^{|J(s,e)|}$. We leave it to the reader to show that the free sets of E and of $P(s,e)|Q(s,e)$ coincide and that thus $J'(s,e)$, which is by definition a maximal-size free set of the latter one, is also a maximal-size free set of E. We infer

$$|J'(s,e)| \;\geq\; \left\lfloor \frac{\log |E|}{\log |Q(s,e)|} \right\rfloor \;\geq\; \left\lfloor \frac{|J(s,e)|}{\log c + \log |J(s,e)|} \right\rfloor \tag{13}$$

where c is the norm of the reduction G_j encoded by $e = \langle i, j\rangle$. The first inequality in (13) follows by Corollary 4 and because $J'(s,e)$ is a maximal-size free set of E, while the second inequality is obtained by substituting the bounds for the size of E and of $Q(s,e)$ from the preceding discussion and from (12), respectively.

According to (13), in order to ensure $|J'(s,e)| \geq k_1$ it suffices to choose $|J(s,e)|$ large enough, that is, by Claim 8, it suffices if $|P(s,e)| \geq k_0$ for some appropriate constant k_0. By choosing the size of $P(s,e)$ large enough, say, at least 3, we will also achieve that $Q(s,e)$ has at least two elements and thus while showing (13) the assumption of Corollary 4 is indeed satisfied. □

In connection with Claim 10, for all $e > s+1$, we let $P(s,e) = P(s,s+1)$ and $J(s,e) = J(s,s+1)$.

Claim 10. For all $e \geq 1$, there is a polynomial time algorithm which, given an input of length n_s, $s \geq 1$, computes $J(s,e)$ and a list of the the graphs of the partial characteristic functions in $P(s,e)$.

Proof. We exploit that for $s \geq 1$, $2^{l_s} \leq n_s$ according to (11) and that thus all kinds of exhaustive searches on $P(s,e)$ and I_s can be done in time polynomial in n_s. The proof is by induction on e. In case $e = 1$, the claim is true by choice of $P(s,1)$. In case $e \geq 2$, by Claim 7, either for almost all stages $P(s,e)$ is set to $P(s,e-1)$ or for almost all stages $P(s,e)$ is set to $P''(s,e-1)$. In the former case, we are done by the induction hypothesis, while in the latter case we argue that, firstly, $P(s,e-1)$ can be computed in polynomial time according to the induction hypothesis while, secondly, l_s is so small compared to n_s that from $P(s,e-1)$ we can compute $P''(s,e-1)$ in time polynomial in n_s. We leave the details of the latter argument to the reader. □

Claim 11. The set A is not computable in polynomial time.

Proof. Assume for a contradiction that A were equal to a set D which is computable in polynomial time. Then, there is a p-btt-reduction G_j such that for all sets X we have $D = G_j(X)$. Thus for $e = \langle 0, j \rangle$, $H_e(X) = G_j(F_0(X))$ also agrees with D for all sets X. So we are done in case R^e is met, because then A differs from $D = H_e(A)$. By Claims 4 and 5, there is a stage $s \geq e$ such that neither Case A nor Case B holds at substage e of stage s. If Case D did not apply at substage e of any stage k less than s, then Case D applies at substage e of stage s (because $P(s,e)$ is not a singleton and thus must contain some β which differs on its domain from D, that is, $P(s,e)$ contains a β which satisfies the condition of Case D). Thus, for some e and s, Case D applies at substage e of stage s and, by Claim 2, R^e is met. □

Together with Claim 1, the next claim shows that if B is in the same p-btt-degree as A, then A is p-m-reducible to B.

Claim 12. Let e be a natural number, where $e = \langle i, j \rangle$, and let $B = F_i(A)$. If A is p-btt-reducible to B via G_j, then A is p-m-reducible to B.

Proof. We construct a p-m-reduction h from A to B. By Claim 11, the set B is not computable in polynomial time and thus in particular it does not contain all strings and we can fix y_0 not in B. For a given argument x, in case x is in none of the intervals I_s, we let $h(x) = y_0$. In case x is in interval I_s, we compute

$P(s,e)$ and $J(s,e)$ according to Claim 10. We exploit that for almost all stages s, $\alpha_s = A|I_s$ has been chosen from $P'(s,e)$ and that hence $A(x)$ can be expressed as truth-table condition on $A|J'(s,e)$. Now, this restriction has the single query property, and so in fact we have $A(x) = A(y)$ for some y in $J'(s,e)$. We leave it to the reader to show that such y can in fact be computed in time polynomial in time $n_s = |x|$, where in the proof one exploits that according to (11) the cardinality of I_s, and hence also the cardinality of its subset $J'(s,e)$ is rather small in comparison with n_s. Now, y is in the subset $J'(s,e)$ of $Q(s,e)$, whence by definition of the latter set there is some z in $Q(J(s,e),G_j)$ with $F_i(z) = y$. Again, we can find such z in time polynomial in n_s and so we can simply let $h(x) = z$. □

