The Isomorphism Conjecture for NP

Manindra Agrawal[*]

December 19, 2009

Abstract

In this article, we survey the arguments and known results for and against the Isomorphism Conjecture.

1 Introduction

The Isomorphism Conjecture for the class NP states that all polynomial-time many-one complete sets for NP are polynomial-time isomorphic to each other. It was made by Berman and Hartmanis [21][1], inspired in part by a corresponding result in computability theory for computably enumerable sets [50], and in part by the observation that all the existing NP-complete sets known at the time were indeed polynomial-time isomorphic to each other. This conjecture has attracted a lot of attention because it predicts a very strong structure of the class of NP-complete sets, one of the fundamental classes in complexity theory.

After an initial period in which it was believed to be true, Joseph and Young [40] raised serious doubts against the conjecture based on the notion of *one-way* functions. This was followed by investigation of the conjecture in relativized worlds [33, 46, 27] which, on the whole, also suggested that the conjecture may be false. However, disproving the conjecture using one-way functions, or proving it, remained very hard (either implies $\mathsf{P} \neq \mathsf{NP}$). Hence research progressed in three distinct directions from here.

The first direction was to investigate the conjecture for complete degrees of classes bigger than NP. Partial results were obtained for classes EXP and NEXP [20, 29].

The second direction was to investigate the conjecture for degrees other than complete degree. For degrees within the 2-*truth-table-complete degree* of EXP, both possible answers to the conjecture were found [44, 43, 41].

The third direction was to investigate the conjecture for reducibilities weaker than polynomial-time. For several such reducibilities it was found that the isomorphism conjecture, or something close to it, is true [16, 1, 8, 2].

These results, especially from the third direction, suggest that the Isomorphism Conjecture for the class NP may be true contrary to the evidence from the relativized world. A recent work [12] shows that if all one-way functions satisfy a certain property then a non-uniform version of the conjecture is true.

An excellent survey of the conjecture and results related to the first two directions is in [45].

[*] N Rama Rao Professor, Indian Institute of Technology, Kanpur. Research supported by J C Bose Fellowship FLW/DST/CS/20060225.

[1] The conjecture is also referred as *Berman-Hartmanis Conjecture* after the proposers.

2 Definitions

In this section, we define most of the notions that we will need.

We fix the alphabet to $\Sigma = \{0,1\}$. Σ^* denotes the set of all finite strings over Σ and Σ^n denotes the set of strings of size n. We start with defining the types of functions we use.

Definition 2.1. *Let r be a resource bound on Turing machines. Function f, $f : \Sigma^* \mapsto \Sigma^*$, is* r-computable *if there exists a Turing machine (TM, in short) M working within resource bound of r that computes f. We also refer to f as an* r-function.

Function f is size-increasing *if for every x, $|f(x)| > |x|$. f is* honest *if there exists a polynomial $p(\cdot)$ such that for every x, $p(|f(x)|) > |x|$.*

For function f, f^{-1} denotes a function satisfying the property that for all x, $f(f^{-1}(f(x))) = f(x)$. We say f is r-invertible *if some f^{-1} is r-computable.*

For function f, its range is denoted as: $range(f) = \{y \mid (\exists x)\ f(x) = y\}$.

We will be primarily interested in the resource bound of *polynomial time*, and abbreviate it as p. We now define several notions of reducibilities.

Definition 2.2. *Let r be a resource bound. Set A* r-reduces *to set B if there exists an r-function f such that for every x, $x \in A$ iff $f(x) \in B$. We also write this as $A \leq_m^r B$ via f. Function f is called an* r-reduction *of A to B.*

Similarly, $A \leq_1^r B$ ($A \leq_{1,si}^r B$; $A \leq_{1,si,i}^r B$) if there exists a 1-1 (1-1 and size-increasing; 1-1, size-increasing and r-invertible) r-function f such that $A \leq_m^r B$ via f.

$A \equiv_m^r B$ if $A \leq_m^r B$ and $B \leq_m^r A$. An r-degree *is an equivalence class induced by the relation $\equiv_m^r$.*

Definition 2.3. *A is* r-isomorphic *to B if $A \leq_m^r B$ via f where f is a 1-1, onto, r-invertible r-function.*

The definitions of complexity classes P, NP, PH, EXP, NEXP etc. can be found in [52]. We define the notion of completeness we are primarily interested in.

