Approximable sets*

Richard Beigel † Martin Kummer ‡ Frank Stephan §
Yale University Universität Karlsruhe Universität Karlsruhe

Abstract

Much structural work on NP-complete sets has exploited SAT's d-self-reducibility. In this paper we exploit the additional fact that SAT is a d-cylinder to show that NP-complete sets are p-superterse unless P = NP. In fact, every set that is NP-hard under polynomial-time $n^{o(1)}$-tt reductions is p-superterse unless P = NP. In particular no p-selective set is NP-hard under polynomial-time $n^{o(1)}$-tt reductions unless P = NP. In addition, no easily countable set is NP-hard under Turing reductions unless P = NP. Self-reducibility does not seem to suffice for our main result: in a relativized world, we construct a d-self-reducible set in NP − P that is polynomial-time 2-tt reducible to a p-selective set.

1 Introduction

Assume we are given a set $A \subseteq \{0,1\}^*$. Even if A is intractable there might be a way to compute some partial information about A efficiently, i.e., in polynomial time.

We are interested in information of the following kind: Given a list of k strings $x_1, \ldots, x_k$ (here $k \geq 1$ is fixed) there are a priori 2^k possibilities of how the characteristic function of A is defined on $x_1, \ldots, x_k$. Can we exclude in polynomial time at least one of these possibilities?

If this is possible for some $k \geq 1$ we call A *approximable*; if this is not possible for all $k \geq 1$ we call A *p-superterse*. (So the approximable sets are just the non-p-superterse sets.)

Let $F_k^A(x_1, \ldots, x_k) = (\chi_A(x_1), \ldots, \chi_A(x_k))$ denote the k-ary membership function of A. A well-known class of approximable sets are the p-selective sets of Selman [31]. There we can exclude one of four possibilities for $F_2^A(x_1, x_2)$.

*An extended abstract of this paper appeared in: *Proceedings of the 9th Conference on Structure in Complexity Theory*, IEEE Computer Society Press, 1994.

†Yale University, Dept. of Computer Science, P.O. Box 208285, New Haven, CT 06520-8285, (Email: `beigel@cs.yale.edu`). Supported in part by the United States National Science Foundation (NSF) under grant CCR-8958528 and the Netherlands Organization for Scientific Research (NWO) under Visitors Grant B 62-403.

‡Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe, D-76128 Karlsruhe, Germany, (Email: `kummer@ira.uka.de`).

§Institut für Logik, Komplexität und Deduktionssysteme, Universität Karlsruhe, D-76128 Karlsruhe, Germany, (Email: `fstephan@ira.uka.de`). Supported by the Deutsche Forschungsgemeinschaft (DFG) grant Me 672/4-1.

The notion p-superterse was introduced by Beigel [6] and was studied in several recent papers, e.g. in [2], [8], [10]. Originally, it was defined via "bounded query classes": A set A is p-superterse iff for every $k \geq 1$ and all oracles X, F_k^A cannot be computed by any polynomial-time oracle Turing machine (OTM) with fewer than k queries to X. Intuitively, there is no way to save a query, regardless of the oracle.

There are various subclasses of approximable sets that have been studied. A set A is *cheatable* [6] iff there is a constant c such that for every k we can compute in polynomial time a set of c possibilities for $F_k^A(x_1, \ldots, x_k)$. A set A is *easily countable* [17] iff for some $k \geq 1$ and all pairwise distinct $x_1, \ldots, x_k$ we can exclude at least one possibility for the cardinality of $A \cap \{x_1, \ldots, x_k\}$. Both are proper subclasses of approximable sets.

Are there natural approximable sets, for instance NP-complete sets? Under the hypothesis that P $\neq$ NP, Beigel [10] proved that SAT is not cheatable, and Hoene and Nickelsen [17] proved that SAT is not easily countable.

In this paper we show that SAT is p-superterse unless P = NP. In fact, we even show that if SAT is tt-reducible to an approximable set by a tt-reduction with $n^{o(1)}$ queries, then P = NP. This also solves an open problem on p-selective sets, since it was not known whether the existence of a p-selective btt-hard set for NP implies P = NP. Furthermore we show that no T-hard set for NP is easily countable unless P = NP. Previously this was only known for m-hard sets.

We also make progress on a question of Krentel [23]: Is every function computable in polynomial time with *parallel* queries to SAT, also computable in polynomial time with $O(\log n)$ *sequential* queries to SAT? We show that if P $\neq$ NP then for every $\epsilon < 1$ there is a function computable in polynomial time with $\log n$ parallel queries to SAT, which is not computable with $\epsilon \cdot \log n$ sequential queries to SAT. (Here log denotes the logarithm with base 2.)

We obtain several other results on the structure of approximable sets, e.g., we investigate different types of approximable cylinders, and we provide relativizations which show that some of our results are optimal (w.r.t. relativizing proof techniques). Recall that every self-reducible set which is 1-tt-reducible to a p-selective set is in P [13]. We show that this result will not be improved by relativizing techniques: there is an oracle relative to which P $\neq$ NP and there exists a d-self-reducible T-complete set which is 2-tt-reducible to a p-selective set in NP.

The main results of Sections 4, 5 have been obtained independently and at the same time by Agrawal and Arvind [1] and Ogihara [29].

All unexplained notations and definitions are standard as e.g. in [5]. FP denotes the set of all polynomial-time computable functions. $\omega = \{0, 1, \ldots\}$ is the set of all natural numbers. For $\sigma \in \{0,1\}^*$ we denote by $\sigma[i]$ the i-th bit of σ, $1 \leq i \leq |\sigma|$. Let λ be the empty string.

Let $\preceq$ be the following proper lexicographic ordering on strings $\sigma, \tau \in \{0,1\}^*$: $\sigma \preceq \tau \Leftrightarrow$ (σ is an initial segment of τ or there exists $i < |\sigma|$ such that σ and τ agree in the first $i-1$ bits and $\sigma[i] = 0, \tau[i] = 1$).

A set A is *self-reducible* iff there is a polynomial-time OTM $M^{()}$ such that $M^A(x) = \chi_A(x)$ and M^A with input x queries only strings of length less than $|x|$. A is *tt-self-reducible* if in addition all queries are made nonadaptively. A is *d-self-reducible* ("disjunctive self-reducible") if $M^A(x)$ accepts iff at least one of the queries is answered positively (w.l.o.g. we may assume that all queries are made nonadaptively). All of our results also hold for the more liberal definition of self-reducibility in [27]; we choose the old definition only to simplify the notation.

2 Basic definitions and facts

Definition 2.1 (Verbose/Superterse) A set A is $(a,b)_p$-*verbose* ($1 \leq a \leq 2^b$) iff we can compute in polynomial-time for each b-tuple $(x_1,\ldots,x_b)$ an a-element set $D \subseteq \{0,1\}^b$ which contains $F_b^A(x_1,\ldots,x_b)$. Trivially, every set is $(2^b,b)_p$-verbose.
A set is *approximable* iff it is $(2^b-1,b)_p$-verbose for some b.
A set is *p-superterse* iff it is not approximable (i.e. not $(2^b-1,b)_p$-verbose for any b).
A set is *k-cheatable* iff it is $(k,k)_p$-verbose.
A set is *cheatable* iff it is k-cheatable for some $k \geq 1$.

The notions "cheatable" and "p-superterse" were introduced by Beigel in [6], the recursion theoretic versions are studied in [11]. The notion "$(a,b)_p$-verbose" was introduced by Beigel, Kummer, and Stephan in [12] who also defined the recursion theoretic version. In [12, Section 7] we provide an algorithm to compute whether "$(a,b)_p$-verbose $\Rightarrow$ $(c,d)_p$-verbose" for any given a,b,c,d.

Definition 2.2 (Frequency Computation) A set A is $(a,b)_p$-*recursive* ($1 \leq a \leq b$) iff there is $f \in \mathrm{FP}$ such that for any pairwise distinct $x_1,\ldots,x_b$, $f(x_1,\ldots,x_b)$ is a b-bit-vector $(y_1,\ldots,y_b) \in \{0,1\}^b$ such that at least a of the numbers y_i agree with $\chi_A(x_i)$. A set is *easily approximable* iff it is $(a,b)_p$-recursive for some a,b with $a > \frac{b}{2}$.

The set A is $(1,b)_p$-recursive iff A is $(2^b-1,b)_p$-verbose: $f(x_1,\ldots,x_b) = (y_1,\ldots,y_b)$ iff the possibility $F_b^A(x_1,\ldots,x_b) = (1-y_1,\ldots,1-y_b)$ is excluded. Thus the approximable sets can also be defined as the sets which are $(1,b)_p$-recursive for some $b \geq 1$.