This completes the proof of Theorem 5. □

The set A constructed in the proof of Theorem 5 bears some resemblance to super sparse sets as introduced in [2]. In particular, it can be shown that in the lower p-T-cone of A the relations $\leq_{\mathrm{T}}^{\mathrm{p}}$ and $\leq_{\mathrm{tt}}^{\mathrm{p}}$ coincide, that is, for all sets B and C which are p-T-reducible to A, we have $B \leq_{\mathrm{T}}^{\mathrm{p}} C$ iff $B \leq_{\mathrm{tt}}^{\mathrm{p}} C$. Consequently, A is not only btt-m-collapsing, but is also T-tt-collapsing.

3 Limiting Results

In this section, we give some consequences of the different types of degree collapses. In particular, we will prove some bounds on the absolute or relative complexity of collapsing sets.

We first review Selman's result [15] that no tally set is T-m-collapsing. By similar but somewhat more involved arguments, we can show that any T-m-collapsing set – if there is any – is recursive and that stronger collapses of the p-T-degree of a set A to its p-1- or even p-1-li-degree imply that A is computable within nondeterministic or deterministic, respectively, exponential time bounds.

Theorem 6. *(Selman [15].) A tally set is not* T-m-*collapsing.*

The proof of Theorem 6 uses the notion of p-selectivity introduced in [15], which is the polynomial-time bounded analog of Jockusch's concept of semi-recursiveness [10]. A set A is p-*selective* if there is a polynomial-time computable function $f : \Sigma^* \times \Sigma^* \to \Sigma^*$ such that $f(x,y)$ is always in $\{x,y\}$ and $f(x,y)$ is in A if $\{x,y\} \cap A \neq \emptyset$. A function f as above is called a p-*selection function* for A. Theorem 6 then is an easy consequence of the following two lemmas.

Lemma 7. *(Selman [15].) If a set is* p-*selective and* btt(1)-m-*collapsing, then it is computable in polynomial time.*

Proof. Suppose that a set B is btt(1)-m-collapsing and is p-selective with selection function sel_B. Since $\overline{B}$ is in the same p-btt(1)-degree as B, $B \leq_{\mathrm{m}}^{\mathrm{p}} \overline{B}$, say via f. Then, x is in B iff $sel_B(x, f(x)) = x$. Thus, B is in $\mathbf{P}$. □

Lemma 8. *(Selman [15].) If A is a tally set, then there is a* p*-selective set in* $deg_T^p(A)$.

Proof. Define $B = \{x : x \leq_{lex} A(0)A(00)A(000)\ldots\}$. Then, B is p-selective via the selection function $sel_B(x, y)$ that returns the lesser of x and y in lexicographic order, and it is easy to see that $B =^{\mathrm{p}}_{\mathrm{T}} A$. □

Proof of Theorem 6. If A is a tally set, then, by Lemma 8, there is a p-selective set B in the p-T-degree of A. But if the latter degree collapses to a p-m-degree, then B satisfies the assumption of Lemma 7, whence B is in $\mathbf{P}$. Thus, A is in $\mathbf{P}$. But no set A in $\mathbf{P}$ is T-m-collapsing, since the p-T-degree of such a set is just $\mathbf{P}$, whereas $\mathbf{P}$ consists of the three p-m-degrees $\{\emptyset\}$, $\{\Sigma^*\}$ and $P - \{\emptyset, \Sigma^*\}$. □

From the proof of Theorem 6, we can isolate the following idea for proving a set not to be T-m-collapsing: for any set S, the set

$$L(S) = \{x : x \leq_{\mathrm{lex}} S(s_0)S(s_1)S(s_2)\ldots\}$$

is p-selective. Moreover, as one can easily check, $L(S)$ is p-T-equivalent to the set $I(S) = \{S(s_0)\ldots S(s_n) : n \geq 0\}$ of finite prefixes of (the characteristic string of) S. Hence, by Lemma 7, any set A which is p-T-equivalent to $I(S)$ for some S is not T-m-collapsing. Now, every tally set A is p-T-equivalent to the set of prefixes of A itself if we omit the trivial instances of A outside of 0^*, i.e., $A =^{\mathrm{p}}_{\mathrm{T}} \{A(0)\ldots A(0^n) : n \geq 0\}$. For an arbitrary set A, however, in general we can't relate the instances of A to finite prefixes of A, since coding $A(y)$ for every hard instance y among the predecessors of a string x might require a string of approximate length $2^{|x|}$. But then, polynomial time does not suffice to extract $A(x)$ from the coded prefixes of A. Still, by relating the elements of A to prefixes of A, we can obtain some limiting results on the complexity of sets which are r-r′-collapsing for r in $\{\mathrm{T}, \mathrm{tt}\}$ and r′ in $\{\mathrm{m}, 1, \text{1-li}\}$. For this sake, we consider the sets introduced in Definition 9, in which an element x of A is combined with prefixes of A of length $|x|$ and length $\lfloor \log(|x|) \rfloor$, respectively.

