Kolmogorov Complexity and Degrees of Tally Sets

Eric Allender
Department of Computer Science
Rutgers University
New Brunswick, NJ 08903

Osamu Watanabe
Department of Mathematics
University of California
Santa Barbara, CA 93106

ABSTRACT: We show that either

\[ E^{m}(TALLY) = E^{s}_{m}(TALLY) \]

or

\[ E^{s}_{k}(TALLY) \subseteq E^{s}_{k-1}(TALLY) \subseteq \ldots \subseteq E^{s}_{1}(TALLY) \]

where \( E^{s}_{k}(TALLY) \) denotes the class of sets which are equivalent to a tally set under \( \leq^{s}_{k} \) reductions. Furthermore, the question of whether or not \( E^{m}(TALLY) = E^{s}_{m}(TALLY) \) is equivalent to the question of whether or not \( NE \) predicates can be solved in deterministic exponential time. The proofs use the techniques of generalized Kolmogorov complexity. As corollaries to some of the main results, we obtain new results about the Kolmogorov complexity of sets in \( P \).

1 Introduction

A recent paper by Tang and Book [TB-88] initiated a study of the classes of sets which are equivalent to tally sets or sparse sets, under varying notions of reducibility. A number of interesting results are proved in [TB-88], and many additional questions are posed and left open. This paper investigates some of these questions and shows that they are equivalent to each other, and shows that they are also closely related to other important open questions in complexity theory.

A tally set is a subset of \( 0^{\ast} \). Tally sets, and other sparse sets, have been the focus of a great deal of attention over the last several years. For example, a basic result of circuit complexity is that a set has polynomial-size circuits if and only if it is efficiently reducible (via Turing or truth-table reductions) to a tally set (see, e.g., [Sc-85]).

Restricted forms of truth-table reductions have also been studied frequently in recent years [CH-86, KLD-86, KMR-86]. For any constant \( k \), the class of \( \leq^{k}_{\text{tt}} \) reductions consists of truth-table reductions such that for all inputs, at most \( k \) queries are made to the oracle. A bounded truth-table reduction \( (\leq^{k}_{\text{tt}}) \) is a truth-table reduction which is a \( \leq^{k}_{m} \) reduction for some \( k \).

It was shown in [BK-87] that, for all \( k \), the class of sets \( \leq^{k+1}_{\text{tt}} \)-reducible to sparse sets properly includes the class of \( \leq^{k}_{\text{tt}} \)-reducible to sparse sets, but that, in contrast, every set which is \( \leq^{k}_{m} \) reducible to a tally set is already \( \leq^{k}_{m} \) reducible to a tally set. In this paper, as in [TB-88], we consider the class of sets which are equivalent to tally sets, rather than the class of sets which are reducible to tally sets.

Let \( r \) denote a class of reductions, such as polynomial-time many-one reductions, or truth-table reductions, etc. (See, e.g., [LLS-75] for a discussion of these notions of reducibility.) Let \( E^{s}_{r}(TALLY) \) denote the class of all sets \( L \) such that, for some set \( T \subseteq 0^{\ast} \), \( T \leq^{r}_{s} L \) and \( L \leq^{s}_{r} T \). The class \( E^{s}_{r}(\text{SPARSE}) \) is defined similarly to be the class of sets which are equivalent under \( \leq^{r}_{s} \) reductions to some sparse set.

Motivation for studying classes of the form \( E^{s}_{r}(TALLY) \) comes from two recent results from [BB-86] and [AR-87]:

1. \( L \) has self-producible circuits iff \( L \in E^{s}_{r}(TALLY) \). [BB-86]
2. \( K[\log, \text{poly}] = E^{s}_{m}(TALLY) \). [AR-87]
(The class of sets with self-producible circuits was studied in [HH-86] and [BB-86]. $K[\log, \text{poly}]$ was defined in [BB-86] to be $\{L : \exists k L \subseteq K[\log n, n^k]\}$. $A \leq_p \text{re}$ reduction is a P-isomorphism; for definitions, see [BH-77].)

In [TB-88], it was shown that

$$E^p_{\text{spar}}(\text{SPARSE}) \subseteq E^p_{\text{re}}(\text{SPARSE})$$

and for all $k$,

$$E^p_{\text{re}}(\text{SPARSE}) \subseteq E^{p+1}_{\text{re}}(\text{SPARSE}).$$

However, the following questions were left open:

1. Is $E^p_n(\text{TALLY}) = E^{p+1}_{\text{re}}(\text{TALLY})$?
2. Is $E^p_n(\text{TALLY}) = E^p_{\text{re}}(\text{TALLY})$?
3. Does $E^p_{\text{re}}(\text{TALLY}) = E^{p+1}_{\text{re}}(\text{TALLY})$ for some $k$?

In this note, it is shown that these questions are all equivalent. In fact, all of these questions are equivalent to the following statement: every honest function $f : \Sigma^* \rightarrow 0^*$ computable in polynomial time is weakly invertible (i.e., there is a function $g$ computable in polynomial time such that $f(g(x)) = x$ for all $x \in \text{image}(f)$). We will call this condition Q throughout the rest of the paper.

Condition Q is closely related to the condition "$E = \text{NE}$" and the condition "every infinite set in P has an infinite P-printable subset". In particular, condition Q is true iff for every nondeterministic Turing machine $M$ which runs in exponential time, there is a deterministic exponential-time algorithm which, for all inputs $x$, will find an accepting computation of $M$ on $x$ if one exists. Also, Q is true iff every polynomial-time machine must accept a Kolmogorov-simple string of length $n$ if it accepts any string of length $n$.

Similar observations allow us to conclude that if all infinite sets in P have infinite P-printable subsets, then no set in NE is immune with respect to E.

There is also a close connection between condition Q and a conjecture by Sewelson [Se-83] that E=NE implies E=\text{NP}. If Sewelson’s conjecture is true, then condition Q and questions about $\leq_{\text{re}}^p$ degrees of tally sets are equivalent to $E = \text{NE}$. These connections are discussed in section 3.