The notion "$(a,b)_p$-recursive" was introduced by Kummer and Stephan in [25], the study of the recursion theoretic version goes back to Trakhtenbrot [34]. There is an algorithm to decide whether "$(a,b)_p$-recursive $\Rightarrow$ $(c,d)_p$-recursive", see [25].

It is easy to see that if A,B are $(a,b)_p$-recursive via the same f and $a > \frac{b}{2}$ then χ_A and χ_B differ in at most $2(b-a) < b$ arguments. This observation is due to Trakhtenbrot [34].

Definition 2.3 (Easily Countable) A set A is *easily k-countable* iff there is a function $f \in \mathrm{FP}$ such that $f(x_1,\ldots,x_k) \in \{0,\ldots,k\}$ and $f(x_1,\ldots,x_k) \neq \chi_A(x_1)+\ldots+\chi_A(x_k)$ for all *pairwise different* $x_1,\ldots,x_k$. A set is *easily countable* iff it is k-countable for some $k \geq 1$.

Hoene and Nickelsen introduced the notion "easily countable" in [17][1], Kummer [24] treats the recursion theoretic version. The countability classes have some strange properties, e.g., there is an easily 5-countable set which is not easily 6-countable [30].

[1]They defined "easily countable" in a slightly more restrictive way omitting the condition "pairwise different".

Definition 2.4 (P-selective) A is *p-selective* iff there is a function $f \in \mathrm{FP}$ such that $f(x,y) \in \{x,y\}$ and $x \in A \vee y \in A \Rightarrow f(x,y) \in A$ for all x, y.

The notion "p-selective" was introduced by Selman in [31], the recursion theoretic version "semirecursive" is due to Jockusch [20]. Note that if A is p-selective then A is $(1,2)_p$-recursive. Thus, the p-selective sets form a subclass of the approximable sets.

The recursion theoretic counterparts of "cheatable", "easily approximable" and "easily countable" are all equivalent to "recursive". We refer the reader to [14, 26] for further background on the proof of this equivalence. In complexity theory the picture is quite different. The following implications hold.

Fact 2.5 (1) *cheatable* $\Rightarrow$ *easily countable* [17].
(2) *easily approximable* $\Rightarrow$ *easily countable.*
(3) *easily countable* $\Rightarrow$ *approximable* [17].

(1) and (3) follow directly from the definitions. If A is easily approximable, i.e. if it is $(a,b)_p$-recursive via f and $a > \frac{b}{2}$, then for all pairwise distinct $x_1, \ldots, x_b$ either 0^b or 1^b differs from $f(x_1, \ldots, x_b)$ in at least $\frac{b}{2}$ components. Thus we can exclude either 0 or b as a possible cardinality of $|A \cap \{x_1, \ldots, x_b\}|$, so A is easily b-countable.

Hoene and Nickelsen [17] noted that the first and third implication cannot be reversed. We shall show below that "cheatable" and "easily approximable" are incomparable notions. Thus, there are no other implications besides those listed above.

The next fact gives an important combinatorial property of approximable sets. If A is approximable then we can compute in polynomial time *for all* $k \geq 1$ and $x_1, \ldots, x_k$ a subset of $\{0,1\}^k$ which contains $F_k^A(x_1, \ldots, x_k)$. See [12, Section 2] for a detailed discussion of the combinatorial background.

Fact 2.6 [6, 7] *If A is $(2^b - 1, b)_p$-verbose then there is a polynomial-time computable function which computes for any k numbers $x_1, \ldots, x_k$ a set of at most*

$$S(k,b) = \binom{k}{0} + \ldots + \binom{k}{b-1}$$

elements from $\{0,1\}^k$ which contains $F_k^A(x_1, \ldots, x_k)$. For fixed b, $S(k,b)$ is a polynomial in k of degree $b-1$.

Corollary 2.7 *Assume that $f \leq_{\mathrm{tt}}^p A$ for some approximable set A. Then there is a polynomial time algorithm to compute for each x a set D with $f(x) \in D$.*

Part (1) of the following fact will be generalized in Lemma 6.6 below. Part (2) follows in a straightforward way from Fact 2.6.

Fact 2.8 (1) *Cheatable sets are closed under* $\leq_{\mathrm{T}}^p$ [2, Lemma 19].
(2) *Approximable sets are closed under* $\leq_{\mathrm{btt}}^p$ [6].

Note that every paddable set which is easily approximable is in P (using majority voting). Since there exist easily approximable sets not in P, it follows that the easily approximable sets are not closed under $\le_{\mathrm{m}}^{p}$. Also the cylinders of cheatable sets, which are again cheatable, are not easily approximable. (As usual, the cylinder of A is the set $\{(x, y) : x \in A \ \wedge \ y \in \{0,1\}^*\}$.)

On the other hand, the easily approximable set B constructed in Theorem 2.9 below is not cheatable (since a p-superterse set is tt-reducible to B).

We now show that easily approximable sets are not closed under tt-reduction, even worse: There is a p-superterse set which is tt-reducible to an easily approximable set.

Theorem 2.9 *There are sets A, B such that A is p-superterse, B is $(2,3)_p$-recursive, and $A \le_{\mathrm{tt}}^{p} B$.*

Proof: We construct A as a subset of the union of a very sparse set of intervals I_k. Similar constructions appear in [3], [9].

Identify each number k with a coded pair $\langle e, b\rangle$, $b < k$. Let I_k be an interval containing b strings of length $tow(k)$; where $tow(0) = 1$ and $tow(k+1) = 2^{tow(k)}$ for all k. We define $A \cap I_k$ as follows: Simulate $M_e(x_1, \ldots, x_b)$ for the first $2^{tow(k)}$ steps, where M_e is the e-th polynomial-time machine and $I_k = \{x_1, \ldots, x_b\}$. If the computation converges and outputs a set $D \subset \{0,1\}^b$ then choose $v \in \{0,1\}^b - D$ and let $F_b^A(x_1, \ldots, x_b) = v$. In this case we have diagonalized M_e, so that A is not $(2^b - 1, b)_p$-verbose via M_e. This ensures that A is a p-superterse set in DTIME(2^n).

On interval I_k, χ_A can only take $2^b \le tow(k)$ possible values. These are coded by a 1-out-of-2^b-code into corresponding intervals $J_k = \{1\}^{tow(k)} \times \{0,1\}^b$ of some set $B \subseteq \cup\{J_k : k \in \omega\}$. B is also 2^n-time-computable.

So for $x \in I_k$ the value $\chi_A(x)$ depends only on the values of B on the interval J_k; so $\chi_A(x)$ can be computed by $|x|$ parallel queries to J_k. Since $\chi_A(x) = 0$ for all x outside these intervals, $A \le_{\mathrm{tt}}^{p} B$.

B is $(2,3)$-recursive: Given x_1, x_2, x_3 we can compute the intervals to which they belong. Let J_k be the interval with the largest index. Since $2^{|x_i|} \le (tow(k))^2$ for $x_i \in J_1 \cup \cdots \cup J_{k-1}$ and since $\chi_B(x_i) = 0$ for $x_i \notin J_1 \cup \cdots \cup J_k$, we can compute $\chi_B(x_i)$ in polynomial time if $x_i \notin J_k$. For $x_i \in J_k$ we output 0. Since $|J_k \cap B| \le 1$ this procedure makes at most one error. ▮

3 Approximable cylinders

A set A is a *bd-cylinder* iff there is $f \in$ FP such that $(\forall x, y)[x \in A \ \vee \ y \in A \ \Leftrightarrow f(x, y) \in A]$. A *bc-cylinder* is defined similarly: $\vee$ is replaced by $\wedge$. A set is a *bptt-cylinder* iff it is a bd-cylinder and a bc-cylinder. Finally, A is a *btt-cylinder* iff A is a bptt-cylinder and $A \le_{\mathrm{m}}^{p} \overline{A}$.

If A is a bptt-cylinder then for every fixed k there is a polynomial-time algorithm which takes as input a monotone k-ary Boolean function h (given by a table) and k strings $x_1, \ldots, x_k$, and outputs x with $\chi_A(x) = h(\chi_A(x_1), \ldots, \chi_A(x_k))$. If A is a btt-cylinder then there is a such an algorithm which works for arbitrary k-ary Boolean functions.

In [6, Theorem 5.6.2] it is shown that every approximable btt-cylinder is cheatable. We shall now give a refined version of this result, with optimal bounds.

