ABSTRACT

The existence of minimal degrees is investigated for several polynomial reducibilities. It is shown that no set has minimal degree with respect to polynomial many-one or Turing reducibility. This extends a result of Ladner [L] where recursive sets are considered. An "honest" polynomial reducibility, $<_{\text{h}}$, is defined which is a strengthening of polynomial Turing reducibility. We prove that no recursive set (or r.e. and P-immune set) has minimal $<_{\text{h}}$-degree. However, proving this same fact for all sets (or even all $\Delta^0_3$ sets) would imply $P = NP$. Finally, a partial converse of this result is obtained, proving that if a certain class of one-way functions exists then no set has minimal $<_{\text{h}}$-degree.

1. Introduction

This paper presents some applications of recursion theory to complexity theory. Many of the basic concepts of and major motivations for concrete complexity theory have come from analogies with recursion theory. These include the various complexity-theoretic hierarchies and the study of structural properties of complexity classes. Most of the results in these theories have come from variants of simple arguments; mainly diagonal arguments and simulations. The more powerful methods of recursion theory, for example priority arguments and effective forcing, have not seemed to be applicable to this field.

The results here relate problems of complexity theory to classical recursion theory. They utilize some of the more powerful methods from recursion theory and indicate that the two theories may have more to say about each other than has previously been believed.

The problems considered here concern minimal degrees for polynomial-time reducibilities. Sets of minimal degree have the property that they are not in polynomial time and are indivisible in the sense that any simpler set must be polynomial. A new but natural reducibility, honest polynomial reducibility, $<_{\text{h}}$, is defined. This reducibility is a strengthening of polynomial Turing reducibility. It is shown that if no set is of minimal $<_{\text{h}}$-degree then $P = NP$. On the other hand, no recursive set (or set which is recursively enumerable and p-immune) has minimal $<_{\text{h}}$-degree. Hence even though $<_{\text{h}}$ is a polynomial time reducibility, it has different (or at least much more difficult to prove) properties when applied to recursive and non-recursive sets. Moreover, the properties of this reducibility on non-recursive sets are related to the $P = NP$ problem.

The first result above, that no minimal $<_{\text{h}}$-degree implies $P = NP$ might be misleading in that it is possible one could prove outright that there are sets of minimal $<_{\text{h}}$-degree. However, a partial converse to this result is given which shows that is unlikely. Namely, assuming a certain class of one-way functions exist then no set is of minimal $<_{\text{h}}$-degree.

The results presented here are for honest-Turing reducibility. An analogous notion of
honest-many-one reducibility is presented as well. Similar properties can be shown for this reducibility, generally with minor changes required in the proofs. The reader is referred to [H1] and [H2] for these results as well as for complete proofs of the theorems presented here.

The methods used in these proofs come from recursion theory. In particular, a priority argument is used for the result on r.e. sets and a primitive effective forcing argument is used to prove that if there are no minimal \( \leq^p \)-degrees then \( \text{P} \neq \text{NP} \). It is also worth noting that this result seemingly does not relativize. This is because the reducibility \( \leq^p \) limits quite strongly the queries which can be made during a computation and with this restriction the relativizations of the computations do not have the same properites as the unrelativized versions.

The existence of sets of minimal degree is also considered for the usual polynomial Turing and polynomial many-one reducibilities. Here Ladner [L] showed that no recursive set can have minimal polynomial Turing or polynomial many-one degree. A different argument shows that this same fact is true for nonrecursive sets as well.

2. The Reducibilities

We consider computations on (oracle) Turing machines. Without loss of generality the tape alphabet of all (oracle) Turing machines is \( \Sigma = \{0,1\} \), and all languages considered will be subsets of \( \Sigma^* \). For \( x \in \Sigma^* \), \( |x| \) denotes the length of \( x \).

We first recall the two notions of polynomial reducibility due to Karp [K] and Cook [C]. These are time bounded versions of many-one and Turing reducibility. We say a set \( A \) is many-one reducible to a set \( B \) in polynomial time (\( A \leq_{m}^p B \)) if there is a polynomial time computable function \( f \) such that \( x \in A \iff f(x) \in B \).

