Six Hypotheses in Search of a Theorem

Harry Buhrman* Lance Fortnow† Leen Torenvliet‡

Sir, we are truly six special and interesting characters. Believe us. However we have gone lost.
—“Six Characters in Search of an Author,” Luigi Pirandello.

Abstract

We consider the following six hypotheses:

- $P = NP$.
- $SAT$ is truth-table reducible to a $P$-selective set.
- $SAT$ is truth-table reducible to a $k$-approximable set for some $k$.
- $FP_{NP}^{NP} = FP_{NP}^{[log]}$
- $SAT$ is $O(\log n)$-approximable.
- Solving $SAT$ is in $P$ on formulae with at most one assignment.

We discuss their importance and relationships among them.

*URL: http://www.cwi.nl/~Harry.Buhrman.html
E-mail: buhrman@cwi.nl. Partially supported by the Dutch foundation for scientific research (NWO) by SION project 612-34-002, and by the European Union through Neuro-COLT ESPRIT Working Group Nr. 8556, and HC&M grant nr. ERB4050PL93-0516. CWI, Kruislaan 413, 1098SJ Amsterdam, The Netherlands.

†URL: http://www.cs.uchicago.edu/~fortnow. Email: fortnow@cs.uchicago.edu. Supported in part by NSF grant CCR 92-53582, the Dutch Foundation for Scientific Research (NWO) and a Fulbright Scholar award. CWI and University of Chicago, Department of Computer Science 1100 E. 58th. St. Chicago, IL 60637, USA.

‡URL: http://turing.wins.uva.nl/~leen/. E-mail: leen@wins.uva.nl. University of Amsterdam, Department of Computer Science, Plantage Muidergracht 24, 1018TV Amsterdam, The Netherlands.

1 Introduction

Complexity theorists have put considerable effort into investigating the structure and properties of sets in $NP$. This research led to various hypotheses. In this survey paper we put together, for the first time, six hypotheses that we encountered in our own research as well as in the literature. We believe that these hypotheses are important and are closely related to each other.

The first hypothesis is: "$P = NP$.” This is the most famous and important one and does not need any further introduction.

Most sets in $NP$ that arise from practice turn out to be $NP$-complete. Moreover since complete sets reflect the structure of a complexity class they receive close attention. Three of our six hypotheses concern sets that are complete or hard for $NP$.

Selman [Sel82] introduced the $P$-selective sets in analogue of recursion theory. A set is called $P$-selective if there exists a polynomial time computable function that from two strings $x$ and $y$ selects one that (if at least one belongs to $A$) is in $A$. He investigated the possibility for $NP$ to have hard sets that are $P$-selective. He showed [Sel82] that this can not be the case for many-one reductions (unless $P = NP$). This was later improved to $\leq^p_1$ reductions by Buhrman and Torenvliet [BT96b]. The hypothesis we are interested in is: "$NP$ has a truth-table hard set that is $P$-selective.”

Beigel [Bei87a], looking at properties of bounded queries to sets (in $NP$), developed a generalization of $P$-selective sets later dubbed the approximable sets. A set $A$ is $k$-approximable if there exists a polynomial time computable function that with $k$ strings $x_1, \ldots, x_k$ as input, generates $k$ bits $b_1, \ldots, b_k$ such that for at least 1 bit it is true that $b_i \neq \chi_A(x_i)$. That is from the $2^k$ possible settings of $x_1, \ldots, x_k$ one is excluded. Beigel, Kummer and Stephan [BKS95], Agrawal and Arvind [AA96], and Ogihara [Ogi95]
shown that $\text{NP}$ can not have $\leq^p_{\text{det}}$-hard sets that are $k$-approximable for some $k$ (unless $P = \text{NP}$). Since $P$-selective sets are in fact 2-approximable sets this result also improves the bound for $P$-selective sets.

The hypothesis related to this work is: “$\text{NP}$ has a truth-table hard set that is $k$-approximable for some $k$.”

Ogihara [Ogi95] working on the hypothesis that $\text{NP}$ has a truth-table hard $P$-selective set, took it one step further and considered $f(n)$-approximable sets for non-constant functions $f(n)$. He showed that if $\text{SAT}$ is not $a \log(n)$-approximable for $a < 1$ unless $P = \text{NP}$, this result subsumes the results on truth-table reductions to $k$-approximable sets (see Section 3). The hypothesis connected to this work is: “$\text{SAT}$ is $O(\log(n))$-approximable.”

The next hypothesis states that it is possible to compute $\text{SAT}$ in polynomial time when we only consider formulæ with at most one satisfying assignment. It is possible to phrase this in terms of sets as: “unique-$\text{SAT} \in P”$ (see Section 2). Valiant and Vazirani [VV86] showed that this set problem for $\text{SAT}$ is hard for $\text{NP}$ under randomized reductions.