Definition 2.4. *Set A is* r-complete for NP *if $A \in \mathsf{NP}$ and for every $B \in \mathsf{NP}, B \leq_m^r A$. For $r = p$, set A is called* NP-complete *in short. The class of r-complete sets is also called the* complete r-degree *of* NP.

Similarly one defines complete sets for other classes.

The *Satisfiability problem* (SAT) is one of the earliest known NP-complete problems [25]. SAT is the set of all satisfiable propositional Boolean formulas.

We now define one-way functions. These are p-functions that are not p-invertible on most of the strings. One-way functions are one of the fundamental objects in cryptography.

Without loss of generality (see [30]), we can assume that one-way functions are honest functions f for which the input length determines the output length, i.e., there is a *length function* ℓ such that $|f(x)| = \ell(|x|)$ for all $x \in \Sigma^*$.

Definition 2.5. *Function f is a* $s(n)$-secure one-way function *if (1) f is a p-computable, honest function and (2) the following holds for every polynomial-time randomized Turing machine M and for all sufficiently large n:*

$$\Pr_{x \in_U \Sigma^n} [\, f(M(f(x))) = f(x) \,] < \frac{1}{s(n)}.$$

In the above, the probability is also over random choices of M, and $x \in_U \Sigma^n$ mean that x is uniformly and randomly chosen from strings of size n.

We impose the property of honesty in the above definition since a function that shrinks length by more than a polynomial is trivially one-way.

It is widely believed that 2^{n^ϵ}-secure one-way functions exist for some $\epsilon > 0$. We give one example. Start by defining a modification of the multiplication function:

$$\mathrm{Mult}(x, y) = \begin{cases} 1z & \text{if } x \text{ and } y \text{ are both prime numbers and } z \text{ is the product } x * y \\ 0xy & \text{otherwise} \end{cases}$$

In the above definition, $(\cdot, \cdot)$ is a *pairing function*. In this paper, we assume the following definition of $(\cdot, \cdot)$: $(x, y) = xy\ell$ where $|\ell| = \lceil \log |xy| \rceil$ and ℓ equals $|x|$ written in binary. With this definition, $|(x, y)| = |x| + |y| + \lceil \log |xy| \rceil$. This definition is easily extended for m-tuples for any m.

Mult is a p-function since testing primality of numbers is in P [13]. Computing the inverse of Mult is equivalent to factorization, for which no efficient algorithm is known. However, Mult is easily invertible on most of the inputs, e.g., when any of x and y is not prime. The density estimate for prime numbers implies that Mult is p-invertible on at least $1 - \frac{1}{n^{O(1)}}$ fraction of inputs. It is believed that Mult is $(1 - \frac{1}{n^{O(1)}})$-secure, and it remains so even if one lets the TM M work for time 2^{n^δ} for some small $\delta > 0$. From this assumption, one can show that arbitrary concatenation of Mult:

$$\mathrm{MMult}(x_1, y_1, x_2, y_2, \ldots, x_m, y_m) = \mathrm{Mult}(x_1, y_1) \cdot \mathrm{Mult}(x_2, y_2) \cdots \mathrm{Mult}(x_m, y_m)$$

is a 2^{n^ϵ}-secure one-way function [30](page 43).

One-way functions that are 2^{n^ϵ}-secure are not p-invertible almost anywhere. The weakest form of one-way functions are *worst-case* one-way functions:

Definition 2.6. *Function f is a* worst-case one-way function *if (1) f is a p-computable, honest function, and (2) f is not p-invertible.*

3 Formulation and Early Investigations

The conjecture was formulated by Berman and Hartmanis [21] in 1977. Part of their motivation for the conjecture was a corresponding result in computability theory for computably enumerable sets [50]:

Theorem 3.1 (Myhill)**.** *All complete sets for the class of computably enumerable sets are isomorphic to each other under computable isomorphisms.*

The non-trivial part in the proof of this theorem is to show that complete sets for the class of computably enumerable sets reduce to each other via 1-1 reductions. It is then easy to construct isomorphisms between the complete sets. In many ways, the class NP is the resource bounded analog of the computably enumerable class, and polynomial-time functions the analog of computable functions. Hence it is natural to ask if the resource bounded analog of the above theorem holds.

