SOME COMMENTS ON FUNCTIONAL SELF-REDUCIBILITY AND THE NP HIERARCHY

A. Borodin*
A. Demers
TR 76-284

July 1976

Department of Computer Science
Cornell University
Ithaca, New York 14853

* Allan Borodin on leave from the University of Toronto.
SOME CONJECTURES ON FUNCTIONAL SELF-REDUCIBILITY
AND THE NP HIERARCHY

A. Borodin
A. Demers

Abstract:

In Valiant [11] and Schnorr [9], concepts of "functional self-reducibility" are introduced and investigated. We concentrate on the class NP and on the NP hierarchy of Meyer and Stockmeyer [7] to further investigate these ideas. Assuming that the NP hierarchy exists (specifically, assuming that $P \subsetneq \Sigma^P_1 \subsetneq \Sigma^P_2$) we show that, while every complete set in $\Sigma^P_2$ is functionally self-reducible, there exist sets in $\Sigma^P_2$ which are not functionally self-reducible.
I. Introduction

Following the work of Valiant [11] and Schnorr [9], we wish to motivate and define a property which we call "functional self-reducibility". As in Valiant's study of "evaluation vs. checking", our main concern lies in knowing if and when a functional problem can be more difficult that the corresponding set recognition problem. In this regard, we study a very general definition of self-reducibility. It should be noted that, as usual, reducibility implies relative complexity bounds but not conversely.

Schnorr [9] studies a notion of reducibility that is more restrictive than ours, in order to derive results concerning the existence of optimal programs for certain sets. For example, consider the satisfiability problem A and the "natural" functional problem f associated with A - namely, to compute a satisfying assignment if one exists. Schnorr observes that f is reducible to A in the following way. Let x be a propositional statement on k variables. We can determine a truth value for the variable \( v_1 \) by substituting "true" and "false" for \( v_1 \) in x. If x is satisfiable, then one of these substitutions (at least) must yield a satisfiable formula with \( k-1 \) variables; using the oracle A we can determine which one. Continuing in this way we can compute a satisfying assignment, using \( k \) oracle questions which are decreasing in length. It follows that, if one defines \( T(n) \)
to be the maximum time complexity on inputs of length \( n \), then 
\[ T_{f}(n) = O(nT_{A}(n)) \]. Obviously, \( T_{A} = O(T_{f}(n)) \). Thus, using a
result of Levin [6] that \( f \) has optimal program (for Kolmogorov-
Uspensky machines) on the set \( \{x|x \in A\} \), \( A \) must have a "nearly"
optimal algorithm. (It is still possible, say, for \( A \) to have a
complexity sequence \( T, T/\log T, T/\log^{2} T, \ldots \).

Schnorr also states that "any polynomial complete problem
in \( NP \) can be encoded such that the self-reducible property holds".
Thus, using Levin [6] again, such sets have optimal programs.
Certain sets which are not seemingly complete - e.g., the graph
isomorphism problem - also have this property, but it remains
an open question whether all sets in \( NP \) have this property.

Our interest lies in the \( NP \) hierarchy and the study of
self-reducibility. It follows from Valiant [11], Baker, Gill
and Solovay [1], Schnorr [9], or the development here that if
\( P = NP \) then every set is complete and hence self-reducible. We
henceforth assume that the hierarchy exists (in particular, that
\( P \neq NP \neq \mathcal{P}_{1} \neq \mathcal{P}_{2} \)).

II. Definitions and Notation

For our purpose, the precise model of computation is
unimportant, but we might as well choose Turing machines. As
usual \( P \) (respectively \( NP \)) is the class of languages (say over
\( \Sigma = \{0,1\} \)) acceptable in deterministic (resp. nondeterministic)
polynomial time. More generally, let us recall the definition
of the NP hierarchy (Meyer and Stockmeyer [7]).