The last hypothesis deals with functions that are computable in polynomial time relative to some set in $\text{NP}$. There are essentially three different ways to define this. The most unrestrictive way is that the polynomial time computable function has unrestricted access to an $\text{NP}$ oracle and is called $\text{FP}^{\text{NP}}$. The next restriction to the oracle mechanism is that the queries have to be non-adaptive: $\text{FP}^{\text{NP}[\text{log}]}$. The last and most restrictive version is that only $O(\log(n))$ queries are allowed on inputs of length $n$: $\text{FP}^{\text{NP}[\text{log}]}$. The last hypothesis can now be stated as: $\text{FP}^{\text{NP}[\text{log}]} = \text{FP}^{\text{NP}}$.

These are the main characters of our paper. We show that these hypotheses are closely related to each other and in Section 3 we show which of these hypotheses implies any of the others. Furthermore we give background information on each of them individually and we indicate which problems are still open. The main open question however is to show that any of two of these six hypotheses are equivalent.

We should note that probably all of the six hypotheses are false since all of them imply that $\text{NP} \subseteq P/\text{poly}$ and this on its turn implies that the polynomial time hierarchy collapses to its second level [KL80].

Until recently no oracles were known that showed that any of these hypotheses are different from each other. However recent progress has been made in this direction (see Section 7).

## 2 Preliminaries

We assume the reader familiar with basic notions of computation and complexity theory as can be found e.g. in [HU79, BDG88, BDG90, GJ79] and many other textbooks.

Central to the six hypotheses in this paper however are the following notions, which we will highlight here by separately defining them.

### Definition 2.1
A set $A$ is called $P$-selective if there exists a polynomial time computable function $f$ (called $p$-selector function) such that for any two strings $x$ and $y$, $f(x,y) \in \{x,y\}$ and if $x$ or $y$ is in $A$ then $f(x,y)$ is in $A$.

For a set $A$ we will identify $A$ with its characteristic function. Hence for a string $x$, $A(x) \in \{0,1\}$ and $A(x) = 1$ iff $x \in A$. For two strings $x$ and $y$ and a $P$-selective set $A$, a $p$-selector excludes one of the four possibilities for the string $A(x)A(y)$ (either $01$ or $10$ is impossible). A generalization extends this exclusion to one of the possible settings for the string $A(x_1) \ldots A(x_k)$ for some function $k(n)$. For constant $k$, this notion was called “approximability” of sets (see Beigel et al. [BK95]).

### Definition 2.2
A function $g$ is called an $f$-approximator for a set $A$ if for every $x_1, \ldots, x_m$ with $m \geq f(\max\{|x_1|, \ldots, |x_m|\})$,

\[ g(x_1, \ldots, x_m) \in \{0,1\}^m \]

and

\[ (A(x_1), \ldots, A(x_m)) \neq g(x_1, \ldots, x_m) \]

A set $A$ is then called $f$-approximable if it has an $f$-approximator. $A$ is bounded-approximable, or $A \in \text{bAPP}$ if $A$ is $k$-approximable for some constant $k$.

The notion $f$-approximability was called $f$-membership comparability by Ogihara [Ogi95] who was the first to consider this notion for nonconstant functions. Beigel [Bei87a] uses the term “approximable” to represent $\text{bAPP}$. Sets which are not in $\text{bAPP}$ Beigel calls superterse.

Amir, Beigel and Gasarch [ABG90] show that every $\text{bAPP}$ language is in $P/\text{poly}$. Ogihara [Ogi95] notices that their proof generalizes.

### Theorem 2.3
(Amir-Beigel-Gasarch-Ogihara)
If $A$ is $f(n)$-approximable for any polynomial $f(n)$ then $A$ is in $P/\text{poly}$.

We use the function $F_{\text{SAT}}$ which on input $\phi_1, \ldots, \phi_n$, returns a string $x \in \{0,1\}^n$, where $x_i = 1$ iff $\phi_i \in \text{SAT}$. We will also need classes of functions
that are computable by queries to \textsc{Sat}. Depending on the number of queries and the type of oracle access these are defined as follows.

**Definition 2.4** A function \(f\) is in \(\text{FP}^{\text{NP}}\) if there exists a polynomial time bounded oracle machine \(M\) that computes \(f\) with non-adaptive queries to some language in \(\text{NP}\).

Note that \(\text{F}_{\text{SAT}}\) is \(\text{FP}^{\text{NP}}\) complete. A set is sparse if there exists a polynomial \(p\) such that for each length \(n\) it contains at most \(p(n)\) strings. Let \(\text{SPARSE}\) denote the class of all sparse sets.

A truth-table reduction from \(A\) to \(B\) is disjunctive \((A \leq_{\text{dt}} B)\) if it accepts if one of it queries is in \(B\).

**Definition 2.5** A function \(f\) is in \(\text{FP}^{\text{NP}[\log]}\) if there is a polynomial time bounded oracle machine that computes \(f\) using \(O(\log n)\) (adaptive) queries to some language in \(\text{NP}\).