Berman and Hartmanis noted that the requirement for p-isomorphisms is stronger. Sets reducing to each other via 1-1 p-reductions does not guarantee p-isomorphisms as p-functions do not have sufficient time to perform exponential searches. Instead, one needs p-reductions that are 1-1, size-increasing, and p-invertible:

Theorem 3.2 (Berman-Hartmanis)**.** *If $A \leq_{1,si,i}^p B$ and $B \leq_{1,si,i}^p A$ then A is p-isomorphic to B.*

They defined the *paddability* property which ensures the required kind of reductions.

Definition 3.3. *Set A is* paddable *if there exists a p-computable* padding function *p, $p : \Sigma^* \times \Sigma^* \mapsto \Sigma^*$, such that:*

- *Function p is 1-1, size-increasing, and p-invertible,*
- *For every x, $y \in \Sigma^*$, $p(x,y) \in A$ iff $x \in A$.*

Theorem 3.4 (Berman-Hartmanis)**.** *If $B \leq_m^p A$ and A is paddable, then $B \leq_{1,si,i}^p A$.*

Proof. Suppose $B \leq_m^p A$ via f. Define function g as: $g(x) = p(f(x), x)$. Then, $x \in B$ iff $f(x) \in A$ iff $g(x) = p(f(x), x) \in A$. By its definition and the fact that p is 1-1, size-increasing, and p-invertible, it follows that g is also 1-1, size-increasing, and p-invertible. □

Berman and Hartmanis next showed that the known complete sets for NP at the time were all paddable and hence p-isomorphic to each other. For example, the following is a padding function for SAT:

$$p_{SAT}(x, y_1 y_2 \cdots y_m) = x \ \wedge \ \bigwedge_{i=1}^{m} z_i \ \bigwedge_{i=1}^{m} c_i$$

where $c_i = z_{m+i}$ if bit $y_i = 1$ and $c_i = \bar{z}_i$ if $y_i = 0$ and the Boolean variables z_1, z_2, …, z_{2m} do not occur in the formula x.

This observation led them to the following conjecture:

Isomorphism Conjecture. *All* NP*-complete sets are p-isomorphic to each other.*

The conjecture immediately implies $\mathsf{P} \neq \mathsf{NP}$:

Proposition 3.5. *If the Isomorphism Conjecture is true then* $\mathsf{P} \neq \mathsf{NP}$.

Proof. If $\mathsf{P} = \mathsf{NP}$ then all sets in P are NP-complete. However, P has both finite and infinite sets and there cannot exist an isomorphism between a finite and an infinite set. Hence the Isomorphism Conjecture is false. □

This suggests that proving the conjecture is hard because the problem of separating P from NP has resisted all efforts so far. A natural question, therefore, is: Can one prove the conjecture assuming a reasonable hypothesis such as $\mathsf{P} \neq \mathsf{NP}$? We address this question later in the paper. In their paper, Berman and Hartmanis also asked a weaker question: Does $\mathsf{P} \neq \mathsf{NP}$ imply that no *sparse* set can be NP-complete?

Definition 3.6. *Set A is* sparse *if there exist constants $k, n_0 > 0$ such that for every $n > n_0$, the number of strings in A of length $\leq n$ is at most n^k.*

This was answered in affirmative by Mahaney [49]:

Theorem 3.7 (Mahaney)**.** *If* $\mathsf{P} \neq \mathsf{NP}$ *then no sparse set is* NP*-complete.*

Proof Sketch. We give a proof based on an idea of [9, 51, 19]. Suppose there is a sparse set S such that SAT $\leq_m^p S$ via f. Let F be a Boolean formula on n variables. Start with the set $T = \{F\}$ and do the following:

Replace each formula $\hat{F} \in T$ by $\hat{F}_0$ and $\hat{F}_1$ where $\hat{F}_0$ and $\hat{F}_1$ are obtained by setting the first variable of $\hat{F}$ to 0 and 1 respectively. Let $T = \{F_1, F_2, \ldots, F_t\}$. If t exceeds a certain threshold t_0, then let $G_j = F_1 \vee F_j$ and $z_j = f(G_j)$ for $1 \leq j \leq t$. If all z_j's are distinct then drop F_1 from T. Otherwise, $z_i = z_j$ for some $i \neq j$. Drop F_i from T and repeat until $|T| \leq t_0$. If T has only formulas with no variables, then output Satisfiable if T contains a True formula else output Unsatisfiable. Otherwise, go to the beginning of the algorithm and repeat.