Definition 1:

(a) $P(B)$ (respectively $NP(B)$) is the class of languages accepted by deterministic (resp. nondeterministic) oracle machines which run in polynomial time,

(b) If $C = \{B_i\}$ is a class of sets, then

$$P(C) = \bigcup_i P(B_i)$$

$$NP(C) = \bigcup_i NP(B_i)$$

(c) $\Sigma^P_0 = \Pi^P_0 = \Delta^P_0 = P$

$$\Sigma^P_{k+1} = NP(\Sigma^P_k)$$

$$\Pi^P_{k+1} = co-NP(\Sigma^P_k)$$

$$\Delta^P_{k+1} = P(\Sigma^P_k)$$

\begin{align*}
\{ & \Sigma^P_{k+1} = \Pi^P_{k+1} = \Delta^P_{k+1} = P(\Sigma^P_k) \} \\
& k \geq 0
\end{align*}

In particular, $\Sigma^P_1 = NP(P) = NP$

This definition is due to Meyer & Stockmeyer. It corresponds to the arithmetic hierarchy in recursive function theory, where

$\Sigma^P_{k+1}$ = those sets recursively enumerable in $\Sigma^P_k$

$\Delta^P_{k+1}$ = those sets recursive in $\Sigma^P_k$.

We also have the inclusions

$$\Sigma^P_k \cup \Pi^P_k \subseteq \Delta^P_{k+1} \subseteq \Sigma^P_{k+1} \cap \Pi^P_{k+1}$$

$k \geq 0$.

One can only conjecture that these inclusions are proper; indeed,
it is completely open whether the hierarchy exists at all.


$$A \in \Sigma_k^P$$ iff there exists a polynomial \(p(x)\) and a

\((k+1)\)-ary relation \(R(x, y_1, \ldots, y_k) \in \mathbb{P}\) such that

$$A = \{x | \exists y_1 (\forall y_2 \ldots (Q_k y_k) R(x, y_1, \ldots, y_k)\},$$

where \(Q_k\) is \(\exists (\forall)\) if \(k\) is odd (even), and \(y_1, \ldots, y_k\)

range over all words of length at most \(p(\|x\|)\).

In particular,

$$A \in \Sigma_1^P = \text{NP}$$ iff there exist \(R\) and \(p\) such that

$$A = \{x | \exists y \ | y| \leq p(\|x\|) \land R(x, y)\}.$$

In this form, the relation \(R\) defines a set \(A_R\). Of course, many different relations can define the same set. Indeed, if \(A \in \Sigma_k^P\), we can define \(A\) as a \(\Sigma_l^P\) set for any \(l \geq k\). (If \(A \in \Sigma_k^P \cap \Sigma_{k-1}^P\), then a vague but interesting question concerns how "different" the relations \(R(x, y_1, \ldots, y_k)\) defining \(A_R = A\) can be; see Open Problems 1 and 2).

The relation \(R\) also defines a total mapping

$$f_R(x) = \begin{cases} \text{some } y_1 & (\forall y_2) \ldots (Q_k y_k) \\ R(x, y_1, \ldots, y_k) & \text{if no such } y_1 \text{ exists.} \end{cases}$$
That is, $f_R$ must compute an acceptable $y_1$ or say that none exists. Note that the output $y_1$ is not necessarily uniquely defined. Indeed, we believe the following to be open problems:

**Open Problem 1:** (for $k \geq 1$)

(a) \((\exists A \in \Sigma^P_{k-1} (\exists y_1) (\forall y_2) \ldots (\forall y_k) R(x, \ldots, y_k))\)

\[ A = \{ x | (\exists y_1)(\forall y_2) \ldots (\forall y_k) R(x, \ldots, y_k) \} \]

(b) \((\forall A \in \Sigma^P_{k-1} (\exists y_1) (\forall y_2) \ldots (\forall y_k) (\exists R))\)

\[ A = \{ x | (\exists y_1)(\forall y_2) \ldots (\forall y_k) \} \]

The literature is full of excellent natural examples of problems in $\Sigma^P_1 = \text{NP}$, most of which are complete but some seemingly not. The definitive works are obviously Cook [2] and Karp [4]. For the hierarchy in general, the best examples can be found in Stockmeyer [10]. For all these examples, there seems to be no way to uniquely define $y_1$ while keeping $R$ polynomial.

We can now define functional self-reducibility for any set $A \in \Sigma^P_k$.

**Definition 2:**

(a) We say that $(A, R)$ (or just $R$) is **functionally self-reducible** if

(i) $A = A_R$, and

(ii) $f_R \leq_T^P A$ - i.e., $f_R$ is polynomially reducible

(in the sense of Turing or Cook [2]) to $A$. 
(b) \( A \in \Sigma^P_k \rightarrow \Sigma^P_{k+1} \) is **functionally self-reducible** if for every \( k+1 \)-ary polynomially computable relation \( R \) defining \( A \) (as a \( \Sigma^P_k \) set) \( f_R \leq^P_T A \).

