A Note on Bi-immunity and $p$-Closeness of $p$-Cheatable Sets in $P/\text{poly}$*

JUDY GOLDSMITH
University of Manitoba, Winnipeg, Manitoba, Canada R3T 2N2

DEBORAH JOSEPH
University of Wisconsin-Madison,
1210 West Dayton Street, Madison, Wisconsin 53706

AND

PAUL YOUNG
University of Washington, Seattle, Washington 98195

Received March 8, 1988; revised January 15, 1992

In this paper we study the interplay between three measures of polynomial time behavior in sets: $p$-cheatability, $p$-closeness, and membership in $P/\text{poly}$. First we construct $(2^k - 1)$—$p$-cheatable sets that are bi-immune with respect to all recursively enumerable sets. We show that the constructed sets are in $P/\text{poly}$, but can be constructed so that they are not $p$-close. In fact, they can be constructed so that they are not even recursively close. Next, we construct $(n for 2)$—$p$-cheatable sets that are bi-immune with respect to arbitrarily difficult deterministic time classes. These sets are also in $P/\text{poly}$, and they also can be constructed so that they are not $p$-close. Finally, we construct a set that is $(n for 1)$—$p$-cheatable but is not $p$-close, although it, too, is in $P/\text{poly}$. These results show that, although $p$-cheatable, $P/\text{poly}$, and $p$-close sets all exhibit some form of polynomial time behavior, the notions of $p$-cheatability and $p$-closeness are often orthogonal. In addition, the results on $p$-closeness answer conjectures made in Amir and Gasarch (Inform. and Comput. 77 (1988), 37–56).


1. INTRODUCTION

In recent years a popular topic in structural complexity theory has been the extent to which sets can exhibit some measure of polynomial time behavior without actually being in the class, $P$, of sets decidable in polynomial time. In this paper we consider three examples of this phenomenon. One is the property of a set $A$ being $p$-close, that is, there is a polynomially decidable set whose symmetric difference

* This work was supported in part by the National Science Foundation under Grant DCR-8402375, as well as by a grant from AT & T Bell Laboratories. The work was done while the first author was a graduate student at the University of Wisconsin-Madison. The work reported here provides proofs for, and in some cases slightly extends, some of the material in [GJY-87a, GJY-87b] reported at the Second Annual Structure in Complexity Theory Conference.

349
with \( A \) is a polynomially sparse set. Another is of a set being in \( P/poly \), that is, of having polynomial size circuits that test membership in the set. The third is of a set being \( p \)-cheatable in the sense that some large number of membership questions about the set can in polynomial time be reduced to some (much) smaller set of membership questions about the set.

In contrast to measuring the extent to which a set may exhibit polynomial time behavior without actually being in \( P \), one is often interested in some description of how "badly" a set fails to exhibit polynomial time behavior. In addition to being not \( p \)-close, one way of showing that a set does not exhibit polynomial time behavior is to show that it is polynomially immune (or bi-immune), that is, that the set (or the set and its complement) fails to have an infinite polynomially decidable subset.

In this paper we are interested in the interplay of these measures of polynomial time behavior. Roughly speaking, we show that several of these ways of measuring polynomial-like behavior, or the lack thereof, are to a large extent orthogonal to each other. For example, we construct various highly \( p \)-cheatable sets that are highly bi-immune and that are not \( p \)-close.

The portion of our work that deals directly with \( p \)-closeness was stimulated by [A & G-87], where a number of conjectures were made about connections between \( p \)-cheatability and \( p \)-closeness. Straightforward extensions of our work in [GJY-87a, GJY-87b] answer all of these conjectures.

Our first, and hardest, theorem is that there exist \((2^k - 1) \) \( p \)-cheatable sets that are bi-immune with respect to all recursively enumerable (r.e.) sets. This result, which was announced without proof in [GJY-87a], was greeted with some skepticism at the time of announcement. The proof, like all proofs in this paper, is motivated by recursion theoretic ideas. In this case, as announced, our proof is modelled on Jockush's construction of a retraceable, semi-recursive set that is bi-immune [Jo-68].

This result contrasts with the situation for \((2^k \) for \( k \) \( p \)-cheatable sets. Beigel et al. [BGGO-87] have given a subtle argument showing that all \((2^k \) for \( k \))-cheatable sets are decidable (and hence certainly not bi-immune). (A simpler proof of decidability, but one that works only for \((2^k \) for \( k \) \( p \)-cheatable sets, is given in [GJY-87a].) In spite of always being decidable, even \((2^k \) for \( k \) \( p \)-cheatable sets can have arbitrarily high computational complexity [A & G-87]. The second theorem of this paper shows that even more highly \( p \)-cheatable sets cannot only be polynomially bi-immune, but can be bi-immune with respect to any deterministic time class, a result stronger than merely showing that such sets can have arbitrarily high computational complexity. In this second theorem we prove that for any deterministic time class Dtime\((T(m)) \) there are \((k \) for \( 2 \) \( p \)-cheatable sets that are bi-immune with respect to Dtime\((T(m)) \). (The proof of this was given in [GJY-87b]. An independent proof by Beigel is given in [Be-87b].) Note that this result is in contrast to the situation for \((2 \) for \( 1 \) \( p \)-cheatable sets, which can easily be shown to never be even polynomially bi-immune [Be-87b].

