NP Might Not Be As Easy As Detecting Unique Solutions

Richard Beigel[*] Harry Buhrman[§] Lance Fortnow[¶]
Lehigh University CWI University of Chicago

Abstract

We construct an oracle A such that

$$\mathbf{P}^A = \oplus\mathbf{P}^A \text{ and } \mathbf{NP}^A = \mathbf{EXP}^A.$$

This relativized world has several amazing properties:

- The oracle A gives the first relativized world where one can solve satisfiability on formulae with at most one assignment yet $\mathbf{P} \neq \mathbf{NP}$.
- The oracle A is the first where
$$\mathbf{P}^A = \mathbf{UP}^A \neq \mathbf{NP}^A = \mathbf{coNP}^A.$$
- The construction gives a much simpler proof than that of Fenner, Fortnow and Kurtz of a relativized world where all the **NP**-complete sets are polynomial-time isomorphic. It is the first such computable oracle.
- Relative to A we have a collapse of $\oplus\mathbf{EXP}^A \subseteq \mathbf{ZPP}^A \subseteq \mathbf{P}^A$/poly.

We also create a different relativized world where there exists a set L in **NP** that is **NP**-complete under reductions that make one query to L but not complete under traditional many-one reductions. This contrasts with the result of Buhrman, Spaan and Torenvliet showing that these two completeness notions for **NEXP** coincide.

1 Introduction

Valiant and Vazirani [VV86] show the surprising power of solving satisfiability on formulae with at most one satisfying assignment or equivalently detecting unique solutions to **NP** problems.

Theorem 1.1 (Valiant-Vazirani) *If one could detect unique solutions to* **NP** *problems then* $\mathbf{R} = \mathbf{NP}$.

The proof of Theorem 1.1 depends heavily on randomization. They leave open whether detecting unique solutions implies $\mathbf{P} = \mathbf{NP}$.

Hypothesis 1.2 *If one can detect unique solutions to* **NP** *problems then* $\mathbf{P} = \mathbf{NP}$.

Theorem 1.1 relativizes. To help us gauge the difficulty of proving a deterministic version of Theorem 1.1 we will show Hypothesis 1.2 fails in a relativized world.

Theorem 1.3 *There exists a relativized world where we can detect unique solutions for* **NP** *problems yet* $\mathbf{P} \neq \mathbf{NP}$.

To prove Theorem 1.3 we consider the set Q consisting of all formulae with an odd number of satisfying assignments. If we can determine membership in Q than we can detect unique solutions. This set Q also has some nice algebraic properties for our proofs. Note the set Q is $\oplus\mathbf{P}$-complete.

In fact we prove a considerably stronger result.

Theorem 1.4 *There exists an oracle A such that* $\mathbf{P}^A = \oplus\mathbf{P}^A$ *and* $\mathbf{NP}^A = \mathbf{EXP}^A$.

Theorem 1.4 has some other important applications. Berman and Hartmanis [BH77] conjectured that all **NP**-complete sets are polynomial-time isomorphic, i.e., for every pair of **NP**-complete sets A and B there exists a polynomial-time computable and invertible bijection reducing A to B.

Finding a relativized world where this isomorphism conjecture held remained open for many years. Homer and Selman [HS92] noted that if $\mathbf{P} = \mathbf{UP}$ and $\mathbf{NP} = \mathbf{EXP}$ then the isomorphism conjecture holds. They created a relativized world where $\mathbf{P} = \mathbf{UP}$ and $\Sigma_2^p = \mathbf{EXP}$. However, even a relativized world where $\mathbf{P} = \mathbf{UP} \neq \mathbf{NP} = \mathbf{coNP}$ seemed much more difficult to prove. Theorem 1.4 is the first to break this barrier.

Corollary 1.5 *There exists an oracle A such that*

$$\mathbf{P}^A = \mathbf{UP}^A \neq \mathbf{NP}^A = \mathbf{coNP}^A = \mathbf{EXP}^A$$

[*] Research done at Yale and the University of Maryland. Address: Elect Eng Comp Science, 19 Memorial Dr W Ste 2, Bethlehem PA 18015-3084, USA. Supported in part by the National Science Foundation under grants CCR-8958528, CCR-9415410 and CCR-9700417 and by NASA under grant NAG 52895. Email: beigel@eecs.lehigh.edu. http://www.eecs.lehigh.edu/~beigel/

[§] Address: CWI, Kruislaan 413, 1098SJ Amsterdam, The Netherlands. Partially supported by the Dutch foundation for scientific research (NWO) by SION project 612-34-002, and by the European Union through NeuroCOLT ESPRIT Working Group Nr. 8556, and HC&M grant nr. ERB4050PL93-0516. Email: buhrman@cwi.nl. http://www.cwi.nl/~buhrman/

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

Fenner, Fortnow and Kurtz [FFK96] used a very different and complicated approach to resolve the relativized isomorphism conjecture. Their oracle is nonconstructive and makes the polynomial-time hierarchy infinite. Theorem 1.4 is the first to fulfill the Homer and Selman approach. The proof is considerably simpler than Fenner, Fortnow and Kurtz and is the first to achieve a constructible oracle and a collapse of the hierarchy.

