Nonuniform Reductions and NP-Completeness

John M. Hitchcock
Department of Computer Science
University of Wyoming

Hadi Shafei
Department of Mathematics and Computer Science
Northern Michigan University

Abstract

Nonuniformity is a central concept in computational complexity with powerful connections to circuit complexity and randomness. Nonuniform reductions have been used to study the isomorphism conjecture for NP and completeness for larger complexity classes. We study the power of nonuniform reductions for NP-completeness, obtaining both separations and upper bounds for nonuniform completeness vs uniform complessness in NP.

Under various hypotheses, we obtain the following separations:

1. There is a set complete for NP under nonuniform many-one reductions, but not under uniform many-one reductions. This is true even with a single bit of nonuniform advice.
2. There is a set complete for NP under nonuniform many-one reductions with polynomial-size advice, but not under uniform Turing reductions. That is, polynomial nonuniformity cannot be replaced by a polynomial number of queries.
3. For any fixed polynomial $p(n)$, there is a set complete for NP under uniform 2-truth-table reductions, but not under nonuniform many-one reductions that use $p(n)$ advice. That is, giving a uniform reduction a second query makes it impossible to simulate by a nonuniform reduction with fixed polynomial advice.
4. There is a set complete for NP under nonuniform many-one reductions with polynomial advice, but not under nonuniform many-one reductions with logarithmic advice. This hierarchy theorem also holds for other reducibilities, such as truth-table and Turing.

We also consider uniform upper bounds on nonuniform completeness. Hirahara (2015) showed that unconditionally every set that is complete for NP under nonuniform truth-table reductions that use logarithmic advice is also uniformly Turing-complete. We show that under a derandomization hypothesis, every set that is complete for NP under nonuniform truth-table reductions is also uniformly truth-table complete.

1 Introduction

Nonuniformity is a powerful concept in computational complexity. In a nonuniform computation a different algorithm or circuit may be used for each input size, as opposed to a uniform computation in which a single algorithm must be used for all inputs. Alternatively, nonuniform advice may be provided to a uniform algorithm – information that may not be computable by the algorithm but is computationally useful [21]. For example, nonuniformity can be used as a substitute for randomness [1]: every randomized algorithm can be replaced by a nonuniform one (BPP $\subseteq$ P/poly). It is unknown whether the same is true for NP, but the Karp-Lipton Theorem [21] states that if the polynomial-time hiearchy does not collapse, then NP-complete problems have superpolynomial nonuniform complexity (PH is infinite implies NP $\not\subseteq$ P/poly). Hardness versus randomness tradeoffs show that such nonuniform complexity lower bounds imply derandomization (for example, EXP $\not\subseteq$ P/poly implies BPP $\subseteq$ i.o.SUBEXP [9]).

Nonuniform computation can also be used to give reductions between decision problems, when uniform reductions are lacking. The Berman-Hartmanis Isomorphism Conjecture [12] for NP asserts that all NP-complete sets are isomorphic under polynomial-time reductions. Progress towards relaxations of the Isomorphism Conjecture with nonuniform reductions has been made [2, 3, 16] under various hypotheses.

Allender et al. [6] used nonuniform reductions to investigate the complexity of sets of Kolmogorov-random strings. They showed that the sets R_{KS} and R_{Kt} are complete for PSPACE and EXP, respectively, under P/poly-truth-table reductions. The problem R_{Kt} is not complete under uniform polynomial-time truth-table reductions [4]. It is known that R_{KS} is not complete under uniform logspace Turing reducibility [6], but it is open whether R_{KS} is complete under any type of uniform polynomial-time reducibility.

The Minimum Circuit Size Problem (MCSP) [20] is an intriguing NP problem. It is not known to be NP-complete. Proving it is NP-complete would imply consequences we don't yet know how to prove, yet there is really no strong evidence that it isn't NP-complete. Recently Allender [5] has asked if the Minimum Circuit Size Problem [20] is NP-complete under P/poly-Turing reductions.

Buhrman et al. [13] began a systematic study of nonuniform completeness. Among their results are separations of uniform and nonuniform completeness notions for EXP. We work toward a similarly solid understanding of NP-completeness under nonuniform reductions. We give both separation and upper bound results for a variety of nonuniform and uniform completeness notions. We consider the standard polynomial-time reducibilities including many-one ($\leq_{\mathrm{m}}^{\mathrm{P}}$), truth-table ($\leq_{\mathrm{tt}}^{\mathrm{P}}$), and Turing ($\leq_{\mathrm{T}}^{\mathrm{P}}$). We will consider nonuniform reductions such as $\leq_{\mathrm{m}}^{\mathrm{P}/h(n)}$ where the algorithm computing the reduction is allowed $h(n)$ bits of advice for inputs of size n.

Separating Nonuniform Completeness from Uniform Completeness. We show in Section 3 that for NP-completeness, uniform reductions that make multiple queries may not be replaced by nonuniform reductions that make a single query.

This is necessarily done under a hypothesis, for if P = NP, all completeness notions for NP trivially collapse. We use the Measure Hypothesis and the NP-Machine Hypothesis – two hypotheses on NP that have been used in previous work to separate NP-completeness notions [25, 27, 17]. The Measure Hypothesis asserts that NP does not have p-measure 0 [23, 24], or equivalently, that NP contains a p-random set [8, 7]. The NP-Machine Hypothesis [17] has many equivalent formulations and implies that there is an NP search problem that requires exponential time to solve almost everywhere.

We show under the Measure Hypothesis that there is a $\leq_{\mathrm{m}}^{\mathrm{P}/1}$-complete set for NP that is not $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete. In other words, nonuniform many-one reductions are stronger than many-one reductions for NP-completeness, and this holds with even a single nonuniform advice bit.

We also show that if the nonuniform reductions are allowed more advice, we have a separation even from Turing reductions. Under the NP-Machine Hypothesis, there is $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$-complete set that is not $\leq_{\mathrm{T}}^{\mathrm{P}}$-complete. That is, a nonuniform many-one reduction cannot be replaced by a uniform reduction that makes a polynomial number of adaptive queries.

Separating Uniform Completeness from Nonuniform Completeness Next, in Section 4, we give evidence that uniform reductions with multiple queries cannot be replaced by nonuniform reductions with a single query for NP-completeness.