In both Theorems 1 and 2, the sets we construct are in \( P/poly \). To answer Amir and Gasarch's conjectures in [A & G-87], we give easy modifications of the
constructions in Theorems 1 and 2 so that the resulting sets are not p-close. An easy modification of the construction given in the second theorem also yields a third theorem that provides a negative answer to another of these conjectures: in Theorem 3 we prove that there are sets that are not p-close, but which are (n for 1) — p-cheatable. These sets are also in P/poly.

**DEFINITION.** A set A is (n for k) — p-cheatable (n and k fixed constants) if there is a polynomial-time oracle machine M such that if M is given inputs \( \langle x_1, ..., x_n \rangle \) and an oracle for A, then with k or fewer queries to the oracle M determines membership in A for each of \( x_1, ..., x_n \). If n can vary, that is, if the algorithm never makes more than k queries no matter how many inputs it is given, then we say that A is k — p-cheatable. If the machine M is merely computable with respect to the oracle A, but does not necessarily run in polynomial time, then the set A is said to be simply (n for k) — cheatable, or simply k-cheatable if n can be arbitrarily large for a fixed k.¹

**DEFINITION.** An infinite set A is bi-immune if neither A nor \( \overline{A} \) has an infinite r.e. subset. Similarly, an infinite set is polynomially bi-immune if neither the set nor its complement has an infinite polynomially decidable subset.

**DEFINITION.** A set A is recursively-close if there is a recursively enumerable set \( W_j \) such that the symmetric difference of A and \( W_j \) (i.e., \( A - W_j \cup W_j - A \)) has at most some polynomial number of elements of size n for all n (i.e., the symmetric difference is polynomially sparse). The set A is p-close if there is a polynomially decidable set P such that the symmetric difference of A and P is polynomially sparse.

### 2. Constructions

Sets that are \( (2^k - 1 \text{ for } k) - p \)-cheatable were called verbose by Amir and Gasarch in [A & G-87], and in that paper they conjectured that such sets need not be p-close. In Theorem 1 we confirm this conjecture by constructing a \( (2^k - 1 \text{ for } k) - p \)-cheatable set that is bi-immune and not even recursively-close.

The next trivial lemma should serve as motivation for what follows:

**LEMMA.** Suppose \( \leq_T \) is a linear ordering and that \( A \subseteq N \) is a set satisfying \( y \in A \) and \( w \leq_T y \) implies that \( w \in A \). Then given any k and given any \( n < 2^k \) integers we can, given the ordering \( \leq_T \) of the n integers, determine membership with respect to A for these n integers by asking membership questions about A for only k of these integers.

¹ In [Be-87a], in discussing \( (2^k \text{ for } k) \)-cheatable and p-cheatable sets, Beigel allows the oracle to be an arbitrary fixed set, B. In [A & G-87], in discussing \( (2^k - 1 \text{ for } k) - p \)-cheatable sets (which they call verbose sets), Amir and Gasarch define p-cheatable sets A as using A as oracle. For our purposes, either oracle convention will work, since all of our results depend merely on the number of queries to the oracle and are independent of the set actually used as the oracle.
Proof. Just do the obvious binary search for the "break point" or "boundary" for the $n$ integers' membership in $A$. 

With this we can now prove our first result:

**Theorem 1.** There exists a set $A$ that is

(i) $(2^k - 1$ for $k$) - $p$-cheatable for all $k$,

(ii) bi-immune,

(iii) not recursively close,

(iv) in $P/poly$ (i.e., has polynomial size decision circuits).

Proof. We first construct a set that is $(2^k - 1$ for $k$) - $p$-cheatable and bi-immune. We then give an easy modification of the construction to keep the set from being recursively-close, and we then prove that the resulting sets are in $P/poly$.

Let $f$ be any function with domain $N - \{0\}$ and range $N$. Assume that the graph with directed edges given by the pairs $(w, f(w))$ forms a tree, $T$, with root 0, and $f(w)$ as the parent node of $w$. For later purposes we will also assume that $|0| = 0$, but that for all other integers $|n|$ is the length of $n$ in any base other than base one. Then $T$ induces a corresponding linear ordering $\leq_T$ on $N$ by taking the natural "left-to-right" ordering of branches and ordering integers on the same branch by their distance from the root. The ordering $\leq_T$ is defined formally as follows:

(i) If $w$ and $y$ are on the same branch of the tree, we define $w \leq_T y$ iff $f^i(y) = w$ for some $i \geq 0$.