Corollary 1.6 *There exists a recursive set A such that the isomorphism conjecture holds relative to A. In addition the polynomial-time hierarchy relative to A collapses to* $\mathbf{NP}^A$.

Heller [Hel84] and Kurtz [Kur85] give a relativized world where $\mathbf{EXP} = \mathbf{ZPP}$. Theorem 1.4 gives a stronger collapse.

Corollary 1.7 *There exists an oracle A such that*

$$\oplus\mathbf{EXP}^A = \mathbf{ZPP}^A \subseteq \mathbf{P}^A/poly$$

We can generalize Theorem 1.4 to $\mathbf{Mod}_k\mathbf{P}$ classes on two fronts.

Theorem 1.8 *For every prime q there is an oracle A such that for all k not a power of q*

$$\mathbf{P}^A = \mathbf{Mod}_q\mathbf{P}^A \textit{ and } \mathbf{Mod}_k\mathbf{P}^A = \mathbf{NP}^A = \mathbf{EXP}^A.$$

In particular we get a relativized world where both $\mathbf{P} = \oplus\mathbf{P}$ and $\mathbf{Mod}_3\mathbf{P} = \mathbf{EXP}$.

Homer, Kurtz and Royer [HKR93] show that if L is complete for the class **EXP** under 1-truth-table reductions then L is also **EXP**-complete under the standard many-one reducibility. Buhrman, Spaan and Torenvliet [BST93] show that the same result holds for **NEXP**. We give a give a relativized world where this collapse does not hold for **NP**.

Theorem 1.9 *There exists an oracle B and a language L that is 1-tt-complete for* $\mathbf{NP}^B$ *but not m-complete.*

In Theorem 1.9 and Corollary 1.6 we allow the reductions to access the oracle.

1.1 Relativization

Most of the proofs in computational complexity theory relativize: The results hold even if all machines involved have access to the same arbitrary oracle. Thus our paper shows that the relativized results described in this paper cannot be proven false in the unrelativized world unless one uses nonrelativizing techniques.

The only reasonable use of nonrelativizing techniques so far has been in the area of interactive proof systems [LFKN92, Sha92, BFL91, ALM$^+$92]. Complexity theorists have yet to find many other interesting applications of these techniques. At this time we know no nonrelativizing techniques to handle the questions mentioned in this paper.

For a more thorough discussion on relativization see [For94].

2 Preliminaries

We assume the reader familiar with basic notations in complexity theory and classes such as **P** and **NP**.

We use the class **R** to represent probabilistic polynomial-time computation with one-sided error. The class $\mathbf{ZPP} = \mathbf{R} \cap \mathbf{coR}$ is probabilistic computation with zero-sided error running in expected polynomial time.

We let $\mathbf{EXP} = \mathbf{DTIME}[2^{n^{O(1)}}]$.

Let $\#M$ represent the number of accepting computations of a nondeterministic Turing machine M.

A language L is in **UP** if there exists a polynomial-time nondeterministic Turing machine M such that for all x,

- $x \in L \Rightarrow \#M(x) = 1$
- $x \notin L \Rightarrow \#M(x) = 0.$

A language L is in $\oplus\mathbf{P}$ if there exists a polynomial-time nondeterministic Turing machine M such that for all x,

- $x \in L \Rightarrow \#M(x)$ is odd.
- $x \notin L \Rightarrow \#M(x)$ is even.

The class $\oplus\mathbf{EXP}$ has the same definition as $\oplus\mathbf{P}$ except we allow M to use $2^{n^{O(1)}}$ time.

We can generalize $\oplus\mathbf{P}$ by allowing different modula. A language L is in $\mathbf{Mod}_k\mathbf{P}$ if there exists a polynomial-time nondeterministic Turing machine M such that for all x,

- $x \in L \Rightarrow \#M(x) \bmod k \neq 0.$
- $x \notin L \Rightarrow \#M(x) \bmod k = 0.$

Without loss of generality we can replace the "$\neq 0$" in the first condition by "$= 1$" (see [Bei91]).

We can *detect unique solutions* if for every nondeterministic Turing machine M there exists a language A in **P** such that for all x,

- $\#M(x) = 0 \Rightarrow x \notin A$
- $\#M(x) = 1 \Rightarrow x \in A$

We put no conditions on A if $\#M(x) > 1$. Note that detecting unique solutions is a stronger restriction than $\mathbf{P} = \mathbf{UP}$.

Since the famous reduction of Cook [Coo71] preserves the number of solutions, detecting unique solutions is equivalent to solving satisfiability on formulae with at most one satisfying assignment.