**Definition 2.6** Let \(Q\) denote a boolean predicate. we define the set \(\text{Unique-SAT}_Q\) as follows.

For any formula \(x\)

\[
\text{Unique-SAT}_Q(x) = \begin{cases} 
0 & \text{if } x \notin \text{SAT} \\
1 & \text{if } x \text{ has 1 satisfying assignment} \\
Q(x) & \text{Otherwise}
\end{cases}
\]

If there exists a predicate \(Q\) such that \(\text{Unique-SAT}_Q\) is polynomial time computable then we will say “Unique-SAT \(\in \text{P}\).”

The notion of bounded nondeterminism was introduced by Kintala and Fischer in [KF80].

**Definition 2.7** Let \(f\) be any function. We define \(\text{NP}(f(n)) = \{L \mid L \subseteq \{0,1\}^* \text{ and there is a constant } c \text{ such that } L \text{ is accepted by a polynomial time bounded Turing machine making at most } f(n) \text{ c-ary nondeterministic moves}\}\)

Kintala and Fischer denote \(\text{NP}(f(n))\) as \(P_{f(n)}\).

**Definition 2.8** A function \(f(x)\) is \(h(n)\)-enumerable iff there exists a polynomial-time computable function \(g(x) = \{y_1, \ldots, y_{h(n)}\}\) such that for every \(x\), \(f(x) \in g(x)\). A function \(f(x)\) is poly-enumerable \(f(x)\) is \(n^c\) enumerable for some \(c\).

There is a very useful connection between \(\text{FP}^{\text{NP}} = \text{FP}^{\text{NP}[\log]}\) and the enumerability of \(\text{F}_{\text{SAT}}\) [Bei87a].

**Lemma 2.9** (Beigel) \(\text{FP}^{\text{NP}} = \text{FP}^{\text{NP}[\log]}\) if and only if \(\text{F}_{\text{SAT}}\) is poly-enumerable.

**Proof** \(\text{FP}^{\text{NP}} = \text{FP}^{\text{NP}[\log]}\) so by assumption it is in \(\text{FP}^{\text{NP}[\log]}\). There are polynomially possible answers for the oracle queries of the \(\text{FP}^{\text{NP}[\log]}\) machine. Cycling through them yields an enumeration of \(\text{F}_{\text{SAT}}\).

(\(\text{F}_{\text{SAT}}\) is polynomial enumerable \(\Rightarrow \text{FP}^{\text{NP}} = \text{FP}^{\text{NP}[\log]}\))

On input \(\phi_1, \ldots, \phi_i\) each of size at most \(n\) one can enumerate \(n^c\) vectors \(b_1, \ldots, b_{n^c}\) such that \(b_i = \text{F}_{\text{SAT}}\) for some \(i\). Next one can use binary search to some suitable oracle in \(\text{NP}\) to find \(b_i\), using \(\log(n^c) + 1\) queries. \(\Box\)

We will need the following definition of the dimension of a family of sets, called Vapnik-Chervonenkis dimension [VC71]:

**Definition 2.10** Given a family of sets \(\mathcal{F}\) the Vapnik-Chervonenkis dimension of \(\mathcal{F}\) or VC-dimension is the largest number \(d\) such that there exists a set \(A\) with \(|A| = d\) and \(|\{A \cap F \mid F \in \mathcal{F}\}| = 2^d\). If such a \(d\) does not exist the VC-dimension of \(\mathcal{F}\) is \(\infty\).

Sauer [Sau72] and independently Shelah [She72] proved the following lemma. Sauer notes that Paul Erdős originally posed this as a question.

**Lemma 2.11** If \(\mathcal{F}\) is a family of sets with VC-dimension at most \(d\) then for any set \(A\) with \(|A| = n\):

\[
|\{A \cap F \mid F \in \mathcal{F}\}| \leq \sum_{i=0}^{d} \binom{n}{i}
\]

For \(n \geq d \geq 1\), \(\sum_{i=0}^{d} \binom{n}{i}\) is bounded by \(n^d + 1\). Moreover the proof of Lemma 2.11 is constructive: Suppose we have a polynomial-time algorithm that on \(S = x_1, \ldots, x_{d+1}\) computes a subset of \(S\) that is not in \(\{A \cap F \mid F \in \mathcal{F}\}\). Lemma 2.11 gives us a polynomial-time algorithm to compute \(\{A \cap F \mid F \in \mathcal{F}\}\) in time polynomial in \(n\) and the sizes of the elements of \(A\).

## 3 Relations

In this section we will show which of the six hypotheses implies any of the others. The relations are given in Figure 1.