2 Preliminaries

It is expected that the reader will be familiar with basic concepts from complexity theory, such as Turing machines, circuits, and complexity classes such as P, NP, etc. For background and definitions, see e.g. [HU-79, Se-85]. We will use E and NE to refer to DTIME($2^{O(n)}$) and NTIME($2^{O(n)}$), respectively. $E^{\text{NP}}$ denotes the class of languages accepted by deterministic exponential-time oracle Turing machines with an oracle from NP.

For any string $x$, the length of $x$ is denoted by $|x|$. For any set $S$, $|S|$ denotes the cardinality of $S$. All languages considered in this paper are subsets of $\{0,1\}^*$. We will use a one-one pairing function mapping $\{0,1\}^* \times \{0,1\}^*$ onto $\{0,1\}^*$, and for inputs $x$ and $y$ in $\{0,1\}^*$, we will denote the output of the pairing function by $(x,y)$. We will also assume a standard mapping from $\{0,1\}^*$ onto the positive integers; namely the string $x$ will denote the integer whose binary representation is $1x$. Thus, for example, $|x| = \lceil \log |x| \rceil$, and given strings $x$ and $y$, we may say $x \leq y$ (which corresponds to the lexicographic ordering on $\{0,1\}^*$).

We say that $A \leq^p_\text{re} B$ if there is a function $f$ computable in polynomial time, such that for all $x$, $x \in A \iff f(x) \in B$. We say that $A \leq_{\text{re}}^p B$ if there is a function $f$ computable in polynomial time, such that, for all $x$, $f(x)$ is of the form $(f(x)[1], f(x)[2], \ldots, f(x)[k], a(x))$ where $a(x)$ is a string of $2^k$ bits specifying a function from $\{0,1\}^k$ to $\{0,1\}^k$, and $x \in A \iff a(x)[y_1, \ldots, y_k] = 1$, where $y_i = 1 \iff f(x)[i] \in B$.

(Intuitively, $A \leq^p_\text{re} B$ if there is a polynomial-time routine which can accept $A$, given that on each input it is allowed to formulate $k$ questions to ask oracle $B$.) More formal definitions are given in [LLS-75].

We use the notation $f : A \leq p B$ to say that $f$ is a reduction of type $r$ from $A$ to $B$.

$A \leq_r^p$ reduction $f$ is honest if there is some polynomial $p$ such that, for $1 \leq i \leq k, |x| \leq p(|f(x)[i]|)$. We call such a reduction a $\leq_r^p$ reduction. Similarly, honest many-one reductions are $\leq_{\text{re}}^p$ reductions. Honest reductions have been considered before in, e.g. [Ho-87, JY-85].

For technical reasons, we will consider a $\leq^p_\text{re}$ reduction which reduces a tally set $T$ to another set $A$ to be honest if there is some $c$ such that, for all $n$, $|f(n^{c})| > n$. That is, in order to be considered honest, such a reduction $f$ need only be honest on $0^*$. Generalized Kolmogorov complexity provides a framework for talking about the complexity of individual strings.
The definitions we used were introduced in [Ha-83]: Given any Turing machine \( M_n \), we define \( K_n(s(n), t(n)) \) to be the set of all strings \( z \) such that, for some string \( y \) of length at most \( s(|x|) \), \( M_n \) prints out \( z \) on input \( y \) in at most \( t(|x|) \) steps. As was shown in [Ha-83], there is a machine \( M_n \) (a universal Turing machine) such that, for all \( v \) there exists a constant \( c \) such that \( K_n(s(n), t(n)) \subseteq K_n(s(n) + c, c(n)\log t(n) + c) \). Dropping the subscript, we will choose some particular universal Turing machine \( M_n \) and we will let \( K(s(n), t(n)) \) denote \( K_n(s(n), t(n)) \).

Let \( L \) be any subset of \( \Sigma^* \). The ranking function for \( L \), denoted \( r_L \), is defined as follows: \( r_L(x) = \{y \in x \in L\} \).\n
A set \( S \) is said to be \( P \)-printable if there is an algorithm which, on input \( n \), will run in time polynomial in \( n \), and will print out the elements of \( S \) which have length at most \( n \). \( P \)-printable sets were defined in [HY-84] and were studied further in [AR-87], in connection with Kolmogorov complexity. It was shown in [AR-87] that a set is a subset of \( K[\log n, n^4] \) for some \( t \) if it is \( P \)-isomorphic to a tally set, and that a set is \( P \)-printable if it is \( P \)-isomorphic to a tally set in \( P \). One of the lemmas in this paper makes use of a more specific fact about \( K[\log n, n^4] \); we prove that fact here.

**Theorem 1** For all \( t \geq 2 \), \( K[\log n, n^4] \) is \( P \)-isomorphic to \( 0^* \).

**Proof:** Note first that there is some Turing machine \( M_n \) such that \( 0^* = K_n[\log n, 2n] \). Thus, \( 0^* \subseteq K[2\log n, n^4] \). Thus for all \( t \geq 2 \), there are at least \( n \) elements of length at most \( n \) in \( K[\log n, n^4] \). Thus it suffices to show that, if \( S \) is any \( P \)-printable set such that, for some \( c \), \( \{r_{S}(1^n)\} > n \), then \( S \) is \( P \)-isomorphic to \( 0^* \).

Let \( S \) be any such set. Let \( T = \overline{S} \) and let \( L = \Sigma^* - 0^* \). As was pointed out in [GS-85], \( r_S, r_T \), and \( r_\emptyset \) are all computable in polynomial time, and they all have inverses computable in time polynomial in the length of their output. Let \( f \) be defined as follows: \( f(x) = 0^n(v(x)) \) if \( x \in S \), and \( f(x) = r_T^{-1}(r_S(x)) \) if \( x \notin S \). Some straightforward calculations verify that \( f \) is a \( P \)-isomorphism mapping \( S \) onto \( 0^* \).