Definition 9. *For any set A, define*

$$\begin{aligned} LI(A) &= \{\langle A(s_0)\ldots A(s_{n-1}), x\rangle : x \textit{ in } A \textit{ and } x \neq \lambda \textit{ and } n = |x|\} \ , \\ SI(A) &= \{\langle A(s_0)\ldots A(s_{n-1}), x\rangle : x \textit{ in } A \textit{ and } x \neq \lambda \textit{ and } n = \lfloor \log(|x|) \rfloor\} \ . \end{aligned}$$

Lemma 10. *For any set A, $SI(A) =^{\mathrm{p}}_{\mathrm{tt}} A$ and $LI(A) =^{\mathrm{p}}_{\mathrm{T}} A$.*

Proof. Fix A. It is clear that both of $LI(A)$ and $SI(A)$ are p-tt-reducible to A. To see that A is p-tt-reducible to $SI(A)$, it suffices to observe that if $y \neq \lambda$, then y is in A iff $\langle z, y\rangle$ is in $SI(A)$ for some string z of length $\lfloor \log(|y|) \rfloor$, where the number of such strings z is linear in $|y|$.

To see that A is p-T-reducible to $LI(A)$, suppose that a nonnull string y is given. Then, y is in A iff $\langle z, y\rangle$ is in $LI(A)$ for $z = A(s_0)\ldots A(s_{|y|-1})$. So, to compute $A(y)$ from $LI(A)$, it suffices to compute z. This can be done by a straightforward inductive process, since, for all $j > 0$, the string s_j is in A iff $\langle A(s_0)\ldots A(s_{|s_j|-1}), s_j\rangle$ is in $LI(A)$, where $|s_j| - 1 < j$. □

Theorem 11. *Every* tt-m-*collapsing set is recursive.*

Proof. Let A be a tt-m-collapsing set. Then, the set $SI(A)$ and its complement are both in the p-tt-degree of A by Lemma 10 and because p-tt-degrees are closed under complementation. But the p-tt-degree of A collapses to a p-m-degree and thus, in particular, $SI(A)$ is p-m-reducible to its complement, say by the polynomial-time computable function f. For strings $x = \langle y, z\rangle$ and $x' = \langle y', z'\rangle$ let $cp(x, x')$ denote the longest common prefix of y and y'. Note that, for any $x = \langle y, z\rangle$ and for $f(x) = \langle y', z'\rangle$, either x or $f(x)$ is in $SI(A)$. Consequently, at least one of the strings y and y' is extended by the characteristic string of A whence, in particular, $cp(x, f(x))$ is extended by A.

In showing that A is recursive, we distinguish the following two cases. In case the set $\{cp(x, f(x)) : x \text{ in } \Sigma^*\}$ contains arbitrarily long strings, then A is recursive by the following method. Given a string y, compute n with $y = s_n$ and find the least string x with $|cp(x, f(x))| > n$. Then, $A(y) = cp(x, f(x))(n)$. Otherwise, in case for some n we have $|cp(x, f(x))| < n$ for all strings x, we will show that $SI(A)$ is in $\mathbf{P}$. Since $A {=}^{\mathrm{p}}_{\mathrm{m}} SI(A)$, this establishes that A is also in $\mathbf{P}$. Let $\alpha = A(s_0) \dots A(s_{n-1})$. In order to decide whether $x = \langle y, z\rangle$ is in $SI(A)$, compute $f(x) = \langle y', z'\rangle$. Now, by case assumption and choice of n, one of the strings y and y' does not extend α and, accordingly, for one of the strings x or $f(x)$, we can decide membership in $SI(A)$ by looking up a finite table that codes α plus the finitely many strings $\langle u, v\rangle$ in $SI(A)$ for which u does not extend α. This suffices, because x is in $SI(A)$ iff $f(x)$ is not. □

For degree collapses stronger than a collapse of the p-T- or p-tt-degree of a set to its p-m-degree, we can not only show that the set is recursive but we obtain the following complexity bounds.