In this paper we also consider different reductions between sets. Traditionally as defined by Karp [Kar72], we say a set A is complete for a class $\mathcal{C}$ if A is in $\mathcal{C}$ and for all $L \in \mathcal{C}$, there exists a function f in **FP** such that x is in L if and only if $f(x)$ is in A.

To distinguish completeness notions we often use the term m-completeness for this Karp definition. We say A is 1-li-complete if f can always be a length increasing injection.

Cook [Coo71] uses the notion of Turing-completeness where instead of a function f we have a polynomial-time Turing machine M such that x is in L if and only if $M^A(x)$ accepts. We say A is 1-tt-complete if M can make only one query to A.

We say two sets A and B are isomorphic if A m-reduces to B via a polynomial-time computable function that is one-to-one, onto and polynomial-time invertible.

3 Detecting Unique Solutions

Valiant and Vazirani [VV86] show how to randomly map satisfiable formula to those with unique satisfying assignments.

Lemma 3.1 *There exists a probabilistic polynomial-time function f such that for all boolean formulae ϕ in n variables*

- *If $\phi \notin$* **SAT** *then $f(\phi)$ is never satisfiable.*
- *If $\phi \in$* **SAT** *then with probability at least $1/4n$, $f(\phi)$ has exactly one assignment.*

If one takes n^2 independent applications of $f(\phi)$ for some satisfiable ϕ then with extremely high probability one of these outputs will have a unique assignment. Theorem 1.1 follows directly from Lemma 3.1.

Valiant and Vazirani's construction creates random subspaces of the assignments. Mulmuley, Vazirani and Vazirani [MVV87] give an alternate proof looking at the maximal weighted cliques after putting random weights on the edges. Buhrman and Fortnow [BF97] show how Lemma 3.1 follows from earlier work by Sipser [Sip83] on Kolmogorov complexity. Gupta [Gup97] gives a construction for Lemma 3.1 that improves the probability to a constant if we only require $f(\phi)$ to have an odd number of assignments.

Attempts at a relativized counterexample to Hypothesis 1.2 have a long history. Rackoff [Rac82] gives a relativized world where **P** $=$ **UP** $\neq$ **NP** but the proof heavily uses the fact that the **UP** machines must have one accepting path for all inputs.

An easy application of Lemma 3.1 allows one to randomly find a satisfying assignment of a formula making nonadaptive queries to **SAT**. Buhrman and Thierauf [BT96] give a relativized world where this fails deterministically.

Theorem 1.4 gives the first relativized counterexample to Hypothesis 1.2. In fact Theorem 1.4 shows a stronger result. Buss and Hay [BH91] and Wagner [Wag90] show that languages computable with a polynomial number of nonadaptive queries to **SAT** are equivalent to those computable with $O(\log n)$ adaptive queries to **SAT**. For functions the equivalence would imply we can distinguish unique solutions (see [BFT97]). Theorem 1.4 gives a relativized world where the converse fails.

Corollary 3.2 *There exists a relativized world where we can distinguish unique solutions but there is a function computable with a polynomial number of nonadaptive queries to* **SAT** *but not with* $O(\log n)$ *adaptive queries.*

Proof: Combine Theorem 1.4 with the fact that for all relativized worlds, if all functions computable with a polynomial number of nonadaptive queries to **SAT** are equivalent to those computable with $O(\log n)$ adaptive queries to **SAT** and **NP** $=$ **coNP** then **P** $=$ **NP** (see [BFT97]). □

4 The Isomorphism Conjecture

Berman and Hartmanis [BH77] consider whether all **NP**-complete sets are isomorphic.

Conjecture 4.1 (Berman-Hartmanis) *Every pair of* **NP**-*complete sets are polynomial-time isomorphic.*

Berman and Hartmanis [BH77] give a powerful tool to show that sets are isomorphic.

Lemma 4.2 (Berman-Hartmanis) *Sets A and B are polynomial-time isomorphic if there exist length-increasing polynomial-time computable and invertible injections from A to B and from B to A.*

Berman and Hartmanis [BH77] use Lemma 4.2 to show that the known natural **NP**-complete sets of the time were all isomorphic. Proving Conjecture 4.1 would imply that **P** $\neq$ **NP** since otherwise we would have finite **NP**-complete sets isomorphic to infinite ones.

The Isomorphism Conjecture has been the subject of considerable research. We recommend the surveys by Joseph and Young [JY90] and Kurtz, Mahaney and Royer [KMR90].

Berman [Ber77] showed that every m-complete set for **EXP** is complete via one-to-one and length-increasing reductions. Grollmann and Selman [GS88] show that **P** $=$ **UP** is equivalent to every length-increasing polynomial-time computable injection being polynomial-time invertible. Using these results Homer and Selman [HS92] realized an implication that would imply the isomorphism conjecture.

Lemma 4.3 (Berman-Grollman-Selman-Homer) *If* **P** $=$ **UP** *and* **NP** $=$ **EXP** *then all* **NP**-*complete sets are polynomial-time isomorphic.*

Lemma 4.3 relativizes so Homer and Selman tried to create an oracle relative to which the isomorphism conjecture holds by getting **P** $=$ **UP** and **NP** $=$ **EXP**. They showed the following result.