### 3 The Condition \( Q \)

The condition \( Q \) is a statement about the nonexistence of a certain kind of one-way function; it states that every honest function \( f : \Sigma^* \rightarrow 0^* \) computable in polynomial time is weakly invertible (i.e., there is a function \( g \) computable in polynomial time such that \( f(g(z)) = z \) for all \( z \in \text{image}(f) \)). In this section we relate the condition \( Q \) to other questions in complexity theory which have received more attention. In particular, we will relate the condition \( Q \) to the \( E=NE \) question.

The notion of NP-completeness has been extremely useful in characterizing the complexity of many optimization problems. One key to this success is the fact that the recognition problem for NP-complete sets is equivalent to the problem of constructing a solution to an instance of the corresponding optimization problem.

Somewhat surprisingly, it is not known if the same situation holds for nondeterministic exponential time. The following definitions and results help to make this precise.

Let \( M \) be a nondeterministic Turing machine such that every configuration of \( M \) has at most two possible successor states. An accepting computation for a string \( x \) on \( M \) (or a witness for \( x \) on \( M \)) is a binary string encoding the sequence of nondeterministic moves of \( M \) on input \( x \) along some computation path leading to an accepting configuration. That is, if \( x \) is a string encoding a sequence of moves of \( M \), then the \( i \)-th bit of \( x \) will be \( 1 \) (0) if the \( i \)-th move in the sequence which involves a nondeterministic choice is resolved in favor of the higher (lower) numbered state.

**Definition:** An \( NE \) predicate is a binary predicate \( R \) such that, for some nondeterministic Turing machine \( M \) which runs in time \( 2^m \) for some \( c \), \( R(x,y) \iff y \) is an accepting computation for \( x \) on \( M \). An \( NE \) predicate \( R \) is \( E \)-solvable if there is some function \( f \) computable in time \( 2^m \) for some \( c \) such that, for all \( x \exists y R(x,y) \iff R(x,f(x)) \).

(Note that, typically, the complexity of deciding whether or not \( R(x,y) \) is true is linear in the length of \( x, y \).)

That is, \( R \) is \( E \)-solvable if there is an exponential-time routine which, for all \( x \), can find a witness for \( x \) if one exists. One can in a similar way consider "NP predicates" and "P-solvability"; it is well-known that \( P=NP \) iff every
NP predicate is $P$-solvable.

It follows from Theorem 4 (proved later in this section) that NE predicates are closely related to the condition $Q$:

$Q$ if and only if Every NE predicate is $E$-solvable.

It is natural to wonder if $Q$ is equivalent to $E=NE$. This question is closely related to Sewelson's conjecture, as the following proposition shows:

**Proposition 2** $E = E^{NP} \implies Q \implies E = NE$.

**Proof:** It is immediate from the preceding discussion that $Q$ implies $E=NE$. To see $E = E^{NP} \implies Q$, it suffices to show that $E = E^{NP}$ implies every NE predicate is $E$-solvable. Let $R$ be an NE predicate defined by a nondeterministic Turing machine running in time $2^n$. Let $L = \{(0^i, w) : \text{for some } x \text{ of length at most } i', R(i, wx)\}$. $L$ is in NP. An exponential-time machine with an oracle for $L$ can recognize the set $L' = \{(i, j) : \text{the j-th bit of the lexicographically least witness for } i \text{ is 1}\}$. By assumption, $L'$ is in $E$. It now follows easily that $R$ is $E$-solvable.

Thus if Sewelson’s conjecture is true, then all of these conditions are equivalent, and the results about $\leq_{tt}$ degrees of tally sets which are proved in this paper turn out to all be equivalent to $E = NE$.

There is also a close relationship between $Q$ and the question of whether or not every infinite set in $P$ has an infinite $P$-printable subset. This latter question has been considered recently in [Al-87, AR-87]. In order to formulate this relationship, it is necessary to introduce certain notions of immunity.

A set $S$ is said to be immune to a class of sets $C$ if $S \cap L$ is finite for all $L \in C$. (Immunity has been widely studied in complexity theory and recursive function theory, e.g. [Ro-67, BS-85].) Now we wish to extend the notion of immunity to cover predicates as well.

**Definition:** An NE predicate $R$ is said to be $E$-immune if (1) the set $\{x : \exists y R(x, y)\}$ is infinite, and (2) for all $f$ computable in exponential time, the set $\{x : R(x, f(x))\}$ is finite.

It is also necessary to define a function $\text{len} : 2^\Sigma^* \rightarrow 2^N$ such that, for any language $L$, $\text{len}(L) = \{n : \text{there is a string of length } n \text{ in } L\}$.

**Proposition 3**

1. For every infinite set $L \in P$ there is an infinite $P$-printable $S \subseteq L \iff$ No NE predicate is $E$-immune.
2. For every infinite set $L \in P$ there is an infinite $P$-printable $S \subseteq L$ such that $\text{len}(S) = \text{len}(L) \iff Q$ ($\iff$ no NE predicate is $E$-solvable).

**Proof:** Part 2 is proved as part of Theorem 4. The proof of part 1 is very similar to that of part 2.

It seems unlikely that $Q$ is true, since it follows from $Q$ that $E = NE$. Thus the preceding proposition tells us that there are probably sets in $P$ which, for some lengths, contain only “complex” strings.

Note that if $P = NP$, then $E = E^{NP}$ and hence $Q$ is true, and thus every set in $P$ has “simple” strings of every length. This is somewhat surprising, since $P = NP$ implies that sets such as $K[n/2, n^2]$ are in $P$. $K[n/2, n^2]$, in some sense, contains only complex strings; yet if $P = NP$ it also contains infinitely many “simple” strings, i.e., strings in $K[t \log n, n^t]$ for some $t$.

In the proofs of the main results of this paper, we actually will make use of different statements which are equivalent to $Q$. The following result deals with all of the equivalent statements of $Q$ of which we make use in this paper.