We show under a hypothesis on $\mathrm{NP} \cap \mathrm{coNP}$ that adding just one more query makes a reduction more powerful than a nonuniform one for completeness: if $\mu_{\mathrm{p}}(\mathrm{NP} \cap \mathrm{coNP}) \neq 0$, then for any $c \geq 1$, there is a $\leq_{2-\mathrm{tt}}^{\mathrm{P}}$-complete set that is not $\leq_{\mathrm{m}}^{\mathrm{P}/n^c}$-complete. This is an interesting contrast to our separation of $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$-completeness from $\leq_{\mathrm{T}}^{\mathrm{P}}$-completeness (which includes $\leq_{2-\mathrm{tt}}^{\mathrm{P}}$-completeness). The $\mu_{\mathrm{p}}(\mathrm{NP} \cap \mathrm{coNP}) \neq 0$ hypothesis is admittedly strong. However, we note that strong hypotheses on $\mathrm{NP} \cap \mathrm{coNP}$ have been used in some prior investigations [28, 18, 13].

Uniform Completeness Upper Bounds for Nonuniform Completeness Despite the above separations, it is possible to replace a limited amount of nonuniformity by a uniform reduction for NP-completeness. Up to logarithmic advice may be made uniform at the expense of a polynomial number of queries:

1. Every $\leq_{\mathrm{m}}^{\mathrm{P}/\log}$-complete set for NP is also $\leq_{\mathrm{T}}^{\mathrm{P}}$-complete. This proof uses search reduces to decision for SAT.

2. Under a derandomization hypothesis (E has a problem with high NP-oracle circuit complexity), every $\leq_{\mathrm{m}}^{\mathrm{P}/\log}$-complete set for NP is also $\leq_{\mathrm{tt}}^{\mathrm{P}}$-complete. The Valiant-Vazirani lemma [29] gives a randomized algorithm to reduce the satisfiability problem to the unique satisifability problem. Being able to derandomize this algorithm [22] yields a nonadaptive reduction.

These upper bound results are presented in Section 5.

Hierarchy Theorems for Nonuniform Completeness In Section 6, we give hierarchy theorems for nonuniform NP-completeness. We separate polynomial advice from logarithmic advice: if the NP-machine hypothesis is true, then there is a $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$-complete set that is not $\leq_{\mathrm{m}}^{\mathrm{P}/\log}$-complete. This also holds for other reducibilities such as truth-table and Turing.

2 Preliminaries

All languages in this paper are subsets of $\{0,1\}^*$. We use the standard lexicographic enumeration of binary strings, i.e. $s_0 = \lambda, s_1 = 0, s_2 = 1, s_3 = 00, \ldots$ as an order on binary strings. For any language $A \subseteq \{0,1\}^*$ the characteristic sequence of A is defined as $\chi_A = A[s_0]A[s_1]A[s_2]\ldots$ where $A[x] = 1$ or 0 depending on whether the string x belongs to A or not, respectively. We identify every language with its characteristic sequence. For any binary sequence X and natural number n, $X \upharpoonright n$ is the first n bits of X's characteristic sequence.

We use the standard definitions of complexity classes and well-known reductions that can be found in [10, 26]. For any two languages A and B and a function $l : \mathbb{N} \to \mathbb{N}$, we say A is *nonuniform polynomial-time reducible to B with advice $l(n)$*, and we write $A \leq_{\mathrm{m}}^{\mathrm{P}/l(n)} B$, if there exists $f \in \mathrm{PF}$ and $h : \mathbb{N} \to \{0,1\}^*$ with $|h(n)| \leq l(n)$ for all n such that $(\forall x)\ x \in A \ \leftrightarrow f\big(x, h(|x|)\big) \in B$. The string $h(|x|)$ is called the *advice*, and it only depends on the length of the input. For a class $\mathcal{H}$ of functions mapping $\mathbb{N} \to \{0,1\}^*$, we say $A \leq_{\mathrm{m}}^{\mathrm{P}/\mathcal{H}} B$ if $A \leq_{\mathrm{m}}^{\mathrm{P}/l} B$ for some $l \in \mathcal{H}$. The class poly denotes all advice functions with length bounded by a polynomial, and log is all advice functions with length $O(\log n)$. We also use $\leq_{\mathrm{m}}^{\mathrm{P}/1}$ for a nonuniform reduction when $|h(|x|)| = 1$. Nonuniform reductions can similarly be defined with respect to other kinds of reductions like Turing, truth-table, etc.

In most of our proofs we use resource-bounded measure [23] to state our hypotheses. In the following we provide a brief description of this concept. For more details, see [23, 24, 7]. A *martingale* is a function $d : \{0,1\}^* \to [0,\infty)$ where $d(\lambda) > 0$ and $\forall x \in \{0,1\}^*$, $2d(x) = d(x0)+d(x1)$. We say a martingale *succeeds* on a set $A \subseteq \{0,1\}^*$ if $\limsup_{n\to\infty} d(A \upharpoonright n) = \infty$, where $A \upharpoonright n$ is the length n prefix of A's characteristic sequence. One can think of the martingale d as a strategy for betting on the consecutive bits of the characteristic sequence of A. The martingale is allowed to use the first $n-1$ bits of A when betting on the n^{th} bit. Betting starts with the initial capital $d(\lambda) > 0$, and $d(A \upharpoonright n-1)$ denotes the capital after betting on the first $(n-1)$ bits. At this stage the martingale bets some amount a where $0 \leq a \leq d(A \upharpoonright n-1)$ that the next bit is 0 and the rest of the capital, i.e. $d(A \upharpoonright n-1) - a$, that the next bit is 1. If the n^{th} bit is 0, then $d(A \upharpoonright n) = 2a$. Otherwise, $d(A \upharpoonright n) = 2(d(A \upharpoonright n-1) - a)$. For any time bound $t(n)$, we say a language L is $t(n)$-random if no $O(t(n))$-computable martingale succeeds on L. A class of languages C has p-measure 0, written $\mu_{\mathrm{p}}(C) = 0$, if there is a c such that no language in C is n^c-random. Similarly, C has p_2-measure 0, written $\mu_{\mathrm{p}_2}(C) = 0$, if there is a c such that no language in C is $2^{\log^c n}$-random. If C is closed under $\leq_{\mathrm{m}}^{\mathrm{P}}$-reductions, then $\mu_{\mathrm{p}}(C) = 0$ if and only if $\mu_{\mathrm{p}_2}(C) = 0$ [19].

We will use the *Measure Hypothesis* that $\mu_{\mathrm{p}}(\mathrm{NP}) \neq 0$ and the NP-*Machine Hypothesis* [17]: there is an NP machine M and an $\epsilon > 0$ such that M accepts 0^* and no 2^{n^ϵ}-time-bounded Turing machine computes infinitely many accepting computations of M. The Measure Hypothesis implies the NP-Machine Hypothesis [17].