**Theorem 3.1** \(P = \text{NP} \Rightarrow \text{SAT} \leq_{\text{tt}} \text{Psel}\).

**Proof** If \(P = \text{NP}\) then \(\text{SAT}\) is in \(P\) and reduces to any set. \(\Box\)
Theorem 3.2 \( \text{SAT} \leq^p_{tt} \text{Psel} \Rightarrow \text{SAT} \leq^p_{tt} \text{bAPP} \)

**Proof**: Note that every \( \text{P} \)-selective set is 2-approximable. \( \square \)

Theorem 3.3 \( \text{SAT} \leq^p_{tt} \text{bAPP} \Rightarrow \text{FP}^\text{NP} = \text{FP}^\text{NP}[\text{gc}] \)

We first prove the following lemma due to Beigel [Bei87a, Bei87b].

**Lemma 3.4 (Beigel)** If \( A \) is \( k \)-approximable then there exists a function \( f \) which computes for any \( n \) numbers \( x_1, \ldots, x_n \) a set of at most \( \sum_{i=0}^{k-1} \binom{n}{i} \) vectors from \( \{0,1\}^n \) which contains \( F^A_n(x_1, \ldots, x_n) \). Moreover \( f \) runs in time polynomial in \( n \) and the size of the largest string in \( x_1, \ldots, x_n \).

**Proof**: Let \( g \) be the function that \( k \)-approximates \( A \). Define the following family of sets:

\[ \mathcal{F} = \{ B \mid g \text{ is a } k\text{-approximator for } B \} \]

It follows that the VC-dimension of \( \mathcal{F} \) is at most \( k - 1 \). We then apply the constructive version of Lemma 2.11. \( \square \)

We now give the proof of Theorem 3.3.

**Proof**: Let \( M \) witness the fact that \( \text{SAT} \) truth-table reduces to a \( k \)-approximable set \( A \). Let \( f \in \text{FP}^\text{NP} \) via machine \( M_f \). On input \( x \), \( M_f \) computes the following queries \( q_1, \ldots, q_l \) to \( \text{SAT} \), for \( l \) some polynomial. Next reduce each of these queries to \( A \) with \( M \), yielding a set of queries \( q'_1, \ldots, q'_l \), for \( l' \) a polynomial. Next we apply Lemma 3.4 to generate \( l'^k \) many different vectors, containing \( F^A_{l'}(q'_1, \ldots, q'_l) \). From these vectors one can generate \( l^k \) many vectors containing \( F^\text{SAT}(q_1, \ldots, q_l) \). \( \text{FP}^\text{NP} = \text{FP}^\text{NP}[\text{gc}] \) follows from Lemma 2.9. \( \square \)

The following theorem is implicit in [Bei88, Tod91b].

**Theorem 3.5 (Beigel-Toda)** \( \text{FP}^\text{NP} = \text{FP}^\text{NP}[\text{gc}] \Rightarrow \text{Unique-SAT} \) is in \( \text{P} \)

**Proof**: We have to show that there is a polynomial time algorithm that tells formulae with exactly one satisfying assignment apart from ones that are unsatisfiable. Consider the function \( f(\phi) \) that on input \( \phi \) with variables \( x_1, \ldots, x_k \) returns \( b_1 \ldots b_k \) such that \( b_i = 1 \) iff there is a satisfying assignment to \( \phi \) with \( x_i = 1 \). This function is in \( \text{FP}^\text{NP} \) and hence, by assumption in \( \text{FP}^\text{NP}[\text{gc}] \). Suppose we are given a formula \( \phi \) with exactly \( l \) satisfying assignments. Then \( f \) will return exactly this assignment. Since there are only polynomial many possible answers to the \( \log(n) \) queries to \( \text{SAT} \), one can enumerate all the possible values of \( f \) in \( \text{P} \). We can check that one of the generated values is indeed a satisfying assignment to \( \phi \). On the other hand if \( \phi \) was unsatisfiable we would not have generated a satisfying assignment, since none exists. \( \square \)

**Theorem 3.6** \( \text{FP}^\text{NP} = \text{FP}^\text{NP}[\text{gc}] \Rightarrow \text{SAT} \) is \( O(\log(n)) \)-approximable.

**Proof**: By Lemma 2.9 we have that \( F^\text{SAT} \) is \( m^c \) enumerable for some \( c \) where \( m \) is the input length of \( F^\text{SAT} \). Given any \( 2e\log(n) \) formulae \( \phi_1, \ldots, \phi_{2e\log(n)} \) each of size at most \( n \). The size of these \( 2e\log(n) \) formulae is bounded by \( 2e\log(n) \times n \) and thus \( F^\text{SAT}(\phi_1, \ldots, \phi_{2e\log(n)}) \) is \( 2e^2(2e\log(n) \times n) < n^{c+1} \) enumerable. Thus one of the \( n^{2e} \) vectors for \( F^\text{SAT} \) has not been enumerated. \( \square \)

### 4 Selective and Approximable

The question whether sets that have simple structure could be hard for \( \text{NP} \) dates back to the Berman-Hartmanis conjecture [BH77] and subsequent work by Mahaney for sparse sets [Mah82]. Following sparse sets, the first sets of simple structure to be considered were the \( \text{P} \)-selective sets introduced by [Sel79].