**Theorem 4** The following are equivalent:

1. Every NE predicate is $E$-solvable.
2. $Q$
3. For all honest $f$ such that for some $t$, $\text{image}(f) \subseteq K[t \log n, n^t]$, there is some $r$ such that for all $x \in \text{image}(f)$, $f^{-1}(x) \cap K[t \log n, n^t] \neq \emptyset$.
4. For all length-increasing $f : \Sigma^* \rightarrow 0^*$ computable in polynomial time, there exists a $t$ such that for all $x \in \text{image}(f)$, $f^{-1}(x) \cap K[t \log n, n^t] \neq \emptyset$.
5. $\forall L \in P \exists S \subseteq L$ such that $S$ is $P$-printable and $\text{len}(S) = \text{len}(L)$.

**Proof:** (1 $\implies$ 2): Let $f : \Sigma^* \rightarrow 0^*$ be an honest function computable in polynomial time. Since $f$ is assumed to be honest, there is a constant $d$ such that $\forall x \in \Sigma^*$
0^i \implies \exists z (|z| < i^4 \text{ and } f(z) = 0^i)$. Let $R$ be the NE predicate $R(i, z) \iff f(x) = 0^i$ and $|z| < i^4$. Assuming that every NE predicate is E-solvable, it follows that there is a function $h$ computable in time $2^m$ for some $c$ such that \( \exists y R(i, y) \iff R(i, h(i)) \). Thus $f$ can be inverted on $0^i$ in time $n^i$ using the following algorithm: on input $0^i$, compute $h(i)$.

$$
(2 \implies 3): \text{Let } f \text{ be given, } f : \Sigma^* \to K[\log n, n^i]. \text{ By \cite{AR87}, there is some P-isomorphism } \gamma \text{ mapping the image of } f \text{ onto some tally set. Since } \gamma \circ f : \Sigma^* \to 0^* \text{ is honest, by assumption, there is some function } h \text{ computable in polynomial time such that } \gamma(\gamma(f(z))) = z \text{ for all } z \text{ in the image of } \gamma \circ f. \text{ That is, for all } z \in \text{image}(f), h(\gamma(z)) \in f^{-1}(z), \text{ and for all such } z, g(z) \in 0^*, \text{ and thus there is some } r \text{ such that, for all } z \in \text{image}(f), h(\gamma(z)) \in f^{-1}(z) \cap K[r \log n, n^i].$$

$$
(3 \implies 4): \text{immediate.}
$$

$$
(4 \implies 5): \text{Let } L \in P. \text{ Let } f(z) = 0^{|z|+1} \text{ if } z \in L, \text{ and } f(z) = 0^{|z|} \text{ if } z \notin L. \text{ By assumption, there exists a } t \text{ such that for all } z \in \text{image}(f), f^{-1}(z) \cap K[\log n, n^i] \neq \emptyset. \text{ Let } L = \{z \in K[\log n, n^i] : |f(z)| \text{ is odd}\}. \text{ Then } L \subseteq L, \text{ S is P-printable, and } \text{len}(S) = \text{len}(L).$$

$$
(4 \implies 1): \text{Let } R \text{ be an NE predicate defined by a non-deterministic Turing machine which runs in time } 2^m. \text{ Let } L = \{0^i1x10^j : |x| \leq i^2 \text{ and } R(i, x)\}. \text{ L is in P. Let } S \text{ be a P-printable subset of } L \text{ such that } \text{len}(S) = \text{len}(L). \text{ Then the following routine can be executed in exponential time, and on input } i \text{ it will return a string } z \text{ such that } R(i, z) \text{ if any such } z \text{ exists: On input } i, \text{ print the elements of } S \text{ of size at most } i^2 + 2 + i. \text{ If any element in the list has length } i^2 + 2 + i, \text{ it is of the form } 0^i1x10^j \text{ for some } x \text{ and } j \text{ such that } R(i, x). \text{ Output } z.$$

4 Some Basic Lemmas

In this section, we present characterizations of sets which are equivalent to tally sets under honest reductions. These characterizations play a central role in the proofs of the main results. Each characterization has essentially the same flavor: namely, if a set is in the same degree as a tally set, then the set is reducible to itself, via reductions which query only strings of low Kolmogorov complexity.

Lemma 5

1. $A \in E^A_{\log}(\text{TALLY}) \iff \exists A \leq_{\log^A} A \cap K[\log n, n^i]$.
2. $A \in E^A_{\log}(\text{TALLY}) \iff \exists A \leq_{\log^A} A \cap K[\log n, n^i]$.
3. $A \in E^A_{\log}(\text{TALLY}) \iff \exists A \leq_{\log^A} A \cap K[\log n, n^i]$.
4. $A \in E^A_{\log}(\text{TALLY}) \iff \exists A \leq_{\log^A} A \cap K[\log n, n^i]$.
5. $A \in E^A_{\log}(\text{TALLY}) \iff \exists A \leq_{\log^A} A \cap K[\log n, n^i]$.

Proofs: We will prove (4) and (5); these clearly imply (2) and (3), and (1) can be proved in a similar way.

4. $A \leq_{\log^A} A \cap K[\log n, n^i]$. Let $g$ be a P-isomorphism between $K[\log n, n^i]$ and $0^i$. Such an isomorphism exists, by Theorem 1. Let $T$ be $g(A \cap K[\log n, n^i])$. Clearly, $A \leq_{\log^A} T$. Also, choosing some fixed $z \notin L$, we have that the function

$$
\gamma = \begin{cases} 
g^{-1}(z) & \text{if } z \in 0^i \\
z & \text{otherwise} \end{cases}
$$

is a $\leq_{\log^A}$ reduction from $T$ to $A$. (Recall that this reduction needs to be honest only on $0^i$.) Thus $A \in E^A_{\log}(\text{TALLY}).$

5. $A \in E^A_{\log}(\text{TALLY})$. Thus there exists some tally set $T$ and some reductions $f$ and $g$ such that $f : A \leq_{\log^A} T$ and $g : T \leq_{\log^A} A$. Note that, since $g$ is honest and runs in polynomial time, and takes input from $0^i$, all queries made by $g$ are in $K[\log n, n^i]$ for some $t$. Thus $g \circ f$ is a $\leq_{\log^A}$ reduction from $A$ to $A \cap K[\log n, n^i]$. It is natural to wonder if part (5) of the preceding lemma can be improved, by replacing the "$\leq_{\log^A}$" with an "$\leq_{\log}$". We conjecture that no such improvement is possible. In fact, we are able to prove that no such improvement is possible for a restricted class of truth-table reductions, which we shall call good reductions. The proofs of our main results make frequent use of the properties of good reductions.