3 Separating Nonuniform Completeness from Uniform Completeness

Buhrman et al. [13] constructed a set that is a $\leq_{\mathrm{m}}^{\mathrm{P}/1}$-complete set for EXP and is P-bi-immune. Since no $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete set for EXP is P-bi-immune [11], the set of Buhrman et al. is not $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete for EXP. Our first theorem achieves the same separation of $\leq_{\mathrm{m}}^{\mathrm{P}/1}$-completeness for NP from uniform many-one completeness, under the measure hypothesis on NP.

Theorem 3.1. *If $\mu_{\mathrm{p}}(\mathrm{NP}) \neq 0$ then there exists a set $D \in \mathrm{NP}$ that is* NP*-complete with respect to $\leq_{\mathrm{m}}^{\mathrm{P}/1}$-reductions but is not $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete.*

Proof. Let $R \in \mathrm{NP}$ be p-random. We use R and SAT to construct the following set:

$$D = \begin{array}{l} \{\ \langle \phi, 0\rangle : \phi \in \mathrm{SAT}\ \vee\ 0^{|\phi|} \in R\} \\ \bigcup\{\langle \phi, 1\rangle : \phi \in \mathrm{SAT}\ \wedge\ 0^{|\phi|} \in R\} \end{array}$$

It follows from closure properties of NP that $D \in \mathrm{NP}$. It is also easy to see that $\mathrm{SAT} \leq_{\mathrm{m}}^{\mathrm{P}/1} D$ via $\phi \to \langle \phi, R[0^{|\phi|}]\rangle$. Note that $R[0^{|\phi|}]$ is one bit of advice, and it is 1 or 0 depending on whether or not $0^{|\phi|} \in R$. We will prove that D is not $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete for NP. To get a contradiction, assume that D is $\leq_{\mathrm{m}}^{\mathrm{P}}$-complete. Therefore $\mathrm{SAT} \leq_{\mathrm{m}}^{\mathrm{P}} D$ via some polynomial-time computable function f. Then

$(\forall\phi)\ \phi \in \mathrm{SAT} \leftrightarrow f(\phi) \in D$. Based on the value of $\mathrm{SAT}[\phi]$ and the second component of $f(\phi)$ we consider four cases:

1. $\phi \in \mathrm{SAT}\ \wedge f(\phi) = \langle\psi, 0\rangle$, for some formula ψ.
2. $\phi \notin \mathrm{SAT}\ \wedge f(\phi) = \langle\psi, 0\rangle$, for some formula ψ.
3. $\phi \in \mathrm{SAT}\ \wedge f(\phi) = \langle\psi, 1\rangle$, for some formula ψ.
4. $\phi \notin \mathrm{SAT}\ \wedge f(\phi) = \langle\psi, 1\rangle$, for some formula ψ.

In the second case above we have $\mathrm{SAT}[\phi] = \mathrm{SAT}[\psi] \vee R[0^{|\psi|}]$ and $\phi \notin \mathrm{SAT}$. Therefore $\mathrm{SAT}[\psi] \vee R[0^{|\psi|}] = 0$ which implies $R[0^{|\psi|}] = 0$. Consider the situation where the second case happens and $|\psi| \geq |\phi|/2$. The following argument shows that $R[0^{|\psi|}]$ is computable in $2^{5|\psi|}$ time in this situation. We apply f to every string of length at most $2|\psi|$, looking for a formula ϕ of length at most $2|\psi|$ such that $f(\phi) = \langle\psi, 0\rangle$ and $\phi \notin \mathrm{SAT}$. We are applying f which is computable in polynomial time to at most $2^{2|\psi|}$ strings. This can be done in $2^{3|\psi|}$ steps. Checking if $\phi \notin \mathrm{SAT}$ can be done in at most $2^{2|\psi|}$ steps for each ϕ. Therefore the whole algorithm takes at most $2^{5|\psi|}$ steps to terminate. If this case happens for infinitely many ψ's we will have a polynomial-time martingale that succeeds on R which contradicts the p-randomness of R. As a result, there cannot be infinitely many ϕ's such that $\phi \notin \mathrm{SAT}$, $f(\phi) = \langle\psi, 0\rangle$, and $|\psi| \geq |\phi|/2$. This is because if there are infinitely many such ϕ's, then there must be infinitely many n's such that for each n there exists a ϕ satisfying the above properties. Since we assumed $|\psi| \geq |\phi|/2$ it follows that there must be infinitely many such ψ's, but we proved that this cannot happen.

Next we will show that a similar result holds for the third case. In the third case we have $\mathrm{SAT}[\phi] = \mathrm{SAT}[\psi] \wedge R[0^{|\psi|}]$ and $\phi \in \mathrm{SAT}$. Therefore $\mathrm{SAT}[\psi] \wedge R[0^{|\psi|}] = 1$ which implies $R[0^{|\psi|}] = 1$. Consider the situation where the third case occurs and $|\psi| \geq |\phi|/2$. The following argument proves that $R[0^{|\psi|}]$ is computable in $2^{5|\psi|}$ time. We apply f to every string of length at most $2|\psi|$, looking for a formula ϕ of length at most $2|\psi|$ such that $f(\phi) = \langle\psi, 1\rangle$ and $\phi \in \mathrm{SAT}$. We are applying f which is computable in polynomial time to at most $2^{2|\psi|}$ strings. This can be done in $2^{3|\psi|}$ steps. Checking if $\phi \in \mathrm{SAT}$ can be done in at most $2^{2|\psi|}$ steps for each ϕ. Therefore the whole algorithm takes at most $2^{5|\psi|}$ steps to terminate. If this case happens for infinitely many ψ's we will have a polynomial-time martingale that succeeds on R which contradicts the p-randomness of R. As a result, there cannot be infinitely many ϕ's such that $\phi \notin \mathrm{SAT}$, $f(\phi) = \langle\psi, 0\rangle$, and $|\psi| \geq |\phi|/2$. This is because if there are infinitely many such ϕ's then there must be infinitely many n's such that for each n there exits a ϕ satisfying the above properties. Since we assumed $|\psi| \geq |\phi|/2$ it follows that there must be infinitely many such ψ's, but we proved that this cannot happen.