(ii) If $w$ and $y$ are not on the same branch of the tree, then $f^i(w) = f^j(y)$ for some unique minimal pair $i, j > 0$. In this case define $w \leq_T y$ iff $f^{i-1}(w) < f^{j-1}(y)$ (as integers).

Our goal now will be to define a function $f$ which induces a tree $T$ as above. Since $f$ will be polynomial-time computable and length-decreasing, $\leq_T$-comparisons will be polynomial-time computable as well. Thus, given any $2^k - 1$ integers $x_1, \ldots, x_{2^k-1}$, we can in time polynomial in $|x_1| + |x_2| + \cdots + |x_{2^k}-1|$ find the tree ordering of $x_1, \ldots, x_{2^k-1}$ under $\leq_T$. As we construct $f$, we will use this tree ordering to define a bi-immune set $A$ such that for any $w$ and $y$, $w \leq_T y$ and $y \in A$ implies $w \in A$.

If we can accomplish this, the result will be proved. To determine membership in $A$, for all of $x_1, \ldots, x_{2^k-1}$ by the preceding lemma, we need only do a binary search on these elements with respect to the ordering $\leq_T$ to find the boundary point for membership in $A$ within this list.

$A$ will be defined in stages. For each $n$, at the beginning of Stage $n$ we will have defined a node, $\text{TARGET}(n)$, of the tree, such that $|\text{TARGET}(n)| < n$. We will make $f$ the same for all integers of the same length and also make $f$ strictly size decreasing. Specifically, we will define

$$f(w) = \text{TARGET}(n) \quad \text{for all } w \text{ such that } |w| = n$$
and we will define

\[ A_n = \{ y \mid y \leq_T \text{TARGET}(n) \}, \quad \overline{A}_n = \{ y \mid |y| < n \} - A_n. \]

The sets \( A_n \) will be approximations to \( A \) in the sense that we will prove

\[ \lim_{n \to \infty} \text{TARGET}(n) = \infty \]

and will also prove that the set of nodes, \( \{\text{TARGET}(n)\} \), define a unique infinite branch in \( T \), so that we can set

\[ A = \lim_{n \to \infty} A_n, \]

i.e., \( A \) is the set of nodes \( w \) such that \( w < \text{TARGET}(n) \) for almost every \( n \).

It follows from the preceding discussion, that if \( \text{TARGET}(n) \) is defined in a way that is polynomially computable in \( n \) (not in \( |n| \)), then \( f(w) \) will be polynomially computable in \( |w| \), because, since it is length decreasing, \( f \)'s computation can, if necessary, be driven all the way back to zero in at most \( |w| \) iterations. Furthermore, for any \( 2^k - 1 \) elements \( x_1, \ldots, x_{2^k-1} \), the tree ordering \( \leq_T \) of \( x_1, \ldots, x_{2^k-1} \) can be found in time that is polynomial in the sum of the lengths of \( x_1, \ldots, x_{2^k-1} \). Thus, since the set \( A \) (\( \equiv \lim_{n \to \infty} A_n \)) is completely determined by our definition of the function \( \text{TARGET} \), all that remains is to define \( \text{TARGET}(n) \) so that \( T \) is polynomially computable in \( n \) and so that the resulting set \( A \) is bi-immune.

Let \( \{W_j\}_{j \in \mathbb{N}} \) be an effective enumeration of all r.e. sets. We will employ a simple (finite injury) priority argument to guarantee that for each \( j \), if \( W_j \) is infinite, then \( W_j \) intersects both \( A \) and \( \overline{A} \). At any given stage in the construction, we will have a finite number of r.e. sets \( W_j \) such that there is some \( z \in W_j \) that is known to be in \( A_n \) and there may also be some \( z' \in W_j \) that is known to be in \( \overline{A}_n \). In the first of these cases we will say that \( j \) is protected for \( A_{n+1} \) by inclusion of \( z \) and in the second we will say that \( j \) is protected for \( A_{n+1} \) by exclusion of \( z' \). We will try to maintain for all such \( z \) that \( z \leq_T \text{TARGET}(n+1) \) and for all such \( z' \) that \( z' \leq_T \text{TARGET}(n+1) \). While we will not succeed in doing this in each case as we pass from Stage \( n \) to Stage \( n+1 \), we will see that we will succeed in doing this in the limit as \( n \) goes to infinity.