Definition: A $\leq_{\log^A}$ reduction $g$ will be called good if $g(x) = (x, i)$ for all $x \in K[3 \log n, n^i]$, where $i$ is the identity function, and $g(x) = (g(x)[1], g(x)[2], \ldots, g(x)[k], \Theta)$ for all $x \notin K[3 \log n, n^i]$, where $|g(x)[i]| > |x|$ and $g(x)[i] \in K[3 \log n, n^i]$ for all $i, 1 \leq i \leq k$, and $\Theta$ is taken to be the function that takes the value 1 if an even number of the strings in the set $\{g(x)[i] : 1 \leq i \leq k\}$ are in the oracle set. (There is no special significance to the number 3 in this definition. It is sufficient to choose any $t$ such that $0^i+t \leq K[\log n, n^i]$.)
The following lemma should be compared to parts (4) and (5) of the preceding lemma.

Lemma 6 If $A \leq^P_{k-1} A \cap K[3 \log n, n^2]$ via a good reduction, then $A \in E\phi^k_{k-1}(TALLY)$.

Proof: Let $g: A \leq^P_{k-1} A \cap K[3 \log n, n^2]$, where $g$ is a good $\leq^P_{k-1}$ reduction computable in time $q(n)$ for some polynomial $q$.

Let $T = \{ (x_1, x_2, \ldots, x_k) : \text{for all } i, x_i \in K[3 \log n, n^2] \text{ and } g(x_i) \geq (x_1, x_2, \ldots, x_k), \text{ and the set } \{ x_i : x_i \in A \} \text{ has an odd number of elements} \}$. Clearly $T$ is $E^h_{k-1}$ reducible to $A$. We need to show that $A \leq^P_{k-1} T$ reducible to $T$.

Let $h$ be the $\leq^P_{k-1}$ reduction given by

$$h(x) = \begin{cases} \langle (x, \ldots, x), i \rangle & \text{if } g(x) = (x, i) \\ \langle x_1, \ldots, x_k, \ldots, (x_{k+1}), \ldots, x_{2k} \rangle, (i) & \text{if } g(x) = (x_1, \ldots, x_k, \ldots, x_{2k}). \end{cases}$$

It is easy to verify that $h: A \leq^P_{k-1} T$. Also, since $T$ consists only of strings of low Kolmogorov complexity, it follows from [AR-87] that $T$ is P-isomorphic to a tally set. Thus $A \in E\phi^k_{k-1}(TALLY)$.

The lemmas presented so far in this section have dealt with honest reductions and degrees of unrestricted reductions.

Lemma 7 For any class of reduction $\leq k \in \{ \leq^P_{k-1}, \leq^P_{k-2}, \ldots \}$ and any set $A$, $A \in E\phi^k(TALLY) \iff A \times 0^* \in E\phi^k(TALLY)$.

Proof: We prove the result in the case of many-one reductions. The proof in the case of $\leq^P_{k-1}$ reductions is similar.

$(\Rightarrow)$: Let $A \in E\phi^k(TALLY)$. That is, there is a tally set $T$ such that $A \leq^P_{k-1} T$. Assume without loss of generality that the pairing function is such that $(1, y, 0) \geq y$, so that $T \times 0^*$ is a tally set.

Let $f'((x, y)) = f((x, 0), 0((x, y)))$ if $y \in 0^*$, and $f'((x, y)) = (1, 1)((x, y))$ if $y \notin 0^*$. Let $g'((x, y)) = g((x, 0), 0((x, y)))$ if $y \in 0^*$, and $g'((x, y)) = (1, 1)((x, y))$ if $y \notin 0^*$. It is easy to verify that $f': A \times 0^* \leq^P_{k-1} T \times 0^*$, and $g': T \times 0^* \leq^P_{k-1} A \times 0^*$.

$(\Leftarrow)$: Assume $A \times 0^* \in E\phi^k(TALLY)$. That is, there is a tally set $T$ and there are $\leq^P_{k-1}$ reductions $f: A \times 0^* \leq^P_{k-1} T$, and $g: T \leq^P_{k-1} A \times 0^*$. Let $f'((x, y)) = f((x, 0))$, and define $g'((x, y))$ to be equal to $y$ if $g((x, y)) = (x, 0)$ for some $j$, and $g'((x, y)) = g((x, 0))$, for some fixed string $s \notin A$, if $g((x, y)) \notin S \times 0^*$. It is easy to verify that $f': A \leq^P_{k-1} T$, and $g': T \leq^P_{k-1} A$. 

Corollary 8 If, for all $A$,

$$\exists A \leq^P_{k-1} A \cap K[3 \log n, n^2] \implies \exists A \leq^P_{k-1} A \cap K[3 \log n, n^2],$$

then $E\phi^k(TALLY) = E\phi^k_0(TALLY)$.

Proof: Let $A \in E\phi^k(TALLY)$.

By Lemma 7, $A \times 0^* \in E\phi^k_0(TALLY)$ for some $k$. By Lemma 5, $A \times 0^* \leq^P_{k-1} A \times 0^* \cap K[\log n, n^2]$ for some $k$. By assumption $A \times 0^* \leq^P_{k-1} A \times 0^* \cap K[\log n, n^2]$ for some $k$. By Lemma 5, $A \times 0^* \in E\phi^k_0(TALLY)$, and by Lemma 7, $A \in E\phi^k_0(TALLY)$.

Corollary 9 If $\exists A$ such that $A \leq^P_{k-1} A \cap K[3 \log n, n^2]$ and

$$\forall A \times 0^* \in E\phi^k_0(A \times 0^* \cap K[\log n, n^2])$$

then $E\phi^k(TALLY) \subset E\phi^k_0(TALLY)$.

