Strong Reductions and Isomorphism of Complete Sets*

Ryan C. Harkins† John M. Hitchcock* A. Pavan ‡

Abstract

We study the structure of the polynomial-time complete sets for NP and PSPACE under strong nondeterministic polynomial-time reductions (SNP-reductions). We show the following results.

- If NP contains a p-random language, then all polynomial-time complete sets for PSPACE are SNP-isomorphic.
- If NP $\cap$ co-NP contains a p-random language, then all polynomial-time complete sets for NP are SNP-isomorphic.

1 Introduction

The celebrated Berman-Hartmanis isomorphism conjecture [BH77] states that all polynomial-time NP-complete sets are polynomial-time isomorphic. This conjecture can be naturally extended to other complexity classes. The isomorphism conjecture for a class $\mathcal{C}$ states that all polynomial-time complete sets for $\mathcal{C}$ are p-isomorphic. The evidence in support of this conjecture comes from the observation that for every natural complexity class, all known complete sets are polynomial-time isomorphic. On the other hand, it has been hypothesized that if one-way functions exist, then the isomorphism conjecture is false [JY85].

In spite of many years of research, we do not know of a single complexity class for which the isomorphism conjecture is resolved. This naturally led to the study of several variants of the conjecture that can be obtained by varying the resource bounds and types of the reducibilities. In most general terms, the conjecture for a class $\mathcal{C}$ and reducibilities r and s can be phrased as follows: "All r-complete sets for $\mathcal{C}$ are s-isomorphic."

This question has been studied extensively for resource bounds that are much smaller than polynomial-time that led to several exciting results. For example, we now know that all 1-L-complete sets for NP and PSPACE are p-isomorphic [All88, AB93]. Allender, Balcazar, and Immerman showed that all sets that are complete under first-order projections are DLOG-uniform AC^0-isomorphic [ABI97]. This result set the stage to investigate the structure of sets complete under AC^0-reductions. Successive papers [AAR98, AAI+01, Agr01b] improved this result, and this

*A preliminary version of this paper appeared in the proceedings of the 27th International Conference on Foundations of Software Technology and Theoretical Computer Science.

†Department of Computer Science, University of Wyoming. This research was supported in part by NSF grants 0515313, 0652601, and 0917417.

‡Department of Computer Science, Iowa State University. This research was supported in part by NSF grants 0430807, 0830479, and 0916797.

line of research culminated with the result of Agrawal [Agr01a]. This result states that all DLOG-uniform AC^0-complete sets for many natural classes are DLOG-uniform AC^0-isomorphic. Some of these results are surveyed in [KMR90, BT94, All01].

All above mentioned results concern sets that are complete under weaker reductions (i.e., where r has less resources than polynomial-time computation). In this paper, we study the isomorphism conjecture for polynomial-time complete sets. In particular we consider the following question: "Are the polynomial-time complete sets for a class s-isomorphic?"

For a long time, there was almost no progress on this question. Very recently Agrawal and Watanabe [AW09] obtained some exciting results. It has been conjectured that if f is a one-one, one-way function, then $f(\text{SAT})$ is not polynomial-time isomorphic to SAT. Recall that a polynomial-time computable function f is one-way, if every polynomial-time algorithm that attempts to invert f errs on a large fraction of instances. Agrawal and Watanabe observed that all known candidates for one-way functions have the following easy property: Even though we do not know an efficient algorithm that inverts on a large fraction of instances, they all admit easy to invert "cylinders". They showed that if every one-way function admits easy cylinders, then all polynomial-time NP-complete sets are P/poly-isomorphic*. We refer the reader to the original paper of Agrawal and Watanabe for the definition of easy cylinders. Since then, Goldreich [Gol09] exhibited a candidate one-way function that does not appear to have easy cylinders. This led Goldreich to conjecture that easy cylinders conjecture does not hold.

Given that it is not clear whether the easy cylinders conjecture holds or not, can we provide some additional evidence that polynomial-time complete sets admit (somewhat stronger) isomorphisms? In this paper we consider strong nondeterministic isomorphisms and P^{NP} isomorphisms.

Strong nondeterministic polynomial-time reductions (SNP-reductions for short) reductions were introduced by Adleman and Manders [AM77]. They showed that certain number-theoretic problems, which are not known to be polynomial-time NP-complete, are complete under SNP-reductions. Informally, these reductions can be thought as NP $\cap$ co-NP-reductions.

We show that if NP contains a p-random sequence, then all polynomial-time PSPACE-complete sets are SNP-isomorphic. This result also holds for any class that is closed under complement and union, in particular for all Δ-levels of the polynomial-time hierarchy. This hypothesis, which is equivalent to "NP does not have p-measure 0," is one of the most widely studied hypotheses in computational complexity and many plausible consequences are known to follow from it [Lut97, LM99]. From this result it follows that polynomial-time complete sets for PSPACE and all Δ-levels of the polynomial-time hierarchy are P^{NP}-isomorphic. With a stronger hypothesis we obtain similar consequences for the NP-complete sets. We show that if NP $\cap$ co-NP contains a p-random sequence, then all polynomial-time NP-complete sets are SNP-isomorphic.

To establish our isomorphism theorem, we first show that if NP does not have p-measure zero, then all polynomial-time complete sets for PSPACE are also complete via one-one, length-increasing SNP-reductions. This result could be of independent interest. We then use the resource-bounded analogue of the Cantor-Bernstein theorem to exhibit the isomorphism [BH77].

Our proofs use a bound on the longest consecutive run of 0's or 1's in a p-random sequence. In classical probability theory this result is proved using the Borel-Cantelli lemma [Dur04], but the proof does not carry over to polynomial-time randomness. Wang [Wan96] overcame this same problem for the law of the iterated logarithm. We use his technique to prove the bound on longest runs in the polynomial-time setting.

*Agrawal and Watanabe's result also holds for P/poly-complete sets.

This paper is organized as follows. Section 2 contains preliminaries on SNP-reductions and polynomial-time measure and randomness. In section 3 we present our main results. In section 4 we prove the longest runs bound. Section 5 concludes the paper with a discussion.