The invariant maintained during the entire algorithm is that F is satisfiable iff T contains a satisfiable formula. It is true in the beginning, and remains true in each iteration after replacing every formula $\hat{F} \in T$ with $\hat{F}_0$ and $\hat{F}_1$. The threshold t_0 must be such that t_0 is a upper bound on the number of strings in the set S of size $\max_j |f(G_j)|$. This is a polynomial in $|F|$ since $|G_j| \leq 2|F|$, f is a p-function, and S is sparse. If T has more than t_0 formulas at any stage then the algorithm drops a formula from T. This formula is F_1 when all z_j's are distinct. This means there are more than t_0 z_j's all of size bounded by $\max_j |f(G_j)|$. Not all of these can be in S due to the choice of t_0 and hence $F_1 \notin \mathsf{SAT}$. If $z_i = z_j$ then F_i is dropped. If F_i is satisfiable then so is G_i. And since $z_i = z_j$ and f is a reduction of SAT to S, G_j is also satisfiable; hence either F_1 or F_j is satisfiable. Therefore dropping F_i from T maintains the invariant.

The above argument shows that the size of T does not exceed a polynomial in $|F|$ at any stage. Since the number of iterations of the algorithm is bounded by $n \leq |F|$, the overall time complexity of the algorithm is polynomial. Hence $\mathsf{SAT} \in \mathsf{P}$ and therefore, $\mathsf{P} = \mathsf{NP}$. □

The "searching-with-pruning" technique used in the above proof has been used profitably in many results subsequently. The Isomorphism Conjecture, in fact, implies a much stronger density result: All NP-complete sets are *dense*.

Definition 3.8. *Set A is* dense *if there exist constants $\epsilon, n_0 > 0$ such that for every $n > n_0$, the number of strings in A of length $\leq n$ is at least 2^{n^ϵ}.*

Buhrman and Hitchcock [22] proved that, under a plausible hypothesis, every NP-complete set is dense infinitely often:

Theorem 3.9 (Buhrman-Hitchcock)**.** *If PH is infinite then for any NP-complete set A, there exists $\epsilon > 0$ such that for infinitely many n, the number of strings in A of length $\leq n$ is at least 2^{n^ϵ}.*

Later, we show that a stronger density theorem holds if 2^{n^ϵ}-secure one-way functions exist.

4 A Counter Conjecture and Relativizations

After Mahaney's result, there was not much progress on the conjecture although researchers believed it to be true. However, this changed in 1984 when Joseph and Young [40] argued that the conjecture is false. Their argument was as follows (paraphrased by Selman [53]). Let f be any 1-1, size-increasing, 2^{n^ϵ}-secure one-way function. Consider the set $A = f(\mathsf{SAT})$. Set A is clearly NP-complete. If it is p-isomorphic to SAT, there must exist a 1-1, honest p-reduction of SAT to A which is also p-invertible. However, the set A is, in a sense, a 'coded' version of SAT such that on most of the strings of A, it is hard to 'decode' it (because f is not p-invertible on most of the strings). Thus, there is unlikely to be a 1-1, honest p-reduction of SAT to A which is also p-invertible, and so A is unlikely to be p-isomorphic to SAT. This led them to make a counter conjecture:

Encrypted Complete Set Conjecture *There exists a 1-1, size-increasing, one-way function f such that* SAT *and* $f(\mathsf{SAT})$ *are not p-isomorphic to each other.*

It is useful to observe here that this conjecture is false in computable setting: The inverse of any 1-1, size-increasing, computable function is also computable. The restriction to polynomial-time computability is what gives rise to possible existence of one-way functions.

It is also useful to observe that this conjecture too implies $\mathsf{P} \neq \mathsf{NP}$:

Proposition 4.1. *If the Encrypted Complete Set Conjecture is true then* $\mathsf{P} \neq \mathsf{NP}$*.*

Proof. If $\mathsf{P} = \mathsf{NP}$ then every 1-1, size-increasing p-function is also p-invertible. Hence for every such function, SAT and $f(\mathsf{SAT})$ are p-isomorphic. □

The Encrypted Complete Set conjecture fails if one-way functions do not exist. Can it be shown to follow from the existence of strong one-way functions, such as 2^{n^ϵ}-secure one-way functions? This is not clear. (In fact, later we argue the opposite.) Therefore, to investigate the two conjectures further, the focus moved to relativized worlds. Building on a result of Kurtz [42], Hartmanis and Hemachandra [33] showed that there is an oracle relative to which $\mathsf{P} = \mathsf{UP}$ and the Isomorphism Conjecture is false. This shows that *both* the conjectures fail in a relativized world since $\mathsf{P} = \mathsf{UP}$ implies that no one-way functions exist.