**Open Problem 2:**

Suppose \( R_1 \) and \( R_2 \) both define \( A \in \Sigma^P_k \rightarrow \Sigma^P_{k+1} \) as a \( \Sigma^P_k \) set. Does it follow that \( R_1 \) is functionally self-reducible iff \( R_2 \) is functionally self-reducible?

If we assume \( P \neq NP \), then it is not difficult to define \( \Pi^* \) (clearly a \( \Pi^P_0 \) set) as a \( \Pi^P_2 \) set by a relation \( R(x,y,z) \) so that \( f_R \) is not polynomially computable. Thus, in defining functional self-reducibility, one cannot demand that every relation defining set \( A \) be self-reducible. On the other hand, we do not know how to define \( \Pi^* \) as a \( \Pi^P_1 \) set so that the \( R \) defining \( \Pi^* \) is not self-reducible. (This seems to require the assumption that \( \Pi^P_1 \cap \Pi^P_0 \nsubseteq \Pi^P_0 \); see Proposition 5 in Valiant [11]).

Unfortunately, our definition cannot deal well with \( \Pi^P_0 \) sets defined as \( \Sigma^P_1 \) sets, even though such sets may occur naturally. For example, the "natural" way to specify the composite numbers is as a \( \Sigma^P_1 \) set; however, it may turn out that the composite numbers are in \( P \), but producing a nontrivial factor is not polynomially computable (see Miller [8]).
III. Complete Sets are Functionally Self-Reducible

The following development shows that any set $A$ which is complete in $\Sigma^P_k$ is functionally self-reducible. The proof is based on the observation that the predicate $R(x,y) = "y is an accepting computation of x"$ is self-reducible. This observation can be found in Baker, Gill, and Solovay [1] and in Valiant [11]. For $\Sigma^P_1$, the theorem can be viewed as a restatement of Schnorr's [9] remark that "any polynomial complete set can be encoded such that the self-reducible property holds".

Theorem 1

Let $A$ be complete in $\Sigma^P_k$ ($k \geq 1$). Then $A$ is functionally self-reducible.

Proof:

Let $A = \{x | (\exists y_1) \cdots (\exists_{y_k}) R(x,y_1,\ldots,y_k)\}$. We wish to show that $f_R$ can be computed in polynomial time given the oracle $A$.

We know that $A$ is the set accepted by some nondeterministic oracle machine $M$ which uses an oracle $B \in \Sigma^P_{k-1}$. Furthermore, we can insist that $M$ asks only one oracle question and then decides membership in $A$. That is, $M$ nondeterministically chooses a $y_1$ and then consults the oracle $B$, where

$$B = \{<x_1y_1> | (\exists y_2)(\exists y_3)\cdots(\exists_y_k) \rightarrow R(x,y_1,\ldots,y_k)\}$$

If $<x_1y_1> \notin B$, then $M$ accepts; otherwise, $M$ rejects. (For
$k=1, B \in \Sigma_0$ can be taken as $\{ <x,y> | \neg R(x,y) \}$. Now, consider the relation $R_M(x',y_1',...,y_k')$ defined as follows:

(i) $x'$ is of the form $1^t ID_j$

(ii) $y'$ is of the form $m_1...m_t ID_{accept}$

(iii) from $ID_j$, $M$ can proceed in $t$ steps by the sequence of choices $m_1,...,m_t$ to the accepting $ID_{accept}$. The last step consists of the appropriate oracle call.

By definition, the set $A_M$ defined by $R_M$ is a $\Sigma_k^P$ set. It is easy to see that $\Lambda \leq_T A_M'$, and since by hypothesis $A$ is complete in $\Sigma_k^P$, we have $A \equiv_T A_M'$.

Now observe that $R_M$ is functionally self-reducible, even in the strong sense of Schnorr [8]. We can show that $R$ is self-reducible - i.e., $f_R \leq_T A$, as follows:

(i) Transform $x$ to $x' = 1^p(n)ID_0$.

(ii) Compute $f_R(x')$ using the oracle $A$; here is where we use the completeness of $A$, for we replace the oracle calls to $A_M$ by calls to $A$.

(iii) Recover $f_R$ from $f_{R_M}$ in the obvious way.