2 Preliminaries

In this paper we consider both single-valued and multi-valued functions. When f is a multi-valued function, $f(x)$ is a set. Recall that if f is a total, multi-valued function, then $f(x)$ is a *nonempty* set for all x. Unless otherwise mentioned all functions in this paper are total.

Definition. Let f be a multi-valued function. A function g is a *single-valued refinement* of f if g is single-valued function, and for every x, $g(x) \in f(x)$.

Definition. Let f be a multi-valued function. We say that f is *strong nondeterministic polynomial-time computable*, SNP-computable for short, if there is a nondeterministic polynomial-time machine M such that for every x, every path of M on x outputs a member of $f(x)$ or outputs a special symbol $\perp$. At least one path of $M(x)$ outputs a member of $f(x)$.

Definition. Let f be a total, multi-valued function and A and B be two languages. We say A *is reducible to* B *via* f if for every x the following conditions hold:

$$x \in A \Rightarrow f(x) \subseteq B,$$

$$x \notin A \Rightarrow f(x) \cap B = \emptyset.$$

Remark. Since we require the function f to be *total*, $f(x)$ can not be $\emptyset$ even when $x \notin A$.

Definition. A language A is SNP-*reducible* to a language B, if there is a (possibly multi-valued) function f that reduces A to B and f is SNP-computable.

Definition. A single-valued function f is an *isomorphism* from A to B, if f is a reduction from A to B and f is a bijection.

Recall that two languages A and B are *polynomial-time isomorphic* if there is a function f such that f reduces A to B, f^{-1} reduces B to A, both f and f^{-1} are polynomial-time computable, and f is a bijection. We can extend this definition to strong nondeterministic isomorphisms. When f is a multi-valued function $f^{-1}(y)$ is the set of all x for which $y \in f(x)$.

Definition. Let A be B be two languages. We say that A is *strong nondeterministic isomorphic* to B, SNP-isomorphic for short, if there is a (possibly multi-valued) function f such that following conditions hold:

- A reduces to B via f.
- B reduces to A via f^{-1}.
- Both f and f^{-1} are SNP-computable.
- There is a single-valued refinement g of f that is an isomorphism from A to B.

Observe that the definition implicitly requires f^{-1} to be a total function. We remark that there are several alternate ways to define the notion of SNP-isomorphism. We discuss these in section 5. We can define P^{NP}-isomorphisms similarly.

Definition. Two languages A and B are P^{NP}-isomorphic, if there is bijection $f : \Sigma^* \to \Sigma^*$ such that both f and f^{-1} are $\mathrm{FP}^{\mathrm{NP}}$-computable, f reduces A to B and f^{-1} reduces B to A.

We will also use the notion of honest reductions.

Definition. A function $f : \Sigma^* \to \Sigma^*$ is *honest* if there exists a constant $k \geq 1$ such that for all but finitely $x \in \Sigma^*$, $|f(x)| \geq |x|^{1/k}$.

We now review the definition of polynomial-time measure [Lut92]. The *Cantor space* $\mathbf{C}$ is the set of all infinite binary sequences. Each *language* (a subset of $\{0,1\}^*$) is identified with the element of Cantor space that is its characteristic sequence according to the standard enumeration of $\{0,1\}^*$. In this way, each complexity class (a set of languages) is viewed as a subset of Cantor space.

Definition. A function $d : \{0,1\}^* \to [0,\infty)$ is a *supermartingale* if for all $w \in \{0,1\}^*$,

$$d(w) \geq \frac{d(w0) + d(w1)}{2}. \tag{2.1}$$

A function $d : \{0,1\}^* \to [0,\infty)$ is a *martingale* if for all $w \in \{0,1\}^*$,

$$d(w) = \frac{d(w0) + d(w1)}{2}. \tag{2.2}$$

Intuitively, a martingale d starts with an initial amount $d(\lambda)$ of capital. The goal is to attain large values on sequences.

Definition. We say d *succeeds on* a sequence $S \in \mathbf{C}$ if

$$\limsup_{n\to\infty} d(S \upharpoonright n) = \infty.$$

Here $S \upharpoonright n$ is the length n prefix of S. The *success set* of d is

$$S^\infty[d] = \{S \in \mathbf{C} \mid d \text{ succeeds on } S\}.$$

The averaging condition (2.2) allows d to succeed on only a small set of sequences. More precisely, Ville [Vil39] showed that a class $X \subseteq \mathbf{C}$ has Lebesgue measure 0 if and only if there is a martingale d with $X \subseteq S^\infty[d]$. Polynomial-time measure [Lut92] arises from requiring efficiently computable martingales. As a martingale is a real-valued function, its values may not be exactly computable. Instead we allow polynomial-time approximations. Intuitively, the approximation computes $d(w)$ to r bits of precision in time bounded by a polynomial in $|w|$ and r.

Definition. We say $d : \{0,1\}^* \to [0,\infty)$ is $t(n)$-*time computable* if there is an approximation $\hat{d} : \mathbb{N} \times \{0,1\}^* \to \mathbb{Q}$ such that

(i) for all $r \in \mathbb{N}$ and $w \in \{0,1\}^*$, $|\hat{d}(r,w) - d(w)| \leq 2^{-r}$ and

(ii) $\hat{d}(r,w)$ is computable in $t(r + |w|)$ time.

If d is $q(n)$-time computable for some polynomial q, then d is polynomial-time computable.

Definition. Let $X \subseteq \mathbf{C}$.

1. X has *p-measure 0*, written $\mu_{\mathrm{p}}(X) = 0$, if there is a polynomial-time computable martingale d with $X \subseteq S^{\infty}[d]$.

2. X has *p-measure 1*, written $\mu_{\mathrm{p}}(X) = 1$, if $\mu_{\mathrm{p}}(X^c) = 0$.

Supermartingales give an equivalent definition: $\mu_{\mathrm{p}}(X) \neq 0$ if and only if there is a polynomial-time computable supermartingale d with $X \subseteq S^{\infty}[d]$ [Lut92]. We also use the notion of time-bounded randomness [ASTZ97].

Definition. Let L be a language.