Kurtz, Mahaney, and Royer [46] defined the notion of *scrambling functions*:

Definition 4.2. *Function f is* scrambling function *if f is 1-1, size-increasing, p-computable and there is no dense polynomial-time subset in range(f).*

Kurtz et. al. observed that,

Proposition 4.3. *If scrambling functions exist then the Encrypted Complete Set Conjecture is true.*

Proof. Let f be a scrambling function, and consider $A = f(\mathsf{SAT})$. Set A is NP-complete. Suppose it is p-isomorphic to SAT and let p be the isomorphism between SAT and A. Since SAT has a dense polynomial-time subset, say D, $p(D)$ is a dense polynomial time subset of A. This contradicts the scrambling property of f. □

Kurtz et. al. [46] then showed that

Theorem 4.4 (Kurtz, Mahaney, Royer)**.** *Relative to a random oracle, scrambling functions exist.*

Proof Sketch. Let O be an oracle. Define function f as:

$$f(x) = O(x)O(x1)O(x11) \cdots O(x1^{2|x|}),$$

where $O(z) = 1$ if $z \in O$, 0 otherwise. For a random choice of O, f is 1-1 with probability 1. So, f is a 1-1, size-increasing, p^O-computable function. Suppose a polynomial-time TM M with oracle O accepts a subset of range(f). In order to distinguish a string in range of f from those outside,

M needs to check the answer of oracle O on several unique strings. And since M can query only polynomially many strings from O, M can accept only a sparse subset of range(f). □

Therefore, relative to a random oracle, the Encrypted Complete Set Conjecture is true and the Isomorphism Conjecture is false. The question of existence of an oracle relative to which the Isomorphism Conjecture is true was resolved by Fenner, Fortnow and Kurtz [27]:

Theorem 4.5 (Fenner, Fortnow, Kurtz)**.** *There exists an oracle relative to which Isomorphism Conjecture is true.*

Thus, there are relativizations in which each of the three possible answers to the two conjectures is true. However, the balance of evidence provided by relativizations is towards the Encrypted Complete Set Conjecture since properties relative to a random oracle are believed to be true in unrelativized world too.[2]

5 The Conjectures for Other Classes

In search of more evidence for the two conjectures, researchers translated them to classes bigger than NP. The hope was that diagonalization arguments that do not work within NP can be used for these classes to prove stronger results about the structure of complete sets. This hope was realized, but not completely. In this section, we list the major results obtained for classes EXP and NEXP which were the two main classes considered.

Berman [20] showed that,

Theorem 5.1 (Berman)**.** *Let A be a p-complete set for* EXP*. Then for every $B \in$* EXP*, $B \leq^p_{1,si} A$.*

Proof Sketch. Let M_1, M_2, … be an enumeration of all polynomial-time TMs such that M_i halts, on input x, within time $|x|^{|i|} + |i|$ steps. Let $B \in \mathsf{EXP}$ and define $\hat{B}$ to be the set accepted by the following algorithm:

> Input (i, x). Let $M_i(i, x) = y$. If $|y| \leq |x|$, accept iff $y \notin A$. If there exists a z, $z < x$ (in lexicographic order), such that $M_i(i, z) = y$, then accept iff $z \notin B$. Otherwise, accept iff $x \in B$.

The set $\hat{B}$ is clearly in EXP. Let $\hat{B} \leq^p_m A$ via f. Let the TM M_j compute f. Define function g as: $g(x) = f(j, x)$. It is easy to argue that f is 1-1 and size-increasing on inputs of the form $(j, \star)$ using the definition of $\hat{B}$ and the fact that f is a reduction. It follows that g is a 1-1, size-increasing p-reduction of B to A. □

Remark 5.2. A case can be made that the correct translation of the isomorphism result of [50] to polynomial-time realm is to show that the complete sets are also complete under 1-1, size-increasing reductions. As observed earlier, the non-trivial part of the result in the setting of computability is to show the above implication. Inverting computable reductions is trivial. This translation will also avoid the conflict with Encrypted Complete Set Conjecture as it does not require p-invertibility. In fact, as will be shown later, one-way functions help in proving an analog of above theorem for the class NP! However, the present formulation has a nice symmetry to it (both the isomorphism and its inverse require the same amount of resources) and hence is the preferred one.