Theorem 4.4 (Homer-Selman) *There exists an oracle relativize to which* **P** $=$ **UP** *and* $\Sigma_2^p =$ **EXP**.

Theorem 4.4 gives the first relativized world where all Σ_2^p-complete sets are isomorphic.

Later, Fenner, Fortnow and Kurtz [FFK96] used a very different approach to settle the relativized isomorphism conjecture.

Theorem 4.5 (Fenner-Fortnow-Kurtz) *There exists a relativized world where all* **NP**-*complete sets are polynomial-time isomorphic.*

The proof of Fenner, Fortnow and Kurtz requires a complicated, nonconstructive argument using a specialized form of generic oracles with infinite conditions. Relative to their oracle the polynomial-time hierarchy is infinite.

Since for all A, $\mathbf{UP}^A \subseteq \oplus\mathbf{P}^A$, Theorem 1.4 combined with Lemma 4.3 gives us an alternative proof of Theorem 4.5. Our proof is considerably simpler, constructive and collapses the polynomial-time hierarchy to **NP** (Corollary 1.6).

5 Collapsing to ZPP and P/poly

Heller [Hel84] and Kurtz [Kur85] exhibit a relativized world collapsing **EXP** to **ZPP**.

Theorem 5.1 (Heller-Kurtz) *There exists an oracle A such that* $\mathbf{EXP}^A = \mathbf{ZPP}^A$.

If **P** $= \oplus$**P** then by Theorem 1.1 we have that **R** $=$ **NP**. Also by standard padding arguments **P** $= \oplus$**P** implies **EXP** $= \oplus$**EXP**. If also **EXP** $=$ **NP** then we have $\oplus$**EXP** $=$ **R**. Since $\oplus$**EXP** is closed under complement, we have $\oplus$**EXP** $=$ **ZPP**.

This whole argument relativizes. Theorem 1.4 thus gives us a strong improvement of Theorem 5.1 giving an oracle A such that $\oplus\mathbf{EXP}^A = \mathbf{ZPP}^A$ (Corollary 1.7).

Since for all B, $\mathbf{ZPP}^B \subseteq \mathbf{BPP}^B \subseteq \mathbf{P}^B$/poly, we also get that $\oplus\mathbf{EXP}^A \subseteq \mathbf{P}^A$/poly. This gives a complementary result to an oracle C by Heller [Hel86] showing that $\mathbf{EXP}^{\mathbf{NP}^C} \subseteq \mathbf{BPP}^C \subseteq \mathbf{P}^C$/poly.

6 Proof of Main Theorem

In this section we prove Theorem 1.4 showing an oracle A such that $\mathbf{P}^A = \oplus\mathbf{P}^A$ and $\mathbf{NP}^A = \mathbf{EXP}^A$.

Torán [Tor88] constructs the first oracle A such that $\mathbf{NP}^A \not\subseteq \oplus\mathbf{P}^A$ which also follows from Theorem 1.4. Tarui [Tar91] gives an alternate proof of Torán's result using the high degree of the **OR** function over GF[2]. This property of the **OR** function also plays an important role in our proof.

Let M^A be a nondeterministic linear time Turing machine such that the language L^A defined by

$$w \in \mathbf{L}^A \Leftrightarrow \#M^A(w) \bmod 2 = 1$$

is $\oplus\mathbf{P}^A$ complete for every A. We assume without loss of generality that M^A makes at most n queries on any computation path, guesses the answers to all oracle queries and verifies the answers nonadaptively at the end.

Let N^A be a deterministic machine that runs in time 2^n and for all A accepts a language K^A that is $\mathbf{EXP}^A$ complete.

We will construct A such that for all w

$$w \in L^A \quad \Leftrightarrow \quad \langle 0, w, 1^{|w|^2}\rangle \in A \qquad \text{(Condition 0)}$$

$$w \in K^A \quad \Leftrightarrow \quad \exists v\ |v| = |w|^2 \text{ and } \langle 1, w, v\rangle \in A \qquad \text{(Condition 1)}$$

Condition 0 will guarantee that **P** $= \oplus$**P** and Condition 1 will guarantee that **NP** $=$ **EXP**.

We will use the terms 0-strings for all of the strings of the form $\langle 0, w, 1^{|w|^2}\rangle$ and 1-strings for the strings of the form $\langle 1, w, v\rangle$ with $|v| = |w|^2$. All other strings we immediately put in $\overline{A}$.

First we give some intuition for the proof. Condition 0 will be automatically fulfilled by just describing how we set the 1-strings because they force the 0-strings as defined by Condition 0.