1. L is *$t(n)$-random* if no $O(t(n))$-time computable martingale succeeds on L.

2. L is *p-random* if for every polynomial $p(n)$, L is $p(n)$-random.

The following result relates p-measure to p-randomness.

Lemma 2.1. ([ASTZ97, JL95]) *If C is a class that is closed under polynomial-time many-one reductions, then the following are equivalent.*

1. *C does not have p-measure 0.*

2. *C contains a p-random language.*

3 SNP Reductions and Isomorphisms

We prove our main theorem in this section. In our proof we use certain properties of p-random languages. Let R be a p-random language. Given a bit b and a finite string w, let $lr(b, w)$ denote the longest consecutive run of the bit b in w. Let $R \upharpoonright n$ denote the first n bits of the characteristic sequence of R.

Theorem 3.1. *If R is a p-random language, then for each $b \in \{0, 1\}$,*

$$\lim_{n\to\infty} \frac{lr(b, R \upharpoonright n)}{\log n} = 1.$$

The proof of Theorem 3.1 is in section 4.

Given a string y, let $r(y)$ be the rank (in lexicographic order) of y among strings of length $|y|$. Given a length n and index i, let s_i^n denote the string z such that $|z| = n$ and $r(z) = i$. Given a string y of length n, let $b_y = s^{n^2}_{2r(y)n^3}$ and $e_y = s^{n^2}_{2(r(y)+1)n^3-1}$. The following observation follows from Lemma 2.1 and Theorem 3.1.

Observation 3.2. *Assume that* NP *does not have p-measure zero. Then there is a p-random language R in* NP *such that for almost every y, the interval $[b_y, e_y]$ has at least one string from R.*

We say that a multi-valued function f is length-increasing if the length of x is smaller than the length of every string from $f(x)$. We say that a multi-valued function f is one-one if for every x and y with $x \neq y$, $f(x) \cap f(y) = \emptyset$.

We first show that if NP does not have p-measure zero, PSPACE-complete sets are complete via one-one, length-increasing, SNP-reductions.

Lemma 3.3. *If* NP *does not have p-measure 0, then all* PSPACE*-complete sets are complete via one-one, length-increasing* SNP*-reductions.*

Proof. Let L be any PSPACE-complete language. Let K be the standard PSPACE-complete language that is complete via one-one, length-increasing reductions. Observe that K can be decided in time 2^n. It suffices to show that K is reducible to L via a one-one, length-increasing SNP reduction. We first define an intermediate language A in PSPACE, and describe a one-one, length-increasing SNP reduction f from K to A. Then we describe a polynomial-time reduction from A to L that is one-one and length-increasing on $f(\Sigma^*)$. Combining these two reductions we obtain the desired reduction from K to L.

By our hypothesis, there is a n^4-random language R in NP.

$$A = \{\langle x, y\rangle \mid |x| = |y|^2, \text{ and } x \in R \oplus y \in K = 0\},$$

where $\oplus$ denotes the xor operation. Clearly, A is in PSPACE.

Claim 3.4. *There is a one-one, length-increasing* SNP *reduction from K to A.*

Proof. Since R is in NP, there is a polynomial-time computable function h and a polynomial $q(.)$ such that a string x is in R if and only if there is a witness w of length at most $q(|x|)$ for which $h(x, w) = 1$.

The following nondeterministic machine N is a reduction from K to A.

1. Input y, $|y| = n$.
2. Compute b_y and e_y.
3. Guess a string x_y between b_y and e_y and a possible witness w of length at most $q(n^2)$.
4. If $h(x_y, w) = 0$, then Output $\perp$ and this branch stops. If $h(x_y, w) = 1$, then output $\langle x_y, y\rangle$ and stop.

Let f be the function computed by N. We first show that f is a valid reduction from K to A. Observe that N outputs a tuple $\langle x_y, y\rangle$ only if $x_y \in R$. If $x_y \in R$, then y belongs to K if and only if $x_y \in R \oplus y \in K = 0$. Thus $y \in K$ if and only if $\langle x_y, y\rangle \in A$. Next we claim that at least one path of N does not output $\perp$.

By Observation 3.2, at least one string from the interval $[b_y, e_y]$ belongs to R. So at least one path of N guesses such string and a valid witness of that string. The output along this path is not $\perp$.

Thus f is a total, multi-valued function that reduces K to A. For every y, every element of $f(y)$ is of the form $\langle x_y, y\rangle$, where x_y is a string of length n^2. Thus f is length-increasing. Let y and z be two distinct strings. Every element of $f(y)$ is of the form $\langle ., y\rangle$ and every element of $f(z)$ is of the form $\langle ., z\rangle$. Thus $f(y) \cap f(z) = \emptyset$. Thus f is one-one.

This completes proof of Claim 3.4. □

Since A is in PSPACE and L is PSPACE-complete, there is a polynomial-time many-one reduction g from A to L. We now show that g must be one-one and honest on $f(\Sigma^*)$. Observe that every string v in $f(\Sigma^*)$ is of the form $\langle x, y\rangle$, where $|x| = |y|^2$. We first observe that f satisfies the following stronger one-one property.

Observation 3.5. *Let $y_1 < y_2$, $f(y_1) = \langle x_1, y_1\rangle$, and $f(y_2) = \langle x_2, y_2\rangle$. Then $x_1 < x_2$.*

Proof. Since $y_1 < y_2$, $e_{y_1} < b_{y_2}$, the intervals $[b_{y_1}, e_{y_1}]$ and $[b_{y_2}, e_{y_2}]$ are disjoint. Observe that x_1 belongs to the interval $[b_{y_1}, e_{y_1}]$ and x_2 belongs to the interval $[b_{y_2}, e_{y_2}]$. Thus $x_1 < x_2$. □

We first show that g must be one-one on $f(\Sigma^*)$.