Theorem 3.1 *Let $1 \leq a < 2^b$. If A is a btt-cylinder and $(a,b)_p$-verbose then A is a-cheatable.*

Proof: Let A be an $(a,b)_p$-verbose btt-cylinder with $a < 2^b$. We show that A is a-cheatable. Given $x_1, \ldots, x_a$ we need to compute in polynomial time a set D of at most a possibilities for $F_k^A(x_1, \ldots, x_a)$ such that the correct one is among them.

This is done by the following reduction procedure. Start with $D = \{0,1\}^a$. As long as $|D| > a$ select a chain of $a+2$ sets D_i such that $\emptyset = D_0 \subset D_1 \subset \cdots \subset D_{a+1} = D$. Let $m = \min\{i : F_a^A(x_1, \ldots, x_a) \in D_i\} \leq a+1 \leq 2^b$. Since A is a btt-cylinder there is an OTM which computes m in polynomial time with b parallel queries to A, say with $y_1, \ldots, y_b$ (e.g., $\chi_A(y_i)$ is the i-th bit in the binary representation of $m-1$). Since A is $(a,b)_p$-verbose we can compute a set of at most a possible answer vectors, among them the correct one. Each of them gives one possible value for m which can be computed by simulation of the OTM. Thus we find $1 \leq i \leq a+1$ which does not occur, i.e., $i \neq m$. So, if $F_a^A(x_1, \ldots, x_a) \in D_i$ then $F_a^A(x_1, \ldots, x_a) \in D_{i-1}$. Therefore, let $D = D - (D_i - D_{i-1})$, and iterate the procedure.

In each iteration the cardinality of D decreases but $F_a^A(x_1, \ldots, x_a) \in D$ is maintained. We end up with $|D| \leq a$ and $F_a^A(x_1, \ldots, x_a) \in D$. Each iteration runs in polynomial time.

The whole algorithm runs in polynomial time since there are a constant number of iterations. So A is $(a,a)_p$-verbose. ∎

Remark: The bound cannot be improved: In [12] we proved that for every $a \geq 2$ there is an a-cheatable set which is not $(a-1)$-cheatable. The btt-cylinder of an a-cheatable set is again a-cheatable (this uses that every $(a,a)_p$-verbose set is also $(a,b)_p$-verbose for $b \geq 1$). Hence there is an a-cheatable btt-cylinder A which is not $(a-1)$-cheatable.

Theorem 3.2 *Let A be a bptt-cylinder.*
(1) A is easily countable iff A is cheatable.
(2) A is easily k-countable iff A is k-cheatable.

Proof: (1) follows from (2) and the "if"-direction of (2) is obvious [17]. For the last case, the "only-if"-direction of (2), assume that A is a bptt-cylinder and A is easily k-countable via $f \in \mathrm{FP}$. We show that A is $(k,k)_p$-verbose.

Given $x_1, \ldots, x_k$ we need to enumerate in polynomial time a set D of at most k possibilities for $F_k^A(x_1, \ldots, x_k)$ such that the correct one is among them.

Using that A is a bptt-cylinder we can compute for every string $\sigma \in \{0,1\}^k$ a value $y(\sigma)$ such that

$$y(\sigma) \in A \;\Leftrightarrow\; \sigma \preceq F_k^A(x_1, \ldots, x_k).$$

To see this consider for each σ the Boolean function h_σ with $h_\sigma(b_1, \ldots, b_k) = 1 \Leftrightarrow \sigma \preceq (b_1, \ldots, b_k)$, and notice that h_σ is monotone. Since $\preceq$ is a linear ordering we get:

$$(*) \;\; \tau \preceq \sigma \;\wedge\; y(\sigma) \in A \;\Rightarrow\; y(\tau) \in A.$$

Similarly as in the previous proof we finally use a reduction procedure to compute a set D with $|D| \leq k$ and $F_k^A(x_1, \ldots, x_k) \in D$.

Algorithm:
Initialize $D = \{0,1\}^k$.
While $|D| > k$ do:

(I) If D contains σ, τ with $\sigma \prec \tau$ and $y(\sigma) = y(\tau)$ then let $D = D - \{\sigma\}$.
(This is correct since $\sigma \neq F_k^A(x_1, \ldots, x_k)$. Otherwise $y(\sigma) \in A$, so $y(\tau) \in A$, and $\tau \preceq F_k^A(x_1, \ldots, x_k) = \sigma$, a contradiction.)

(II) If D contains $k+1$ elements $\sigma_0 \prec \sigma_1 \prec \cdots \prec \sigma_k$ such that the $y(\sigma_i)$-values are pairwise different then compute $c = f(y(\sigma_1), \ldots, y(\sigma_k))$. Let $D = D - \{\sigma_c\}$.
(This is correct since $\sigma_c \neq F_k^A(x_1, \ldots, x_k)$. Otherwise, by the hypothesis on f and $(*)$, we get $\sigma_0, \ldots, \sigma_c \in A$, so $|A \cap \{y(\sigma_1), .., y(\sigma_k)\}| = c$, a contradiction.) □

In each iteration the cardinality of D decreases but $F_k^A(x_1, \ldots, x_k) \in D$ is maintained. After a constant number of iterations we have $|D| \leq k$, as required. Since all computations run in polynomial time, we get that A is $(k,k)_p$-verbose. ▮

Remark: (1) There exist approximable bptt-cylinders which are not cheatable, e.g., every p-selective set is a bptt-cylinder, but there are non-cheatable p-selective sets.

(2) The theorem does not hold if we require that A is a bc-cylinder (or bd-cylinder) instead of bptt-cylinder: The set B from Theorem 2.9 is an easily 2-countable bc-cylinder but is not cheatable.

SAT is a bptt-cylinder. Thus, by Theorem 3.2 (1), if SAT is easily countable then SAT is cheatable, and hence, by [10, Corollary 5.10], SAT $\in$ P. This is an alternative proof of the result of Hoene and Nickelsen [17, Corollary 9] that SAT is not easily countable unless P = NP. In the following sections we generalize this result in two directions: We show that SAT is not approximable unless P = NP, and we show that no T-hard set for NP is easily countable unless P = NP.

4 Are NP-hard sets p-superterse?

Since a positive answer to this question implies P $\neq$ NP we can only hope to answer it under some reasonable hypothesis. Also note that for the notion "NP-hard" we have to specify a polynomial-time reduction. Previously, it was not even known whether an m-complete set for NP must be p-superterse if P $\neq$ NP. The following results were the best known.

Fact 4.1
(1) *If* SAT *is* $\leq_{\mathrm{tt}}^p$*-reducible to an approximable set then* R = NP *and* P = UP [8].
(*Independently Toda* [33] *obtained the special case for p-selective sets.*)
(2) *If there is an approximable set that is* $\leq_{k\text{-tt}}^p$*-hard for the Boolean hierarchy then* P = NP [2, Corollary 39].
(3) *If* SAT *is* $\leq_{1\text{-tt}}^p$*-reducible to a p-selective set then* P = NP [13, Corollary 15].
(4) *If* SAT *is* $\leq_{\mathrm{btt}}^p$*-reducible to a p-selective set then* NP $\subseteq$ DTIME$(2^{n^{O(1/\sqrt{\log n})}})$ [32].

In the following we generalize those three results for the case of sets that are NP-hard under $n^{o(1)}$-tt reductions. In particular, we show that every btt-hard set for NP is p-superterse unless P = NP. Since p-selective sets are approximable, this implies that SAT is not btt-reducible to a p-selective set unless P = NP.

First we illustrate the technique by showing that every d-self-reducible bd-cylinder is p-superterse unless it is in P.

Theorem 4.2 *Let A be d-self-reducible and a bd-cylinder. Then either $A \in \mathrm{P}$ or A is p-superterse.*

Proof: Assume that A is a d-self-reducible bd-cylinder. Further suppose that A is $(2^k-1,k)_p$-verbose. We describe a polynomial-time decision procedure for A.

Let A be d-self-reducible via $M^{()}$ and let $p(n)$ be a polynomial that bounds the run-time of $M^{()}$. For each x we get a d-self-reduction tree T such that x is the root of T and the successors of each inner node y are the strings z, $|z| < |y|$, queried by M^A on input y. Furthermore $y \in A$ iff at least one of its successors is in A. There are at most $|x|$ many levels of T.

We expand T level by level. But before we expand level $i+1$ we apply a pruning algorithm until at most 2^k nodes remain in level i. Since each node has at most $p(|x|)$ many successors we have to deal in each level with at most $2^k \cdot p(|x|)$ many nodes. The pruning algorithm will need polynomial time to prune one node. Thus the whole procedure runs in polynomial time. In the end we know $M^A(y)$ for all leaves of the pruned tree. Then $x \in A \Leftrightarrow\ M^A(y) = 1$ for some leaf y. The pruning algorithm is based on the following fact.