Theorem 12. *(i) Every* T-1-*collapsing set is in* $\mathbf{NTIME}(2^{O(n)})$.
(ii) Every tt-1-*collapsing set is in* $\mathbf{NTIME}(2^{2^{O(n)}})$.
(iii) Every T-(1-li)-*collapsing set is in* $\mathbf{DTIME}(2^{O(n)})$.
(iv) Every tt-(1-li)-*collapsing set is in* $\mathbf{DTIME}(2^{2^{O(n)}})$.

Proof. We will prove claims (i) and (iii). For a proof of (i), assume that A is T-1-collapsing. Since $A \oplus \Sigma^* {=}^{\mathrm{p}}_{\mathrm{T}} A$ and, by Lemma 10, $LI(A) {=}^{\mathrm{p}}_{\mathrm{T}} A$, the fact that A is T-1-collapsing implies that $A \oplus \Sigma^* \leq^{\mathrm{p}}_{1} LI(A)$, say via the one-to-one polynomial-time computable function f. So $f(1x)$ is in $LI(A)$ for all strings x. By definition of $LI(A)$, this implies that $(f(1x))_0 \sqsubseteq A$ for all strings x. Hence, given s_n, to compute $A(s_n)$ it suffices to find a string x such that

$$|(f(1x))_0| > n \quad . \tag{14}$$

Then, we can read off $A(s_n)$ from $(f(1x))_0$, namely $A(s_n) = (f(1x))_0(n)$. So, in order to show that A is in $\mathbf{NTIME}(2^{O(n)})$ it suffices to prove that for a given string s_n there is a string x of length $O(n)$, hence of length $2^{O(|s_n|)}$, such that (14) holds. But this holds by the following simple counting argument. Since there are 2^n strings x of length n, by injectivity of f, $F_n = \{f(1x) : |x| = n\}$ is a subset of $LI(A)$ of cardinality 2^n. On the other hand, all strings in $LI(A)$ are of the form $\langle (y)_0, (y)_1 \rangle$ where $(y)_0$ is the prefix of the characteristic string of A of length $|(y)_1|$, whence the cardinality of $S_n = \{y \text{ in } LI(A) : |(y)_0| \leq n\}$ is bounded by $2^{n+1} - 1$. So, for some string x of length $n+1$, $f(1x)$ is in $LI(A) - S_n$ and thus (14) holds for x.

For a proof of (iii), by the stronger hypothesis that A is T-(1-li)-collapsing, we may assume that the function f above is length increasing, i.e., $|f(x)| > |x|$ for all strings x. This implies that, given $x = s_n$, the value of $A(s_n)$ can be read off from $f(1y)$ for every sufficiently large string y, whence the nondeterministic search in the proof of (i) becomes superfluous here. To be more precise, recall that we assume that $|\langle u, v \rangle| \leq 2(|u| + |v|)$ for all strings u, v, whence, for $z = \langle u, v \rangle$ in $LI(A)$, $|z| \leq 4 \cdot |u|$ by $|u| = |v|$. It follows that $A(s_n)$ is equal to $(f(11^{4n}))_0(n)$, whence A is in $\mathbf{DTIME}(2^{O(n)})$.

The proofs of parts (ii) and (iv) are similar to those of parts (i) and (iii), respectively. Here, however, we have to replace $LI(A)$ by $SI(A)$. The fact that an element x of $SI(A)$ codes only information about an initial segment of A of size logarithmic in $|x|$ – not of size linear in $|x|$ as in the case of $LI(A)$ – is responsible for the double exponential-time bound for A here. $\square$

For partial collapses in the range between p-T and p-m, we have not been able to prove any absolute complexity bounds for the collapsing sets, but we can give some relative complexity bounds. We will show that, for every pair r and r′ in $\{\text{btt}(1), \text{btt}, \text{tt}, \text{T}\}$ where r′ is stronger than r, an **EXP**-hard set cannot be r-r′-collapsing. In the case of p-T and p-tt, we show this by a Kolmogorov complexity argument, while the other cases, which we will prove first, rely on the existence of certain generic sets in **EXP**. By the latter fact, it then suffices to show that sets above such generic sets are non-collapsing as required.

There is a large variety of resource-bounded genericity concepts in the literature (see [3] for a survey), most of which are suitable for our task. In order to obtain quite general results, we introduce a new concept here, which is a weak variant of Fenner's concept [8] and which is weaker than most of the concepts in the literature (see [3]).

In connection with Definition 13, recall that strings are partial characteristic functions with domain equal to an initial segment of Σ^* and that in particular $X \restriction x$ is a string for all sets X and all strings x.