Claim 3.6. *For all but finitely many strings u and v in $f(\Sigma^*)$, $g(u) \neq g(v)$.*

Proof. We have to show that the following set is finite.

$$S = \{u \in f(\Sigma^*) \mid \exists v \in f(\Sigma^*), u \neq v, g(u) = g(v)\}.$$

Suppose that $u \neq v$ for two tuples $u = \langle x_2, y_2\rangle \in f(\Sigma^*)$ and $v = \langle x_1, y_1\rangle \in f(\Sigma^*)$. From Observation 3.5 it follows that $x_1 \neq x_2$. Therefore we can rewrite S as

$$S = \{\langle x_2, y_2\rangle \in f(\Sigma^*) \mid \exists \langle x_1, y_1\rangle \in f(\Sigma^*), x_1 \neq x_2, g(\langle x_1, y_1\rangle) = g(\langle x_2, y_2\rangle)\}.$$

Assume that S is infinite. Then the set

$$T = \{\langle x_2, y_2\rangle \in f(\Sigma^*) \mid \exists \langle x_1, y_1\rangle \in f(\Sigma^*), x_1 < x_2, g(\langle x_1, y_1\rangle) = g(\langle x_2, y_2\rangle)\}$$

is also infinite. We will show that this contradicts the randomness of R.

Consider the following strategy for a martingale d that bets on R. Let $d(n)$ denote the capital d has after betting on strings on length n. If n is not a perfect square, then d does not bet on strings of length n and we have $d(n) = d(n-1)$. Suppose that n is a perfect square. Before betting on strings of length n, d searches for two tuples $\langle x_1, y_1\rangle$ and $\langle x_2, y_2\rangle$ with the following properties.

- $|x_2| = n = |y_2|^2$.
- $x_1 < x_2$, $|x_1| = |y_1|^2$.
- $g(\langle x_1, y_1\rangle) = g(\langle x_2, y_2\rangle)$.

Because T is infinite, d will find such tuples for infinitely many n. If d does not find such tuples, then it does not bet on any string at length n and we have $d(n) = d(n-1)$. Suppose d finds such tuples. Then d does not bet on any string up to x_2. Recall that when d is ready to bet on x_2, it has access to the partial characteristic sequence of R up to x_2. Thus at this point d knows the membership of x_1 in R. Next d computes the membership of y_1 and y_2 in K. Since $g(\langle x_1, y_1\rangle) = g(\langle x_2, y_2\rangle)$,

$$(x_1 \in R \oplus y_1 \in K) = (x_2 \in R \oplus y_2 \in K)$$

Since d knows the values of $x_1 \in R$, $y_1 \in K$, and $y_2 \in K$, it can determine the value of $x_2 \in R$. Thus d bets on x_2 accordingly. This way d can double its capital. Thus we have $d(n) = 2d(n-1)$.

Thus for every n either $d(n) = d(n-1)$ or $d(n) = 2d(n-1)$, and for infinitely many n, $d(n) = 2d(n-1)$. Thus $d(n)$ approaches infinity as n tends to ∞ and d succeeds on R.

Observe that the time taken by d to search for the tuples with desired properties is bounded by $O(2^{4n})$. In addition, d needs at most $O(2^{\sqrt{n}})$ time to decide membership of y_1 and y_2 in K. This is because K is in DTIME(2^n) and lengths of y_1 and y_2 are bounded by $\sqrt{n}$. Recall that the running time of d is measured with respect to the length of the partial characteristic sequence, thus d runs in time $O(n^4)$.

Therefore if S is infinite, R is not n^4-random. Thus g is one-one on strings from $f(\Sigma^*)$ and the proof of Claim 3.6 is complete. □

Next we show that any reduction from A to L must be honest. Since the complete set L is in PSPACE, there is a constant k such that L can be decided in time 2^{n^k}.

Claim 3.7. *Let g be a reduction from A to L. Let $T = \{\langle x, y\rangle \mid |x| = |y|^2\}$. For all but finitely many strings $w = \langle x, y\rangle$ from T, $|g(w)| \geq |x|^{1/k}$.*

Proof. Let U be the set of strings $w = \langle x, y\rangle$ from T for which $|g(w)| < |y|^{1/k}$. We show that if U is infinite, then R is not n^4-random.

Consider the following strategy for a martingale d. Let $d(n-1)$ be the capital that d has before it starts to bet on strings of length n. Before betting on strings of length n, the martingale searches for a tuple $w = \langle x, y\rangle$ in U with $n = |x| = |y|^2$. If no such tuple exists, then d does not bet on any strings at length n. In this case, $d(n) = d(n-1)$.

By our assumption that U is infinite, d finds such a tuple for infinitely many n. Upon finding a tuple in $w \in U$, d determines the membership of $w \in A$ by computing the membership of $g(w) \in L$. Then d knows $x \in R \oplus y \in K$. Now d can decide the membership of y in K and infer the membership of x in R. Thus $d(n) = 2d(n-1)$.

If U is infinite, then $d(n) = 2d(n-1)$ for infinitely many n. Thus d succeeds on R. The time taken by d can be bounded as follows. It takes $O(2^{2n})$ time to search for a tuple w in U. Once w is found, it decides the membership of w in A by deciding the membership of $g(w)$ in L. Since $w \in U$, $|g(w)| < n^{1/k}$. Since L can be decided in 2^{n^k} time, this step takes $O(2^n)$ time. Since $|y| = \sqrt{n}$ and K is in DTIME(2^n), membership of $y \in K$ can be computed in $O(2^n)$ time. Thus the running time of the martingale, when measured with respect to the length of the characteristic sequence, is bounded by $O(n^2)$. Thus R is not n^4-random.

This completes the proof of Claim 3.7. □

Now we will complete the proof of Lemma 3.3. By Claim 3.4, there is a one-one, length-increasing SNP-reduction f from K to A. By Claims 3.6 and 3.7, there is a polynomial-time reduction g from A to L that is one-one and honest on strings from $f(\Sigma^*)$. Combining the reduction f with g, we obtain a one-one, honest reduction from K to L. Since K is paddable, we conclude that there is a one-one, length-increasing, SNP reduction from K to L.

Thus all PSPACE-complete sets are complete via one-one, length-increasing, SNP-reductions under the assumption NP does not have p-measure 0. □

We are now ready to prove our isomorphism theorem for PSPACE. We start with the following easy to prove observation.