Proof: Let $A$ be such that $A \leq^P_{k-1} A \cap K[3 \log n, n^2]$ and $A \times 0^* \in E\phi^k_0(A \times 0^* \cap K[\log n, n^2])$.

By Lemma 5, $A \in E\phi^k_0(TALLY)$, and $A \times 0^*$ is not in $E\phi^k_0(TALLY)$. By Lemma 7, $A \notin E\phi^k_0(TALLY)$.

Corollary 10 If $\forall A$ such that $A$ is reducible to $A \cap K[3 \log n, n^2]$ via a good $\leq^P_{k-1}$ reduction and

$$\forall A \times 0^* \in E\phi^k_0(A \times 0^* \cap K[\log n, n^2])$$

then for all $k$, $E\phi^k_{k-1}(TALLY) \subset E\phi^k_0(TALLY)$.

Proof: Let $k$ be given. By assumption there is a set $A$ such that $A \leq^P_{k-1} A \cap K[3 \log n, n^2]$ via a good $\leq^P_{k-1}$ reduction, and

$$\forall A \times 0^* \in E\phi^k_0(A \times 0^* \cap K[\log n, n^2])$$

By Lemma 6, $A \in E\phi^k_{k-1}(TALLY)$. By Lemma 5, $A \times 0^* \notin E\phi^k_0(TALLY)$, since $k^2 < (k+1)^2 - 1$. By Lemma 7, $A \notin E\phi^k_0(TALLY)$.

The corollaries suggest how the proofs will be structured. We will show that if $Q$ is true, then the hypothesis of Corollary 8 is satisfied, and we will show that if $Q$ is false, then the hypotheses of Corollaries 9 and 10 are satisfied.
5 Main Results

Theorem 11 \( Q \implies E^A_n(TALLY) = E^A_n(TALLY) \).

Proof: By Corollary 8, it suffices to show that if \( Q \) is true, then \( \exists A \leq^A_m A \cap K[\log n, n'] \implies \exists A \leq^A_m A \cap K[\log n, n'] \). Assume \( Q \) is true, and let \( f: A \leq^A_m A \cap K[\log n, n'] \).

Recall that, for all \( z, f(z) \) is of the form \( (f(z)[1], f(z)[2], \ldots, f(z)[k], \alpha(z)) \) where \( \alpha(z) \) is a string of length \( 2^k \). Without loss of generality, we may assume that there is some \( u \) such that for all \( i, f(z)[i] \in K[u \log n, n^n] \). (We can do this since, if \( f(z)[i] \notin K[\log n, n'] \), the value of \( f(z)[i] \in A \cap K[\log n, n'] \) is 0. We can easily find a string \( z \in K[u \log n, n^n] \) and set \( f(z)[i] \) to \( z \). Using this replacement, the truth value of \( f(z) \) remains unchanged.)

Since all of the \( f(z)[i] \) are of low Kolmogorov complexity, and since \( \alpha(z) \) has bounded length, it follows that for some \( \nu \), the range of \( f \) is contained in \( K[\log n, n^n] \).

Since \( Q \) is assumed to be true, it is thus the case that there exists some \( r \) such that for all large \( z \) there is a \( y \in K[r \log n, n^n] \) such that \( f(z) = f(y) \). The routine that, on input \( z \), searches through \( K[r \log n, n^n] \) until it finds such \( y \) and then outputs \( y \), is a \( \leq^A_m \) reduction from \( A \cap K[\log n, n^n] \) to \( A \cap K[r \log n, n^n] \).

The proof of Theorem 11 is not hard; it is more difficult to prove the separation results.

Each of the separation results involves constructing a set \( A \) with certain properties. As is usual in such constructions, the set \( A \) will be built in "stages". In order to explain how we construct the set \( A \), some discussion is necessary.

Let \( g \) be any good \( \leq^A_m \) reduction, and let \( S \) be any subset of \( K[\log n, n^n] \). Notice that there is a (unique) set \( A \) such that \( A \cap K[\log n, n^n] = S \) and \( A \leq^A_m A \cap K[\log n, n^n] \). That is, any subset of \( K[\log n, n^n] \) gives rise to some set which reduces to itself via \( g \), and membership in any such set is entirely determined by the membership of strings of low Kolmogorov complexity. Thus we will build our set \( A \) by specifying membership in \( A \) for certain strings of low Kolmogorov complexity. The next paragraph makes this more precise.

At the start of each stage \( i \), there will be a function \( A_{i-1}: K[\log n, n^n] \to \{0, 1, ?\} \) such that \( A_{i-1}(z) = ? \) for all but finitely many \( z \). During stage \( i \) we will build a function \( A_i \), which is a finite extension of \( A_{i-1} \) (i.e., \( A_i(z) = A_{i-1}(z) \) for all \( z \), except for finitely many \( z \) such that \( A_{i-1}(z) = ? \)).

Let us say that a set \( A \) is consistent with \( A_i \), if \( A \leq^A_m A \cap K[\log n, n^n] \), and \( A_i(z) \neq ? \implies (A_i(z) = 1 \iff z \in A) \). The functions \( A_i \) will be constructed so that there is at least one set \( A \) such that, for all \( i, A \) is consistent with \( A_i \).

Any set which is consistent with all of the functions \( A_i \), will be a witness for the separation result.

During the course of the construction, there will be many strings \( y \in K[\log n, n^n] \) such that, for some stage \( i, A_i(y)[j] \neq ? \) for all \( j, 1 \leq j \leq k \). For any such string \( y \), it follows that membership of \( y \) in \( A \) is determined by \( A_i \), for any set \( A \) which is consistent with \( A_i \). Thus we shall sometimes say that \( A_i \) guarantees that such a string \( y \) is in \( A \) or is not in \( A \).

We are now ready to prove the first separation result.