Lemma 4.3 *For each set $D, |D| = 2^k$, we can compute in polynomial time an element $x \in D$ such that if A intersects D then A intersects $D - \{x\}$.*

Proof: Let $D = \{x_1, \ldots, x_{2^k}\}$ and let $\{\sigma_1, \ldots, \sigma_{2^k}\} = \{0,1\}^k$. We define

$$D_i = \{x_j : \sigma_j[i] = 1\} \text{ for } i = 1, \ldots, k.$$

Since A is a bd-cylinder and the cardinality of D_i is bounded by a constant we can compute in polynomial time $y_1, \ldots, y_k$ such that for $i = 1, \ldots, k$:

$$y_i \in A \Leftrightarrow\ D_i \cap A \neq \emptyset.$$

Since A is $(2^k-1,k)_p$-verbose we find in polynomial time an index j such that $1 \leq j \leq 2^k$ and $F_k^A(y_1, \ldots, y_k) \neq \sigma_j$. Then x_j cannot be the only element of $A \cap D$ since otherwise $(D_i \cap A \neq \emptyset \Leftrightarrow\ \sigma_j[i] = 1)$ for $i = 1, \ldots, k$, so $F_k^A(y_1, \ldots, y_k) = \sigma_j$, a contradiction. Thus if $A \cap D \neq \emptyset$ then $A \cap (D - \{x_j\}) \neq \emptyset$. This completes the proof of the Lemma. ▮

When we expand level i we get a list L of nodes with the invariant that $x \in A\ \Leftrightarrow\ A \cap L \neq \emptyset$. First we discard all duplicates. Then, as long as $|L| \geq 2^k$ we select a subset $D \subseteq L$ with $|D| = 2^k$ and eliminate one element of D according to the Lemma. In this way we reduce $|L|$ and maintain the invariant. After at most $2^k \cdot p(|x|)$ iterations we have $|L| < 2^k$. Now we compute the successors of the nodes in L, which defines the list for level $i+1$. This completes the description of our decision procedure. ▮

Well-known natural examples of d-self-reducible bd-cylinders are SAT, GI (the Graph Isomorphism Problem), and GA (the Graph Automorphism Problem). See [22] for more information on GI and GA. We get the following corollary.

Corollary 4.4 SAT, GI, *and* GA *are either p-superterse or in* P.

Using Fact 2.8 (2), that approximable sets are closed under btt-reductions, we get:

Corollary 4.5 *Every btt-hard set for* NP *is p-superterse unless* P = NP.

How far can this be generalized? We do not know if it holds for tt-hard sets. Our best result in this direction works for tt-reductions with $n^{o(1)}$ parallel queries. Versions of the following theorem were obtained simultaneously and independently by Agrawal and Arvind [1] (who consider only p-selectiveness), by Ogihara [29] (who considers p-selectiveness and a notion closely related to verboseness), and by ourselves. Before seeing the other two papers, we stated our results only for $n^{o(1)}$-tt reductions; however, the proof we give now is essentially the same as our original.

Theorem 4.6 (1) *For every $\epsilon > 0$ and every $c > 1$, there is no $(2^c - 1, c)_p$-verbose and $n^{\frac{1}{c-1}-\epsilon}$-tt-hard set for NP unless P = NP. In particular, every $\leq^p_{n^{o(1)}\text{-tt}}$-hard set for* NP *is p-superterse unless* P = NP.
(2) *If, for some $\epsilon > 0$,* SAT *is $\leq^p_{n^{1-\epsilon}\text{-tt}}$-reducible to a p-selective set, then* P = NP.

Proof: (2) follows from (1), because p-selective sets are $(3, 2)_p$-verbose.

The proof of (1) is a modification of the proof of Theorem 4.2. Let B be $(2^c-1, c)_p$-verbose for some $c > 1$. Assume there exists $\epsilon > 0$ such that SAT $\leq^p_{g(n)\text{-tt}} B$, where $g(n) = n^{\frac{1}{c-1}-\epsilon}$. We show that SAT $\in$ P, i.e., P = NP. Note that SAT is not only a bd-cylinder but a *d-cylinder*, i.e, for every finite set E we can compute in polynomial time a formula x such that $x \in \text{SAT} \Leftrightarrow E \cap \text{SAT} \neq \emptyset$. This allows us to prove a version of Lemma 4.3 where k may depend on the size of D.

Claim: There is a polynomial-time algorithm to compute for each set D with $|D| = 2^k$ where $k = \lceil \frac{1}{(c-1)\epsilon} \log n \rceil$ and $n = \max\{|y| : y \in D\}$, an element $x \in D$ such that if $\text{SAT} \cap D \neq \emptyset$ then $\text{SAT} \cap (D - \{x\}) \neq \emptyset$.

Proof of claim: We proceed exactly as in Lemma 4.3, i.e., we compute the sets D_i and the corresponding y_i. In the case of SAT we simply compute the disjunction of all formulas in D_i, thus, $|y_i| \leq 2^k n$. Now we want to exclude one possibility for $F_k^{\text{SAT}}(y_1, \ldots, y_k)$. Since SAT $\leq^p_{g(n)\text{-tt}} B$, $\chi_{\text{SAT}}(y_i)$ can be computed with $g(|y_i|)$ many parallel queries to B. Thus, $F_k^{\text{SAT}}(y_1, \ldots, y_k)$ is determined by $m = k \cdot g(2^k n)$ many parallel queries to B. By Fact 2.6, there is a polynomial p of degree $c-1$ such that we can compute a set of at most $p(m)$ possibilities of how B is defined on the set of all queries. Hence, we get at most $p(m)$ possibilities for $F_k^{\text{SAT}}(x_1, \ldots, x_k)$. Note that $p(m) = O((k(n2^k)^{1/(c-1)-\epsilon})^{c-1})$. A straightforward computation shows that for all sufficiently large n we have $p(m) < 2^k$. Now we can continue as in the proof of Lemma 4.3. ∎

Using the Claim each level of the d-self-reduction tree for x can be pruned until at most $|x|^{\frac{1}{(c-1)\epsilon}}$ nodes remain. Hence, also in this case the decision procedure runs in polynomial time. ∎

Remark: Note that the proof of Theorem 4.6 actually shows that if a d-self-reducible d-cylinder A is $n^{o(1)}$-tt-reducible to an approximable set, then A is in P.

It is not known whether there exists a relativized world with $\mathrm{P} \neq \mathrm{NP}$ and an approximable set that is tt-hard for NP. For T-reducibility the following is known (see the footnote following Corollary 3.17 in [16]).

Fact 4.7 *In some relativized world with* $\mathrm{P} \neq \mathrm{NP}$ *there exists a* $(1,2)_p$*-recursive (actually p-selective) set* $A \in \mathrm{NP}$ *that is* $\leq_{\mathrm{T}}^{p}$*-hard for* NP*. (See also Theorem 7.1 below for a more general result.)*

Amir, Beigel, and Gasarch proved that there are no approximable T-hard sets for NP unless the Polynomial Hierarchy collapses:

Fact 4.8 [2] *Every approximable set belongs to* P/Poly*.*
If $\Sigma_2^p \neq \Pi_2^p$ *then every* $\leq_{\mathrm{T}}^{p}$*-hard set for* NP *is p-superterse.*

5 Computing functions with queries to NP

The methods from the previous section have a further application: They allow us to improve a result of Krentel [23]. Let $\mathrm{FP}^{\mathrm{SAT}}$ denote the class of all function that can be computed in polynomial time with an oracle for SAT. $\mathrm{FP}^{\mathrm{SAT}}_{g(n)\text{-T}}$ is the class of all functions $f \in \mathrm{FP}^{\mathrm{SAT}}$ that can be computed using at most $g(n)$ *adaptive* queries on inputs of length n. $\mathrm{FP}^{\mathrm{SAT}}_{\mathrm{tt}}$ and $\mathrm{FP}^{\mathrm{SAT}}_{g(n)\text{-tt}}$ are the corresponding *nonadaptive* version.

As in [23] we call a function $g : N \to N$ *smooth* if the function $1^n \mapsto 1^{g(n)}$ is computable in polynomial time and if $g(x) \leq g(y)$ for $x \leq y$. Krentel proved the following result. Actually he showed this just for $c < 1/2$, but his proof can be modified to $c < 1$, see [8].