To this end recall that \( |0| = 0 \), and proceed in stages to construct \( \text{TARGET}(n) \) as follows:

\[ IB_T = \{ y \mid \exists \text{ infinitely many } w \in \text{Range} (\text{TARGET}) \text{ such that } f'(w) = y \text{ for some } i \} \]

then, as we will show, \( IB_T \) will be the unique branch in the tree \( T \) and \( IB_T \) will be retraced by \( f \). Furthermore, \( A \) will be the set of all points that either lie on \( IB_T \) or lie on a branch to the "left" of this infinite branch. That is, \( A = \{ y \mid y \leq z \text{ for some } z \in IB_T \} \). Although we will not explicitly use the fact, the reader may find it useful to keep in mind that \( \leq_T \) will impose on \( N \) (\( = \text{the set of all natural numbers} \)) the order type of all negative and positive rationals of the form \( 1/n \), \( n \) a negative or positive integer, and \( A \) will then correspond to the negative rationals in this ordering.

\[ \]
Stage 0. Define $\text{TAR}\text{G}E\text{T}(1) = 0$.

Stage $n$ ($n \geq 1$).

Step A. For each $j < \log(n)$ such that $W_j$ is not yet known to be protected for both $A_n$ and for $\overline{A}_n$ spend just a few more steps enumerating $W_j$.

Step B. For the smallest $j$ (if any) such that

(i) we have found a $z$ with $|z| < n$, $z \in W_j$, and $j$ not protected for $A_n$ by prior inclusion of some $z' \geq_T z$, and

(ii) no $j' < j$ is currently protected for $A_n$ by some $z'$ with $z' \geq_T z$, and

(iii) no $j' < j$ is currently protected for $\overline{A}_n$ by exclusion of some $z'$ with $z' \leq_T z$,

choose the largest such $z$. If:

Subcase B.1. $j$ is not yet known to be protected for $A_n$:

in this subcase define

$$\text{TAR}\text{G}E\text{T}(n + 1) = \max_{<_T} \{z, \text{TAR}\text{G}E\text{T}(n)\}$$

and say that $j$ is \textit{protected} for $A_{n+1}$ by \textit{inclusion} of $z$. (Note that by our description of our choice of $z$ and by an inductive proof on $\text{TAR}\text{G}E\text{T}(n)$ this does not change any protections for any $j' < j$.)

Subcase B.2. $j$ is known to be protected for $A_n$ by prior inclusion of some $z'$:

in this subcase, by our choice of $z$, we know, among other things, that $z' <_T z$. To keep $z$ excluded from $A$ we want to keep $\text{TAR}\text{G}E\text{T}(n) <_T z$; so in this subcase we define

$$z_{\text{max}} = \max_{<_T} \{z' \mid z' \text{ protects some } j' < j \text{ for } A_n\},$$

and we then define

$$\text{TAR}\text{G}E\text{T}(n + 1) = \max_{<_T} \{z_{\text{max}}, \text{TAR}\text{G}E\text{T}(n)\}$$

and we say that $j$ is \textit{protected} for $\overline{A}_{n+1}$ by \textit{exclusion} of $z$. (Note that $j$ is still protected for $A_n$ by inclusion of the smaller $z'$, and that $z$ and, by induction on $n$, the values $\text{TAR}\text{G}E\text{T}(n)$ are so chosen that this changes no protection for any $j' < j$.)

Step C. If Step B is vacuous, we define $\text{TAR}\text{G}E\text{T}(n + 1) = \text{TAR}\text{G}E\text{T}(n)$.

If Step B is nonvacuous, then before the start of Stage $n + 1$, \textit{terminate} the protection of all $j' > j$ by erasing all protection memories for all $j' > j$.

Go on to Stage $n + 1$. 
This completes our description of how to compute the functions $TARGET(n)$ and $f(w)$, and also our descriptions of the sets $A_n$. It can be seen from our description that $TARGET(n)$ is computable in time polynomial in $n$, and so it remains to show that

$$\lim_{n \to \infty} TARGET(n) = \infty,$$

that the nodes \{TARGET(n)\} define a unique infinite branch, so that the set $A = \lim_{n \to \infty} A_n$ is well-defined, and also to show that the resulting set $A$ is bi-immune. Not surprisingly, these facts are closely interrelated.

We first observe that for any r.e. set $W$, if at some Stage $n$, $j$ is protected for $A_n$ by inclusion of $z$ and $j$ is protected for $A_n$ by exclusion of $z'$, and, if for all Stages $n'$ with $n' > n$ we have

$$z \leq_T TARGET(n') <_T z',$$

then we are guaranteed that for all Stages $n'$ with $n' > n$ we will have that $j$ is protected for $A_{n'}$ by inclusion of $z$ and that $j$ is protected for $A_{n'}$ by exclusion of $z'$. Assuming that $\lim_{n \to \infty} TARGET(n) = \infty$ and that $A$ is well defined, this would imply that $W_j$ intersects both $A$ and $\overline{A}$.