We have shown that:

1. For almost every ϕ, if $\phi \notin \mathrm{SAT}\ \wedge f(\phi) = \langle\psi, 0\rangle$, then $|\psi| < |\phi|/2$.
2. For almost every ϕ, if $\phi \in \mathrm{SAT}\ \wedge f(\phi) = \langle\psi, 1\rangle$, then ψ, $|\psi| < |\phi|/2$.

It follows from these two facts that for almost every ϕ, if $|\psi| \geq |\phi|/2$, then $\mathrm{SAT}[\phi]$ can be computed in polynomial time:

1. If $f(\phi) = \langle\psi, 0\rangle$ and $|\psi| \geq |\phi|/2$, then $\phi \in \mathrm{SAT}$.
2. If $f(\phi) = \langle\psi, 1\rangle$ and $|\psi| \geq |\phi|/2$, then $\phi \notin \mathrm{SAT}$.

Note that the only computation required in the algorithm above is computing f on ϕ which can be done in polynomial time. To summarize, for every formula ϕ it is either the case that when we apply f to ϕ the new formula ψ satisfies $|\psi| < |\phi|/2$ or $\mathrm{SAT}[\phi]$ is computable in polynomial time. In the following we use this fact and the many-one reduction from SAT to D to introduce a $(\log n)$-tt-reduction from SAT to R.

The many-one reduction from SAT to D implies that $(\forall \phi)\ \phi \in \mathrm{SAT} \leftrightarrow f(\phi) \in D$. In other words:

$$(\forall \phi)\ f(\phi) = \langle \psi_1, i \rangle \ \text{ and } \ \mathrm{SAT}[\phi] = SAT[\psi_1] \diamond_1 R[0^{|\psi_1|}] \tag{3.1}$$

where $\diamond_1$ is $\vee$ or $\wedge$ when $i = 0$ or 1 respectively.

Fix two strings a and b such that $a \in R$ and $b \notin R$. If $|\psi_1| \geq |\phi|/2$ then $\mathrm{SAT}[\phi]$ is computable in polynomial time, and our reduction maps ϕ to either a or b depending on $\mathrm{SAT}[\phi]$ being 1 or 0 respectively. To put it differently, the right hand side of (3.1) will be substituted by $R[a]$ or $R[b]$ respectively.

On the other hand, if $|\psi_1| < |\phi|/2$ then we repeat the same process for ψ_1. We apply f to ψ_1 to get

$$\mathrm{SAT}[\psi_1] = \mathrm{SAT}[\psi_2] \diamond_2 R[0^{|\psi_2|}] \tag{3.2}$$

By substituting this in (3.1) we will have:

$$\mathrm{SAT}[\phi] = (\mathrm{SAT}[\psi_2] \diamond_2 R[0^{|\psi_2|}]) \diamond_1 R[0^{|\psi_1|}] \tag{3.3}$$

Again, if $|\psi_2| \geq |\psi_1|/2$ then $\mathrm{SAT}[\psi_1]$ is computable in polynomial time, and its value can be substituted in (3.2) to get a reduction from SAT to R. On the other hand, if $|\psi_2| < |\psi_1|/2$ then we use f again to find ψ_3 such that:

$$\mathrm{SAT}[\psi_2] = \mathrm{SAT}[\psi_3] \diamond_3 R[0^{|\psi_3|}] \tag{3.4}$$

By substituting this in (3.3) we will have:

$$\mathrm{SAT}[\phi] = \big((\mathrm{SAT}[\psi_3] \diamond_3 R[0^{|\psi_3|}]) \diamond_2 R[0^{|\psi_2|}]\big) \diamond_1 R[0^{|\psi_1|}] \tag{3.5}$$

We repeat this process up to $(\log n)$ times where $n = |\phi|$. If there exists some $i \leq (\log n)$ such that $|\psi_{i+1}| \geq |\psi_i|/2$, then we can compute $\mathrm{SAT}[\psi_i]$ in polynomial time and substitute its value in the following equation:

$$\mathrm{SAT}[\phi] = \big((\mathrm{SAT}[\psi_i] \diamond_k R[0^{|\psi_i|}]) \diamond_{i-1} R[0^{|\psi_{i-1}|}]\big) \ldots \diamond_1 R[0^{|\psi_1|}] \tag{3.6}$$

This gives us an i-tt-reduction from SAT to R for some $i < (\log n)$.

On the other hand, if $|\psi_{i+1}| < |\psi_i|/2$ for every $i \leq (\log n)$ then we will have:

$$\mathrm{SAT}[\phi] = \big((\mathrm{SAT}[\psi_{(\log n)}] \diamond_{(\log n)} R[0^{|\psi_{(\log n)}|}]) \diamond_{(\log n)-1} R[0^{|\psi_{(\log n)-1}|}]\big) \ldots \diamond_1 R[0^{|\psi_1|}] \tag{3.7}$$

It follows from the construction that the length of ψ_i's is halved on each step. Therefore $|\psi_{(\log n)}|$ must be constant in n. As a result $\mathrm{SAT}[\psi_{(\log n)}]$ is computable in constant time. If we compute the value of $\mathrm{SAT}[\psi_{(\log n)}]$, and substitute it in (3.7) we will have a $(\log n)$-tt-reduction from SAT to R. In any case, we have shown that if SAT is many-one reducible to D, we can use this reduction to define a polynomial time computable $(\log n)$-tt-reduction from SAT to R. This means that R is $(\log n)$-tt-complete for NP. Buhrman and van Melkebeek [14] showed that complete sets for NP under $\leq^{\mathrm{P}}_{n^{\alpha}-\mathrm{tt}}$-reductions have p_2-measure 0. Since this complete degree is closed under $\leq^{\mathrm{P}}_{\mathrm{m}}$-reductions, it also has p-measure 0 [19]. Therefore the $(\log n)$-tt-completeness of R contradicts its p-randomness, which completes the proof. □

This next theorem is based on a result of Hitchcock and Pavan [16] that separated strong nondeterministic completeness from Turing completeness for NP.

Theorem 3.2. *If the* NP*-machine hypothesis holds, then there exists a* $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$*-complete set in* NP *that is not* $\leq_{\mathrm{T}}^{\mathrm{P}}$*-complete.*

Proof. We follow the setup in [16, Theorem 4.2]. Assume the NP-machine hypothesis holds. Then it can be shown there exists an NP-machine M that accepts 0^* such that no 2^{n^3}-time bounded Turing machine can compute infinitely many of its computations. Consider the following NP set:

$$A = \{\langle \phi, a\rangle \mid \phi \in \mathrm{SAT} \text{ and } a \text{ is an accepting computation of } M(0^{|\phi|})\} \tag{3.8}$$

The mapping $\phi \to \langle \phi, a\rangle$ where a is the first accepting computation of $M(0^{|\phi|})$ is a $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$-reduction from SAT to A. Note that a only depends on the length of ϕ and $|a|$ is polynomial in $|\phi|$. Therefore A is $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$-complete for NP. It is proved in [16] that A is not $\leq_{\mathrm{T}}^{\mathrm{P}}$-complete. □

Because the measure hypothesis implies the NP-machine hypothesis, we have the following corollary.