[2] There are notable counterexamples of this though. The most prominent one being the result IP = PSPACE [48, 54] which is false relative to a random oracle [24].

For the class NEXP, Ganesan and Homer [29] showed that,

Theorem 5.3 (Ganesan-Homer)**.** *Let A be a p-complete set for* NEXP*. Then for every $B \in$* NEXP*, $B \leq_1^p A$.*

The proof of this uses ideas similar to the previous proof for EXP. The result obtained is not as strong since enforcing the size-increasing property of the reduction requires accepting complement of a NEXP set which cannot be done in NEXP unless NEXP is closed under complement, a very unlikely possibility. Later, the author [5] proved the size-increasing property for reductions to complete sets for NEXP under a plausible hypothesis.

While the two conjectures could not be settled for the complete p-degree of EXP (and NEXP), answers have been found for p-degrees *close* to the complete p-degree of EXP. The first such result was shown by Ko, Long and Du [41]. We need to define the notion of truth-table reductions to state this result.

Definition 5.4. *Set A* k-truth-table reduces *to set B if there exists a p-function f, $f : \Sigma^* \mapsto \underbrace{\Sigma^* \times \Sigma^* \times \cdots \times \Sigma^*}_{k} \times \Sigma^{2^k}$ such that for every $x \in \Sigma^*$, if $f(x) = (y_1, y_2, \ldots, y_k, T)$ then $x \in A$ iff $T(B(y_1)B(y_2)\cdots B(y_k)) = 1$ where $B(y_i) = 1$ iff $y_i \in B$ and $T(s)$, $|s| = k$, is the sth bit of string T.*

Set B is k-truth-table complete *for* EXP *if $B \in$* EXP *and for every $A \in$* EXP*, A k-truth-table reduces to B.*

The notion of truth-table reductions generalizes p-reductions. For both EXP and NEXP, it is known that complete sets under 1-truth-table reductions are also p-complete [38, 23], and not all complete sets under 2-truth-table reductions are p-complete [55]. Therefore, the class of 2-truth-table complete sets for EXP is the smallest class properly containing the complete p-degree of EXP.

Ko, Long, and Du [41] related the structure of certain p-degrees to the existence of worst-case one-way functions:

Theorem 5.5 (Ko-Long-Du)**.** *If there exist worst-case one-way functions then there is a p-degree in* EXP *such that the sets in the degree are not all p-isomorphic to each other. Further, sets in this degree are 2-truth-table complete for* EXP*.*

Kurtz, Mahaney, and Royer [43] found a p-degree for which the sets are unconditionally not all p-isomorphic to each other:

Theorem 5.6 (Kurtz-Mahaney-Royer)**.** *There exists a p-degree in* EXP *such that the sets in the degree are not all p-isomorphic to each other. Further, sets in this degree are 2-truth-table complete for* EXP*.*

Soon afterwards, Kurtz, Mahaney, and Royer [44] found another p-degree with the opposite structure:

Theorem 5.7 (Kurtz-Mahaney-Royer)**.** *There exists a p-degree in* EXP *such that the sets in the degree are all p-isomorphic to each other. Further, this degree is located inside the 2-truth-table complete degree of* EXP*.*

The set of results above on the structure of complete (or nearly complete) p-degree of EXP and NEXP do not favour any of the two conjectures. However, they do suggest that the third possibility, viz., both the conjectures being false, is unlikely.

6 The Conjectures for Other Reducibilities

Another direction to approach the two conjectures is to weaken the power of reductions instead of the class NP. The hope being that for reductions substantially weaker than polynomial-time, one can prove unconditional results. For several weak reductions, this was proven correct and in this section we summarize the major results in this direction.

The two conjectures for r-reductions can be formulated as:

r-Isomorphism Conjecture. *All r-complete sets for NP are r-isomorphic to each other.*

r-Encrypted Complete Set Conjecture. *There is a 1-1, size-increasing, r-function f such that SAT and $f(\mathsf{SAT})$ are not r-isomorphic to each other.*

Weakening p-reductions to *logspace-reductions* (functions computable by TMs with read-only input tape and work tape space bounded by $O(\log n)$, n is the input size) does not yield unconditional results as any such result will separate NP from L, another long-standing open problem. So we need to weaken it further. There are three major ways to doing this.