Fact 5.1 [23, Theorem 4.2] *Let g be smooth and $g(n) \leq c \cdot \log n$ for some $c < 1$.*
If $\mathrm{FP}^{\mathrm{SAT}}_{g(n)\text{-T}} \subseteq \mathrm{FP}^{A}_{(g(n)-1)\text{-T}}$ *for some A then* $\mathrm{P} = \mathrm{NP}$*.*

Informally, if $g(n)+1$ adaptive queries to SAT can be simulated by $g(n)$ adaptive queries to any oracle then $\mathrm{P} = \mathrm{NP}$. We shall now show that the conclusion holds already under the weaker hypothesis that $g(n)+1$ *nonadaptive* queries to SAT can be simulated by $g(n)$ adaptive queries.

Theorem 5.2 *Let g be smooth and $g(n) \leq c \cdot \log n$ for some $c < 1$.*
If $\mathrm{FP}^{\mathrm{SAT}}_{g(n)\text{-tt}} \subseteq \mathrm{FP}^{A}_{(g(n)-1)\text{-T}}$ *for some A then* $\mathrm{P} = \mathrm{NP}$*.*

Proof: Let g be smooth and $g(n) \leq c \cdot \log n$ for $c < 1$. Define a function f as follows:

$$f(x) = \begin{cases} F_k^{\mathrm{SAT}}(x_1, \ldots, x_k), & \text{if } x = x_1\# \ldots \#x_k\#^l \wedge k \leq g(|x|) \wedge l \geq 0; \\ \#, & \text{otherwise.} \end{cases}$$

Note that x may be padded by a suffix of #'s. Clearly $f \in \mathrm{FP}^{\mathrm{SAT}}_{g(n)\text{-tt}}$. Suppose that $f \in \mathrm{FP}^{A}[g(n)-1]$ via M^A. We will show that $\mathrm{SAT} \in \mathrm{P}$.

Claim: There is a procedure to compute in polynomial time for $x = x_1\#\dots\#x_k$ with $k \geq g(|x|)$ a string $(c_1,\dots,c_k) \in \{0,1\}^k$ with $F_k^{\mathrm{SAT}}(x_1,\dots,x_k) \neq (c_1,\dots,c_k)$.

Proof of claim: Let $k' = g(|x|)$ and let $x' = x_1\#\dots\#x_{k'}\#^l$ such that $|x'| = |x|$. Simulate the computation tree of $M^{()}$ with input x'. We may assume that $M^X(x')$ makes at most $g(|x'|) - 1 = k' - 1$ queries for every oracle X. Hence there are at most $2^{k'-1}$ different paths and at most this many different outputs. So we can find a string $(c_1,\dots,c_{k'}) \in \{0,1\}^{k'}$ which does not occur as an output of any path. Thus $F_k^{\mathrm{SAT}}(x_1,\dots,x_k) \neq (c_1,\dots,c_{k'},0,\dots,0)$. Since $k' \leq \log|x|$ the simulation can be done in polynomial time. ■

Now we can use an appropriate modification of the algorithm in Theorem 4.2 to decide SAT in polynomial time:

On input x, $|x| = n$, we expand the d-self-reduction tree of x until we get in some level more than 2^k nodes, then we discard nodes until we are left with less than 2^k nodes, then we expand the next level, etc. Here k is a function of n that will be determined later.

We select a set of 2^k nodes, form the corresponding sets D_i and compute the y_i. Here y_i is a disjunction of at most 2^k formulas of length at most n. Let $y = y_1\#\dots\#y_k$. We have $|y_i| \leq 4 \cdot n \cdot 2^k$ and $|y| \leq 16 \cdot k \cdot n \cdot 2^k$. In order to apply the Claim and to exclude one possibility for $F_k^{\mathrm{SAT}}(y_1,\dots,y_k)$ we need that $g(|y|) \leq k$.

Since $g(|y|) \leq c \cdot \log|y|$ and g is monotone, we may choose $k = \lceil \frac{2c}{1-c}(\log n + 4) \rceil$. As $k = O(\log n)$ the algorithm runs in polynomial time. ▮

It is open whether Theorem 5.2 holds for $c \geq 1$. Under this weaker hypothesis, Amir, Beigel, and Gasarch [2, Corollary 14, i.] proved that $\Sigma_3^p = \Pi_3^p$.

Krentel [23, Theorem 4.1] proved that if $\mathrm{FP}^{\mathrm{SAT}} \subseteq \mathrm{FP}^A_{O(\log n)\text{-T}}$ for some A then $\mathrm{P} = \mathrm{NP}$. The question whether the hypothesis can be weakened to $\mathrm{FP}^{\mathrm{SAT}}_{\mathrm{tt}} \subseteq \mathrm{FP}^A_{O(\log n)\text{-T}}$ is open, see [19] for a recent survey of related work. It is connected with the existence of tt-hard approximable sets for NP: Using Fact 2.6 it is easy to see that if there is such a set then $\mathrm{FP}^{\mathrm{SAT}}_{\mathrm{tt}} \subseteq \mathrm{FP}^A_{O(\log n)\text{-T}}$ for some A.

6 On self-reducibility and easily countable sets

In this section we look at self-reducible sets that are approximable in some strong sense. The following fact suggests these sets are likely to be in P.

Fact 6.1 (1) *Every self-reducible cheatable set is in* P [2, Theorem 20].
(2) *Every d-self-reducible easily countable set is in* P [17, Theorem 8].
(3) *Every self-reducible p-selective set is in* P [13, Corollary 6].

We show that every self-reducible easily approximable set is in P, and extend (2) by showing that every d-self-reducible set which is T-reducible to an easily countable set is in P.

First we show that every easily countable self-reducible set A is already tt-self-reducible. To this end we need a way to convert a T-reduction to A into a tt-reduction. We will apply a combinatorial tool which was used in recursion theory for the proof of the Cardinality Theorem in [24]: The trees of bounded rank.

Definition 6.2 Let B_r be the full binary tree of depth r. The *rank* of a tree T is the greatest r such that B_r is embeddable into T.

As we shall show, binary trees of bounded rank and polynomial depth have polynomial size. Hence if the computation tree of a T-reduction has bounded rank then we can compute the whole tree in polynomial time. This will allow us to convert T-reductions into truth-table reductions.

In the next lemma we determine $f(r, d, l)$, the maximal number of nodes in a tree with rank less than r, degree d and depth l. The degree is the maximal number of successors of a node, the depth is the length of the longest branch.

Lemma 6.3 *Let $f(r, d, l)$ be the maximal number of nodes of any d-ary tree of depth l and rank less than r. Then $f(r, d, l) = \sum_{i=0}^{r-1}(d-1)^i\binom{l+1}{i+1}$. In particular, $f(r, d, l)$ is a polynomial in d and l, for fixed r.*

Proof: Given any d-ary tree T of depth l and rank less than r, let a be the root node and let $T_1, \ldots, T_d$ be the subtrees below a. If two of them have rank $r-1$, i.e. if B_{r-1} is embeddable into two different subtrees below a, then B_r is embeddable into T (where the root of B_r is mapped to a). Thus only one of the subtrees may have rank $r-1$. So the number of nodes is bounded by $1+f(r, d, l-1)+(d-1)\cdot f(r-1, d, l-1)$, where we count a plus the subtree of rank up to $r-1$ plus the $d-1$ other subtrees of rank up to $r-2$. Conversely, there is a d-ary tree of depth l and rank $r-1$ with this many nodes.

So the recursive equation $f(r, d, l) = 1 + f(r, d, l-1) + (d-1) \cdot f(r-1, d, l-1)$ holds for f. The formula of the theorem follows by induction on l. ∎

We want to deal with the following situation: Suppose we are given $x_1, \ldots, x_k$ and a finite set $D \subseteq \{0,1\}^k$ of possible values of the characteristic function of A on $x_1, \ldots, x_k$. Furthermore we have a query tree T with queries "$x \in A$?" for $x \in \{x_1, \ldots, x_k\}$. All branches of T are consistent with D (i.e. for every branch there is a vector $v \in D$ such that every query "$x_j \in A$?" on this branch is answered by $v[j]$).

We want to bound the rank of T in terms of D. Therefore we define the *width* of D as the maximal possible rank of any such T. This is the greatest i such that we can assign elements from $\{x_1, \ldots, x_k\}$ to the inner nodes of $B_i = \{0,1\}^{\leq i}$ such that the resulting query tree is consistent with D. The formal definition is as follows.

Definition 6.4 Let $D \subseteq \{0,1\}^k$. The *width* of D ($wd(D)$ for short) is the greatest i such that there is a mapping $\psi : \{0,1\}^{<i} \to \{1, \ldots, k\}$ such that for every $\sigma \in \{0,1\}^i$ there is $v \in D$ with $v[\psi(\sigma[0]\sigma[1]\ldots\sigma[j-1])] = \sigma[j]$ for $j = 1, \ldots, i$.