Definition 13. *A* bounded p-extension function f *is a (total) function f from strings to finite partial characteristic functions which can be computed in polynomial time and such that, for some number k and for all strings α, the domain of $f(\alpha)$ has cardinality at most k and does not intersect the domain of α.*

A set A meets f at x *if $f(A \restriction x) \sqsubseteq A$; and A* meets f *if A meets f at some x. A set G is* p-generic *if G meets every bounded* p*-extension function.*

A p-generic set has all properties which can be ensured by polynomial-time bounded finite extension arguments with extensions of constant length. Here every such property corresponds to a bounded p-extension function. Note that our concept differs from Fenner's original concept by the restriction to extensions of constant length.

We say that f *forces a property* P *at* x if every set which meets f at x has property P. Similarly, we say that f *forces* P if every set which meets f has property P.

Theorem 14. *Let G be* p*-generic and let A be any set.*

(i) If $G \leq^{\mathrm{p}}_{\mathrm{btt}} A$ then A is not btt-btt(1)*-collapsing.*
(ii) If $G \leq^{\mathrm{p}}_{\mathrm{tt}} A$ then A is not tt-btt*-collapsing.*

Proof. We first prove the second part of the theorem and then show how the proof can be modified to obtain the first part.

Fix A and G such that G is p-generic and $G \leq^{\mathrm{p}}_{\mathrm{tt}} A$, and, for a contradiction, assume that A is tt-btt-collapsing. Define B by

$$\begin{aligned} B = \quad & \{1^j 0x : x \text{ in } A \text{ and } 1^j 0x \text{ in } G \text{ and } 2 \leq j \leq |x|\} \\ \cup \quad & \{10x : x \text{ in } A \text{ and } 1^j 0x \text{ is not in } G \text{ for } j = 2, \ldots, |x|\} \ . \end{aligned}$$

It is easily shown that $A \leq^{\mathrm{p}}_{\mathrm{tt}} B$ and $B \leq^{\mathrm{p}}_{\mathrm{tt}} A \oplus G$, whence, by $G \leq^{\mathrm{p}}_{\mathrm{tt}} A$, the sets A, B, and $A \oplus G$ are all p-tt-equivalent. By our assumption that A is tt-btt-collapsing, this implies that $G \leq^{\mathrm{p}}_{\mathrm{btt}} B$. So, for the remainder of the argument, fix a p-btt-reduction $\tau = (h, g_1, \ldots, g_k)$ from G to B. Then, for all x,

$$G(x) = h(x, B(g_1(x)), \ldots, B(g_k(x))) \tag{15}$$

and w.l.o.g. we can assume $g_1(x) < g_2(x) < \ldots < g_k(x)$ for all x. To get the desired contradiction, we will define a bounded p-extension function f which forces the failure of (15) for some x. Then, by p-genericity of G, (15) will actually fail for some x.

For the definition of f, we first introduce some notation isolating some relevant features of the reduction τ. For any string x and any number n, let $Q(x) = \{g_1(x), \ldots, g_k(x)\}$ be the set of queries of τ on input x and let $Q^{\leq n}(x) = \{y \text{ in } Q(x) : \ |y| \leq n\}$ be the set of queries of length at most n. Moreover let

$$\begin{aligned} Q_1(x) &= \{10u \ : \ 10u \text{ in } Q(x)\} \ , \\ Q_2(x) &= \{1^j 0u \ : \ 1^j 0u \text{ in } Q(x) \text{ and } 2 \leq j \leq |u|\} \ . \end{aligned}$$

The following implications, which are immediate by definition of B, show that for every given string y queried by τ we can enforce $B(y) = 0$ via appropriately specifying G.

$$G \cap \{1^j 0u : \ 2 \leq j \leq |u|\} \neq \emptyset \quad \text{implies} \quad B(10u) = 0 \ . \tag{16}$$

$$y \text{ in } Q_2(x) \text{ and } G(y) = 0 \quad \text{implies} \quad B(y) = 0 \ . \tag{17}$$

$$y \text{ in } Q(x) - (Q_1(x) \cup Q_2(x)) \quad \text{implies} \quad B(y) = 0 \ . \tag{18}$$

Let h_x denote the k-ary Boolean function used in τ to evaluate $Q(x)$, i.e., $h_x(i_1, \ldots, i_k) = h(x, i_1, \ldots, i_k)$ for $i_1, \ldots, i_k$ in Σ. Finally, call two strings x

and x' *n-equivalent* if $h_x = h_{x'}$ and $Q^{\leq n}(x) = Q^{\leq n}(x')$. Note that for almost all n, the number $e(n)$ of n-equivalence classes is bounded by

$$e(n) \leq 2^{2^k} \cdot (k+1) \cdot (2^{n+1}-1)^k < 2^{(k+1)n} \ . \tag{19}$$

Namely, there are 2^{2^k} choices for the k-ary Boolean function h_x, $k+1$ choices for the cardinality of $Q^{\leq n}(x)$, since $|Q^{\leq n}(x)| \leq k$, and for each element y of $Q^{\leq n}(x)$ there are at most $2^{n+1}-1$ choices, since $|y| \leq n$.