Oracle Turing machines run in polynomial time if there is a polynomial \( p \) such that for any oracle set and any input of length \( n \) the machine halts within \( p(n) \) steps. A set \( A \) is Turing reducible to \( B \) in polynomial time (\( A \leq^p B \)) if there is a polynomial time oracle Turing machine \( T \) with oracle \( B \) such that \( x \in A \iff T^B \) accepts \( x \). The oracle machine \( T \) is sometimes referred to as a reduction procedure as it reduces \( A \) to \( B \).

We want to consider here two different polynomial reducibilities which arise from the notion of polynomial honesty. A polynomial-time compatible function \( f \) is honest if there is a polynomial \( q \) such that for all \( x \), \( q(|f(x)|) \geq |x| \). Polynomial honesty has previously arisen in several places in complexity theory. One of the first examples can be found in Stockmeyer [St]. More recently the work of Young [Yo] and Joseph and Young [JY] has made significant use of this notion.

The concept of polynomial honesty gives rise to two new reducibilities. The "honest reducibilities" defined here are strengthenings of the usual polynomial reducibilities. A set \( A \) will be honest-many-one reducible to a set \( B \) if \( A \leq_{m}^p B \) via an honest polynomial time function.

The motivation behind the corresponding Turing reductions is a bit harder to come by. \( \leq^p_{m} \) has the property that if a set \( A \) is reducible to a set \( B \), then there is a polynomial \( p \) such that to answer questions about \( A \) of length \( n \) requires knowledge of questions about \( B \) of length at most \( p(n) \). For the corresponding honest reducibility we will require that to answer length \( n \) questions about \( A \) we need only ask questions of \( B \) of length in some polynomial sized interval about \( n \). That is, \( A \) will be \( \leq_{m}^{p} \)-Turing reducible to \( B \) if \( A \) is polyno-
mial Turing reducible to $B$ as above and in addition there is a polynomial $q$ such that questions about $A$ of length $n$ use knowledge of questions $x$ about $B$ only for $x$ with $q(|x|) \geq n$.

Formally, a set $A$ is honest-many-one reducible to a set $B$ $(A \leq^h_m B)$ if there is a polynomial time computable function $f$ and a polynomial $q$ such that for all $x$,

1. $x \in A$ iff $f(x) \in B$ and
2. $q(|f(x)|) \geq |x|$.

$A$ is honest-Turing reducible to $B$ $(A \leq^h_T B)$ if there is an oracle Turing machine $M$ such that

1. $A \leq^h_T B$ via oracle Turing machine $M$, and
2. there is a polynomial $q$ such that for all $x$, if $M_B(x)$ queries $B$ about a string $y$ then $q(|y|) \geq |x|$.

Note that both $\leq^h_m$ and $\leq^h_T$ are reflexive, transitive relations and so we can define the equivalence relations $\equiv^h_m(\equiv^h_T)$ by $A \equiv^h_m B (A \equiv^h_T B)$ iff $A \leq^h_m B$ and $B \leq^h_m A (A \leq^h_T B$ and $B \leq^h_T A)$. The equivalence classes of these relations are called honest-many-one degrees and honest-Turing degrees respectively. Questions concerning the properties of $\leq^h_m$ and $\leq^h_T$ often reduce to questions about the corresponding degrees.

We want to consider minimal degrees for the four reducibilities above. Let $\leq$ be any of these four polynomial time reducibilities. Note that if $A$ is recognizable in polynomial time then $A \leq B$ for any set $B$, where $B \neq \emptyset$, $B \neq \Sigma^*$. Hence the polynomial time recognizable sets form the least degree with respect to each of the reducibilities (ignoring the sets $\emptyset$ and $\Sigma^*$ here.) Minimal degrees are those which are not polynomial time but there are no sets strictly between the degree and polynomial time. We say that a set $C$ is minimal with respect to $\leq$ if

1. $C \in P$ and
2. for any set $D$ if $D \leq C$ (that is $D \subseteq C$ and $C \not\equiv D$) then $D \not\in P$. A minimal degree for $\leq$ is one which is made up of minimal sets. (Note that if $C$ is minimal and $D \not\in C$ then $D$ is minimal as well.)