Corollary 3.3. *If* $\mu_{\mathrm{p}}(\mathrm{NP}) \neq 0$*, then there exists a* $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$*-complete set in* NP *that is not* $\leq_{\mathrm{T}}^{\mathrm{P}}$*-complete.*

4 Separating Uniform Completeness from Nonuniform Completeness

In the following proof, we use the construction of a $\leq_{2-\mathrm{tt}}^{\mathrm{P}}$-complete set that was previously used to separate $\leq_{2-\mathrm{tt}}^{\mathrm{P}}$-completeness from $\leq_{1-\mathrm{tt}}^{\mathrm{P}}$-completeness [28] and $\leq_{2-\mathrm{tt}}^{\mathrm{P}}$-autoreducibility from $\leq_{1-\mathrm{tt}}^{\mathrm{P}}$-autoredicibility [18].

Theorem 4.1. *If* $\mu_{\mathrm{p}}(\mathrm{NP} \cap \mathrm{coNP}) \neq 0$ *then for every* $c \geq 1$*, there exists a set* $A \in \mathrm{NP}$ *that is* $\leq_{2-\mathrm{tt}}^{\mathrm{P}}$*-complete but is not* $\leq_{\mathrm{m}}^{\mathrm{P}/n^c}$*-complete.*

Proof. We know that $\mu_{\mathrm{p}}(\mathrm{NP}\cap\mathrm{coNP}) \neq 0$ implies $\mu_{\mathrm{p}_2}(\mathrm{NP}\cap\mathrm{coNP}) \neq 0$ [19]. Therefore we can assume there exists $R \in \mathrm{NP}\cap\mathrm{coNP}$ that is p_2-random. We fix $c \geq 1$, and define $A = 0(R\cap\mathrm{SAT})\cup 1(\bar{R}\cap\mathrm{SAT})$, where $\bar{R}$ is R's complement. It follows from closure properties of NP that $A \in \mathrm{NP}$. We can define a polynomial-time computable 2-tt-reduction from SAT to A as follows: on input x we make two queries $0x$ and $1x$ to A, and we have $x \in \mathrm{SAT} \leftrightarrow (0x \in A \vee 1x \in A)$. Therefore A is $\leq_{2-\mathrm{tt}}^{\mathrm{P}}$-complete in NP. We will show that A is not $\leq_{\mathrm{m}}^{\mathrm{P}/n^c}$-complete. To get a contradiction, assume A is $\leq_{\mathrm{m}}^{\mathrm{P}/n^c}$-complete in NP. This implies that $R \leq_{\mathrm{m}}^{\mathrm{P}/n^c} A$ via functions $f \in \mathrm{PF}$ and $h : \mathbb{N} \to \{0,1\}^*$ where $(\forall n)\ |h(n)| = n^c$. In other words:

$$(\forall x)R[x] = A[f\big(x, h(|x|)\big)] \text{ where } |h(n)| = n^c \tag{4.1}$$

For each length n the advice a_n has length n^c. As a result, there are 2^{n^c} possibilities for a_n. For each length n we define 2^{n^c} martingales such that each martingale assumes one of these possible strings is the actual advice for length n, and uses (4.1) to bet on R. We divide the capital into 2^{n^c} equal shares between these martingales. In the worst case, the martingales that do not use the right advice lose their share of the capital. We define these martingales such that the martingale that uses the right advice multiplies its share by 2^{n^c+1}. We will also show that this happens for infinitely many lengths n, which gives us a p_2-strategy to succeed on R. Note that based on the argument above, we can only focus on the martingale that uses the right advice for each length. To say it differently, in the rest of the proof we assume that we know the right advice for each length, but the price that we have to pay is to show that our martingale can multiply its capital by 2^{n^c+1}.

For each length n we first compute $\mathrm{SAT}[z]$ for every string z of length n. In particular, we are interested in the following set:

$$A_n = \{z \mid |z| = n \text{ and } z \notin \mathrm{SAT}\}$$

If $|A_n| < n^{2c}$ we do not bet on any string of length n. It follows from paddability of SAT that there must be infinitely many n's such that $|A_n| \geq n^{2c}$. Assume n is a length where $|A_n| \geq n^{2c}$, and let a_n be the right advice for length n. For any string x, let $v(0x) = v(1x) = x$. Consider the following set:

$$B_n = \{z \mid |z| = n,\ z \notin \mathrm{SAT},\ \text{and } v(f(z, a_n)) \leq z\}$$

Claim 4.1.1. *There can only be finitely many n's such that $|B_n| \geq n^c + 1$.*

Proof. To prove this claim, assume there are infinitely many n's such that $|B_n| \geq n^c + 1$. This means there are infinitely many n's such that for each such n there are $n^c + 1$ strings of length n, $z_1, z_2, ..., z_{n^c+1}$, satisfying the following property:

$$(\forall\ 1 \leq i \leq n^c + 1)\ R[z_i] = A[f(z_i, a_n)]\ \wedge\ v(f(z_i, a_n)) \leq z_i \tag{4.2}$$

It follows from the definition of A that $A[y] = (\tilde{R} \cap \mathrm{SAT})[v(y)]$ where $\tilde{R}$ is R or $\bar{R}$ depending on whether y starts with a 0 or 1 respectively. Therefore (4.2) turns into:

$$(\forall\ 1 \leq i \leq n^c + 1)\ R[z_i] = (\tilde{R} \cap \mathrm{SAT})[v(f(z_i, a_n))]\ \wedge\ v(f(z_i, a_n)) \leq z_i \tag{4.3}$$