Definition 6.5 Let A be any set. $f \in \mathrm{FP}$ is called an *A-approximation of rank i* iff for all k, $x_1, \ldots, x_k$, on input $(x_1, \ldots, x_k)$, f outputs a set $D \subseteq \{0,1\}^k$ of width less than i which contains $F_k^A(x_1, \ldots, x_k)$.

Remark: If there is an A-approximation of bounded rank then A is recursive (cf. [24, Lemma 1]).

Lemma 6.6 *If A is cheatable, easily approximable, or easily countable then there is an A-approximation of bounded rank.*

Proof: Since the easily approximable sets and the cheatable sets are easily countable it suffices to prove the Lemma for A easily countable.

By definition there is a constant c and a function $f \in \mathrm{FP}$ such that $f(x_1, \ldots, x_c) \in \{0, \ldots, c\}$ and $f(x_1, \ldots, x_c) \neq \chi_A(x_c) + \cdots + \chi_A(x_c)$ for all pairwise distinct $x_1, \ldots, x_c$. Now for any set $\{x_1, \ldots, x_k\}$ let $D(x_1, \ldots, x_k)$ be the set of all vectors v such that $v[i_1] + \cdots + v[i_c] \neq f(x_{i_1}, \ldots, x_{i_c})$ for all pairwise distinct indices $i_1, \ldots, i_c \in \{1, \ldots, k\}$. Obviously $F_k^A(x_1, \ldots, x_k) \in D(x_1, \ldots, x_k)$.

We now show that the width of D is bounded by $4^c - 2$. Suppose for a contradiction that $wd(D(x_1, \ldots, x_k)) \geq 4^c - 2$. Then by a straightforward modification of Lemma 3 in [24] there are pairwise distinct indices $i_1, \ldots, i_c \in \{1, \ldots, k\}$ and $c + 1$ vectors $v_0, v_1, \ldots, v_c \in D(x_1, \ldots, x_k)$ such that $v_j[i_1] + \cdots + v_j[i_c] = j$ for $j = 0, \ldots, c$. But then $v_j \in D(x_1, \ldots, x_k)$ for $j = f(x_{i_1}, \ldots, x_{i_c})$, a contradiction. Hence $wd(D(x_1, \ldots, x_k)) < 4^c - 2$.

Note that D can be computed in polynomial time: Compute inductively the set $D(x_1, \ldots, x_j)$ for $j = 1, \ldots, k$. Since $D(x_1, \ldots, x_j, x_{j+1}) \subseteq D(x_1, \ldots, x_j) \times \{0, 1\}$ and since we can check in polynomial time whether a vector $v \in D(x_1, \ldots, x_j) \times \{0, 1\}$ belongs to $D(x_1, \ldots, x_j, x_{j+1})$, the complexity of the inductive step is polynomial in the size of $D(x_1, \ldots, x_j)$ which is bounded by a polynomial in j. So this algorithm gives an A-approximation of bounded rank. ∎

Remark: (1) If A is cheatable, say A is $(c, c)_p$-verbose, then there is an A-approximation of rank $\lceil \log(c) \rceil + 1$ since for any $x_1, \ldots, x_k$ one can compute in polynomial time a set of at most c strings containing $F_k^A(x_1, \ldots, x_k)$ (cf. [10, Lemma 5.8]).

If A is (a, b)-recursive with $a > \frac{b}{2}$ then there is an A-approximation of rank b, by the argument in [14, p. 683].

(2) Let $g : \{0, 1\}^c \to D$ be a fixed finite function. For a given set A consider the task to compute for all pairwise different $x_1, \ldots, x_c$ a finite set $E(x_1, \ldots, x_c) \subseteq D$ which contains $g(\chi_A(x_1), \ldots, \chi_A(x_c))$. We call A $(k, g)_p$-approximable iff there is such a procedure running in polynomial time with $|E(x_1, \ldots, x_c)| \leq k$, for some fixed k.

For example, let id_c be the identity function, then A is k-cheatable iff A is (k, id_k)-approximable. Let $\#_c(b_1, \ldots, b_c) = b_1 + \cdots + b_c$, then A is easily k-countable iff A is $(k, \#_k)$-approximable.

Assume that A is $(k, g)_p$-approximable. For which values of k can we conclude that A is recursive?

This question is answered in [26]: Let $s(g)$ be the maximal number of different values of g on a set $\{v_0, \ldots, v_c\}$ where $v_0 = 0^c$ and v_{i+1} is obtained from v_i by flipping one component from 0 to 1. It is shown in [26] that there is an $(s(g), g)_p$-approximable nonrecursive set, and that every $(s(g) - 1, g)_p$-approximable set is recursive.

For example $s(id_k) = s(\#_k) = k + 1$. Since there exist cheatable sets not in P, there may in general exist $(k, g)_p$-approximable sets A with $k < s(g)$ which are not in P (in fact one can show that this is the case for $k > 1$). Let us call a set A g-easily approximable if A is $(s(g) - 1, g)_p$-approximable. Using the proof of [26, Theorem 3.4] instead of the proof of the Cardinality Theorem, Lemma 6.6 can be generalized as follows:

$(*)$ *If A is g-easily approximable then there is an A-approximation of bounded rank.*

Since the existence of an A-approximation of bounded rank implies that A is recursive, we cannot weaken the hypothesis of $(*)$ to "$(s(g), g)$-approximable". Hence $(*)$ is tight. Corollary 6.10 (2), (3) below hold more generally for "g-easily approximable" instead of "easily countable".

In the following we say that a set A is *T-easy* if every T-reduction to A can be turned into a tt-reduction, i.e., $f \leq_{\mathrm{T}}^{p} A$ implies $f \leq_{\mathrm{tt}}^{p} A$, or in other words, $\mathrm{FP}^{A} \subseteq \mathrm{FP}_{\mathrm{tt}}^{A}$. For instance, every tally set is T-easy.

Lemma 6.7 *If there exists an A-approximation of bounded rank then A is T-easy.*

Proof: Assume that $f = M^{A}$ where $M^{()}$ is a polynomial time bounded oracle machine. Let g be an A-approximation of rank r. Then we can compute in polynomial time for every x a subtree of the query tree of $M^{()}(x)$ which contains the path determined by oracle A. The queries in this subtree are then used in the tt-reduction.

The subtree is computed as follows: On input x, compute the query tree of $M^{()}(x)$ in a breadth-first fashion, but extend only those branches that are consistent with g. Say in step i we expand the nodes in level i. Let $x_1, \ldots, x_k$ be the list of all queries which have been discovered so far. We extend a branch from level i to level $i+1$ only if there exists a string $v \in g(x_1, \ldots, x_k)$ such that every x_j which is queried on the branch receives the answer $v[j]$.

Let T_i be the subtree consisting of all branches that are extended to level $i+1$. Since these branches are consistent with g it follows that the rank of T_i is less than r. By Lemma 6.3, T_i has at most $f(r, 2, i)$ many nodes. Since i is bounded by a polynomial in $|x|$ we can do step i, and hence the whole construction, in polynomial time. ∎

Note that in the tt-reduction from the previous proof only queries from the computation tree of $M^{()}$ appear. If A is self-reducible via $M^{()}$ then we may assume w.l.o.g. that all queries in the computation tree of $M^{()}(x)$ are shorter than x. So, the corresponding tt-reduction is a tt-self-reduction of A. Thus, we get the following corollary.

Corollary 6.8 *If A is self-reducible and there exists an A-approximation of bounded rank then A is tt-self-reducible.*

Our next results shows that no intractable d-self-reducible set can be T-reduced to an approximable T-easy set.

Theorem 6.9 *Let A, B be any sets with:*
(a) *A is T-easy.*
(b) *A is approximable.*
(c) *$B \leq_{\mathrm{T}}^{p} A$.*
(d) *B is d-self-reducible.*
Then $B \in \mathrm{P}$.

Proof: Assume that the hypotheses hold. Since $B \leq_{\mathrm{T}}^{p} A$ there is a function $f \leq_{\mathrm{T}}^{p} A$ which computes for every $x \in B$ an "accepting path" in the d-self-reduction tree of x. As A is T-easy we get $f \leq_{\mathrm{tt}}^{p} A$. Corollary 2.7 and (b) imply that we can compute for each x in polynomial time a set D which contains $f(x)$. Thus $x \in B$ iff there is an accepting path for x in D. This is a polynomial-time algorithm. ▮

Remark: Theorem 6.9 also holds, by virtually the same proof, if we replace (d) by the weaker condition that "search reduces to decision for B", see [28] for the definition.