3. Non-Minimality Results

We begin by recalling the following theorem of Ladner found in [L].

**Theorem:** Let $C$ be recursive $C \not\in P$. Then there exists a set $A : A \not\in P$ such that $A \leq^P_m C$ and $C \not\in P A$.

Note that because of the two different polynomial reducibilities used in the conclusion, this theorem implies that no recursive set can have minimal $\leq^P_m$ degree or minimal $\leq^P_T$ degree.

What if the condition that $C$ be recursive is dropped from the above theorem? Our first results show that even without this requirement, there are no minimal $\leq^P_m$ or $\leq^P_T$ degrees. It should be noted that the fact that no nonrecursive $C$ can have minimal $\leq^P_T$ degrees can be seen from the proof of Ladner's theorem. For $\leq^P_m$ degrees, this does not seem to be the case. We give here different, and more elementary, proofs of these facts.

**Theorem 1:** Let $C$ be nonrecursive. There exists a set $B$ such that $B$ is not recursive, $B \leq^P_m C$ and $C \not\in P B$.

**Proof:** Define $B = \{xO^{2^{1^x}} | x \in C\}$.

(i) Since $C$ is not recursive, $B$ is not either.
in the computation of $M^B$ on $y$, we have $|x| \leq q(|y|)$ and so $2^z < |x| \leq q(|y|) < 2^{|y|}$ which yields $|z| < |y|$. So the algorithm is called recursively only for inputs $z$ with $|z| < |y|$ and hence must eventually halt. As $C \leq_r B$ via machine $M$ the algorithm is correct.

Hence the above algorithm determines membership in $C$ and so $C$ is recursive contrary to our assumption. □

The above result, together with that of Ladner shows that no set can have minimal $\leq_r$ or $\leq_m$ degree. It might be of interest to see if other properties of recursive $\leq_r$ and $\leq_m$ degrees hold in general as well.

Turning to the reducibility $\leq_r$ we again consider the question of minimal degrees. For recursive sets Ladner's techniques for $\leq_m$ and $\leq_r$ can be applied.

**Theorem 3:** There is no recursive set with minimal $\leq_r$-degree.

Using more complicated arguments this result can be extended to more general classes of sets. Recall that a set is $P$-immune if it does not contain an infinite, polynomial time computable subset. A priority argument can be used to show.

**Theorem 4:** If $A$ is $P$-immune and recursively enumerable then $A$ is not $\leq_r$-minimal.

Sketch of Proof: A set $B$ is constructed such that $B \not\leq_r P$, $B$ is r.e. and $B <^h A$. The underlying method used is delayed diagonalization as formalized by Ladner [L]. The assumption of

(i) $B \leq_m^r C$

This is true as $y \in B \iff (y$ is of the form $xO^2 z \land z \in C)$.

(ii) $C \leq_p^r B$

To show this assume it is false, say $C \not\leq_p B$ via an oracle machine $M$ which runs in time $q(x)$, $q$ a polynomial. We will show that in this case $C$ is recursive.

Choose an integer $b$ such that $\forall m \geq b, q(m) < 2^m$. Let $D$ be the set of strings in $C$ of length $\leq b$. The following recursive algorithm computes whether or not a string $y$ is in $C$ and hence implies that $C$ is recursive.

**Algorithm:** Let $y$ be the input. We determine if $y \in C$.

1. If $|y| \leq b$ test if $y \in D$.
   If $y \in D$, output 'yes'.
   If $y \not\in D$, output 'no'.
   Halt.

2. If $|y| > b$ run machine $M^B$ on input $y$ until $M^B$ tells us whether or not $y \in C$.
   Each time this computation queries $B$ about some string $x$ we see if $x$ is of the form $zO^2 z$, for some $z$.

   (i) If $x$ is not of this form then $x \not\in B$ and go on with the computation.

   (ii) If $x = zO^2 z$, recursively input $z$ to the algorithm and when the answer is returned go on with the computation.