We use (4.3) to define a martingale that predicts $R[z_i]$ for every $1 \leq i \leq n^c + 1$. Since we know $R[z_i] = \tilde{R}[v(f(z_i, a_n))] \wedge \mathrm{SAT}[v(f(z_i, a_n))]$ our martingale computes $\tilde{R}[v(f(z_i, a_n))] \wedge \mathrm{SAT}[v(f(z_i, a_n))]$ and bets on $R[z_i]$ having the same value as $\tilde{R}[v(f(z_i, a_n))] \wedge \mathrm{SAT}[v(f(z_i, a_n))]$. Now we need to show why a polynomial time martingale has enough time to compute $\tilde{R}[v(f(z_i, a_n))] \wedge \mathrm{SAT}[v(f(z_i, a_n))]$. Note that we know $v(f(z_i, a_n)) \leq z_i$ so it is either the case that $v(f(z_i, a_n)) < z_i$ or $v(f(z_i, a_n)) = z_i$. In the first case, the martingale has access to $\tilde{R}[v(f(z_i, a_n))]$, and has enough time to compute $\mathrm{SAT}[v(f(z_i, a_n))]$. In the second case we know that $\mathrm{SAT}[v(f(z_i, a_n))] = 0$ therefore $R[z_i] = 0$. This implies that we can double the capital for each z_i. As a result, the capital can be multiplied by 2^{n^c+1}. If this happens for infinitely many n's we have a martingale that succeeds on R which is a contradiction. This completes the proof of Claim 4.1.1. □

The fact that there must be infinitely many n's where $|A_n| \geq n^{2c}$ and the claim above imply that there exist infinitely many n's such that $|C_n| \geq n^{2c} - n^c$ where C_n is defined as follows:

$$C_n = \{z \mid\ |z| = n,\ z \notin \mathrm{SAT},\ \text{and } v\big(f(z, a_n)\big) > z\}$$

The following claim states that when applying f to elements of C_n there cannot be many collisions. Define:

$$D_n = \{z \in C_n \mid (\exists\ y \in C_n)\ y < z\ \wedge\ f(y, a_n) = f(z, a_n)\}$$

Claim 4.1.2. *There cannot be infinitely many n's such that $|D_n| \geq n^c + 1$.*

Proof. To get a contradiction, assume there are infinitely many n's such that $|D_n| \geq n^c + 1$. Let $t_1, t_2, ..., t_{n^c+1}$ be the first such strings. Then we have:

$$(\forall\, 1 \leq i \leq n^c + 1)\ (\exists\, r_i)\ r_i \in C_n\ \wedge\ r_i < t_i\ \wedge\ f(r_i, a_n) = f(t_i, a_n)$$

It follows that:

$$(\forall\, 1 \leq i \leq n^c + 1)\ (\exists\, r_i)\ r_i \in C_n\ \wedge\ r_i < t_i\ \wedge\ R[r_i] = R[t_i]$$

We can define a martingale that looks up the value of $R[r_i]$, and bets on $R[t_i]$ based on the equation above. This means that we can double the capital by betting on $R[t_i]$ for every $1 \leq i \leq n^c + 1$. As a result, the capital will be multiplied by 2^{n^c+1}. If this happens for infinitely many n's we will have a martingale that succeeds on R which is a contradiction. This completes the proof of Claim 4.1.2. □

Assume n is a length where $|C_n| \geq n^{2c} - n^c$. We have shown that there are infinitely many such n's. We claim that for infinitely many of these n's, since R is p_2-random, there must be at least $(n^{2c} - n^c)/4$ strings in C_n that also belong to R.

Claim 4.1.3. *There must be infinitely many n's such that $|C_n| \geq n^{2c} - n^c$, and $|C_n \cap R| \geq (n^{2c} - n^c)/4$.*

Proof. We have shown that there exist infinitely many n's where $|C_n| \geq n^{2c} - n^c$. For a contradiction, assume that for all but finitely many of these n's $|C_n \cap R| < (n^{2c} - n^c)/4$. We use this assumption to define a polynomial time martingale that succeeds on R. Note that finding n's where $|C_n| \geq n^{2c} - n^c$ consists of computing SAT for every string of length n, and counting the number of negative answers, which can be done in at most 2^{3n} steps, followed by applying f to these strings and comparing $v(f(z, a_n))$ and z, which can be done in at most 2^{2n} steps. This means a polynomial-time martingale has enough time to detect C_n's where $|C_n| \geq n^{2c} - n^c$. After detecting these C_n's we know that most of the strings in C_n do not belong to R by assumption above. Therefore a simple martingale that for every string z in C_n bets 2/3 of the capital on $R[z] = 0$ and the rest on $R[z] = 1$ will succeeds on R, which is a contradiction. This completes the proof of Claim 4.1.3. □

Let n be a length where $|C_n \cap R| \geq (n^{2c} - n^c)/4$, and consider the image of $C_n \cap R$ under $f(\cdot, a_n)$:

$$I_n = \{f(z, a_n) \mid z \in C_n \cap R\}$$

It follows from Claim 4.1.2 that $|I_n| \geq [(n^{2c} - n^c)/4] - n^c$. If we consider the image of I_n under $v(\cdot)$ we have:

$$V_n = \{v(f(z, a_n)) \mid z \in C_n \cap R\}$$

It is easy to see that $|V_n| \geq |I_n|/2$. Therefore for large enough n we have $|V_n| \geq n^c + 1$. Now if we use (4.3) we have $R[z] = (\tilde{R} \cap \mathrm{SAT})[v(f(z, a_n))]$. We know that $z \in R$. This implies that $\tilde{R}[v(f(z, a_n))] = 1$. Therefore a martingale that bets on $\tilde{R}[v(f(z, a_n))] = 1$ can double the capital each time. Since $|V_n| \geq n^c + 1$ this martingale multiplies the capital by 2^{n^c+1}. As a result, we have a martingale that succeeds on R, which completes the proof. □

5 Uniform Upper Bounds on Nonuniform Completeness

In this section, we consider whether nonuniformity can be removed in NP-completeness, at the expense of more queries.

Buhrman et al. [13] proved that every $\leq_{\mathrm{T}}^{\mathrm{P}/\log}$-complete set for EXP is also $\leq_{\mathrm{T}}^{\mathrm{P}}$-complete using a tableaux method. Hirahara [15] proved a more general result that implies the same for holds NP. Specifically, Hirahara showed that for any paddable, downward self-reducible language L and any oracle R, $L \in \mathrm{P}^R/\log$ implies $L \in \mathrm{P}^R$. Since SAT is paddable and downward self-reducible, the following theorem is immediate from [15].

Theorem 5.1. *(Hirahara [15]) Every* $\leq_{\mathrm{T}}^{\mathrm{P}/\log}$*-complete set in* NP *is* $\leq_{\mathrm{T}}^{\mathrm{P}}$*-complete.*

Valiant and Vazirani [29] proved that there exists a randomized polynomial-time algorithm that given any formula ϕ, outputs a list of formulas l such that:

1. Every assignment that satisfies a formula in l also satisfies ϕ.
2. If ϕ is satisfiable, then with high probability at least one of the formulas in l is uniquely satisfiable.