\( \text{P} \)-selective sets, though of arbitrary complexity, are structurally simple sets. The \( \text{p} \)-selector function induces an ordering that reduces the number of possible "membership configurations" of two strings. For a \( \text{P} \)-selective set \( A \) and two strings \( x \) and \( y \) either \( x \in A \land y \not\in A \land x \not\in A \land y \not\in A \) is ruled out.
by the p-selector. This property makes P-selective sets structurally as simple as being Turing equivalent to tally sets [Sel82]. Generalizing the structural restriction: “Not all $2^n$ membership configurations of $n$ strings are possible” has induced many related notions. Among the many notions that pertain to the idea are: P-selective sets [Sel79, HHN+95], near-testable sets [GHJY91], k-approximable sets (see below), (a, b)$_p$-recursive sets [KS91], Easily countable sets [HN93], Cheatable sets [Beigel97a, BGGO93], (a, b)$_p$-verbose sets [BKS], and Membership comparable sets [Ogihara95].

Because of the structural relation between P-selective sets and sparse sets, one might not be too surprised that hardness of P-selective sets for NP is as unlikely as hardness for NP of sparse sets. It is quite easy to see that SAT itself cannot be P-selective unless P = NP. Buhrman and Torenvliet [BT96a] showed that SAT cannot be 1-tt reducible to a P-selective set.

Toda [Toda91a], building upon insights provided by Ko [Ko83], proved that in the special case of the existence of only one satisfying assignment, reduction to a P-selective set would imply polynomial time decidability. In fact Toda’s results hold for the more general k-approximable sets. In this section we cite all results for k-approximable sets. Since P-selective sets are k-approximable sets with $k = 2$, all these results also hold for P-selective sets. Similar ideas were obtained independently by Beigel [Beigel88].

Theorem 4.1 (Beigel-Toda)

1. P = UP if and only if UP $\leq^P_{tt}$ bAPP.

2. Unique-SAT $\in$ P if and only if Unique-SAT$_Q$ $\leq^P_{tt}$ bAPP for some Q.

3. P = NP if and only iff $\Delta^P_2$ $\leq^P_{tt}$ bAPP

4. P = PSPACE if and only if PSPACE $\leq^P_{tt}$ bAPP.

5. EXP $\leq^P_{tt}$ bAPP

The Turing reduction of bAPP sets to sparse sets (Theorem 2.3) allows us to apply the famous Karp-Lipton theorem [KL80] showing a collapse of the polynomial-hierarchy if SAT is Turing-reducible to a sparse sets.

Theorem 4.2 (Karp-Lipton) If SAT $\leq^P_T$ bAPP then PH = $\Sigma^P_2$

or in its currently sharpest form proved in [BCG+96, KW95].

Theorem 4.3 (BCGKTKW) If SAT $\leq^P_T$ bAPP then PH = ZPP$^{NP}$

Both directions of strengthening the consequence of SAT $\leq^P_T$ bAPP and weakening the reduction type $r$ in SAT $\leq^P_r$ bAPP $\Rightarrow$ P = NP are currently the subject of active research. Of course in the present context the latter type is the more interesting. In 1994 a major breakthrough was achieved by three independent sets of authors: Beigel, Kummer and Stephan [BK895], Agrawal and Arvind [AA96] and Ogihara [Ogihara95].

Theorem 4.4 (AABKOS) If SAT $\leq^P_{tt}$ bAPP then P = NP

Or in its currently strongest form

Theorem 4.5 (AABKOS) If SAT $\leq^P_{tt}$ bAPP

Theorem 4.6 If SAT $\leq^P_{tt}$ bAPP

Proof: Note that in Lemma 3.4 the number of vectors is actually bounded by $k \times n^b$ 1. Hence if we have $r \log n$ formulae $\phi_1, \ldots, \phi_{r \log n}$ we can reduce these to a $k$-approximable set $A$ via a reduction that produces $n^\alpha$ queries for a total of $(r \log n)n^{\beta} < n^{\beta}$ where $\beta < \frac{1}{\log n}$. Applying Lemma 3.4 gives $(r \times n^{\beta})^k$ 1 vectors including the characteristic vector of these formulae. Hence if $1 > r > \frac{1}{k \log n}$ we can exclude at least one possibility, which means that SAT is $r \log n$-approximable.

We can then apply the following result from [AA96, BK895, Ogi95].

Theorem 4.7 (AABKOS) If SAT is $r \log n$-approximable for some $r < 1$ then P = NP.

To give a flavor of the proof we prove the following weaker result.

Theorem 4.8 If SAT is 2-approximable, then P = NP.