For the following, fix $n_0 > 2k+2$ such that (19) holds for $n \geq n_0$. Then, for $n \geq n_0$, we may fix x_n and x'_n with $x_n < x'_n$ to be the least pair of n-equivalent strings such that both have length $(k+2)n$ and are in $0\Sigma^*$. (Since the latter set contains $2^{(k+2)n-1} > 2^{(k+1)n}$ strings of length $(k+2)n$, such x, x' exist by (19).)

Now, the idea for the definition of the extension function f forcing the failure of (15) is as follows. Given an initial segment $X \restriction y$ with $|y| = n \geq n_0$ we define $f(X \restriction y)$ in such a way that f forces at y both of

$$G(x_n) \neq G(x'_n) \ , \tag{20}$$

$$h_{x_n}(B(g_1(x_n)), \ldots, B(g_k(x_n))) \ = \ h_{x'_n}(B(g_1(x'_n)), \ldots, B(g_k(x'_n))) \ , \tag{21}$$

whence (15) will fail for $x = x_n$ or $x = x'_n$.

The definition of $f(X \restriction y)$ will only depend on the length $|y|$ of y and not on the particular initial segment $X \restriction y$, i.e., for every n, we define a partial characteristic function γ_n and let $f(X \restriction y) = \gamma_n$ for all strings y of length n. (So, in order to make f total, for the finitely many initial segments $X \restriction y$ with $|y| < n_0$, we can let $f(X \restriction y) = f(X \restriction 0^{n_0})$ for any extension $X \restriction 0^{n_0}$ of $X \restriction y$ up to the string 0^{n_0}.) Here, the domain of γ_n does not contain any string of length less than or equal to n and its cardinality is bounded by some constant which does not depend on n. The definition of γ_n consists of the following three parts, where it is to be understood that γ_n is undefined on all strings for which its value is not explicitly specified here:

Let $\gamma_n(x_n) = 1$ and $\gamma_n(x'_n) = 0$. (22)

For every $10u$ in $Q_1(x_n) \cup Q_1(x'_n)$ with $|10u| > n$, let $\gamma_n(1^{j_u}0u) = 1$, where j_u is the least number j with $2 \leq j \leq |u|$ and 1^j0u not in $Q(x_n) \cup Q(x'_n)$. (23)

For every y in $Q_2(x_n) \cup Q_2(x'_n)$ with $|y| > n$, let $\gamma_n(y) = 0$. (24)

Note that the definition of γ_n is consistent because x_n and x'_n have been chosen in $0\Sigma^*$. To show that f has the desired properties, note that if G meets f at y for some string y of length $n \geq n_0$, then, by (22), $G(x_n) = 1$ differs from $G(x'_n) = 0$, whence (20) holds. Moreover, by (23) and (16), $B(y) = 0$ for all y in $Q_1(x_n) \cup Q_1(x'_n)$ with $|y| > n$, and, by (24) and (17), $B(y) = 0$ for all y in $Q_2(x_n) \cup Q_2(x'_n)$ with $|y| > n$. By (18) and because x_n and x'_n are n-equivalent this implies (21).

It remains to show that f is a bounded p-extension function. But this is immediate by the observations that the domain of γ_n has cardinality at most

$2k+2$ and that γ_n can be computed in $2^{O(n)}$ steps, and hence in time polynomial in the length of the argument string.

For a proof of the first assertion, assume G is p-btt-reducible to A. Then, we can argue as above if we make the following straightforward changes. In the definition of B, use $\{2,3\}$ as range for j instead of $\{2,\ldots,|x|\}$. Then $A{=}^{\mathrm{p}}_{\mathrm{btt}}B$, whence it suffices prove that $A{\not=}^{\mathrm{p}}_{\mathrm{btt}(1)}B$. This is done by showing that G is not p-btt(1)-reducible to B. Just as above, for a contradiction we assume that there is a p-btt(1)-reduction τ from G to B, i.e., that (15) holds for $k=1$. Then, a p-extension function f forcing the failure of (15) is defined as above. We only have to adapt $Q_2(x)$, (16), and (23) to the modified definition of B by shrinking the range of j to $\{2,3\}$. Note that the number j_u required in (23) exists, since the set $Q(x_n)\cup Q(x'_n)$ contains $10u$ and, by $k=1$, has cardinality at most 2, whence it can contain at most one of the strings $110u$ and $1110u$. □

By the existence of p-generic sets in **EXP** (see [8] or [3]), Theorem 14 implies the following corollary.