Theorem 12 If \( Q \) is false, then

\[ E^A_n(TALLY) \subset E^A_{n+1}(TALLY). \]

Proof: As suggested by Corollary 9, the strategy will be to show that, if \( Q \) is false, then \( \exists A \) such that \( A \leq^A_{n+1} A \cap K[\log n, n^n] \) and

\[ \forall A \times 0^* \leq^A_m A \times 0^n \cap K[\log n, n^n]. \]

Recall that, if \( Q \) is false, then there is some length-increasing \( f: \Sigma^* \to \Sigma^* \) computable in polynomial time such that for all \( t \), there exists infinitely many \( x \in \text{image}(f) \) such that \( f^{-1}(x) \cap K[\log n, n^n] = \emptyset \).

The \( \leq^A_{n+1} \) reduction from \( A \) to \( A \cap K[\log n, n^n] \) will be given by the function \( g \) defined by \( g(z) = (f(z), \emptyset) \) if \( z \notin K[\log n, n^n] \) and \( g(z) = (\emptyset, i) \) for all \( z \in K[\log n, n^n] \). Clearly, \( g \) is a good \( \leq^A_{n+1} \) reduction. (Notice that if \( g(z) = (f(z), \emptyset) \), then \( z \in A \iff f(z) \notin A \).)

Let \( f_1, f_2, \ldots \) be an enumeration of \( \leq^A_n \) reductions, and let \( p_i \) be a polynomial bounding the running time of a machine computing \( f_i \). Let \( q \) be a polynomial such that \( f \) is computable in time bounded by \( q \).

We will build \( A \) in stages. At stage \( s = (i,t) \) we will guarantee that \( f_i \) is not a \( \leq^A_n \) reduction of \( A \times 0^* \cap K[\log n, n^n] \).
Initially, set $A_0(x) = ?$ for all $x$.

At stage $s = (i,t)$, choose $r$ so that, for all $y$, if $(y,0^m) \in K(t \log n, n^q)$ and $m \leq p_i(2q(|y|))$, then $y \in K[r \log n, n^q]$. Choose $z$ so that
\[ A_{z-1}(f(z)) = ? \quad \text{and} \quad f^{-1}(f(z)) \cap K[r \log n, n^q] = \emptyset. \]
(This is possible since $A_{z-1}$ is only defined on finitely many strings.)

If either $f_1((x,0)) \not\subseteq K[t \log n, n^q]$ or $f_1((x,0))$ is not of the form $(y,0^m)$, then set $A_t(f(z)) = 0$. This guarantees that $z \in A_t$ and thus $(x,0) \in A \times 0^*$, but $f_t((x,0)) \not\subseteq A \times 0^* \cap K[t \log n, n^q]$.

Consider the other case: namely $f_1((x,0)) \subseteq K[t \log n, n^q]$ and $f_1((x,0)) = (y,0^m)$ for some $y$ and some $m \leq p_i(|(x,0)|)$. There are three cases now, according to whether $A_{z-1}(y) = 0$, $A_{z-1}(y) = 1$, or $A_{z-1}(y) = ?$.

- If $A_{z-1}(y) = 0$, then let $A_t(f(z)) = 0$. This guarantees that $z \in A_t$, and thus $(x,0) \in A \times 0^*$, but $f_t((x,0)) = (y,0^m) \in A \times 0^* \cap K[t \log n, n^q]$.

- If $A_{z-1}(y) = 1$, then let $A_t(f(z)) = 1$. This guarantees that $z \not\in A_t$, and thus $(x,0) \not\in A \times 0^*$, but $f_t((x,0)) = (y,0^m) \in A \times 0^* \cap K[t \log n, n^q]$.

- If $A_{z-1}(y) = ?$, then there are two cases, depending on whether or not $y \in K[3 \log n, n^q]$. If $y \in K[3 \log n, n^q]$, then set $A_t(y) = 1$ and $A_t(f(z)) = 1$. (Note that this is possible even if $f(z) = y$.) This guarantees that that $z \not\in A_t$ and $(x,0) \not\in A \times 0^*$, but $f_t((x,0)) = (y,0^m) \in A \times 0^* \cap K[3 \log n, n^q]$.

If $y \not\in K[3 \log n, n^q]$, then note that $f(y) \neq f(x)$. This is because if $f(y) = f(x)$, then $y \not\in K[r \log n, n^q]$ (by choice of $z$). Also, $g(|y|) \geq f(|x|) \geq |z|$, and thus $m \leq p_i(|(x,0)|) \leq p_i(2q(|y|)) \leq p_i(2q(|y|))$. Thus $(y,0^m) \not\subseteq K[t \log n, n^q]$ (by choice of $r$), contrary to assumption. Thus we may set $A_t(f(z)) = 1$ and $A_t(f(y)) = 0$. This guarantees that $z \not\in A_t$ and $y \in A_t$. Thus $(x,0) \not\in A \times 0^*$, but $f_t((x,0)) = (y,0^m) \in A \times 0^* \cap K[t \log n, n^q]$.

Theorem 13 If $Q$ is false, then for all $k$, $E_{z-1}(TALLY) \subseteq E_{z-1}(TALLY)$.

Proof: By Corollary 10, it suffices to show that if $Q$ is false, then for all $k$, $\exists A$ such that $A \leq E_{z-1}(TALLY)$ via a good $E_{z-1}$ reduction, and $A \times 0^* \not\subseteq A \times 0^* \cap K[t \log n, n^q]$.

Recall that, if $Q$ is false, then there is some length-increasing $f : \Sigma^* \to \Sigma^*$ computable in polynomial time such that for all $t$ there exist infinitely many $z \in \text{image}(f)$ such that $f^{-1}(z) \cap K[t \log n, n^q] = \emptyset$.

The $E_{z-1}$-reduction from $A$ to $A \cap K[3 \log n, n^q]$ will be given by the function $g$ defined by
\[ g(x) = \begin{cases} (f(x), f(x)1, f(x)11, \ldots, f(x)k) & \text{if } x \not\subseteq K[3 \log n, n^q] \\ (x, z) & \text{otherwise} \end{cases} \]

Clearly, $g$ is a good $E_{z-1}$-reduction.