In the case of $\Sigma_1^P = NP$, our intuition about all the known complete sets confirms Theorem 1. Now, consider $\Sigma_2^P$. The first complete set for this class was given by Meyer and Stockmeyer [7]; namely
$B_2 = \{ R | R \text{ is a well-formed propositional formula over the variables } y_{11}, \ldots, y_{12}, y_{21}, \ldots, y_{2j}, \text{ and } \exists y_{11} \cdots \exists y_{12} (\forall y_{21}) \cdots (\forall y_{2j}) R(y_{11}, \ldots, y_{2j}) \}$

Clearly, $R$ is functionally self-reducible. On the other hand, Stockmeyer [10] shows that $\text{INEQ} = \{ <E,F> | E \text{ and } F \text{ are "integer expressions" and the sets of integers defined by } E \text{ and } F \text{ are different.} \}$ is complete in $\Lambda_2^P$. In this case, the "natural" function to be computed is an integer which exhibits the inequality. It is not as obvious here that any relation defining $\text{INEQ}$ should be self-reducible.

IV. A $\Lambda_2^P$ Set Which is Not Functionally Self-Reducible

Consider the $\Lambda_2^P$ set $\text{NMIN}$ as defined in Meyer and Stockmeyer [7]: $\text{NMIN} = \{ F | F \text{ is a propositional formula which is not minimal} \}$. That is, $F$ is in $\text{NMIN}$ if there exists a shorter (say in terms of primitive propositional languages characters) equivalent formula. It is not know whether $\text{NMIN}$ is complete for $\Lambda_2^P$. We also do not know whether $\text{NMIN}$ - or the "natural" relation $R(x, y_1, y_2) = " y_1 \text{ is a shorter formula than } x \text{ with the same truth value on assignment } y_2 "$ - which defines $\text{NMIN}$ - is functionally self-reducible.

Consider also the satisfiability problem $S$ in $\Sigma_1^P$. Again it is not clear how to reduce $S$ to $\text{NMIN}$, or $\text{NMIN}$ to $S$. Although we cannot prove anything about $S$ and $\text{NMIN}$, by Theorem
2 the possibility exists that S and NMIN are incomparable.

On the other hand, it is easy to reduce satisfiability to the "natural" functional problem associated with NMIN, namely, \( f(F) = F' \) where \( F' \) is a shorter formula equivalent to \( F \). Given \( F \), we can keep computing shorter and shorter equivalent formulas until a minimal formula \( F_x \) is derived. \( F \) is satisfiable if and only if \( F_x \) is not \( v \land \neg v \). Theorem 3 shows that whenever \( A \in \Sigma_2^p \) is sufficiently dense, we can reduce any \( B \in \Sigma_1^p \) to some functional problem associated with \( A \).

Using Theorems 2 and 3, we can show that there are sets \( A \in \Sigma_2^p \) which are not functionally self-reducible.

The next result is based directly on the proof techniques of Ladner [5].

**Theorem 2:**

Suppose for some \( B \not\in \Sigma_0^p \) and \( C \not\in \Sigma_1^p \) we have \( B \preceq_T^p C \) but \( C \preceq_T^p B \). Then there exists \( D \in \Pi_1^p \) such that \( A = C \cap D \) and \( B \) are incomparable; furthermore \( A \not\in \Pi_1^p \).

**Proof:**

We follow Ladner and construct a polynomial time bounded transducer \( T \) such that \( D = \{ x \mid |T(x)| \text{ is even} \} \). \( T \) is trying to satisfy the following conditions:

1) \( B \neq M_i(A) \)

2) \( A \neq M_i(B) \)

3) \( A \neq NP_i \)

where \( M_i(A) \) denotes the language recognized by the \( i \)th deterministic
polynomial time bounded Turing machine with oracle \( A \), and \( \text{NP}_i \) is the language recognized by the \( i \)th nondeterministic polynomial time bounded TM. To satisfy 1) we make \( D \) look empty. To satisfy 2) and 3) we make \( D \) look like \( \Sigma^* \). (\( \Sigma = \{0,1\} \).) \( T \) operates as follows:

\[
T(\lambda) = \lambda
\]

\[
T(x) = T(0^n) \text{ for } x \in \Sigma^n - \{0^n\}
\]

On input \( 0^n \), \( T \) first tries, for \( n \) moves, to reconstruct the sequence \( T(\lambda), T(0), T(00), \ldots \) as far as possible. Let \( T(0^m) \) be the last value computed. Then

(a) If \( |T(0^m)| = 4i \), we are trying to make \( A \neq M_i(B) \).