Fulfilling Condition 1 requires a bit more care since $N^A(x)$ can query exponentially long 0- and 1-strings. We consider each 1-string $\langle 1, w, v\rangle$ as a variable $y_{\langle w,v\rangle}$ whose value determines whether $\langle 1, w, v\rangle$ is in A. We will show that the computation $N^A(x)$ can be represented by a low-degree polynomial over these variables in the field of two elements. To encode the computation properly we use the fact that the **OR** function has high degree.

We will assign a polynomial p_z over GF[2] to all of the 0-strings and 1-strings z. We ensure that for all z

1. If $p_z = 1$ then z is in A.
2. If $p_z = 0$ then z is not in A.

First for each 1-string $z = \langle 1, w, v\rangle$ we let p_z be the single variable polynomial $y_{\langle w,v\rangle}$.

We assign polynomials to the 0-strings recursively. Note that $M^A(x)$ can only query 0-strings with $|w| \leq \sqrt{|x|}$. Consider an accepting computation path π of $M(x)$ (assuming the oracle queries are guessed correctly). Let $q_{\pi,1}, \ldots, q_{\pi,m}$ be the queries on this path and $b_{\pi,1}, \ldots, b_{\pi,m}$ be the query answers with $b_{\pi,i} = 1$ if the query was guessed in A and $b_{\pi,i} = 0$ otherwise. Note that $m \leq n$.

Let $\mathcal{P}$ be the set of accepting computation paths of $M(x)$. We then define the polynomial p_z for $z = \langle 0, x, 1^{|x|^2}\rangle$ as follows:

$$p_z = \sum_{\pi\in\mathcal{P}} \prod_{i:1\leq i\leq m} (p_{q_{\pi,i}} + b_{\pi,i} + 1) \qquad (1)$$

Remember that we are working over GF[2] so addition is parity.

Setting the variables $y_{\langle w,v\rangle}$ (and thus the 1-strings) forces the values of p_z for the 0-strings. We have set things up properly so the following lemma is straightforward.

Lemma 6.1 *For each* 0*-string* $z = \langle 0, x, 1^{|x|^2}\rangle$ *we have* $p_z = \#M^A(x) \bmod 2$ *and Condition 0 can be satisfied. The polynomial* p_z *has degree at most* $|x|^2$.

Proof: Simple proof by induction on $|x|$. □

We would like to create a polynomial r_x that captures the value of $N^A(x)$. Consider a nondeterministic machine $N'(x)$ that simulates $N^A(x)$ by first guessing the oracle queries and verifying them at the end. Similar to Equation (1), we can sum up over all the paths of the machine. We then define r_x by

$$r_x = \sum_{\pi\in\mathcal{P}} \prod_{i:1\leq i\leq m} (p_{q_{\pi,i}} + b_{\pi,i} + 1)$$

where the terms have similar meaning as in Equation 1. Here we have $m \leq 2^{|x|}$.

Lemma 6.2 *The value* r_x *is exactly* 1 *when* $N^A(x)$ *accepts and* 0 *otherwise. The degree of* r_x *is at most* $2^{3|x|}$.

Proof: Since N is deterministic, N' can have at most one accepting path.

To bound the degree note that the queries of q made by $N^A(x)$ have length at most $2^{|x|}$ so the degree of p_q is bounded by $(2^{|x|})^2 = 2^{2|x|}$. Since $m \leq 2^{|x|}$ this gives us a total degree of $2^{3|x|}$. □

To properly encode to fulfill Condition 1, we need the following lemma about the **OR** function.

Lemma 6.3 *The function* **OR**$(u_1, \ldots, u_m)$ *as a multivariate polynomial over GF[2] requires degree exactly* m.

Proof: Every function over GF[2] has a unique representation as a multivariate multilinear polynomial.

Note that **AND** is just the product so by using De Morgan's laws we can write **OR** as

$$\mathbf{OR}(u_1, \ldots, u_m) = 1 + \prod_{1\leq i\leq m} (1 + u_i). \ \Box$$

First let us discuss how to fulfill condition 1 in isolation. Let us assume that the only variables that r_x depends on are $y_{\langle x,v\rangle}$ for some v. We have two cases.

Case (1): $r_x(\vec{0}) = 0$: We just set all the variables $y_{\langle x,v\rangle} = 0$.

Case (2): $r_x(\vec{0}) = 1$: In this case there must be a $\vec{t} \neq \vec{0}$ such that $r_x(\vec{t}) = 1$. Otherwise $1 - r_x$ computes the **OR** function on the $y_{\langle x,v\rangle}$. But the degree of $1 - r_x$ is at most $2^{3|x|}$ and by Lemma 6.3 the **OR** function has degree $2^{|x|^2}$. We encode using this $\vec{t}$.

However r_x may, of course, depend on variables $y_{\langle w,v\rangle}$ for $w \neq x$. So we must encode r_x in stages.

Stage x: We assume all the variables $y_{\langle w,v\rangle}$ for $w < x$ are encoded. For $s \leq x$ define the polynomials r_s^x as the polynomials r_s where we replace the variables $y_{\langle w,v\rangle}$ for $w < x$ with their encoded value and variables $y_{\langle w,v\rangle}$ for $w > x$ with 0. The only remaining variables of r_s^x are of the form $y_{\langle x,v\rangle}$.