Observation 3.8. *Let f be a length-increasing* SNP*-computable function. There is a nondeterministic polynomial-time machine M such that for every y that has an inverse, every path of $M(y)$ either outputs $\perp$ or outputs a member of $f^{-1}(y)$, and at least one path outputs a member of $f^{-1}(y)$. If $f^{-1}(y)$ does not exist, then every path of M outputs $\perp$.*

Theorem 3.9. *If* NP *does not have p-measure zero, then all polynomial-time many-one complete sets for* PSPACE *are* SNP*-isomorphic.*

Proof. Let A and B be any two PSPACE-complete sets. By Lemma 3.3, there is a one-one, length-increasing SNP-reduction f from A to B, and similarly there is a one-one, length-increasing SNP-reduction g from B to A.

Consider the following multi-valued function h: If if $g^{-1}(x)$ exists, $h(x) = f(x) \cup \{g^{-1}(x)\}$, else $h(x) = f(x)$. Observe that since g is a one-one function, $g^{-1}(x)$, if it exists, is unique.

By Observation 3.8, there is a nondeterministic machine N that computes g^{-1}. Consider the following nondeterministic machine. On input x, it guesses a bit $b \in \{0, 1\}$. If $b = 0$, then it simulates the SNP-machine that computes f. If $b = 1$, then it simulates N. If $g^{-1}(x)$ exists, then the output set of this machine is exactly $f(x) \cup \{g^{-1}(x)\}$. If $g^{-1}(x)$ does not exist, then the output set of this machine is $f(x)$. Thus h is SNP-computable. Observe that $h^{-1}(x) = g(x) \cup f^{-1}(x)$. Thus it follows that h^{-1} is also SNP-computable.

The value of $h(x)$ is either $f(x)$ or $f(x) \cup \{g^{-1}(x)\}$. Since f is a reduction from A to B and g is a reduction from B to A, it follows that h is a reduction from A to B, and h^{-1} is a reduction from B to A.

We now exhibit a single-valued refinement of h that is an isomorphism between A and B. Let $f_s(x)$ denote the smallest element of $f(x)$, and $g_s(x)$ denote the smallest element of $g(x)$. Observe the f_s and g_s are one-one, length-increasing, single-valued functions.

Given a string x of length n, consider the following sequence.

$$S_x = g_s^{-1}(x), f_s^{-1}(g_s^{-1}(x)), g_s^{-1}(f_s^{-1}(g_s^{-1}(x))), \cdots$$

The sequence stops when either g_s^{-1} or f_s^{-1} does not exist. Since both f_s and g_s are length-increasing, f_s^{-1} and g_s^{-1} are length-decreasing. Thus the above sequence contains at most n strings.

Consider the following function e. If S_x has even number of elements then $e(x) = f_s(x)$, else $e(x) = g_s^{-1}(x)$. Clearly, e is single-valued. Consider the case S_x has odd number of elements. In this case $g^{-1}(x)$ must exist. Thus $h(x) = f(x) \cup \{g^{-1}(x)\}$. Hence, if S_x has odd number of elements, then $e(x) \in h(x)$. Observe that for every x, $f(x) \subseteq h(x)$. Thus if S_x has even number of elements, then $e(x) = f_s(x) \in h(x)$. Thus e is a single-valued refinement of h.

It remains to show that e is an isomorphism from A to B. The proof of this is exactly the same as the proof given by Berman and Hartmanis [BH77], so we omit the details here.

Thus A and B are SNP-isomorphic. This completes the proof of Theorem 3.9. □

We observe that the isomorphism exhibited in the above proof can be computed in P^{NP}. This yields the following result.

Theorem 3.10. *If* NP *does not have p-measure zero, then all polynomial-time* PSPACE*-complete sets are* P^{NP}*-isomorphic.*

Observe that the above proof applies to any class that is closed under $\oplus$ operation. In particular, it applies to all Δ-levels of the polynomial-time hierarchy.

Theorem 3.11. *Assume that* NP *does not have p-measure zero. For every $k \geq 2$, all sets that are polynomial-time complete for Δ_k^{P} are* SNP*-isomorphic and* P^{NP}*-isomorphic.*

We next consider whether we can prove a similar result for NP-complete sets. We need a stronger hypothesis to do this.

Theorem 3.12. *If* NP ∩ co-NP *does not have p-measure zero, then all polynomial-time complete sets for* NP *are* SNP*-isomorphic.*

For the most part, the the structure of the proof is similar to the proof of Theorem 3.9. We can first prove that all NP-complete sets are complete via one-one, length-increasing, SNP-reductions. For this we define an intermediate language A and argue that there is a one-one, length-increasing reduction from SAT to A and a one-one, length-increasing reduction from A to the desired NP-complete language. The main difference is in definition of the intermediate language A. Here we define the intermediate language A as

$$A = \{\langle x, y, z\rangle \mid |x| = |z| = |y|^2, \mathrm{maj}\{x \in R, y \in \mathrm{SAT}, z \in R\} = 1\}.$$

This ensures that A is also in NP. The remainder of the proof uses similar ideas.

4 Longest Runs and Polynomial-Time Randomness

In this section we establish bounds on the number of consecutive zeros or ones that appear in the characteristic sequence of a p-random language and prove Theorem 3.1.

Define the function $lr(b, w)$ as the function that, given a finite string w, returns the length of the longest consecutive run of the bit b in w. For each bit b, let

$$X_b = \left\{ S \in \{0,1\}^\infty \;\middle|\; \lim_{n\to\infty} \frac{lr(b, S\restriction n)}{\log n} = 1 \right\},$$

and let $X = X_0 \cap X_1$. The Borel-Cantelli lemma can be used to show that X has Lebesgue measure 1 [Dur04]. While there is a polynomial-time version of the Borel-Cantelli lemma [Lut92], it does not apply to show that X has p-measure 1 because of problems with polynomial-time convergence. Wang [Wan96] observed a similar problem with adapting the law of the iterated logarithm to p-measure and developed another approach. We now use Wang's technique to prove that X has p-measure 1.

Theorem 4.1. $\mu_{\mathrm{p}}(X) = 1$.

Theorem 3.1 is immediate from Theorem 4.1.