Klivans and van Melkebeek [22] showed that the Valiant-Vazirani lemma can be derandomized if E contains a problem with exponential NP-oracle circuit complexity. This yields a deterministic polynomial-time algorithm that given any formula ϕ, outputs a list of formulas l such that:

1. Every assignment that satisfies a formula in l also satisfies ϕ.
2. If ϕ is satisfiable, then one of the formulas in l is uniquely satisfiable.

Theorem 5.2. *If* E *contains a problem with* NP*-oracle circuit complexity* $2^{\Omega(n)}$*, then every* $\leq_{\mathrm{m}}^{\mathrm{P}/1}$*-complete set in* NP *is* $\leq_{\mathrm{tt}}^{\mathrm{P}}$*-complete.*

Proof. Let A be an arbitrary $\leq_{\mathrm{m}}^{\mathrm{P}/1}$-complete set in NP, and define $\widehat{\mathrm{SAT}}$ as before. We know that $\widehat{\mathrm{SAT}} \leq_{\mathrm{m}}^{\mathrm{P}/1} A$ via some $f \in \mathrm{PF}$ and some $h : \mathbb{N} \to \{0,1\}$ where $(\forall \phi)\ \widehat{\mathrm{SAT}}[\phi] = A[f(\phi, h(|\phi|))]$. We will define a $\leq_{\mathrm{tt}}^{\mathrm{P}}$-reduction from SAT to A.

Before describing the rest of the algorithm, observe that the process of reducing search to decision for a Boolean formula can be done using independent queries in the case that the formula is uniquely satisfiable. This is due to the fact that if a formula $\psi(y_1, \ldots, y_m)$ is uniquely satisfiable, then for each $1 \leq j \leq m$ exactly one of the formulas $\psi \wedge x_j$ and $\psi \wedge \neg x_j$ is satisfiable. Therefore the unique satisfying assignment can be found by making m independent queries to SAT, i.e. $\psi \wedge x_1, \ldots, \psi \wedge x_m$.

Using the hypothesis to derandomize the Valiant-Vazirani algorithm [22], we have a deterministic algorithm that on input $\phi(x_1, \ldots, x_n)$ outputs a list containing polynomially many formulas ψ_1,...,ψ_m satisfying properties described above. For each formula $\psi_j(y_1^j, ..., y_{n_j}^j)$ consider $\psi_j \wedge y_k^j$'s for every $1 \leq k \leq n_j$, and use padding in $\widehat{\mathrm{SAT}}$ to turn these formulas into formulas of the same length. We denote the padded version of $\psi_j \wedge y_k^j$ by ψ_j^k for simplicity. For each ψ_j we make n_j independent queries to A: $q_1^j = f(\psi_j^1, 0), \ldots, q_{n_j}^j = f(\psi_j^{n_j}, 0)$. For each one of these queries if the answer is positive we set the respective variable to 1 and 0 otherwise. We repeat this process using 1 as advice, and we will have $2m$ assignments. We argue that ϕ is satisfiable if and only if at least one of these assignments satisfies it. If ϕ is not satisfiable then obviously none of these assignments will satisfy it. On the other hand, if $\phi \in \mathrm{SAT}$ then at least one of the ψ_j's must be uniquely satisfiable. In this case the process described above will find this unique satisfying assignment. Again, by the Valiant-Vazirani lemma we know that every assignment that satisfies at least one of the ψ_j's must also satisfy ϕ, which means one of the $2m$ assignments produced by the algorithm above will satisfy ϕ in the case that ϕ is satisfiable. It is evident from the algorithm that the queries are independent. It is also easy to see that the reduction runs in polynomial time in $|\phi|$ since we are applying a polynomial-time computable function f to arguments about the same length as ϕ, and we are doing this $2m$ times which is polynomial in $|\phi|$. Therefore this algorithm defines a polynomial-time truth-table reduction from SAT to A. □

If the nonuniform reduction in the theorem above uses k bits of advice instead of considering two cases in the proof there are 2^k cases to be consider. If $k \in O(\log n)$ then this can be done in polynomial time. Also note that the nonuniform reduction can be a truth-table reduction instead of a many-one reduction, and the same proof still works.

Theorem 5.3. *If* E *contains a problem with* NP*-oracle circuit complexity* $2^{\Omega(n)}$*, then every* $\leq_{\mathrm{tt}}^{\mathrm{P}/\log}$*-complete set in* NP *is* $\leq_{\mathrm{tt}}^{\mathrm{P}}$*-complete.*

6 Hierarchy Theorems for Nonuniform Completeness

Hirahara [15] proved unconditionally that every $\leq_{\mathrm{m}}^{\mathrm{P}/\log}$-complete set in NP is $\leq_{\mathrm{T}}^{\mathrm{P}}$-complete. On the other hand, we showed that under the NP-machine hypothesis there exists a $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$-complete set in NP that is not $\leq_{\mathrm{T}}^{\mathrm{P}}$-complete. This results in a separation of $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$-completeness from $\leq_{\mathrm{m}}^{\mathrm{P}/\log}$-completeness under the NP-machine hypothesis.

Theorem 6.1. *If the* NP*-machine hypothesis is true, then there exists a set in* NP *that is* $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$*-complete, but is not* $\leq_{\mathrm{T}}^{\mathrm{P}/\log}$*-complete.*

Proof. Assume the NP-machine hypothesis. From Theorem 3.2, we obtain a set that is $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$-complete but not $\leq_{\mathrm{T}}^{\mathrm{P}}$-complete. By Theorem 5.1, this set cannot be $\leq_{\mathrm{T}}^{\mathrm{P}/\log}$-complete. □

We have the following corollary because the measure hypothesis implies the NP-machine hypothesis.

Corollary 6.2. *If* $\mu_{\mathrm{p}}(\mathrm{NP}) \neq 0$*, then there exists a set in* NP *that is* $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$*-complete, but is not* $\leq_{\mathrm{T}}^{\mathrm{P}/\log}$*-complete.*

We note that while Theorem 6.1 is stated for many-one vs. Turing, it applies to any reducibility in between.