Proof: Given a formula $\phi$, apply the standard self-reduction to produce four formulae $\phi_1, \phi_2, \phi_3, \phi_4$ with the property that $\phi$ is satisfiable iff at least one
of these formulae is satisfiable. Now let \( f \) be a 2-
approximator and let \( f(\phi_1 \lor \phi_2, \phi_1 \lor \phi_3) = (b_1, b_2) \).
If \( b_1 = b_2 = 0 \) then \( \phi \) is satisfiable and we’re done. If \( (b_1, b_2) \) is \( (1, 0) \) then \( \phi_2 \) can not be the only satisfiable
formula. If \( (b_1, b_2) = (0, 1) \) then \( \phi_3 \) can not be the
only satisfiable formula. Finally, if \( (b_1, b_2) = (1, 1) \) then \( \phi_1 \) is not satisfiable.

In all cases one formula in the self-reduction can be
discarded and the corresponding branch in the self-
reduction tree ends. Hence the self-reduction can be
decided trivially.

A polynomial (even fixed) number of queries in
Theorem 4.5 is not yet in sight, nor does the proof
technique seem to be extensible to obtain such a result.
On the other hand there is no known oracle where \( P \neq NP \) and \( SAT \leq^P \text{Psel} \).

The notion of \( P \)-selectivity has been extended to other
types of selector functions ([HHN+95]) for these
(mostly nondeterministic) selector types similar results
are known. These are however outside the scope of
this paper.

The value \( r < 1 \) seems to be a real bottleneck of the
technique ([Ogi95]) used for the proof, but on the other hand no oracle is known where \( P \neq NP \) and \( SAT \) is \( O(\log n) \)-approximable.

5 \( \text{FP}^\text{NP} = \text{FP}^\text{NP}[\log] \)

At first glance one might think that \( \text{FP}^\text{NP} = \text{FP}^\text{NP}[\log] \) since this is true for the language classes:
\( \text{P}^\text{NP} = \text{P}^\text{NP}[\log] \) [BH91, Wagu90]. Indeed this result
yields that \( \text{FP}^\text{NP} = \text{FP}^\text{NP}[\log] \) when only functions
are considered that compute \( \log(n) \) output bits (i.e.
functions from \( \{0,1\}^n \rightarrow \{0,1\}^{O(\log(n))} \}). However
\( \text{FP}^\text{NP} = \text{FP}^\text{NP}[\log] \) implies \text{Unique-SAT} in \( P \)
and this implies that the polynomial hierarchy collapses
(see Section 6). For overview papers on functions
classes and related problems see [JTN95, JT97, Sel96].

In Lemma 2.9 we saw that \( \text{FP}^\text{NP} = \text{FP}^\text{NP}[\log] \)
is equivalent to \( \text{F}_{\text{SAT}} \) being polynomial enumerable.
We can use these ideas to get equivalences of \( \text{FP}^\text{NP} = \text{FP}^\text{NP}[\log] \)
to many other hypotheses.

Theorem 5.1 The following are equivalent:

- \( \text{FP}^\text{NP} = \text{FP}^\text{NP}[\log] \)

- \( \text{FP}^\text{NP} \subseteq \text{FP}^X[\log] \) for some oracle \( X \). [Bei88]

- \( \text{F}_{\text{SAT}} \) is polynomial enumerable.

- Every \( \text{NPSV} \) function is polynomial enumerable.

where \( \text{NPSV} \) is the class of single-valued nondeter-
nomistic functions (see [Sel96]).

Some progress has been made on showing the
equivalence with \( P = NP \). Jenner and Toran [JT95] showed that \( \text{FP}^\text{NP}[\log] \) implies that \( \text{SAT} \)
can be computed in less than \( 2^n \) time. They also showed that languages recognized by nondeterministic
polynomial time machines that make \( \log k(n) \) non-
deterministic moves are in \( P \).

Theorem 5.2 (Jenner-Toran) If \( \text{FP}^\text{NP}[\log] = \text{FP}^\text{NP}[\log] \) then

1. \( \text{NP} \subseteq \text{DTIME}(2^{O(1/\log(\log(n)))}) \).

2. \( \text{NP}(\log(n)) \subseteq P \).

Buhrman and Fortnow showed that the \( \text{FP}^\text{NP}[\log] \) question can be phrased as a question on
resource bounded Kolmogorov complexity [BF97].

Theorem 5.3 (Buhrman-Fortnow) The following are equivalent:

1. \( \text{CND}^{\log y(x \mid y)} \leq O^{\log(x)} + O(\log(\log(x))) \).

2. \( \text{CND}^{\log y(x \mid y)} \leq O^{\log(x)} + O(\log(\log(x))) \).

3. \( \text{FP}^\text{NP} = \text{FP}^\text{NP}[\log] \).