Proof of Theorem 4.1. It suffices to show that $\mu_{\mathrm{p}}(X_0^c) = 0$ and $\mu_{\mathrm{p}}(X_1^c) = 0$. We will only show the latter, as the other statement follows by a symmetric argument.

Let

$$X^+ = \left\{ S \in \{0,1\}^\infty \;\middle|\; \limsup_{n\to\infty} \frac{lr(1, S\restriction n)}{\log n} > 1 \right\}$$

and

$$X^- = \left\{ S \in \{0,1\}^\infty \;\middle|\; \liminf_{n\to\infty} \frac{lr(1, S\restriction n)}{\log n} < 1 \right\},$$

so that $X_1^c = X^+ \cup X^-$.

For any $S \in X^+$, there is an $\epsilon > 0$ such that $lr(1, S\restriction n) > (1+\epsilon)\log n$ for infinitely many n. Therefore

$$X^+ = \bigcup_{\epsilon>0} X_{(1+\epsilon)\log n},$$

where

$$X_f = \{S \in \{0,1\}^\infty \mid (\exists^\infty n \in \mathbb{N})\ S \upharpoonright n \text{ ends in at least } f(n) \text{ 1's}\}.$$

More generally, we have $X^+ \subseteq X_{\log n + 2\log\log n}$, so the following lemma implies $\mu_{\mathrm{p}}(X^+) = 0$.

Lemma 4.2. *Let $f(n) = \log n + 2\log\log n$. Then $\mu_{\mathrm{p}}(X_f) = 0$.*

Proof of Lemma 4.2. We define a sequence of martingales $\{d_n\}_{n\geq 2}$. The initial capital of d_n is $d_n(\lambda) = 2^{-f(n)}$. For any string w with $|w| < n - f(n)$ or $|w| \geq n$,

$$d_n(w0) = d_n(w1) = d_n(w).$$

For a string w with $n - f(n) \leq |w| < n$,

$$\begin{aligned} d_n(w0) &= 0, \\ d_n(w1) &= 2d_n(w) \end{aligned}$$

In other words, each d_n only bets on the last $f(n)$ bits on a string of size n. For each of these $f(n)$ bits, d_n bets its entire capital on 1. For any w of length n, we have $d_n(w) = 0$ or $d_n(w) = 1$:

$$d_n(w) = \begin{cases} 1 & \text{if } lr(1,w) \geq f(n), \\ 0 & \text{otherwise.} \end{cases}$$

It is clear that each d_n is a p-martingale. Let

$$d(w) = \sum_{n=2}^{\infty} d_n(w).$$

For a sequence $S \in X_f$, since there are infinitely many n such that $lr(1, S\upharpoonright n) \geq f(n)$, there will be infinitely many d_n that reach a capital of 1 on S. Thus $X_f \subseteq S^\infty[d]$. However, we need to show that d is a p-martingale. The averaging condition is immediate by linearity and

$$d(\lambda) = \sum_{n=2}^{\infty} d_n(\lambda) = \sum_{n=2}^{\infty} 2^{-f(n)} \leq \sum_{n=2}^{\infty} 2^{-\log n - 2\log\log n} = \sum_{n=2}^{\infty} \frac{1}{n\log^2 n} < \infty.$$

However, d is not p-approximable because this series converges too slowly to be p-convergent: the number of terms required to approximate the sum within 2^{-r} is not bounded by a polynomial in r. We circumvent this problem by defining a supermartingale $\hat{d}$ that is p-approximable and $S^\infty[d] \subseteq S^\infty[\hat{d}]$.

Given any input length, only a finite number of the martingales d_n bet on the next bit. Let $h(i)$ be the largest index such that $d_{h(i)}$ bets on the $(i+1)^{\text{th}}$ bit. In other words, $h(i)$ is the largest m such that $d_m(w1) \neq d_m(w)$ when $|w| = i$. By definition, d_m bets on the $(i+1)^{\text{th}}$ bit if and only if $m > i \geq m - f(m)$. In particular, for sufficiently large i and all $m \geq i^2$, we have $m - f(m) > i$ and d_m does not bet. Therefore $h(i) \leq i^2$ is an upper bound.

Note that if $m > h(i)$ and $|w| = i$, then $d_m(w) = 2^{-f(m)}$.

Because $h(i)$ is bounded by a polynomial, we can directly calculate the sum of the first $h(i)$ martingales in polynomial time. As demonstrated by Wang [Wan96], the sum total of the other

martingales can be overestimated with an integral. If we consider each discrete value to be a rectangular area in a Riemann sum, we have:

$$\sum_{n=h(i)+1}^{\infty} 2^{-f(n)} \leq \int_{h(i)}^{\infty} 2^{-\log x - 2\log\log x} dx.$$

We then define our supermartingale $\hat{d}$ by

$$\hat{d}(w) = \sum_{n=2}^{h(|w|)} d_n(w) + \int_{h(|w|)}^{\infty} \frac{1}{x \log^2 x} dx.$$

The sum has a polynomial number of terms which can be approximated in polynomial time. The integral evaluates to

$$\frac{\ln 2}{\log h(|w|)},$$

which can be approximated in polynomial time. Therefore $\hat{d}$ has a polynomial-time approximation.