Proceeding inductively, assume now that for some fixed $j$ we have reached a Stage $n$ of the construction such that for all $j' < j$ no protection of $j'$ will ever be initiated or terminated after Stage $n$. Then any protection of $j$ that is in place at Stage $n$ or initiated after Stage $n$ cannot be terminated, because the termination of a protection of $j$ requires the initiation of some protection for some $j' < j$. From this it follows that:
(i) Any protection of \( j \) that is initiated after Stage \( n \) cannot be terminated since this would require an initiation of a protection of some \( j' < j \). Thus there is some stage after which no protection of \( j \) is ever initiated or terminated. This completes the inductive proof that protections must stabilize, but we also see:

(ii) For any \( W_j \), once all protections for all \( j' < j \) have stabilized, if we set

\[
\z_{\text{max}} \overset{\text{def}}{=} \max_{w \in T} \{ w \mid w \text{ protects some } j' < j \text{ by inclusion of } w \},
\]

and

\[
\z_{\text{min}} \overset{\text{def}}{=} \min_{w \in T} \{ w \mid w \text{ protects some } j' < j \text{ by exclusion of } w \},
\]

then from that point on for all sufficiently large \( n' \)

\[
\z_{\text{max}} \leq_T \TARGET(n') <_T \z_{\text{min}},
\]

which guarantees that all sufficiently large integers \( z \) must satisfy

\[
\z_{\text{max}} <_T z <_T \z_{\text{min}},
\]

But this means that if \( W_j \) is infinite then, after a sufficiently large stage, there must be some \( z \in W_j \) that satisfies

\[
\z_{\text{max}} <_T z <_T \z_{\text{min}},
\]

and this enables us to protect \( j \) for \( A_n \) by inclusion of \( z \). A similar analysis shows that once we have protected \( j \) for \( A_n \) by inclusion of some \( z \), then if \( W_j \) is infinite we must eventually be able to find some \( z' \) that enables us to protect \( j \) for \( A_n \) by exclusion of \( z' \), without changing any protections of any \( j' < j \) or of the existing protection of \( j \) for \( A_n \). This guarantees that the set \( A = \overset{\text{def}}{= \lim_{n \to \infty} A_n} \), if well defined, is bi-immune to all r.e. sets.

(iii) From the previous argument, we see that \( \lim \sup_{n \to \infty} \TARGET(n) \) must be infinite, for otherwise we could not protect all infinite r.e. sets. But in fact once \( z \) is protecting \( j \) we cannot set \( \TARGET(n + 1) < z \) without forcing \( j \)'s protection by \( z \) to be terminated. Thus, \( \lim \inf_{n \to \infty} \TARGET(n) \) must also be infinite.

(iv) Next we can see that the nodes, \( \{ \TARGET(n) \} \), define a unique infinite branch of \( T \). Otherwise, if \( B_1 \) and \( B_2 \) were branches in \( T \), each containing infinitely many nodes from \( \{ \TARGET(n) \} \), then there would be a \( z \) such that, without loss of generality, \( b_1 <_T z <_T b_2 \) for almost all \( b_1 \in B_1, b_2 \in B_2 \). The tree order is polynomial-time computable, so for some \( j \), \( W_j = \{ \omega : z <_T \omega \} \). At some stage \( n \), the protections for \( W_j, j' < j \) will have stabilized, and so \( j \) will be protected for \( A_n \) by inclusion of some \( \omega \in W_j \). For all \( n' > n \), \( \TARGET(n') >_T \omega \). By definition
of \( W_j, w >_T z \), so \( \text{TARGET}(n') > b_1 \) for almost all \( b_1 \in B_1 \), contradicting the assumption on \( B_1 \).

This completes the construction of a (recursively) bi-immune set which is \( 2^k - 1 \) for \( k - p \)-cheatable for all \( k \).

We now explain how to modify the construction to keep the set \( A \) from being recursively close. For each \( j \), the preceding construction had two goals: to manipulate the definition of \( \text{TARGET}(n) \) so that if \( W_j \) is infinite, then in the limit \( W_j \) always intersects \( A_n \), and hence intersects \( A \), and also to manipulate the definition of \( \text{TARGET}(n) \) so that if \( W_j \) is infinite, then in the limit \( W_j \) always intersects \( A_n \), and hence intersects \( \overline{A} \).