First note that for any input $y$ the algorithm eventually halts since if $|y| \leq b$ it halts immediately in step 1 while if $|y| > b$, then $q(|y|) < 2^y$ and for any $x = zO^2 z$ queried
P-immunity is quite strong. It allows us, since we construct \( B \) as a subset of \( A \), to ensure \( B \upharpoonright P \) by simply making \( B \) infinite. It is also used in making \( A \upharpoonright P \). The requirements which ensure this fact will sometimes be injured and a finite injury priority argument is used to guarantee that each requirement is eventually satisfied.

We build a set \( B \subseteq A \) in stages. \( B \) will satisfy the following requirements:

\[ L_i: |B| \geq i \]
\[ R_i: A \neq M_i^B \]

(Here \( M_i \) is the \( i^{th} \) honest polynomial reduction procedure.)

The \( L_i \) will ensure that \( B \) is infinite, the \( R_i \) that \( A \upharpoonright P \). That \( B \subseteq P A \) will follow from the construction.

In order to satisfy \( L_i \) we copy \( A \) into \( B \) until we see, via a recursive enumeration of \( A \), that \( |B| \geq i \). To satisfy \( R_i \), after some initial segment of \( B \) has been determined, we make \( B \) look like the empty set \( (\emptyset) \). This is done until it is seen that for some \( x \), \( A(x) \neq M_i^B(x) \). By the honesty of \( M_i \) we can as well require that \( M_i^B(x) = M_i^\emptyset(x) \). If for this \( x \), we have \( x \in A \) (i.e. \( A(x) = 1 \)) when the above inequality is found, then it is preserved throughout the construction and \( R_i \) is permanently satisfied. However it may happen that we find \( O = A(x) \neq M_i^B(x) \), but at a later stage of the construction \( x \) is enumerated into \( A \), giving equality. At this stage we say \( R_i \) is injured and we have to attempt to satisfy it again. However, the \( P \)-immunity of \( A \) implies that \( R_i \) can only be injured finitely often and is eventually satisfied.

The assumption that an r.e. set be \( P \)-immune is quite strong. Nonetheless, it seems difficult, using this same assumption, to prove nonminimality for any class of non-r.e. sets. In the next theorem we have an even stronger assumption, namely that both the set and its complement are \( P \)-immune. In this case no definability assumption is necessary for \( A \). The proof is a straightforward delayed diagonalization and is omitted here. The details can be found in [HI].

**Theorem 5:** Let \( A \upharpoonright P \) be such that both \( A \) and \( A \) are \( P \)-immune. Then there is a \( B \upharpoonright P \) with \( B \subseteq P A \).

4. A Connection with the \( P=\text{NP} \) Problem

The situation for general nonrecursive sets seems different and difficult. The main theorem says that if there is no set of minimal \( \leq P \)-degree then \( P \neq \text{NP} \). By the above, if a \( \leq P \)-minimal set exists it must be nonrecursive. The theorem says that we need only consider sets which are in low levels of the arithmetic hierarchy.

**Theorem 6:** If there is no set in the class \( \Delta^0_2 \) (recursive in \( 0^n \)) of the arithmetic hierarchy which is \( \leq P \)-minimal then \( P \neq \text{NP} \).

**Idea of proof:**

We assume \( P=\text{NP} \) and using this assumption construct a set of minimal \( \leq P \)-degree which is recursive in \( 0^n \). The construction is based on the construction of a minimal Turing degree originally proved by Spector [Sp]. The method can be viewed as a primitive forcing argument. The extreme effectiveness of the reductions considerably complicates the argument. The complete proof is presented in [H2]. An infinite sequence of trees \( T_0, T_1, T_2, T_3, \ldots \)
(with $T_0$ = the full binary tree) is constructed. For all $s$, $T_{s-1}$ is a proper subtree of $T_s$. Each node of each tree is labelled with a binary string and the set $C$ is the unique path through $T_0$ determined by the roots of all the succeeding $T_i$.