That is, for \( n \) moves \( T \) tries to find a \( z \) such that \( A(z) \neq M_i(B)(z) \). If no such \( z \) is found, \( T \) prints \( 4i \); otherwise it prints \( 4i+1 \).

In doing this simulation, \( T \) only needs to look at those \( z \) of length \( \leq m \). Also, as part of the \( n \) moves allowed for the simulation we count the time required to compute each oracle question using some fixed, perhaps very inefficient deterministic algorithm for \( B \).

(b) If \( |T(0^m)| = 4i+1 \), we are trying to make \( B \neq M_{21}(A) \).

Again, for \( n \) moves \( T \) tries to find a \( z \) such that \( B(z) \neq M_{21}(A)(z) \). In doing so, \( T \) looks only at \( z \) and \( A \) oracle questions of length \( \leq m \). If no such \( z \) is found, \( T \) outputs \( 4i+1 \); otherwise \( 4i+2 \).

(c) If \( |T(0^m)| = 4i+2 \), we are trying to satisfy \( A \neq \text{NP}_i \).

For \( n \) moves, \( T \) tries to find a \( z \) such that \( A(z) \neq \text{NP}_i(z) \). Here \( T \) must try all computation paths for \( \text{NP}_i(z) \). If no such \( z \) is
found, T outputs 4i+2; otherwise 4i+3.

(d) If \(|T(0^m)| = 4i+3\), we are trying to satisfy \(B \not= M_{2i+1}(A)\) (as in case b). If T can find a \(x\) such that \(B(x) \not= M_{2i+1}(A)(x)\) then it outputs \(4i+4 = 4(i+1)\); otherwise 4i+3.

If range (T) = \(1^*\), then the set \(A\) must be as desired.

By cases, we can show that T cannot "get stuck" on 4i, 4i+1, 4i+2, or 4i+3 for any \(i\).

Corollary:

(Assuming \(\Sigma^p_2 - \Sigma^p_1 \neq \emptyset\)) Let \(B\) be in \(\Sigma^p_1 - \Sigma^p_0\) and not complete for \(\Sigma^p_1\). Then there exists \(A \in \Sigma^p_2 - \Sigma^p_1\) such that \(A\) and \(B\) are incomparable.

Proof:

In Theorem 2, choose \(C\) to be complete in \(\Sigma^p_2\). Then \(B \leq_T C\) (since \(\Sigma^p_2 \supseteq \Sigma^p_1\)) and \(C \geq_T B\) (since \(B\) is not complete for \(\Sigma^p_2\)). \(A = C \land D\) is still in \(\Sigma^p_2 - \Sigma^p_1\), and by the theorem \(A\) and \(B\) are incomparable.

Remark:

If we know \(\Sigma^p_2 - \Delta^p_2 \neq \emptyset\) (as in recursive function theory) then in the Corollary we would not have to require that \(B\) is not complete for \(\Sigma^1_1\).

Open Problem 3:

Assuming \(\Sigma^p_k - \Sigma^p_{k-1} \neq \emptyset\), \(\exists A \in \Sigma^p_k - \Sigma^p_{k-1}\) \(\forall B \in \Sigma^p_{k-1}\) B \(\not\leq_T A\)?
As constructed in Theorem 2, the set $A$ may be sparse; that is,
\[
\lim_{n \to \infty} \frac{\#\{x \in A, |x| < n\}}{\#\{x \in \Sigma^*, |x| < n\}} = 0.
\]

However, we can construct $A$ join $H$, for any set $H$, by letting $A' = A \cup H = \{0x | x \in A\} \cup \{1y | y \in H\}$. Clearly $A'$ is dense if $H$ is dense; moreover, if $A$ satisfies the Corollary and $H$ is in $P$, then $A'$ satisfies the Corollary also. We may as well let $H$ be $\Sigma^*$, though in what follows all we need is that $H$ (and thus $A'$) has an unbounded regular subset.

Lemma:

(Hopcroft [3]). If $A$ has an unbounded regular subset, then there exist $r, u, v, s$ in $\Sigma^*$ such that $u$ and $v$ do not commute and $r(u,v)s \subseteq A$.

The impact of this lemma is that every word in $\{0,1\}^*$ can be encoded as a word in $A$. That is, $b_1 \ldots b_k$ is encoded as $k(b_1 \ldots b_k) = rz_1 \ldots z_ks$, where $z_i = uv$ (if $b_i = 0$) or $vu$ (if $b_i = 1$). Clearly $k$ is injective, and any sets $B$ and $kB = \{kx | x \in B\}$ are of essentially the same complexity.