In order for $\hat{d}$ to define a supermartingale, it must hold that $2\hat{d}(w) \geq \hat{d}(w0) + \hat{d}(w1)$, and in order to be useful, it must hold that $S^{\infty}[d] \subseteq S^{\infty}[\hat{d}]$. If

$$\hat{d}(w) - d(w) = \int_{h(|w|)}^{\infty} \frac{1}{x \log^2 x} dx \ - \sum_{n=h(|w|)+1}^{\infty} \frac{1}{n \log^2 n} > 0,$$

then whenever d is unbounded on a sequence, $\hat{d}$ must also be unbounded. Since the sum is difficult to evaluate, we instead consider the difference of a unit interval of the integral and a single term of the sum:

$$\int_{n-1}^{n} \frac{1}{x \log^2 x} dx - \frac{1}{n \log^2 n} = \frac{\ln 2}{\log(n-1)} - \frac{\ln 2}{\log n} - \frac{1}{n \log^2 n}.$$

This difference is greater than zero, which is easily seen as each unit interval of the integral by definition overapproximates each term of the sum. As the integral and the sum each evaluate to finite amounts, the sum of the differences is also finite. Thus $\hat{d}(w) \geq d(w)$. To show that $2\hat{d}(w) \geq \hat{d}(w1) + \hat{d}(w0)$, we have:

$$\begin{aligned}
\hat{d}(w0) + \hat{d}(w1) &= \sum_{n=2}^{h(|w|+1)} d_n(w0) + \sum_{n=2}^{h(|w|+1)} d_n(w1) + 2\int_{h(|w|+1)}^{\infty} \frac{1}{x \log^2 x} dx \\
&= 2\sum_{n=2}^{h(|w|+1)} d_n(w) + 2\int_{h(|w|+1)}^{\infty} \frac{1}{x \log^2 x} dx \\
&= 2\sum_{n=2}^{h(|w|)} d_n(w) + 2\sum_{n=h(|w|)+1}^{h(|w|+1)} d_n(w) + 2\int_{h(|w|+1)}^{\infty} \frac{1}{x \log^2 x} dx \\
&\leq 2\sum_{n=2}^{h(|w|)} d_n(w) + 2\int_{h(|w|)}^{h(|w|+1)} \frac{1}{x \log^2 x} dx + 2\int_{h(|w|+1)}^{\infty} \frac{1}{x \log^2 x} dx \\
&= 2\sum_{n=2}^{h(|w|)} d_n(w) + 2\int_{h(|w|)}^{\infty} \frac{1}{x \log^2 x} dx \\
&= 2\hat{d}(w).
\end{aligned}$$

(In the case when $h(|w|) = h(|w|+1)$, the fourth line holds with equality.)

Therefore $\hat{d}$ is a polynomial-time supermartingale such that $S^\infty[d] \subseteq S^\infty[\hat{d}]$ and the proof of Lemma 4.2 is complete. □

In a similar fashion, we define

$$Y_f = \{w \in \{0,1\}^\infty \mid (\exists^\infty n \in \mathbb{N}) lr(1, w\restriction n) < f(n)\},$$

and note that $X^- = \bigcup_{\epsilon>0} Y_{(1-\epsilon)\log n}$. Also, for $f > g$, $Y_g \subseteq Y_f$, so the following is sufficient to show $\mu_{\mathrm{p}}(X^-) = 0$.

Lemma 4.3. *Let $f(n) = \log n - 2\log\log n$. Then $\mu_{\mathrm{p}}(Y_f) = 0$.*

Proof of Lemma 4.3. We define a sequence of martingales $\{d_n\}_{n\geq 1}$ such that d_n only bets on the first n bits of a string via a simple polynomial-time computable betting strategy. Each martingale d_n divides the first n bits of the string into blocks of size $f(n)$ (if there is a remainder, it is assumed to be at the head of the string). For each block, d_n divides its capital evenly among all possible combinations of bits that are not all 1's. Thus if that block somewhere contains a 0, d_n will have a small return, but if the block is all 1's, d_n will lose all of its capital.

Formally, for $|w| \geq n$, if each block contains a 0, we have by definition that

$$d_n(w) = d_n(\lambda)\left(\frac{2^{f(n)}}{2^{f(n)}-1}\right)^{\frac{n}{f(n)}}.$$

Setting the initial capital to be

$$d_n(\lambda) = \left(\frac{2^{f(n)}-1}{2^{f(n)}}\right)^{\frac{n}{f(n)}},$$

we have $d_n(w) = 1$. We then define

$$d(w) = \sum_{n=2}^{\infty} d_n(w).$$

For d to be a martingale, we need only show that $\sum d_n(\lambda)$ converges. We have

$$\left(\frac{2^{f(n)}-1}{2^{f(n)}}\right)^{\frac{n}{f(n)}} = \left(\frac{\frac{n}{\log^2 n}-1}{\frac{n}{\log^2 n}}\right)^{\frac{n}{\log n - 2\log\log n}} = 2^{-n\left[\frac{\log n - \log(n-\log^2 n)}{\log n - 2\log\log n}\right]}.$$

We note that for $0 < \epsilon < 1$,

$$\frac{1}{n^\epsilon} \in o\left(\left[\frac{\log n - \log(n-\log^2 n)}{\log n - 2\log\log n}\right]\right)$$

so that

$$2^{-n\left[\frac{\log n - \log(n-\log^2 n)}{\log n - 2\log\log n}\right]} \leq 2^{-\frac{n}{n^\epsilon}} = 2^{-n^{1-\epsilon}} = 2^{-n^\delta}.$$

Since $\sum_{n=2}^{\infty} 2^{-n^\delta}$ converges for all $\delta > 0$, we have $\sum_{n=2}^{\infty} d_n(\lambda) < \infty$.

Thus d is a martingale, and it is clear that $Y_f \subseteq S^\infty[d]$. It follows that d is p-computable, because the series $\sum_{n=2}^{\infty} 2^{-n^\delta}$ is p-convergent [Lut92].

Thus $\mu_{\mathrm{p}}(Y_f) = 0$, completing the proof of Lemma 4.3. □

Since $\mu_{\mathrm{p}}(X^+) = 0$ and $\mu_p(X^-) = 0$, we have that $\mu_{\mathrm{p}}(X_1^c) = 0$, completing the proof of Theorem 4.1. □

5 Discussion

This paper initiates the study of structure of polynomial-time complete sets under more powerful SNP reductions. We now briefly discuss a few interesting questions raised by our results.

As mentioned in the preliminaries, there are several ways of defining the notion of SNP-isomorphism. Our current definition asks for a function h such that both h and h^{-1} are SNP-computable and some single valued-refinement of h is an isomorphism. Perhaps a more natural definition would the following: A set A is SNP-isomorphic to B if there is a (multi-valued) function h such that h reduces A to B, h^{-1} reduces B to A, both h and h^{-1} are SNP-computable, and h is bijection. A multi-valued function $h : \Sigma^* \to \Sigma^*$ is a bijection if every $y \in \Sigma^*$ has an inverse and $h(x) \cap h(y) = \emptyset$ for every x that is not equal to y. Another way of defining SNP-isomorphism is to require that h is a *single-valued* SNP-computable function.