We now add one more goal for each \( j \): We want to find at least one length, \( m \), such that if at least half of the strings of length \( m \) are in \( W_j \) then all but one of the strings of length \( m \) are in \( A \), but if over half of the strings of length \( m \) are in \( W_j \), then all strings of length \( m \) are in \( A \). If we can accomplish this, then, bearing in mind that every r.e. set \( W_j \) really has infinitely many indices, \( A \) cannot be recursively close since infinitely often at least half of the strings of any given length are in the symmetric difference of \( W_j \) and \( A \). But if we could choose the length \( m \) for which we want this to happen, we could then easily succeed by initially setting \( \text{TARGET}(n) \) to be the largest number of length \( m \), and then, as \( n \) increases and more members of \( W_j \) are discovered as we enumerate \( W_j \), if we discover that more than half the numbers of size \( m \) are in \( W_j \) we then switch \( \text{TARGET}(n) \) to the smallest number of length \( m \).

Now in the construction of the \( p \)-cheatable bi-immune set which we have just given, in order to make the set bi-immune, strings of every length must be available to diagonalize against the possibility that, if infinite, \( W_j \) is a subset of \( A \) or of \( \overline{A} \). These two requirements were realized by our attempts to protect \( j \) for \( A \) and to protect \( j \) for \( \overline{A} \). We now add a third requirement for \( j \), namely that for some \( m \), if at least half of the strings of length \( m \) are in \( W_j \) then all but the smallest string of length \( m \) is in \( A \), while if over half of the strings of length \( m \) are in \( W_j \), then all strings of length \( m \) are in \( A \). This third requirement for each \( j \) is blended with the other requirements, and it achieves highest priority at any stage when all requirements for all smaller \( j \)' or for the two other requirements for \( j \) itself have already been met or cannot be met at that level.

We can blend this additional diagonalization on all sets \( W_j \) into the preceding priority argument to keep the resulting set \( A \) from being recursively close: For each \( j \), the desired length \( m \) for this diagonalization varies, becoming associated in the priority argument with this third requirement for \( W_j \). The same induction as above then shows that the priorities eventually stabilize, so that eventually the target length \( m \) for meeting this third requirement for \( W_j \) becomes fixed, and the requirement is then met as \( W_j \) is enumerated.

To finally complete the proof, we must still show that the set \( A \) is in \( P/poly \). But this is known to be equivalent to being polynomial time Turing reducible to a polynomially sparse set. Intuitively, it seems clear that the set \( A \) should be reducible
to a suitable encoding of the single infinite branch, \(IB_T\), through the tree \(T\), since tree elements on the branch or to the left of the branch are in \(A\) while elements to the right of the branch are in \(\overline{A}\). This intuition suggests the following: for each length \(m\), either there are no elements of length \(m\) in \(A\), or there is an element \(x = b_1 b_2 \ldots b_m\) of length \(m\) that lies on \(IB_T\), or all elements of length \(m\) are in \(A\). (Here \(b_1 b_2 \ldots b_m\) is the binary representation of \(x\). We will use \(2^m\) as the standard notation when \(b_1\) is one with trailing bits all zero.) For each \(m\), define the set \(A'_m\) by

\[
A'_m = \begin{cases} 
\langle 2^m, 0 \rangle & \text{if } A \text{ contains no elements of length } m, \\
\langle 2^m, m + 1 \rangle & \text{if } A \text{ contains all elements of length } m, \\
\{\langle 2^m, i \rangle \mid b_i = 1\} & \text{otherwise.}
\end{cases}
\]

Now define \(A' = \bigcup_m A'_m\). Clearly \(A'\) is polynomially sparse. Furthermore, \(A\) is Turing reducible to \(A'\) in polynomial time: given an element \(x\) of length \(m\), two quick checks to \(A'\) tell whether all or no elements of length \(m\) are in \(A\). If these two checks do not immediately tell that \(x\) is or is not in \(A\), then \(m\) direct questions to \(A'\) determine the binary representation of the \(\preceq_T\) "break point" for membership in \(A\) for elements of length \(m\). In this case, \(x\) is in \(A\) if and only if \(x\) is \(\preceq_T\) this break point.

One might still ask whether the [A & G-87] conjecture that there are \(2^k - 1\) for \(k\) - \(p\)-cheatable sets that are not \(p\)-close might become false if the set \(A\) were required to be r.e., but even here the conjecture can be shown to be true: the preceding construction can be modified to make the set \(A\) r.e. and the set \(\overline{A}\) recursively immune, while still keeping the set \(A\) polynomially immune. In this case with our definitions, we cannot keep \(A\) from being recursively close, but we can use the same construction to keep \(A\) from being polynomially close. In fact, given any deterministic time class \(\text{Dtime}(T(m))\) we can construct a recursively enumerable set \(A\) so that it is not "close" to any set in \(\text{Dtime}(T(m))\). Giving up the full immunity of \(\overline{A}\) will similarly allow us to make \(A\) decidable in time roughly exponential in time \(T(m)\), while keeping it \((2^k - 1) - \text{p-cheatable and not } T(m)\)-close.