Given incomparable sets $A$ and $B$ (as in the Corollary), we can now construct a relation $R$ defining $A$ as a $\mathcal{P}_2$ set such that $B \triangleleft_T P R$. 
Theorem 3:
Let \( B \in \Sigma^P_1 \), \( A \in \Sigma^P_2 \), and \( B \subseteq A \). (Note: by using the previous lemma, we can replace the condition "\( B \subseteq A \)" by "\( A \) has an unbounded regular subset"). Then there exists a polynomial computable predicate \( \tilde{R}(x,y,z) \) defining \( A \) as a \( \Sigma^P_2 \) set so that \( B \subseteq^P_T \tilde{R} \).

Proof:
Let \( R \) define \( B \) as a \( \Sigma^P_1 \) set, and let \( R' \) define \( A \) as a \( \Sigma^P_2 \) set; that is,
\[
A = \{ x \mid (\exists y)(\forall z) R'(x,y,z) \}
\]
\[
B = \{ x \mid (\exists y) R(x,y) \}.
\]
Without loss of generality, we may assume

1) \( R(x,\cdot) \) is false for all \( x \). Otherwise, define \( \tilde{R}(x,by') = R(x,y) \) for any \( b \in \Sigma \), and \( \tilde{R}(x,\lambda) = \text{false} \). Clearly \( \tilde{R} \) defines \( B \) as a \( \Sigma^P_1 \) set just as \( R \) does.

2) \( \langle y_1, y_2 \rangle \) represents a pairing of \( y_1 \) and \( y_2 \).

Now define
\[
\tilde{R}(x,\tilde{y},\tilde{z}) = \tilde{R}(x, \langle y_1, y_2 \rangle, \langle z_1, z_2 \rangle)
\]
\[
= R'(x,y_1,z_1) \land ((y_2 \neq \lambda \land R(x,y_2))) \lor (y_2 = \lambda \land \neg R(x,z_2)).
\]

We first need to show that \( \tilde{R} \) defines \( A \).

1) \( (\exists R \subseteq A_R \uparrow = A) \). We must show
\[
(\exists y)(\forall z) R'(x,y,z) \Rightarrow (\exists \tilde{y})(\forall \tilde{z}) \tilde{R}(x,\tilde{y},\tilde{z}).
\]
Case 1: \( x \in B \subseteq A \). Then there exists \( y_2 \neq \lambda \) such that \( R(x,y_2) \) is true, since \( x \in B \). Similarly, \( (\exists y_1)(\forall z) R'(x,y_1,z) \). Thus,
letting \( \tilde{y} = \langle y_1, y_2 \rangle \) we have \((\forall z) \tilde{R}(x, \tilde{y}, z)\) as required.

**Case 2:** \( x \in A - B \). Then \((\exists y_1)(\forall z_1) R'(x, y_1, z_1)\), but \((\forall z_2) R(x, z_2)\) (i.e., \( x \notin B \)). Letting \( \tilde{y} = \langle y_1, \lambda \rangle \), we have \((\forall z) \tilde{R}(x, \tilde{y}, z)\).

2) \((A = A_R, \subseteq A_{\tilde{R}})\). Suppose \( x \in A \). Then \((\exists \tilde{y})(\forall z) \tilde{R}(x, \tilde{y}, z)\). Let \( \tilde{y} = y_1, y_2 \); then \((\forall z) R'(x, y_1, z_1)\), and thus \((\forall z_1) R'(x, y_1, z_1)\), as desired.

We now show that \( B \leq_T^P f_R \).

1) If \( f_R(x) = \text{"no"} \) then \( x \notin A \), and hence \( x \notin B \).

2) If \( f_R(x) = \tilde{y} = \langle y_1, \lambda \rangle \) then \((\forall z) \neg R(x, z)\) (by definition of \( f_R \) and \( \tilde{R} \)), thus \( x \notin B \).

3) If \( f_R(x) = \tilde{y} = \langle y_1, y_2 \rangle \), \( y_2 \neq \lambda \), then \( R(x, y_2) \) is true, and thus \( x \in B \).

We can now prove the main result of this section; namely, that there is a non-self-reducible set in \( \Sigma^P_2 \).

**Theorem 4:**

(Assuming \( \Sigma^P_2 \) \( \neq \phi \)) There exists a set \( A \in \Sigma^P_2 \), and a relation \( R(x, y, z) \) defining \( A \) as a \( \Sigma^P_2 \) set, such that \( f_R \leq_T^P A \).