For each $s \leq x$, let $b_s^x = r_s^x(\vec{0})$. There are two cases again.

Case (1): $b_x^x = 0$: In this case just set the variables $y_{\langle x,v\rangle}$ to zero.

Case (2): $b_x^x = 1$. Consider the polynomial

$$R_x = \mathbf{OR}_{s\leq x}(r_s^x + b_s^x).$$

R_x has degree at most

$$2^{|x|+1}(\max_{s\leq x} 2^{3|s|}) = 2^{|x|+1}2^{3|x|} = 2^{4|x|+1}.$$

By the definition of b_s^x we have $R_x(\vec{0}) = 0$. There must be some vector $\vec{t} \neq \vec{0}$ such that $R_x(\vec{t}) = 0$. Otherwise R_x computes the **OR** function over $2^{|x|^2}$ variables.

We use this $\vec{t}$ as our encoding.

Theorem 1.4 follows from the following lemma.

Lemma 6.4 *After all the stages, Condition 1 is fulfilled for every* x.

Proof: We will show by induction that after every stage $\ell \geq x$ the polynomial r_x is properly encoded assuming that all unencoded strings are zero. After stage ℓ for $|\ell| > 2^{|x|}$ all the variables used by r_x have been encoded and will no longer change.

Note that immediately after stage ℓ we have that the value of $r_s^\ell = r_s$ for all $s \leq \ell$ assuming the unencoded string are zero.

Base Case ($\ell = x$): If $b_x^x = r_x^x(\vec{0}) = 0$ then we encode using $\vec{0}$ so Condition 1 is satisfied. If $b_x^x = r_x^x(\vec{0}) = 1$ then since $R_x(\vec{t}) = 0$, $r_x^x(\vec{t}) = b_x^x = 1$ so by using $\vec{t}$ we once again satisfy Condition 1.

Inductive Case ($\ell > x$): In stage $\ell - 1$ we have r_x properly encoded assuming all of the unencoded strings are zero. Therefore $b_x^\ell = r_x^\ell(\vec{0}) = r_x$ immediately before stage ℓ. If in stage ℓ we only encode with $\vec{0}$ then nothing changes. If we use $\vec{t}$, since $R_\ell(\vec{t}) = 0$, we still have $r_x^\ell(\vec{t}) = b_x^\ell = r_x$ as desired. □

7 Extension of Main Theorem

In this section we give a proof sketch showing that for any prime q there is an oracle A such that for all k not a power of q, $\mathbf{P}^A = \mathbf{Mod}_q\mathbf{P}^A$ and $\mathbf{Mod}_k\mathbf{P}^A = \mathbf{NP}^A = \mathbf{EXP}^A$.

For all k and j such that k divides j, $\mathbf{Mod}_k\mathbf{P} \subseteq \mathbf{Mod}_j\mathbf{P}$ (see [Bei91]). Thus we can assume that k is a prime different from q.

To get an oracle A such that $\mathbf{P}^A = \mathbf{Mod}_q\mathbf{P}^A$ and $\mathbf{NP}^A = \mathbf{EXP}^A$ is a simple variation of the proof in Section 6: One works over GF[q] instead of GF[2].

We will encode **EXP** into $\mathbf{Mod}_k\mathbf{P}$ similarly to the way we encoded **EXP** into **NP** in Section 6. We will add the following Condition k for each prime $k \neq q$.

$$w \in K^A \Leftrightarrow |\{v \ : \ |v| = |w|^2 \text{ and } \langle k, w, v\rangle \in A\}| \bmod k = 1.$$

We will fulfill all of the conditions $k \geq 1$ for each string w by a standard dovetailing through pairs (k, w). We will only encode pairs (k, w) where $k < \log\log |w|$ to keep the number of conditions we must fulfill at any length low.

To fulfill condition k without interfering with the other conditions we need to show that the Mod_k function has high degree over GF[q]. We will use the following lemma implicitly proven by Barrington, Beigel and Rudich [BBR94].

Lemma 7.1 (Barrington-Beigel-Rudich) *Let r be a polynomial in binary variables $x_1, \ldots, x_N$. Let q be a prime. Suppose that r satisfies:*

- $r(x_1, \ldots, x_N) \not\equiv 0 \bmod q$ *if* $x_1 + \ldots + x_N = 0$*, and*
- $r(x_1, \ldots, x_N) \equiv 0 \bmod q$ *if* $x_1 + \ldots + x_N$ *is a power of* q

Then the degree of r is at least $N/2q$.

If we have $r(0, \ldots, 0) \not\equiv 0 \bmod q$ and $r(x_1, \ldots, x_N) \equiv 0 \bmod q$ whenever $x_1 + \ldots + x_N \not\equiv 0 \bmod k$ then the degree of r is at least $N/2q$.