Let $f_1, f_2, \ldots$ be an enumeration of $E_{z-1}$ reductions, and let $p_i$ be a polynomial bounding the running time of a machine computing $f_i$. Let $g$ be a polynomial such that $f$ is computable in time bounded by $g$.

We will build $A$ in stages. At stage $s = (i,t)$ we will guarantee that $f_i(z) \not\subseteq A \cap K[t \log n, n^q]$.

Initially, set $A_0(x) = ?$ for all $x$.

At stage $s = (i,t)$, choose $r$ so that, for all $z$, if $(z,0^m) \in K(t \log n, n^q)$ and $m \leq p_i(|(z,0)|)$, then $z \in K[r \log n, n^q]$. Choose $z$ so that
\[ f_i((y,0^m)) \not\subseteq K[r \log n, n^q] \quad \text{or} \quad A_{z-1}(y) \neq ? \]

Without loss of generality, we can assume that $f_i((y,0^m))$ is of the form $(y_1,0^m), (y_2,0^m), \ldots, (y_k,0^m), a_l$ for some $y_1, y_2, \ldots, y_k, m, a_l \in K[t \log n, n^q]$.

Let $z = \{x_1, x_2, \ldots, x_{k+1}\}$ be a set such that $f_i((z,0^m)) \not\subseteq A \cap K[t \log n, n^q]$. Notice that if we construct $A_z$ in such a way that it determines whether $z \in A$ for each $z \in Z$, then we will have determined the truth value of $f_i(z)$ when the oracle is $A \times 0^* \cap K[t \log n, n^q]$, where $A$ is any set consistent with $A_z$.

Notice that, for all $z \in Z$, $f_i(z) \neq f(z)$. This is because if $f(z) = f_i(z)$, then $x \not\subseteq K[r \log n, n^q]$ (by choice of $z$). Also, $g(|z|) \geq f(|z|) \geq |z|$. Let $m$ be the integer such that, for some $j$, $f_i((z,0)^j) = (z,0^m)$. Then we have $m \leq p_i(|(z,0)|) \leq p_i(2q(|z|))$. Thus $(z,0^m) \not\subseteq K[t \log n, n^q]$ (by choice of $r$), which contradicts the fact that $(z,0^m) \subseteq K[t \log n, n^q]$ (since $z \in Z$).

Notice also that there is some $h, 0 \leq h \leq k$ such that $f(z)1^h \not\subseteq \{z_1, \ldots, z_{k+1}\}$. (There are $k + 1$ choices for $h$, but there are only $l \leq k$ $z$'s.) Fix $h$. Also, since range($f$)

109
and since \( f(x) \neq f(z) \) for all \( x \in Z \), it follows that \( f(x)|^k \notin \{z, f(x), f(z), \ldots, f(z)|^k\} \) for all \( z \in Z \).

We are now ready to define \( A_\tau \). Let \( Y = \{f(x)|^k : j \neq h\} \cup \{f(x), f(z), \ldots, f(z)|^k : z \in Z \text{ and } z \notin K[\log n, n^3]\} \). Set \( A_\tau(y) = 1 \) for all \( y \in Y \); the construction of \( A_\tau \) will be complete once an assignment is made for \( A_\tau(f(x)|^k) \). (It is important to note, using the observations in the preceding paragraphs, that \( f(x)|^k \) is not in \( Y \), and thus we are still free to set \( A_\tau(f(x)|^k) \) to 0 or 1.)

6 Conclusions and Questions

We have presented a condition \( Q \) which has a natural interpretation in terms of one-way functions, in terms of the \( E = NE \) problem, and in terms of the Kolmogorov complexity of sets in P. If P=NP, then \( Q \) is true, whereas if \( E \neq NE \), then \( Q \) is false. Furthermore

\[ Q \implies E^T_{\text{NP}}(\text{TALLY}) = \ldots = E^T_{\text{NP}}(\text{TALLY}) \]

and

\[-Q \implies E^T_{\text{NP}}(\text{TALLY}) \subseteq E^T_{\text{NP}}(\text{TALLY}) \subseteq E^T_{\text{NP}}(\text{TALLY}) \subseteq \ldots \subseteq E^T_{\text{NP}}(\text{TALLY}).\]

Since there are oracles relative to which P=NP (and thus \( Q \) is true) and oracles relative to which \( E \neq NE \) (and thus \( Q \) is false), it follows that the question of whether or not \( E^T_{\text{NP}}(\text{TALLY}) = E^T_{\text{NP}}(\text{TALLY}) \) cannot be resolved by any proof technique which relativizes.

We mention in closing some other interesting open problems from [TB-88] regarding the classes of sets which are equivalent to sparse and tally sets:

1. Is \( E^T_{\text{NP}}(\text{SPARSE}) = \text{P/poly} \)?
2. Is \( E^T_{\text{NP}}(\text{SPARSE}) = E^T_{\text{NP}}(\text{SPARSE}) \)?
3. Is \( E^T_{\text{NP}}(\text{TALLY}) \subseteq E^T_{\text{NP}}(\text{SPARSE}) \)?

Finally, recall that, if Sewelson's conjecture that \( E = NE \) \( \implies \) \( E = \text{NP} \) is true, then \( Q \) is equivalent to \( E = NE \). Also recall that

\[ Q \implies \text{Every infinite set in P has an infinite P-printable subset.} \]

Thus it is relevant to ask if

\[ E = NE \implies \text{Every infinite set in P has an infinite P-printable subset.} \]

In light of Proposition 3, this is equivalent to asking if \( E = NE \) implies that no infinite NE predicate is E-immune.

References


J. Hartmanis and Y. Yesha, Computation times of NP sets of different densities, Theoretical Computer Science 34, 17–32.

S. Homer, Minimal degrees for polynomial reducibilities, J. ACM 34, 480–491.


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


H. Rogers, Theory of Recursive Functions and Effective Computability, MIT Press, Cambridge, MA.

U. Schöning, Complexity and Structure, Lecture Notes in Computer Science 211.


S. Tang and R. Book, Separating polynomial-time Turing and truth-table degrees of tally sets, To appear in Proc. 15th International Collo-