**Proof:**

Using Theorems 2 and 3, we can construct sets \( A \in \Sigma^P_2 \) and \( B \in \Sigma^P_1 \), and a relation \( R(x, y, z) \) defining \( A \) as a \( \Sigma^P_2 \) set, such that \( B \leq_T^P f_R \) but \( A \) and \( B \) are incomparable. By transitivity of \( \leq_T^P \), \( B \leq_T^P f_R \) and \( B \leq_T^P A \) imply \( f_R \leq_T^P A \), so \( A \) has the desired
properties.

It is easy to translate the above results upward to show that for any $k > 1$ there exists a set $A \in \mathcal{P}_{k-1}$ and a relation $R$ defining $A$ as a $\mathcal{P}_k$ set such that $(A, R)$ is not self-reducible. Unfortunately, the existence of non-self-reducible sets in $\mathcal{P}_1$ remains an open question.

Valiant [11, Proposition 5] shows that, assuming $\mathcal{P}_1 \cap \mathcal{P}_1 = \mathcal{P}_1 \neq \emptyset$, i.e., assuming $P \not\in \mathcal{NP} \cap \text{co-NP}$, there exists a total function which can be checked but not evaluated in polynomial time. In our terminology, he gives a polynomial binary relation $R(x, y)$ which defines $\mathcal{F}^*$ as a $\mathcal{P}_1$ set and which is not self-reducible. Using the techniques of this section, we can strengthen Valiant’s result by showing the existence of a non-self-reducible binary relation defining a set which is properly in $\mathcal{P}_1$.

Theorem 5:

Assume $\mathcal{P}_1 \cap \mathcal{P}_1 = \mathcal{P}_1 \neq \emptyset$. Then there exists a polynomial binary relation $R(x, y)$ such that $A_R \in \mathcal{P}_1 - \mathcal{P}_0$ and $f_R \not\in \mathcal{T} A_R$.

Proof:

Choose $B \in (\mathcal{P}_1 \cap \mathcal{P}_1) - \mathcal{P}_1$. Then there exist polynomial relations $R_B$ and $\bar{R}_B$ such that

$B = \{x \mid \exists y \ R_B(x, y)\}$

$\bar{B} = \{x \mid \exists y \ R_B(x, y)\}$.
By Ladner [5, Theorem 1], there exists a set $A \in \Sigma^P_1 - \Sigma^P_0$ such that $A \not\leq_T^P B$ but $B \not\leq_T^P \Lambda$. Let

$$A' = A \cup \{0,1\}^* \cup \{y \mid y \in \{0,1\}^*\}$$

Clearly $A \not\leq_T^0$, since $A \leq_T^P A'$; moreover $B \not\leq_T^P A'$. We next construct a relation $R'$ defining $A'$ such that $B \not\leq_T^P f_{R'}$. Of course, let

$$\Lambda = \{x (\exists y) R_A(x,y)\}.$$  

We define $R'$ by

$$R'(ax,y) = \begin{cases} R_A(x,y) & \text{if } a=0 \\ R_B(x,y) \lor R_B(x,y) & \text{if } a=1 \end{cases}$$

Clearly $R'$ defines $A'$ as a $\Sigma^P_1$ set. Also, $B \not\leq_T^P f_{R'}$, since

$$x \in B \iff R_B(x, f_{R'}(1x)).$$

Since $B \not\leq_T^P A'$, it follows that $f_{R'} \not\leq_T^P A'$, and $R'$ is the desired relation.

Like Theorem 4, this result can be generalized to any $k \geq 1$.

With Theorems 4 and 5 as evidence, we close with the following:

**Conjecture:**

There exists a set $A \in \Sigma^P_1 - \Sigma^P_0$ and a polynomial relation $R(x,y)$ defining $A$ (as a $\Sigma^P_1$ set) such that $(A,R)$ is not self-reducible.
References


for Context-free Languages", Mathematical Systems Theory 3:2, 
1969.

Complexity of Computer Computations, R.E. Miller and 


Regular Expressions with Squaring Requires Exponential Space", 
Proc. 13th SWAT, October 1972.

[8] G. Miller, "Riemann's Hypothesis and Tests for Primality", 
Proc. 7th STOC, May 1975.

manuscript, September 1975.


manuscript, Harvard Univ., 1975.