Our next theorem gives very strong examples of highly cheatable sets that are bi-immune and are not \(p\)-close:

**Theorem 2.** Let \(\text{Dtime}(T(m))\) be any deterministic time class. There is a \(2 - \text{p-cheatable set } A\) that is bi-immune with respect to \(\text{Dtime}(T(m))\). That is, neither \(A\) nor \(\overline{A}\) has an infinite \(\text{Dtime}(T(m))\) decidable subset.\(^3\) Furthermore, the set \(A\) can be taken so that it is not \(p\)-close. In fact, \(A\) can be modified so it is not close to any set in \(\text{Dtime}(T(m))\). Finally, \(A \in P/poly\).

**Proof.** We first show how to construct a \(2 - \text{p-cheatable set that is bi-immune with respect to } \text{Dtime}(T(m))\). We then explain how to modify the construction to keep the set \(A\) from being close to any member of \(\text{Dtime}(T(m))\).

\(^3\) As mentioned above, this result was proved in [GJY-87b] and independently in [Be-87b].
Let \( \{ M_i \}_{i \in \mathbb{N}} \) be a canonical enumeration of programs that contains all programs that always halt within \( \text{Dtime}(T(m)) \) and let \( L(M_i) = \{ x : M_i(x) = 1 \} \). In addition, let \( f(n) \) be a monotonically increasing function such that

(i) \( f(n) \) is polynomially honest; that is, for all \( n \) the complexity of computing \( f(n) \) is polynomially related to the length of \( f(n) \),

(ii) \( f(n) \) bounds the summation of the run times of all programs \( M_i, i < n \), on all inputs of lengths less than or equal to \( f(n - 1) \), plus a little additional time to cover the overhead of the simulation.

We will divide the strings in \( \{0, 1\}^* \) into intervals \( I_n = (\frac{1}{f(n-1)}, \frac{1}{f(n)}) \) using the lexicographic ordering of the strings. At stage \( n \) in the construction all strings in \( I_n \) will be placed into either \( A \) or \( \overline{A} \). The stages of the construction will perform a diagonalization to ensure that if \( L(M_i) \) is infinite, then it is not a subset of \( A \) or of \( \overline{A} \).

**Stage 0.** Assume that \( I_0 = \{0, 1\} \), let \( I_0 \subseteq A \) and place \( M_0 \) on the active lists for \( A \) and for \( \overline{A} \).

**Stage \( n \).**
1. Place \( M_n \) onto the active lists for \( A \) and for \( \overline{A} \).
2. Run all programs on the active lists on all inputs in the interval \( I_n \). Let \( n_0 \) be the smallest index of an active program such that \( M_{n_0}(z) = 1 \) for some \( z \in I_n \), if such a program exists. If \( M_{n_0} \) is on the active list for \( A \), then place \( I_n \) into \( \overline{A} \), ensuring that \( L(M_{n_0}) \not\subseteq A \), and remove \( M_{n_0} \) from \( A \)'s active list. Otherwise, place \( I_n \) into \( \overline{A} \), ensuring that \( L(M_{n_0}) \not\subseteq \overline{A} \), and remove \( M_{n_0} \) from \( \overline{A} \)'s active list. If no program \( M_{n_0} \) exists, place \( I_n \) into \( A \).

**Bi-immunity.** By induction, if \( L(M_n) \) is infinite, then there is a pair of stages \( (n_1, n_2) \) such that \( L(M_n) \) contains elements in the intervals \( I_{n_1} \) and \( I_{n_2} \) and at these stages it is the smallest active program to contain elements in the intervals. During the first of these stages we will have ensured that \( L(M_n) \) is not a subset of \( A \) by placing \( I_{n_1} \) into \( \overline{A} \) and, during the second, we will have similarly ensured that \( L(M_n) \) is not a subset of \( \overline{A} \) by placing \( I_{n_2} \) into \( A \).

**P-cheatability.** We must give a polynomial time oracle algorithm that, when given inputs \( (z_1, \ldots, z_k) \), \( k \) a variable, decides membership in \( A \) for each of the inputs and makes at most two queries to \( A \).

Assume that the inputs are sorted lexicographically and that \( z_k \in I_n \). Consider the positions of the inputs in the intervals used to construct \( A \). Since \( f \) is polynomially honest, in polynomial time we can determine the interval in which each \( z_i \) is contained.

Note that \( z_k \) is large enough that our entire diagonalization construction up through interval \( I_{n-2} \) can be recomputed in time polynomial in \( |z_k| \). (This is because \( |z_k| \geq f(n - 1) \) and \( f(n - 1) \) was explicitly defined so that all membership questions about all elements less than \( f(n - 2) \) for all members of \( \text{Dtime}(T(m)) \) could be decided in time \( f(n - 1) \).) Since the intervals are each entirely contained within \( A \) or \( \overline{A} \), to decide membership for all the \( z_i \)'s in the intervals \( I_0, \ldots, I_{n-2} \), we
simply repeat the construction. To decide membership in the intervals $I_{n-1}$ and $I_n$ requires only that we query $A$ on the two elements $1^{f(n)}$ and the next smaller element, $1^{f(n)-1}0$. This shows $k$ for $2-p$-cheatability for all $k$.