Corollary 15. *A* p-btt-*hard set for* **EXP** *is not* btt-btt(1)-*collapsing. A* p-tt-*hard set for* **EXP** *is not* tt-btt-*collapsing.*

To establish the analogous result that no p-T-hard set for **EXP** can be T-tt-collapsing, we will apply a different kind of argument. Watanabe shows in [18] that no p-T-complete set for **E** (and hence for **EXP**, which is the downwards closure of **E** under $\leq^{\mathrm{p}}_{\mathrm{T}}$) can be T-tt-collapsing and the proof uses a special kind of coding involving resource-bounded Kolmogorov complexity. Here, we will use a similar coding technique to show that any T-tt-collapsing set which is p-T-hard for **EXP** has to be a member of **EXP** itself. But then, Watanabe's result implies that no such set exists.

We will give only an informal notion of resource-bounded Kolmogorov complexity; for precise definitions see [13]. For a time bound t and a string x, let

$$\begin{aligned}K^t(x) = min\{|M| \,:\, &M \text{ is a Turing machine which on input } \lambda\\ &\text{computes } x \text{ in at most } t(|x|) \text{ steps}\}\end{aligned}$$

and call a string x K^t-random if $K^t(x)\geq|x|$. A simple counting argument shows that for any t and n there is a K^t-random string of length n. Further, it is also easy to see that for $t(n)=2^n$ there is a function r where $r(n)$ has length n, is computable in time exponential in n, and where

$$K^{2^n}(r(n)) \geq |r(n)| = n \ . \tag{25}$$

Theorem 16. *If* A *is* p-T-*hard for* **EXP** *and* T-tt-*collapsing, then* A *is in* **EXP**.

Proof. Assume that A is p-T-hard for **EXP** and T-tt-collapsing and let

$$\begin{aligned}B &= \{\langle r(2|x|+1), x\rangle \,:\, x \text{ in } A\} \ ;\\ C &= \{0^n1^m \,:\, 1\leq m\leq n \text{ and the } m\text{-th bit of } r(n) \text{ is } 1\} \ .\end{aligned}$$

Then, $A \leq^{\mathrm{p}}_{\mathrm{T}} B \oplus C$, since on input x of length n, the string $r(2n+1)$ can be computed by $2n+1$ many queries to C, whence $A(x)$ can then be computed by one more query to B. On the other hand, $B \leq^{\mathrm{p}}_{\mathrm{T}} A \oplus C$ by definition of B, and $C \leq^{\mathrm{p}}_{\mathrm{T}} A$ because C is in **EXP** and A is p-T-hard for **EXP**. Hence $B \oplus C =^{\mathrm{p}}_{\mathrm{T}} A$.

Since A is T-tt-collapsing, this implies that A is p-tt-reducible to $B \oplus C$, say via the p-tt-reduction M. Now consider a string of the form $0\langle y, z\rangle$ in $Q(x, M)$ with $|y| \geq 2|x| + 1$. If the running time of M is bounded by the polynomial p, then

$$K^{2^n}(y) \leq |x| + \log(p(|x|)) + c$$

for some constant c. Here, the code of a Turing machine which computes y will contain the constant size code of M, as well as the argument x and the index of y among all queries in $Q(x, M)$. Observe that this Turing machine computes y in time polynomial in $|x|$ and hence, by choice of y, in time polynomial in $|y|$.

But this implies, by (25), that for almost all x, the set $Q(x, M)$ does not contain strings of the form $0\langle r(n), z\rangle$ with $|r(n)| \geq 2|x|+1$. Hence, for almost all x and all queries $0\langle y, z\rangle$ in $Q(x, M)$, either the query can be easily answered in the negative because $|y|$ differs from $2|z|+1$ or $|z| \geq |x|$ or else we have $|y| = 2|z|+1$ and $|z| < |x|$. Queries of the latter type can be answered by querying C (in order to check whether y is equal to $r(2|z|+1)$) plus querying A at z and consequently A is self-reducible with oracle C. More precisely, $A \leq^{\mathrm{p}}_{\mathrm{T}} A \oplus C$ via a reduction N with the property that for all x, if a string $0y$ is in $Q(x, N)$, then y has length strictly less than $|x|$. Using this reduction N and an exponential time algorithm for C, we now can compute A by an inductive procedure in exponential time. Here, given $A{\restriction}x$, we let $A(x)$ be equal to $N(A \oplus C, x) = N((A{\restriction}x) \oplus C, x)$. □

Corollary 17. *If A is* p-T-*hard for* **EXP** *then A is not* T-tt-*collapsing.*

Proof. By the previous theorem, a T-tt-collapsing set which is p-T-hard for **EXP** must be p-T-complete for **EXP**. So the existence of such a set contradicts Watanabe's above-mentioned separation result in [18]. □

4 Conclusion

We have shown that there is a btt-m-collapsing set. Besides the previously known existence of T-btt(1)-collapsing sets, this is the strongest known collapse in the range between p-m- and p-T-reducibility. Of course, this leads to the question of whether these results can be combined to obtain a full collapse of the p-T-degree of a set to its p-m-degree. This question remains open and similarly for the somewhat weaker problem of whether there is a tt-m-collapsing set.