Corollary 6.3. *If the* NP*-machine hypothesis is true, then for any reducibility* $\mathcal{R}$ *where* $\leq_{\mathrm{m}}^{\mathrm{P}}$*-reducibility implies* $\mathcal{R}$*-reducility and* $\mathcal{R}$*-reducility implies* $\leq_{\mathrm{T}}^{\mathrm{P}}$*-reducibility, there is a set in* NP *that is* $\leq_{\mathcal{R}}^{\mathrm{P/poly}}$*-complete, but is not* $\leq_{\mathcal{R}}^{\mathrm{P}/\log}$*-complete.*

It is natural to ask if we can separate completeness notions above P/poly many-one. We observe that for this, we will need stronger hypotheses than we have considered in this paper.

Proposition 6.4. *If there is a* $\leq_{\mathrm{T}}^{\mathrm{P/poly}}$*-complete set that is not* $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$*-complete in* NP*, then* NP $\not\subseteq$ P/poly*.*

Proof. If NP $\subseteq$ P/poly, then every set in NP is $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$-complete. □

The measure hypothesis and the NP-machine hypothesis are not known the imply NP $\not\subseteq$ P/poly. If it is possible to separate completeness notions above $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$, it appears an additional hypothesis at least as strong as NP $\not\subseteq$ P/poly – such as PH is infinite – would be needed.

7 Conclusion

We conclude by mentioning some open questions.

In Theorem 3.2, we separated $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$-completeness from $\leq_{\mathrm{T}}^{\mathrm{P}}$-completeness under the NP-machine hypothesis. Are these completeness notions incomparable?

Question 7.1. *Does the* NP*-machine hypothesis (or even* $\mu_{\mathrm{p}}(\mathrm{NP}) \neq 0$*) imply there is a* $\leq_{\mathrm{T}}^{\mathrm{P}}$*-complete set in* NP *that is not* $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$*-complete?*

Does the separation in Theorem 3.2 hold unconditionally for a larger complexity class?

Question 7.2. *Is there a* $\leq_{\mathrm{m}}^{\mathrm{P/poly}}$*-complete set in* EXP *that is not* $\leq_{\mathrm{T}}^{\mathrm{P}}$*-complete?*

Acknowledgement. We thank the anonymous referees for many helpful comments and suggestions.

References

[1] L. Adleman. Two theorems on random polynomial time. In *Proceedings of the 19th IEEE Symposium on Foundations of Computer Science*, pages 75–83, 1978.

[2] M. Agrawal. Pseudo-random generators and structure of complete degrees. In *Proceedings of the Seventeenth Annual IEEE Conference on Computational Complexity*, pages 139–147. IEEE Computer Society, 2002.

[3] M. Agrawal and O. Watanabe. One-way functions and the Berman-Hartmanis conjecture. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity*, pages 194–202, 2009.

[4] E. Allender. When worlds collide: Derandomization, lower bounds, and Kolmogorov complexity. In *Proceedings of the 21st Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 1–15. Springer-Verlag, 2001.

[5] E. Allender. The complexity of complexity. In *Computability and Complexity - Essays Dedicated to Rodney G. Downey on the Occasion of His 60th Birthday*, volume 10010 of *Lecture Notes in Computer Science*, pages 79–94, 2017.

[6] E. Allender, H. Buhrman, M. Koucký, D. van Melkebeek, and D. Ronneburger. Power from random strings. *SIAM Journal on Computing*, 35:1467–1493, 2006.

[7] K. Ambos-Spies and E. Mayordomo. Resource-bounded measure and randomness. In A. Sorbi, editor, *Complexity, Logic and Recursion Theory*, Lecture Notes in Pure and Applied Mathematics, pages 1–47. Marcel Dekker, New York, N.Y., 1997.

[8] K. Ambos-Spies, S. A. Terwijn, and X. Zheng. Resource bounded randomness and weakly complete problems. *Theoretical Computer Science*, 172(1–2):195–207, 1997.

[9] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. BPP has subexponential simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3:307–318, 1993.

[10] J. L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* Springer-Verlag, Berlin, second edition, 1995.

[11] L. Berman. On the structure of complete sets: Almost everywhere complexity and infinitely often speedup. In *Proceedings of the Seventeenth Annual Conference on Foundations of Computer Science*, pages 76–80, 1976.

[12] L. Berman and J. Hartmanis. On isomorphism and density of NP and other complete sets. *SIAM Journal on Computing*, 6(2):305–322, 1977.

[13] H. Buhrman, B. Hescott, S. Homer, and L. Torenvliet. Non-uniform reductions. *Theory of Computing Systems*, 47(2):317–341, 2010.

[14] H. Buhrman and D. van Melkebeek. Hard sets are hard to find. *Journal of Computer and System Sciences*, 59(2):327–345, 1999.

[15] S. Hirahara. Identifying an honest $\mathrm{EXP}^{\mathrm{NP}}$ oracle among many. In *Proceedings of the 30th Conference on Computational Complexity (CCC 2015)*, pages 244–263, 2015.

[16] J. M. Hitchcock and A. Pavan. Comparing reductions to NP-complete sets. *Information and Computation*, 205(5):694–706, 2007.

[17] J. M. Hitchcock and A. Pavan. Hardness hypotheses, derandomization, and circuit complexity. *Computational Complexity*, 17(1):119–146, 2008.

[18] J. M. Hitchcock and H. Shafei. Autoreducibility of NP-complete sets under strong hypotheses. *Computational Complexity*, 27(1):63–97, 2018.

[19] D. W. Juedes and J. H. Lutz. Weak completeness in E and E_2. *Theoretical Computer Science*, 143(1):149–158, 1995.

[20] V. Kabanets and J. Y. Cai. Circuit minimization problem. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 73–79, 2000.

[21] R. M. Karp and R. J. Lipton. Turing machines that take advice. *Enseign. Math.*, 28:191–201, 1982.

[22] A. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM Journal on Computing*, 31(5):1501–1526, 2002.

[23] J. H. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Sciences*, 44(2):220–258, 1992.

[24] J. H. Lutz. The quantitative structure of exponential time. In L. A. Hemaspaandra and A. L. Selman, editors, *Complexity Theory Retrospective II*, pages 225–254. Springer-Verlag, 1997.

[25] J. H. Lutz and E. Mayordomo. Cook versus Karp-Levin: Separating completeness notions if NP is not small. *Theoretical Computer Science*, 164(1–2):141–163, 1996.

[26] C. H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[27] A. Pavan. Comparison of reductions and completeness notions. *SIGACT News*, 34(2):27–41, June 2003.

[28] A. Pavan and A. L. Selman. Bi-immunity separates strong NP-completeness notions. *Information and Computation*, 188(1):116–126, 2004.

[29] L. Valiant and V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47(3):85–93, 1986.