Corollary 6.10 (1) *No tally approximable set is* $\leq_{\mathrm{T}}^{p}$*-hard for* NP *unless* P = NP.
(2) *If A is easily countable* (*or easily approximable*) *then any d-self-reducible set B with $B \leq_{\mathrm{T}}^{p} A$ is in* P.
(3) *There is no easily countable* $\leq_{\mathrm{T}}^{p}$*-hard set for* NP *unless* P = NP.

Corollary 6.10 (2) answers an open question of Hoene and Nickelsen [17] who proved the version for m-reducibility instead of T-reducibility.

We do not know whether every self-reducible and easily countable set is in P, but we can show the corresponding result for easily approximable sets:

Theorem 6.11 *If A is self-reducible and easily approximable then $A \in$* P.

Proof: Let A be self-reducible and $(a, b)_p$-recursive via f with $a > \frac{b}{2}$. By Corollary 6.8 we may assume that A is tt-self-reducible. The following algorithm decides A in polynomial time.

On input x compute b iterations of the tt-self-reduction. We get a tree of depth b with x at the root where the direct successors of each inner node y are the elements to which y is tt-self-reduced. Let $\{x_1, \ldots, x_k\}$ be the set of all elements in this tree, with $x = x_1$. Compute all characteristic strings $\sigma \in \{0,1\}^k$ that are consistent with f (i.e., for all pairwise distinct $x_{i_1}, \ldots, x_{i_b}$, $f(x_{i_1}, \ldots, x_{i_b})$ and σ agree in at least a components) and consistent with the self-reduction (i.e., the values at level i are computed from the values at level $i+1$ by the corresponding truth-tables). We claim that $\sigma[1] = \chi_A(x)$ independent of σ. Hence we may just output $\sigma[1]$ for some consistent σ. Since we can compute the set of all consistent strings inductively for $\{x_1, \ldots, x_i\}$, $i = 1, \ldots, k$, we get a polynomial-time algorithm: Any two consistent strings differ in less than b components (cf. Trakhtenbrot's observation which we mentioned after Definition 2.2). So in the i-th iteration we get $O(i^b) = O(k^b)$ many strings which is polynomial in $|x|$ as k is polynomial in $|x|$.

Suppose for a contradiction that there exist two consistent strings σ_1, σ_2 with $\sigma_1[1] \neq \sigma_2[1]$. Let $i_1 = 1$. Because σ_1, σ_2 are consistent with the self-reduction there exists x_{i_2} $|x_{i_2}| < |x|$ on level 1 with $\sigma_1[i_2] \neq \sigma_2[i_2]$. Continuing in this fashion we find $x_{i_3}, \ldots, x_{i_b}$ on levels $2, \ldots, b-1$ with $\sigma_1[i_j] \neq \sigma_2[i_j]$ and $|x_{i_j}| < |x_{i_{j-1}}|$ for $j = 3, \ldots, b$. But then σ_1 and σ_2 differ in b components which is impossible.

Since $(\chi_A(x_1), \ldots, \chi_A(x_k))$ is a consistent string, the algorithm is correct. ▮

7 Some relativized counterexamples

We show that Theorem 6.11 cannot be improved – in a relativizable way – to $(3,2)_p$-verbose (or $(1,2)_p$-recursive) sets. Our result is formulated as general as possible and so it provides relativized counterexamples to several other plausible conjectures.

Theorem 7.1 *In some relativized world with* $\mathrm{P} \neq \mathrm{NP}$ *there exists a* $(3,2)_p$*-verbose, d-self-reducible, sparse c-cylinder* A *which is* $\leq^p_{\mathrm{T}}$*-complete for* NP. *In addition,* A *is* $\leq^p_{2\text{-tt}}$*-reducible to some p-selective set in* NP.

Proof: The proof uses a modification of the proof of Theorem 6.5 in [4]. We start with an oracle B such that $\mathrm{P}^B = \mathrm{NP}^B$. Relative to B we construct by a standard diagonalization a supersparse set C such that $\mathrm{Length}(C) = \{0^n : (\exists x \in C)[|x| = n]\}$ is in $\mathrm{NP}^{B\oplus C} - \mathrm{P}^{B\oplus C}$, $\mathrm{Length}(C) \subseteq \{0^{tow(k)} : k \in \omega\}$ and C contains at most one element of each length. Here $tow(0) = 1$ and $tow(k+1) = 2^{tow(k)}$. Let

$$A = \mathrm{Prefix}(C) = \{(0^n, y) : (\exists z)[\,|yz| = n \wedge \ yz \in C]\}.$$

Obviously $\mathrm{Prefix}(C) \in \mathrm{NP}^{B\oplus C}$. It is shown in [4, Theorem 6.5] that $\mathrm{Prefix}(C)$ is $\leq^p_{\mathrm{T}}$-hard for $\mathrm{NP}^{B\oplus C}$: The main idea is that one can compute in polynomial time all strings in C of length at most n using a prefix search with $\mathrm{Prefix}(C)$ as an oracle. If all strings from C that an $\mathrm{NP}^{B\oplus C}$-machine may ask on input x are known then we can compute in polynomial time (using B as oracle) whether the machine accepts x (since $\mathrm{P}^B = \mathrm{NP}^B$).

Prefix(C) is sparse and d-self-reducible: If $|y| < n$ then $[(0^n, y) \in \mathrm{Prefix}(C) \ \Leftrightarrow\ (0^n, y0) \in \mathrm{Prefix}(C) \ \vee\ (0^n, y1) \in \mathrm{Prefix}(C)]$. (To simplify notation we have d-self-reducibility only in the liberal sense of Meyer and Paterson [27] with respect to some polynomially related well-ordering; of course we can also easily modify the definition of A such that it becomes self-reducible in the classical sense.) Note that Prefix(C) is even 2-d-self-reducible.

Prefix(C) is $(3,2)_p$-verbose (relative to $B \oplus C$): Given $(0^n, y), (0^m, y')$, $|y| \leq n$, $|y'| \leq m$. Since Length(C) is supersparse we can decide relative to C in polynomial time the membership of at least one of the inputs if $n \neq m$. Now suppose that $n = m$. If y is a prefix of y' then $[(0^n, y') \in \mathrm{Prefix}(C) \ \Rightarrow\ (0^n, y) \in \mathrm{Prefix}(C)]$, so we exclude $(0,1)$. If y' is a prefix of y we exclude $(1,0)$. If they are incomparable we exclude $(1,1)$.

Similarly it is shown that Prefix(C) is c-cylinder.

Finally we define $S = \{(0^n, y) : (\exists z \in C)\,[\,|y| \leq |z| = n \ \wedge\ y \preceq z]\}$. Clearly $S \in \mathrm{NP}^{B\oplus C}$.

S is p-selective: Given $(0^n, y), (0^m, y')$, $|y| \leq n, |y'| \leq m$. As above, if $n \neq m$ we can decide membership of at least one of the inputs in polynomial time. If $n = m$ the selector function outputs the minimum of y, y' w.r.t. $\preceq$.

$\mathrm{Prefix}(C) \leq^p_{2\text{-tt}} S$ (it is even a 2-parity reduction): Given y, $|y| \leq n$. If $y \in \{1\}^*$ then $[(0^n, y) \in \mathrm{Prefix}(C) \ \Leftrightarrow\ (0^n, y) \in S]$. Otherwise $y \notin \{1\}^*$, say $y = y'01^s$. Then $[(0^n, y) \in \mathrm{Prefix}(C) \ \Leftrightarrow\ (0^n, y) \in S \ \wedge\ (0^n, y'1) \notin S]$. ∎

Remark: (1) Buhrman, v. Helden, and Torenvliet show in [13, Corollary 13] that if A is self-reducible, B is p-selective, and $A \leq^p_{1\text{-tt}} B$, then $A \in \mathrm{P}$. Theorem 7.1 shows that the generalization to $\leq^p_{2\text{-tt}}$ fails in some relativized world.

(2) Theorem 7.1 also shows that Theorem 4.2 fails in a relativized world if we replace in the hypothesis "bd-cylinder" by "c-cylinder".