6.1 Restricting the Input Head Movement

Allowing the input head movement in only one direction leads to the notion of 1-L-functions.

Definition 6.1. *A* 1-L-function *is computed by deterministic TMs with read-only input tape, the workspace bounded by $O(\log n)$ where n is the input length, and the input head restricted to move in one direction only (left-to-right by convention). In other words, the TM is allowed only one scan of its input. To ensure the space bound, the first $O(\log n)$ cells on the work tape are marked at the beginning of the computation.*

These functions were defined by Hartmanis, Immerman and Mahaney [34] to study the complete sets for the class L. They also observed that the "natural" NP-complete sets are also complete under 1-L-reductions. Structure of complete sets under 1-L-reductions attracted a lot of attention, and the first result was obtained by Allender [14]:

Theorem 6.2 (Allender)**.** *For the classes PSPACE and EXP, complete sets under 1-L-reductions are p-isomorphic to each other.*

While this shows a strong structure of complete sets of some classes under 1-L-reductions, it does not answer the 1-L-Isomorphism Conjecture. After a number of extensions and improvements [29, 37, 10], the author [1] showed that,

Theorem 6.3 (Agrawal)**.** *Let A be a 1-L-complete set for NP. Then for every $B \in \mathsf{NP}$, $B \leq_{1,si,i}^{1\text{-}L} A$.*

Proof Sketch. We first show that A is also complete under *forgetful* 1-L-reductions. Forgetful 1-L-reductions are computed by TMs that, immediately after reading a bit of the input, forget its value. This property is formalized by defining configurations: A *configuration* of a 1-L TM is a tuple $\langle q, j, w\rangle$ where q is a state of the TM, j its input head position, and w the contents of its worktape including the position of the worktape head. A forgetful TM, after reading a bit of the input and before reading the next bit, reaches a configuration which is independent of the value of the bit that is read.

Let $B \in \mathsf{NP}$, and define $\hat{B}$ be the set accepted by the following algorithm:

> Input x. Let $x = y10^b1^k$. Reject if b is odd or $|y| \neq tb$ for some integer t. Otherwise, let $y = y_1y_2\cdots y_t$ with $|y_i| = b$. Let $v_i = 1$ if $y_i = uu$ for some u, $|u| = \frac{b}{2}$; $v_i = 0$ otherwise. Accept iff $v_1v_2\cdots v_t \in B$.

The set $\hat{B}$ is a 'coded' version of set B and reduces to B via a p-reduction. Hence, $\hat{B} \in \mathsf{NP}$. Let f be a 1-L-reduction of $\hat{B}$ to A computed by TM M. Consider the workings of M on inputs of size n. Since M has $O(\log n)$ space, the number of configurations of M will be bounded by a polynomial, say $q(\cdot)$, in n. Let $b = k\lceil \log n \rceil$ such that $2^{b/2} > q(n)$. Let C_0 be the initial configuration of M. By the Pigeon Hole Principle, it follows that there exist two distinct strings u_1 and u_1', $|u_1| = |u_1'| = \frac{b}{2}$, such that M reaches the same configuration, after reading either of u_1 and u_1'. Let C_1 be the configuration reached from this configuration after reading u_1. Repeat the same argument starting from C_1 to obtain strings u_2, u_2', and configuration C_2. Continuing this way, we get triples (u_i, u_i', C_i) for $1 \leq i \leq t = \lfloor \frac{n-b-1}{b} \rfloor$. Let $k = n - b - 1 - bt$. It follows that the TM M will go through the configurations C_0, C_1, ..., C_t on any input of the form $y_1y_2\ldots y_t10^b1^k$ with $y_i \in \{u_iu_i, u_i'u_i\}$. Also, that the pair (u_i, u_i') can be computed in logspace without reading the input.

Define a reduction g of B to $\hat{B}$ as follows. On input v, $|v| = t$, compute b such that $2^{b/2} > q(b + 1 + bt)$, and consider M on inputs of size $b + 1 + bt$. For each i, $1 \leq i \leq t$, compute the pair (u_i, u_i') and output u_iu_i if the ith bit of v is 1, output u_iu_i' otherwise. It is easy to argue that the composition of f and g is a forgetful 1-L-reduction of B to A.