The connection with Kolmogorov complexity enables
one to use Theorem 5.2 to prove:

Theorem 5.4 (Buhrman-Fortnow) If \( \text{FP}^\text{NP}[\log] = \text{FP}^\text{NP}[\log] \) then the class of languages accepted by
nondeterministic polynomial time machines that have
at most \( 2^{\log^\beta(n)} \) accepting paths on inputs of length \( n \)
is included in \( P \).

On the other hand it follows from [Ogi95] that

Theorem 5.5 If \( \text{FP}^\text{NP}[\alpha \log n] \subseteq \text{FP}^\text{NP}[\alpha \log n] \) for some
\( 1 > \beta > \alpha \) then \( P = NP \).

All the above results have not established the
equivalence with \( P = NP \). We note here that in
order to obtain an equivalence it is sufficient to prove
that \( \text{FP}^\text{NP} = \text{FP}^\text{NP}[\log] \Rightarrow P^{NP} = P \) by the
following theorem.

Theorem 5.6

\( P^{NP} = P \) and \( \text{FP}^\text{NP} = \text{FP}^\text{NP}[\log] \implies P = NP \)
Proof: If $P^{NP} = P^{NP|log}$ then the leftmost satisfying assignment can be computed in $FP^{NP|log}$ and hence via assumption in $FP^{NP[log]}$. We can then cycle through all the possible oracle queries as in the proof of Lemma 2.9 and find the assignment in $FP$. □

Corollary 5.7 If $NP = coNP$ and $FP^{NP} = FP^{NP[log]}$ then $P = NP$.

This argument shows that it is actually sufficient to prove that $FP^{NP} = FP^{NP[log]}$ implies that some satisfying assignment can be found in $FP^{NP}$. See [WT93, BT96a] for this question. Watanabe and Toda show that relative to a random oracle it is the case that some satisfying assignment can be found in $FP^{NP}$. However relative to a random oracle all of the six hypotheses fail (see Section 7).

6 Unique-SAT is in P

The hypotheses “Unique-SAT is in P” is a promise problem. It states the existence of a polynomial time algorithm that, under the promise that a formula has either no or a single satisfying assignment, decides whether this formula is satisfiable. Valiant and Vazirani showed that SAT is randomly reducible to Unique-SAT$_Q$ for any predicate $Q$.

Theorem 6.1 (Valiant-Vazirani) There is a polynomial time randomized procedure that given a formula $\phi$ of length $n$ produces a list of $n^i$ formulae $\phi_1, \ldots, \phi_{n^i}$ with the property that:

- $\phi \in SAT$ then with probability $(1 - 2^{-n})$ there is an $i$ such that $\phi_i$ has exactly 1 satisfying assignment.

- $\phi \notin SAT$ then for all $i$, $\phi_i \notin SAT$.

Theorem 6.1 is the key to show that Unique-SAT in P implies that NP = R.

Theorem 6.2 (Valiant-Vazirani) If Unique-SAT is in P then NP = R and the polynomial hierarchy collapses.

Proof: Given a formula $\phi$, use Theorem 6.1 to randomly produce a list of $n^i$ formulae. Next for each of these formula $\phi_i$, use that Unique-SAT is in P algorithm to try generate a satisfying assignment for $\phi_i$. This can be done using the selfreducibility of SAT (See [BDG89] for details). If a satisfying assignment has been found accept $\phi$ and if for every $i$ no assignment was found reject. The fact that the polynomial hierarchy collapses follows since it is known that $R \in P/poly$ and $NP \in P/poly$ implies that the polynomial hierarchy collapses [KL85]. □

Another consequence of Unique-SAT $\in P$ is that FewP, the class of languages that are accepted by nondeterministic polynomial time Turing machines that have at most a polynomial number of accepting paths, is in P. This was essentially proved in Toda’s paper [Tod91a].

Theorem 6.3 (Toda) If Unique-SAT $\in P$ then FewP = P

Fortnow and Kummer [FK96] showed that the assumption that Unique-SAT is in P is linked to resource bounded Kolmogorov complexity:

Theorem 6.4 (Fortnow-Kummer) Unique-SAT is in P if and only if

$$CD^{poly}(x \mid y) \leq C^{poly}(x \mid y) + O(\log |x|)$$

We mentioned before that all the six hypotheses imply that NP $\subseteq P/poly$. This is equivalent to SAT $\subseteq^p_{tt}$ SPARSE. Ogihara and Watanabe [OW91] showed that if SAT $\subseteq^p_{tt}$ SPARSE then P = NP.

With a slightly weaker hypothesis Cai, Naik, and Sivakumar [CNS96] proved the following:

Theorem 6.5 (Cai-Naik-Sivakumar) If SAT $\subseteq_{tt}^p$ SPARSE then Unique-SAT $\in P$.