This allows us to use the same kind of argument to fulfill Condition $k > 1$ as we used to fulfill Condition 1 in Section 6.

8 1-tt-completeness for NP

Homer, Kurtz and Royer [HKR93] show that every 1-tt-complete set for **EXP** is also m-complete. If one can show that there exists a 1-tt-complete set for **NP** that is not m-complete this would separate **NP** from **EXP**. Theorem 1.9 shows a relativized world where this possibility holds.

Buhrman, Spaan and Torenvliet [BST93] show that every 1-tt-complete set for **NEXP** is also m-complete. Buhrman and Fortnow [BF96] give a relativized world where a 1-tt-complete set for **PSPACE** is not m-complete.

Proof of Theorem 1.9: We will construct an oracle B relative to which there exists a 1-tt-complete set for **NP** that is not m-complete. Not just the classes but the reductions themselves must have access to the oracle.

We use the generic oracle approach along the lines of Fortnow and Rogers [FR94]. We show that the theorem is true for what Fortnow and Rogers call $\mathbf{UP} \cap \mathbf{coUP}$-generic oracles.

For the construction we assume that $\mathbf{P} = \mathbf{PSPACE}$. Since the proof below relativizes, we can remove this assumption by first relativizing to an oracle that makes $\mathbf{P} = \mathbf{PSPACE}$.

We guarantee that our oracle B contains exactly one string at lengths that are towers of 2. At all other lengths the oracle B is empty.

We can create a nondeterministic machine $M^B(x)$ that runs in linear time, queries at most $|x|$ strings whose length is less than $|x|$ and $M^B(x)$ accepts an $\mathbf{NP}^B$-complete set under unrelativized reductions.

Fortnow and Rogers [FR94] show that $\mathbf{NP}^B = \mathbf{coNP}^B = \mathbf{PSPACE}^B$ for these oracles. Thus we have a polynomial $p(n)$-time computable function f such that $M^B(x)$ accepts if and only if $M^B(f(x))$ rejects. The function f does not depend on B.

Let m be the largest tower of 2 at most $p(|x|)$. Let C be the set of strings in B of length less than m. Since every string in C has length at most $\log m$, we can enumerate C in polynomial-time with access to B by brute-force search.

We call an input x *good* if the number of z of length m such that $M^{C\cup\{z\}}(x)$ accepts is at least 2^{m-1}. Since $\mathbf{P} = \mathbf{PSPACE}$ we can determine whether an input x is good in polynomial time.

Consider the set L^B defined as

$$L^B = \{x \ : \ M^B(x) \text{ accepts and } x \text{ is good}\}.$$

We will show that L^B is in $\mathbf{NP}^B$, 1-tt-hard for $\mathbf{NP}^B$ but not m-complete where the reductions can access the oracle.

L^B is in $\mathbf{NP}^B$ since M is a nondeterministic polynomial-time Turing machine and testing goodness is in polynomial time.

If a string x is not good then $f(x)$ is good since for all z, $M^{C\cup\{z\}}(f(x))$ accepts exactly when $M^{C\cup\{z\}}(x)$ rejects.

We create a 1-tt reduction from $L(M^B)$ to L^B as follows: If x is good then accept if $x \in L^B$ otherwise accept if $f(x) \notin L^B$.

We still need to show that L^B is not m-complete for $\mathbf{NP}^B$. Consider the set

$$S^B = \{1^m \ : \ \text{There exists a } y \text{ of length } m-2 \text{ such that } 00y \in B\}$$

We show how to diagonalize over a potential many-one reduction g from S^B to L^B.

Fix m a large tower of 2. Consider $r = g^B(1^m)$ putting any unencoded query made by $g^B(1^m)$ out of B. If $|r| \leq m$ then $M^B(r)$ cannot query any strings of length m so we can easily diagonalize.

Otherwise we have two cases:

Case (1): r is not good: In this case we put some unencoded string y of length m that starts with 00 in B. We have 1^m in S^B but r is not in L^B.

Case (2): r is good: Since at least 2^{m-1} strings y of length m make $M^{C\cup\{y\}}(r)$ accept there must be such an unencoded y that does not start with 00. Putting y in B gives us that $r \in L^B$ but $1^m \notin S^B$ finishing the proof. □

Acknowledgments

We thank Alexis Maciel for helpful discussions and Dieter van Melkebeek for helpful comments on the write-up.

References

[ALM+92] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. In *Proceedings of the 33rd IEEE Symposium on Foundations of Computer Science*, pages 14–23. IEEE, New York, 1992.

[BBR94] D. Barrington, R. Beigel, and S. Rudich. Representing Boolean functions as polynomials modulo composite integers. *Computational Complexity*, 4(4):367–382, 1994.

[Bei91] R. Beigel. Relativized counting classes: relations among thresholds, parity and mods. *Journal of Computer and System Sciences*, 42(1):76–96, 1991.

[Ber77] L. Berman. *Polynomial Reducibilities and Complete Sets*. PhD thesis, Cornell University, 1977.