$P/poly$. Since the set $A$ is reducible to the sparse set $A \cap \bigcup_n \{1^{f(n)}, 1^{f(n)-1}0\}$, $A$ is in $P/poly$.

Non-$p$-closeness. The proof that $A$ can be kept from being close to any set in $\text{Dtime}(T(m))$ proceeds much as the corresponding part of the proof of Theorem 1. We assume that each machine $M_n$ occurs with infinitely many indices in our list of machines for $\text{Dtime}(T(m))$, and we require not only that, if infinite, the language accepted by $M_n$ should intersect each of $A$ and $\overline{A}$, but also that, for at least one interval $I_p$, if over half the elements of $I_p$ are accepted by $M_n$ then $I_p \subseteq \overline{A}$, while if at least half the elements of $I_p$ are not in the language accepted by $M_n$, then $I_p \subseteq A$.

Just as in the proof of Theorem 1, for each $M_n$ this new requirement can clearly be blended with the requirements for bi-immunity given above. It is also clear that satisfying these requirements will not affect the proof that the resulting set, $A$, is in $P/poly$.

In contrast to what is shown for $(2^k-1$ for $k)$ and for $(k$ for $2) - p$-cheatability in Theorems 1 and 2, in [A & G-87] it is conjectured that if a set is $(k$ for $1) - p$-cheatable, then it must be $p$-close. The proof of Theorem 2 is easily modified to show that this conjecture is false.

**Theorem 3.** Let $\text{Dtime}(T(m))$ be any deterministic time class. There is a $1-p$-cheatable set $A$ that is not close to any set in $\text{Dtime}(T(m))$. Furthermore, $A \in P/poly$.

**Proof.** The machines $M_i$ and the languages $L(M_i)$ are defined exactly as in the proof of Theorem 2, and the intervals $I_n$ are also constructed exactly as in that proof. For each $n$, the interval $I_{2n}$ is placed into $\overline{A}$. For each $n$, the interval $I_{2n+1}$ is placed into $\overline{A}$ if at least half the elements of $I_{2n+1}$ are in $L(M_n)$, while $I_{2n+1}$ is placed into $A$ if less than half the elements of $I_{2n+1}$ are in $L(M_n)$. This construction clearly keeps the set $A$ from being close to any set in $\text{Dtime}(T(m))$.

Just as in proof of Theorem 2, if $x_1, x_2, ..., x_k$ are any $k$ elements sorted in increasing order and $x_k \in I_n$, then membership in $I_0, I_1, ..., I_{n-2}$ can be tested in time polynomial in $|x_k|$ simply by running the entire construction over these intervals. Because all even indexed intervals are known to be in $A$, a single query to either $1^{f(n)}$ or to $1^{f(n-1)}$, depending on whether $n$ is even or odd, will determine membership for both intervals $I_{n-1}$ and $I_n$. This shows, first, that $A$ is $(k$ for $1) - p$-cheatable for all $k$ and, second, that $A$ is in $P/poly$ since this reduction reduces $A$ to the sparse set $A \cap \bigcup_n \{1^{f(n)}\}$.

Since, as remarked earlier, $1-p$-cheatable sets can easily be shown to never be polynomially bi-immune [Be-87], Theorem 3 seems to be a very strong result for witnessing non-polynomial time behavior in $1-p$-cheatable sets.

Finally, for completeness, we remark that the remaining conjecture in
[A & G-87], namely that any \((2^k \text{ for } k) \text{- } p\)-cheatable set that is not in \(P\) is not close to a polynomially decidable set, is easily shown to be false by techniques related to, but simpler than, those employed in this paper: the techniques used in the proofs of Theorems 2 and 3 can also be used to construct very sparse sets (ones that contain only isolated elements spread very far apart) that are \((k \text{ for } 1) \text{- } p\)-cheatable. But, in fact, since one can keep a set from being in \(P\) by diagonalizing directly on a very sparse set of elements that are determined a priori, it is easy to construct such very sparse sets that are not in \(P\). By definition, all polynomially sparse sets are \(p\)-close, so the most obvious diagonalizations which construct very sparse sets not in \(P\) always construct \(p\)-close sets that are not in \(P\) but that are \((k \text{ for } 1) \text{- } p\)-cheatable.

The fact that all \(p\)-cheatable sets constructed in this paper fall naturally into \(P/poly\) suggests that there may be an interesting relationship between \(p\)-cheatability and membership in \(P/poly\).

REFERENCES