7 Relativization

To understand the difficulty of proving results about the six hypotheses, it is useful to turn to the theory of relativization. All of the results in this paper relativize, i.e., hold if every machine has access to the same oracle. See Fortnow [For94] for a discussion of the importance and limitations of relativization results.

In order to relativize some of the questions related to the six hypotheses we need a relativized version of SAT developed by Goldsmith and Joseph [GJ93]. Relativized SAT$^A$ has several extra predicates $A_0, A_1, \ldots$ such that $A_m(x_1, \ldots, x_m)$ has the property that

$$x_1 \ldots x_m \in A \iff A_m(x_1, \ldots, x_m)$$

For every oracle $A$, SAT$^A$ has the following properties:
1. SAT$^4$ is NP$^4$ complete.

2. Whether $\phi$ is in SAT$^4$ depends only on strings in $A$ of length less than $|\phi|$.

Baker, Gill and Solovay [BGS75] in their seminal paper on relativization give an oracle $A$ such that P$^A = NP^A$. Relative to this oracle all of the six hypotheses are true.

All of the six hypotheses imply that NP has polynomial-size circuits (Theorems 2.3 and 6.2) and thus $P^2 = \Sigma^2_2$. Baker and Selman [BS79] give a relativized world where $P^2 \neq \Sigma^2_2$ and thus all of the six hypotheses are false. The six hypotheses also fail relative to generic and random oracles.

Creating relativized worlds where some of the six hypotheses are true while others fail appears considerably more difficult. Recently Beigel, Buhrman, and Fortnow [BBF97] have made some progress in this direction.

**Theorem 7.1 (Beigel-Buhrman-Fortnow)**

There exists an oracle $A$ such that

$$P^A = \oplus P^A \neq NP^A = \text{EXP}^A$$

One can use $\oplus P$ to solve Unique-SAT questions. Toda [Toda91b] uses this fact in his celebrated proof that $PH \subseteq P^\#P$. Combined with Corollary 5.7, this gives us the following conclusion.

**Corollary 7.2 (Beigel-Buhrman-Fortnow)**

There exists a relativized world where Unique-SAT is in P and $FP_{||}^{NP} \neq FP_{||}^{NP[\log n]}$.

Other relativized separations of the six hypotheses remain important open problems.

We can get some more relativized separations if we weaken some of the hypotheses.

**Theorem 7.3** Let $f(n) = \omega(\log n)$. There exists a relativized world where SAT is $f(n)$-approximable but $P \neq NP$.

The proof uses ideas from Homer and Longpré [HL94].

**Proof:** First start with an oracle that makes $P = \text{PSPACE}$. We build a new oracle on top of this one.

Define the language $L(A) = \{1^n \mid \text{There exists a string } x \text{ of length } n \text{ in } A \}$

For all $A$ we have $L(A) \in \text{NP}^A$.

We diagonalize $P$ from $\text{NP}$ in the same way as Baker, Gill and Solovay [BGS75]. However we will guarantee that we put at most one string in at every length and the string we put in will be among the first $2^{f(n)} - 2$ strings of length $n$. Since $2^{f(n)} - 2$ is greater than every polynomial, we are able to use the Baker, Gill and Solovay diagonalization technique.

Suppose we are given $f(n)$ formulae. Note there are only $2^{f(n)} - 1$ possibilities for the oracle strings of length $n$ (the oracle could be empty). Using our $P = \text{PSPACE}$ base oracle we can compute some possible setting of the $f(n)$ formulae that cannot occur. □

Generalizing these techniques we get additional relativized worlds.

**Theorem 7.4** Let $f(n) = \omega(\log n)$. There exists a relativized world where

1. $P \neq NP$.

2. SAT is $f(n)$-Turing reducible to a $P$-selective set and thus a $k$ approximable set for $k \geq 2$.

3. $FP^{NP} \subseteq FP^{NP}[f(n)]$.

**8 Open Problems**

In this section we summarize the open problems. For most of these problems it is not even known whether there are relativized worlds where they fail, so relativized results are welcome too.

The main open problems are the following.

1. Show that any two of the six hypotheses are equivalent to each other.

2. Show that SAT is $O(\log n)$-approximable implies Unique-SAT is in P or vice versa.

3. Show that if $FP_{\Sigma_2^{\log n}} \mu \subseteq FP^{NP[\log n]}$ then $P = NP$. This is the stronger version of the hypothesis $FP_{||}^{NP} = FP_{||}^{NP[\log n]}$.

4. Show that if there is a $\Sigma_2^p$-complete set that is $O(\log n)$-approximable then $P = NP$. (Similar for PSPACE).

5. (related to Section 6) Is $\Sigma_2^p = UP^{NP}$.

6. Show that if SAT $\leq_{\text{det}} SPARSE$ then $P = NP$.

**Acknowledgments**

We like to thank Richard Beigel for proof reading an earlier version of our paper. We thank Dieter van Melkebeek for helpful discussions and Sophie Laplante for useful legwork.
References