[BF96] H. Buhrman and L. Fortnow. Two queries. Technical Report CS 96-20, University of Chicago Department of Computer Science, 1996.

[BF97] H. Buhrman and L. Fortnow. Resource-bounded kolmogorov complexity revisited. In *Proceedings of the 14th Symposium on Theoretical Aspects of Computer Science*, volume 1200 of *Lecture Notes in Computer Science*, pages 105–116. Springer, Berlin, 1997.

[BFL91] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1(1):3–40, 1991.

[BFT97] H. Buhrman, L. Fortnow, and L. Torenvliet. Six hypotheses in search of a theorem. In *Proceedings of the 12th IEEE Conference on Computational Complexity*, pages 2–12. IEEE, New York, 1997.

[BH77] L. Berman and J. Hartmanis. On isomorphism and density of NP and other complete sets. *SIAM Journal on Computing*, 1:305–322, 1977.

[BH91] S. Buss and L. Hay. On truth-table reducibility to SAT. *Information and Computation*, 90(2):86–102, February 1991.

[BST93] H. Buhrman, E. Spaan, and L. Torenvliet. Bounded reductions. In K. Ambos-Spies, S. Homer, and U. Schöning, editors, *Complexity Theory*, pages 83–99. Cambridge University Press, December 1993.

[BT96] H. Buhrman and T. Thierauf. The complexity of generating and checking proofs of membership. In *Proceedings of the 13th Symposium on Theoretical Aspects of Computer Science*, volume 1046 of *Lecture Notes in Computer Science*, pages 75–86. Springer, Berlin, 1996.

[Coo71] S. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd ACM Symposium on the Theory of Computing*, pages 151–158. ACM, New York, 1971.

[FFK96] S. Fenner, L. Fortnow, and S. Kurtz. The isomorphism conjecture holds relative to an oracle. *SIAM Journal on Computing*, 25(1):193–206, 1996.

[For94] L. Fortnow. The role of relativization in complexity theory. *Bulletin of the European Association for Theoretical Computer Science*, 52:229–244, February 1994.

[FR94] L. Fortnow and J. Rogers. Separability and one-way functions. In *Proceedings of the 5th Annual International Symposium on Algorithms and Computation*, volume 834 of *Lecture Notes in Computer Science*, pages 396–404. Springer, Berlin, 1994.

[GS88] J. Grollmann and A Selman. Complexity measures for public-key cryptosystems. *SIAM Journal on Computing*, 17:309–355, 1988.

[Gup97] S. Gupta. Isolating an odd number of elements and applications in complexity theory. *Theory of Computing Systems*, 1997. To appear.

[Hel84] H. Heller. Relativized polynomial hierarchies extending two levels. *Mathematical Systems Theory*, 17(2):71–84, May 1984.

[Hel86] H. Heller. On relativized exponential and probabilistic complexity classes. *Information and Computation*, 71:231–243, 1986.

[HKR93] S. Homer, S. Kurtz, and J. Royer. A note on many-one and 1-truth table complete sets. *Theoretical Computer Science*, 115(2):383–389, July 1993.

[HS92] S. Homer and A. Selman. Oracles for structural properties: The isomorphism problem and public-key cryptography. *Journal of Computer and System Sciences*, 44(2):287–301, 1992.

[JY90] D. Joseph and P. Young. Self-reducibility: Effects of internal structure on computational complexity. In A. Selman, editor, *Complexity Theory Retrospective*, pages 82–107. Springer, 1990.

[Kar72] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[KMR90] S. Kurtz, S. Mahaney, and J. Royer. The structure of complete degrees. In A. Selman, editor, *Complexity Theory Retrospective*, pages 82–107. Springer, 1990.

[Kur85] S. Kurtz. Sparse sets in NP$-$P: Relativizations. *SIAM Journal on Computing*, 14(1):113–119, February 1985.

[LFKN92] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992.

[MVV87] K. Mulmuley, U. Vazirani, and V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

[Rac82] C. Rackoff. Relativized questions involving probabilistic algorithms. *Journal of the ACM*, 29(1):261–268, 1982.

[Sha92] A. Shamir. IP = PSPACE. *Journal of the ACM*, 39(4):869–877, 1992.

[Sip83] M. Sipser. A complexity theoretic approach to randomness. In *Proceedings of the 15th ACM Symposium on the Theory of Computing*, pages 330–335. ACM, New York, 1983.

[Tar91] J. Tarui. Degree complexity of Boolean functions and its applications to relativized separations. In *Proceedings of the 6th IEEE Structure in Complexity Theory Conference*, pages 382–390, New York, 1991. IEEE.

[Tor88] J. Torán. *Structural properties of the counting hierarchies*. PhD thesis, Facultat d'Informàtica, UPC Barcelona, January 1988.

[VV86] L. Valiant and V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.

[Wag90] K. Wagner. Bounded query classes. *SIAM Journal on Computing*, 19(5):833–846, 1990.