Can we prove that PSPACE-complete sets or NP-complete sets are SNP-isomorphic using these definitions? One way to achieve this is to strengthen Lemma 3.3 to the following: If the p-measure of NP is not zero, then PSPACE-complete sets are complete via monotone, length-increasing, SNP reductions.

We note that we can obtain an affirmative answer to this question for EXP. It is known that polynomial-time EXP-complete sets are complete via one-one, length-increasing reductions [Ber77]. A function f is monotone if $f(x) < f(y)$ whenever $x < y$. It is easy to modify Berman's proof to show that polynomial-time EXP-complete sets are complete via monotone, polynomial-time reductions. Thus we unconditionally obtain that all EXP-complete sets are single-valued SNP-isomorphic.

Ideally, we would like the resource bounds of isomorphisms and the reductions to be the same. Can we show that all SNP-complete sets for PSPACE are SNP-isomorphic? How about p-isomorphisms? Can we prove or disprove the isomorphism conjecture under the measure hypothesis?

Finally, can we show that NP-complete sets or PSPACE-complete sets are complete via one-one, length-increasing, polynomial-time computable reductions? There have been several partial results on this question [Agr02, HP07, BHHT10, GHP12].

Acknowledgments. We thank the anonymous referees for helpful comments.

References

[AAI+01] A. Agrawal, E. Allender, R. Impagliazzo, T. Pitassi, and S. Rudich. Reducing the complexity of reductions. *Computational Complexity*, 10:117–138, 2001.

[AAR98] M. Agrawal, E. Allender, and S. Rudich. Reductions in circuit complexity: An isomorphism theorem and a gap theorem. *Journal of Computer and System Sciences*, 57(2):127–143, 1998.

[AB93] M. Agrawal and S. Biswas. Polynomial-time isomorphism of 1-L complete sets. In *Proceedings of Structure in Complexity Theory*, pages 75–80, 1993.

[ABI97] E. Allender, J. Balcazar, and N. Immerman. A first-order isomorphism theorem. *SIAM Journal on Computing*, 26:557–567, 1997.

[Agr01a] M. Agrawal. The first-order isomorphism theorem. In *Foundations of Software Technology and Theoretical Computer Science*, pages 70–82, 2001.

[Agr01b] M. Agrawal. Towards uniform AC^0-isomorphisms. In *Proceedings of 16th IEEE Conference on Computational Complexity*, pages 13–20, 2001.

[Agr02] M. Agrawal. Pseudo-random generators and structure of complete degrees. In *17th Annual IEEE Conference on Computational Complexity*, pages 139–145, 2002.

[All88] E. Allender. Isomorphisms and 1-L reductions. *Journal of Computer and System Sciences*, 36:336–350, 1988.

[All01] E. Allender. Some pointed questions concerning asymptotic lower bounds, and new from the isomorphism front. In G. Paun, G. Rozenberg, and A. Salomaa, editors, *Current Trends in Theoretical Computer Science: Entering the 21st Century*, pages 25–41. Scientific Press, 2001.

[AM77] L. Adleman and K. Manders. Reducibility, randomness, and intractability. In *Proceedings of the 9th Annual ACM Symposium on Theory of Computing*, pages 151–163, 1977.

[ASTZ97] 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.

[AW09] M. Agrawal and O. Watanabe. One-way functions and Berman-Hartmanis conjecture. In *24th IEEE Conference on Computational Complexity*, pages 194–202, 2009.

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

[BH77] L. Berman and H. Hartmanis. On isomorphisms and density of NP and other complete sets. *SIAM J. Comput.*, 6:305–322, 1977.

[BHHT10] H. Buhrman, B. Hescott, S. Homer, and L. Torenvliet. Non-uniform reductions. *Theory of Computing Systems*, 47(2):317–341, 2010.

[BT94] H. Buhrman and L. Torenvliet. On the structure of complete sets. In *9th IEEE Annual Conference on Structure in Complexity Theory*, pages 118–133, 1994.

[Dur04] R. Durrett. *Probability: Theory and Examples.* Duxbury Press, third edition, 2004.

[GHP12] X. Gu, J. M. Hitchcock, and A. Pavan. Collapsing and separating completeness notions under average-case and worst-case hypotheses. *Theory of Computing Systems*, 51(2):248–265, 2012.

[Gol09] O. Goldreich. A candidate counter example to the easy cylinders conjecture. Technical Report TR09-028, ECCC, 2009.

[HP07] J. M. Hitchcock and A. Pavan. Comparing reductions to NP-complete sets. *Information and Computation*, 205(5):694–706, 2007.

[JL95] D. W. Juedes and J. H. Lutz. Weak completeness in E and E_2. *Theoretical Computer Science*, 143(1):149–158, 1995.

[JY85] D. Joseph and P. Young. Some remarks on witness functions for non- polynomial and non-complete sets in NP. *Theoretical Computer Science*, 39:225–237, 1985.

[KMR90] S. Kurtz, S. Mahaney, and J. Royer. The structure of complete degrees. In A. Selman, editor, *Complexity Theory Retrospective*, pages 108–146. Springer-Verlag, 1990.

[LM99] J. H. Lutz and E. Mayordomo. Twelve problems in resource-bounded measure. *Bulletin of the European Association for Theoretical Computer Science*, 68:64–80, 1999. Also in *Current Trends in Theoretical Computer Science: Entering the 21st Century*, pages 83–101, World Scientific Publishing, 2001.

[Lut92] J. H. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Sciences*, 44(2):220–258, 1992.

[Lut97] 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.

[Vil39] J. Ville. *Étude Critique de la Notion de Collectif.* Gauthier–Villars, Paris, 1939.

[Wan96] Y. Wang. The law of the iterated logarithm for p-random sequences. In *Proceedings of the Eleventh Annual IEEE Conference on Computational Complexity*, pages 180–189. IEEE Computer Society, 1996.