The construction is carried out in stages. At stage $s+1$ we have $T_s$ and want to construct $T_{s+1}$. We consider the $(s+1)^{st}$ $\leq_T$-reduction procedure (in some fixed enumeration of $\leq_T$ reduction procedures). This reduction procedure is given by an oracle Turing machine $M_s(\cdot)$. Our construction of $T_{s+1}$ ensures that for $C$ as defined above and $B = M_s C$ we will have either $B \in P$ or $C \leq_T B$. Hence this $(s+1)^{st}$ reduction procedure does not violate the minimality of $C$. In addition we build $T_{s+1}$ so that $C \neq P_{s+1}$ (the $(s+1)^{st}$ polynomial time set).

As this construction is carried out for infinitely many stages, we diagonalize through all possible $\leq_T$ reductions and all polynomial time sets and so the constructed set $C$ is $\leq_T$-minimal.

The proof that the above construction works uses the assumption that $P=NP$ at several crucial points. It seems unlikely if can be removed from this proof.

5. A Partial Converse

While Theorem 5 indicates that the question of minimal $\leq_T$-sets is of import for complexity theory this result may be misleading: it may just be the case that one can prove that $\leq_T$-minimal sets exist outright. In this case any real relationship between these ideas and the $P=NP$ question would disappear.

One way to show this is not the case is to prove the converse of this theorem. Once done, proving that $\leq_T$-minimal sets exist would imply $P=NP$. What is shown here is not the full converse but a weaker statement. It is shown that if certain one-way functions exist then there are no minimal $\leq_T$-sets. The assumption is seemingly stronger than $P\neq NP$. Nonetheless, it ties the question of minimal sets more firmly to complexity theory.

**Definition:** Given a set $S \subseteq \{0,1\}^*$, a function $f$ is $S$-1-way if,

1. $f$ is computable in polynomial time
2. $f$ is polynomially honest (that is, there is a polynomial $p$ such that $\forall x. p(|f(x)|) \geq |x|$)
3. $f$ is surjective
4. $S \not\leq_T f^{-1}(S)$.

First note that if any such $f$ exists (for any $S$) then $P\neq NP$. (Pf: If $P=NP$ then in polynomial time one can, given a string $y$, find a string $x$ such that $f(x)=y$. But then, as $f$ is honest, $S \not\leq_T f^{-1}(S)$.)

**Theorem 7:** For any nonrecursive set $S$, if there exists an $S$-1-way function then $S$ is not $\leq_T$-minimal.

Proof: Assume $S \not\in P$ and there exists an $S$-1-way function $f$.

(i) $f^{-1}(S) \not\leq_T S$

This holds since $y \in f^{-1}(S)$ iff $f(y) \in S$ and $f$ is polynomially honest.

(ii) $f^{-1}(S) \not\in P$. (In fact $f^{-1}(S)$ is not recursive.)
If $f^{-1}(S) \in P$ then we could test for membership in $S$ as follows: To see if $w \in S$, compute all values of $f$ until the least $x$ such that $f(x) = w$ is found. Then $w \in S$ iff $x \in f^{-1}(S)$. Hence $S$ is recursive contradicting the assumption on $S$.

By the definition of $f$, $S \notin \frac{\lambda}{\lambda}$ $f^{-1}(S)$ and so with (i) and (ii) above this shows $S$ is not $\leq \frac{\lambda}{\lambda}$-minimal. □

Recall that no recursive set can be $\leq \frac{\lambda}{\lambda}$-minimal. Hence we have,

**Corollary:** Assume that for any nonrecursive set $S$, there is an $S$-1-way function. Then no set has minimal $\leq \frac{\lambda}{\lambda}$-degree.

The above results remain true if conditions (2) and (3) in the definition of $S$-1-way are replaced by

(2') $f$ is length preserving

(3') $f$ is one-one, as (2') and (3') are stronger conditions. When stated in this way the definition appears more similar to the 1-way functions which have proved useful in cryptography (see [GGM] or [Ya]). While no direct connection between these two notions is known, finding such would certainly be of interest. For example it would be useful to know if the existence of 1-way functions (say as defined in [GGM]) implies the existence of $S$-1-way functions as above.

**REFERENCES**


[H1] Homer, S., "Minimal polynomial degrees of nonrecursive sets", submitted for publication.