(3) Krentel [23, Theorem 4.1] proved that if $\mathrm{FP}^{\mathrm{NP}} \subseteq \mathrm{FP}^{\mathrm{NP}}_{O(\log n)\text{-T}}$ then $\mathrm{P} = \mathrm{NP}$. He asked [23, p. 507] whether the bound $O(\log n)$ can be replaced by any asymptotically larger bound. The oracles constructed by Homer and Longpré [18, Theorem 16] provide relativized counterexamples. Though it is not stated explicitly, they show that for every function $f(n)$ such that $f(n)/\log n$ is non-decreasing and unbounded there is an oracle B such that $\mathrm{P}^B \neq \mathrm{NP}^B$ and $\mathrm{FP}^{\mathrm{NP}^B} \subseteq \mathrm{FP}^{\mathrm{NP}^B}_{f(n)\text{-tt}}$. Thus, Krentel's result is optimal, with regard to relativizing techniques.

The technique of Theorem 7.1 produces a sparse T-complete set. Hence, by our next result, it does not suffice to construct an oracle X with $\mathrm{P}^X \neq \mathrm{NP}^X$ such that there is an approximable tt-hard set for NP^X.

Proposition 7.2 *If there is a sparse T-complete set for* NP *and* $\mathrm{P} \neq \mathrm{NP}$ *then no approximable set is tt-complete for* NP. (*This result relativizes.*)

Proof: Assume that A is a sparse T-complete set for NP. By a well-known result of Hartmanis [15] there is a tally set $B \in \mathrm{NP}$ with $A \leq^p_{\mathrm{tt}} B$, so B is a tt-complete tally set. Now suppose that C is an approximable tt-hard set for NP. Then $\mathrm{SAT} \leq^p_{\mathrm{tt}} B \leq^p_{\mathrm{tt}} C$. Now we argue as in the proof of Theorem 6.9: There is a function $f \leq^p_{\mathrm{T}} \mathrm{SAT}$ which computes for every satisfiable formula a satisfying assignment. It follows that $f \leq^p_{\mathrm{T}} B$, and so $f \leq^p_{\mathrm{tt}} B$ because B is T-easy. Hence we get $f \leq^p_{\mathrm{tt}} C$. Since C is approximable we can compute by Corollary 2.7 for each x in polynomial time a set of assignments such that if $x \in \mathrm{SAT}$ then one of the assignments makes x true. This shows that $\mathrm{SAT} \in \mathrm{P}$, i.e., $\mathrm{P} = \mathrm{NP}$.

The proof relativizes because for all oracles X there is a d-self-reducible NP^X-complete set. This can be used instead of SAT in the proof above. ▮

8 Conclusion

We have substantiated a conjecture of Amir, Beigel, and Gasarch [2] that natural sets are either p-superterse or in P. It seems likely that natural sets are either in P or not even reducible to an approximable set. We proved this for btt-reductions and came very close to a proof for tt-reductions. It should be noted that our results are shown without additional hypotheses like the collapse of PH or $\mathrm{NP} \subseteq \mathrm{P/Poly}$.

We do not know of any relativized counterexamples to the following conjectures which may therefore also be tractable, and which we recommend for further research.

Conjecture 8.1
(1) *Every* $\leq^p_{\mathrm{tt}}$*-hard set for* NP *is p-superterse unless* $\mathrm{P} = \mathrm{NP}$.
(2) *If* A *is self-reducible and easily countable then* $A \in \mathrm{P}$.

Acknowledgments. The first author is grateful to CWI and Paul Vitányi for their hospitality during his visit to Amsterdam. We would like to thank two anonymous referees and Bill Gasarch for comments and corrections.

References

[1] M. Agrawal, V. Arvind. Polynomial time truth-table reductions to p-selective sets. In *Proceedings of the 9th Conference on Structure and Complexity Theory*, IEEE Computer Society Press, 24–30, 1994.

[2] A. Amir, R. Beigel, W. I. Gasarch. Some connections between bounded query classes and non-uniform complexity. In *Proceedings of the 5th Conference on Structure and Complexity Theory*, IEEE Computer Society Press, 232–243, 1990.

[3] A. Amir, W. I. Gasarch. Polynomial terse sets. *Information and Computation*, 77:37–56, 1988.

[4] K. Ambos-Spies, J. Kämper. On disjunctive self-reducibility. *CSL'88*, Lecture Notes in Computer Science, Vol. 385, pp. 1–13, 1989.

[5] J. L. Balcázar, J. Diaz, J. Gabarró. *Structural complexity I.* Springer-Verlag, Berlin, 1988.

[6] R. Beigel. Query-limited reducibilities. Ph.D. thesis, Stanford University, 1987.

[7] R. Beigel. A structural theorem that depends quantitatively on the complexity of SAT. In *Proceedings of the 2nd Conference on Structure and Complexity Theory*, IEEE Computer Society Press, 28–32, 1987.

[8] R. Beigel. NP-hard sets are p-superterse unless R = NP. TR 4, Johns Hopkins Univ., Dept. of C.S., 1988.

[9] R. Beigel. Bi-immunity results for cheatable sets. *Theoretical Computer Science*, 73:249–263, 1990.

[10] R. Beigel. Bounded queries to SAT and the Boolean hierarchy. *Theoretical Computer Science*, 83:199–223, 1991.

[11] R. Beigel, W. I. Gasarch, J. Gill, J. C. Owings, Jr. Terse, superterse, and verbose sets. *Information and Computation*, 103:68–85, 1993.

[12] R. Beigel, M. Kummer, F. Stephan. Quantifying the amount of verboseness. To appear in: Information and Computation.

[13] H. Buhrman, P. v. Helden, L. Torenvliet. P-selective self-reducible sets: A new characterization of P. In *Proceedings of the 8th Conference on Structure in Complexity Theory*, 44–51, IEEE Computer Society Press, 1993.

[14] V. Harizanov, M. Kummer, J. C. Owings, Jr. Frequency computation and the cardinality theorem. *J. Symb. Log.*, 57:682–687, 1992

[15] J. Hartmanis. On sparse sets in NP-P. *Information Processing Letters*, 16:55-60, 1983.

[16] L. A. Hemaspaandra, A. Hoene, A. V. Naik, M. Ogiwara, A. L. Selman, T. Thierauf, J. Wang. Selectivity: Reductions, nondeterminism, and function classes. State University of New York at Buffalo, TR 93-28, 1993.

[17] A. Hoene, A. Nickelsen. Counting, selecting, and sorting by query-bounded machines. In *STACS 93*, pp. 196–205, Lecture Notes in Computer Science, Vol. 665, 1993.

[18] S. Homer, L. Longpré. Reductions of NP sets to sparse sets. *Journal of Computer and System Sciences*, 48:324–336, 1994.

[19] B. Jenner, J. Torán. Computing functions with parallel queries to NP. In *Proceedings of the 8th Conference on Structure in Complexity Theory*, 280–291, IEEE Computer Society Press, 1993.

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

[21] K. Ko. On self-reducibility and weak p-selectivity. *J. Comp. Syst. Sci.*, 26:209–221, 1983.

[22] J. Köbler, U. Schöning, J. Torán. *The graph isomorphism problem.* Birkhäuser, Boston, 1993.

[23] M. W. Krentel. The complexity of optimization problems. *J. Comput. Syst. Sci.*, 36:490–509, 1988.

[24] M. Kummer. A proof of Beigel's cardinality conjecture. *J. Symb. Log.*, 57:677–681, 1992.

[25] M. Kummer, F. Stephan. Some aspects of frequency computation. Technical Report Nr. 21/91, Fakultät für Informatik, Universität Karlsruhe, 1991.

[26] M. Kummer, F. Stephan. Effective search problems. *Math. Log. Quart.*, 40:224–236, 1994.

[27] A. Meyer, M. Paterson. With what frequency are apparently intractable problems difficult? MIT Tech. Rep., MIT/LES/TM-126, 1979.

[28] A. V. Naik, M. Ogiwara, A. L. Selman. P-selective sets, and reducing search to decision vs. self-reducibility. In *Proceedings of the 8th Conference on Structure in Complexity Theory*, 52–64, IEEE Computer Society Press, 1993.

[29] M. Ogihara. Polynomial-time membership comparable sets. In *Proceedings of the 9th Conference on Structure in Complexity Theory*, IEEE Computer Society Press, 2–11, 1994.

[30] S. Rogina. Kardinalitätsberechnungen. Studienarbeit, Fakultät für Informatik, Universität Karlsruhe, 1992.

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

[32] T. Thierauf, S. Toda, O. Watanabe. On sets bounded truth-table reducible to p-selective sets. In *STACS 94*, pp. 427–438, Lecture Notes in Computer Science, Vol. 775, 1994.

[33] S. Toda. On polynomial-time truth-table reducibility of intractable sets to p-selective sets. *Math. Systems Theory*, 24:69-82, 1991.

[34] B. A. Trakhtenbrot. On frequency computation of functions. *Algebra i Logika*, 2:25–32, 1963. (Russian)