Our results in Sect. 3 might be taken as some evidence that no T-m-collapsing and no tt-m-collapsing sets exist. In particular, we can argue that, for reducibilities restricted by some class of time bounds, if there is a T-m-collapsing set then this fact depends on the particular time bounds for the reductions we consider. Namely, by an analysis of Selman's result that tally sets are not T-m-collapsing, which is similar to the considerations following the proof of Theorem 6, every set A which is T-equivalent to the set $I(A)$ of its finite prefixes is not T-m-collapsing. On the other hand, however, every set A is T-equivalent to $I(A)$ by exponential-time bounded reductions. So, for instance, for elementary recursive reducibility, no set is T-m-collapsing. In fact, for elementary recursive reducibility, the techniques used here and in previous papers suffice to completely specify the maximum collapses that might occur.

By adapting the notion of super sparseness to elementary time, the arguments in [2] carry over directly to show that there are (nonelementary) T-btt(1)-collapsing sets again. Similarly, our proof of Theorem 5 can be adapted to show that there are btt-m-collapsing sets too. Now, any resource-bounded T-reduction can be simulated by a tt-reduction with exponential overhead, whence in case of elementary time, tt-reducibility and T-reducibility coincide, i.e., every set is T-tt-collapsing. Thus the discussion above shows that for elementary time the maximum collapse of T, or, equivalently, of tt, to m does not occur, while both of the next smaller collapses are realized, i.e., the collapse of T to btt(1) and that of btt to m.

References

1. E. Allender. News from the isomorphism front. *Bulletin of the European Association for Computer Science* 66:73–81, 1998.
2. K. Ambos-Spies. Inhomogeneities in the polynomial-time degrees: the degrees of super sparse sets. *Information Processing Letters* 22:113–117, 1986.
3. K. Ambos-Spies. Resource-bounded genericity. In: S.B. Cooper et al., editors, *Computability, Enumerability, Unsolvability: Directions in Recursion Theory.* London Mathematical Society Lecture Note Series 224, 1–59. Cambridge University Press, 1996.
4. L. Berman. *Polynomial reducibilities and complete sets.* Ph.D. dissertation, Cornell University, 1977.
5. L. Berman, J. Hartmanis. On isomorphisms and density of NP and other complete sets. *SIAM Journal on Computing* 6:305–322, 1977.
6. H. Buhrman. *Resource Bounded Reductions.* Ph.D. dissertation, Universiteit van Amsterdam, Amsterdam, 1993.
7. H. Buhrman, L. Torenvliet. Complete sets and structure in subrecursive classes. In: J.M. Larrazabal, D. Lascar, G. Mints, editors, *Logic Colloquium '96*, 45–77, Lecture Notes in Logic 12, Springer Verlag, 1998.
8. S.A. Fenner. Notions of resource-bounded category and genericity. *Proceedings of the 6th Annual Structure in Complexity Theory Conference, 1991,* 196–212, IEEE Computer Society Press, 1991.
9. S. Homer, S. Kurtz, and J. Royer. On 1-truth-table-hard languages. *Theoretical Computer Science.* 115:383–389, 1993.
10. C.G. Jockusch Jr. Semirecursive sets and positive reducibility. *Transactions of the American Mathematical Society* 131:420–436, 1968.
11. S.A. Kurtz, S.R. Mahaney, and J.S. Royer. The structure of complete degrees. In: [16], 108–146.
12. R.E. Ladner, N.A. Lynch, A.L. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science* 1:103–123, 1975.

13. M. Li, P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, 1997.
14. N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13:145–147, 1972.
15. A.L. Selman. P-selective sets, tally languages, and the behavior of polynomial time reducibilities on NP. *Mathematical Systems Theory* 13:55–65, 1979.
16. A.L. Selman, editor. *Complexity Theory Retrospective.* Springer-Verlag, 1990.
17. P. Young. Juris Hartmanis: fundamental contributions to isomorphism problems. In [16], 28–58.
18. O. Watanabe. A comparison of polynomial time completeness notions. *Theoretical Computer Science* 54:249-265, 1987.