Define another set B' as accepted by the following algorithm:

> Input x. Reject if $|x|$ is odd. Otherwise, let $x = x_1x_2\cdots x_ns_1s_2\cdots s_n$. Accept if exactly one of s_1, s_2, ..., s_n, say s_j, is zero and $x_j = 1$. Accept if all of s_1, s_2, ..., s_n are one and $x_1x_2\cdots x_n \in B$. Reject in all other cases.

Set $B' \in \mathsf{NP}$. As argued above, there exists a forgetful 1-L-reduction of B' to A, say h. Define a reduction g' of B to B' as: $g'(v) = v1^{|v|}$. It is easy to argue that $h \circ g'$ is a size-increasing, 1-L-invertible, 1-L-reduction of B to A and $h \circ g'$ is 1-1 on strings of size n for all n. Modifying this to get a reduction that is 1-1 everywhere is straightforward. □

The above result strongly suggests that the 1-L-Isomorphism Conjecture is true. However, the author [1] showed that

Theorem 6.4 (Agrawal)**.** *1-L-complete sets for* NP *are all 2-L-isomorphic to each other but not 1-L-isomorphic.*

The *2-L-isomorphism* above is computed by logspace TMs that are allowed two left-to-right scans of their input. Thus, the 1-L-Isomorphism Conjecture fails and a little more work shows that the 1-L-Encrypted Complete Set Conjecture is true! However, the failure of the Isomorphism Conjecture here is for a very different reason: it is because 1-L-reductions are not powerful enough to carry out the isomorphism construction as in Theorem 3.2. For a slightly more powerful reducibility, 1-NL-reductions, this is not the case.

Definition 6.5. *A* 1-NL-function *is computed by TMs satisfying the requirements of definition 6.1, but allowed to be nondeterministic. The nondeterministic TM must output the same string on all paths on which it does not abort the computation.*

For 1-NL-reductions, the author [1] showed, using proof ideas similar to the above one, that:

Theorem 6.6 (Agrawal)**.** *1-NL-complete sets for* NP *are all 1-NL-isomorphic to each other.*

The author [1] also showed similar results for *c-L-reductions* for constant c (functions that are allowed at most c left-to-right scans of the input).

6.2 Reducing Space

The second way of restricting logspace reductions is by allowing the TMs only *sublogarithmic* space, i.e., allowing the TM space $o(\log n)$ on input of size n; we call such reductions *sublog-reductions*. Under sublog-reductions, NP has no complete sets, and the reason is simple: every sublog-reduction can be computed by deterministic TMs in time $O(n^2)$ and hence if there is a complete set for NP under sublog-reductions, then $\mathsf{NTIME}(n^{k+1}) = \mathsf{NTIME}(n^k)$ for some $k > 0$, which is impossible [26]. On the other hand, each of the classes $\mathsf{NTIME}(n^k)$, $k \geq 1$, has complete sets under sublog-reductions.

The most restricted form for sublog-reductions is *2-DFA-reductions*:

Definition 6.7. *A* 2-DFA-function *is computed by a TM with read-only input tape and no work tape.*

2-DFA functions do not require any space for their computation, and therefore are very weak. Interestingly, the author [4] showed that sublog-reductions do not add any additional power for complete sets:

Theorem 6.8 (Agrawal)**.** *For any $k \geq 1$, sublog-complete sets for* $\mathsf{NTIME}(n^k)$ *are also 2-DFA-complete.*

For 2-DFA-reductions, the author and Venkatesh [11] proved that:

Theorem 6.9 (Agrawal-Venkatesh)**.** *Let A be a 2-DFA-complete set for* $\mathsf{NTIME}(n^k)$ *for some $k \geq 1$. Then, for every $B \in \mathsf{NTIME}(n^k)$, $B \leq_{1,si}^{2DFA} A$ via a reduction that is* mu-DFA-invertible.

muDFA-functions are computed by TMs with no space and multiple heads, each moving in a single direction only. The proof of this is also via forgetful TMs. The reductions in the theorem above are not 2-DFA-invertible, and in fact, it was shown in [11] that:

Theorem 6.10 (Agrawal-Venkatesh)**.** *Let $f(x) = xx$. Function f is a 2-DFA-function and for any $k \geq 1$, there is a 2-DFA-complete set A for* $\mathsf{NTIME}(n^k)$ *such that $A \not\leq_{1,si,i}^{2DFA} f(A)$.*

The above theorem implies that 2DFA-Encrypted Complete Set Conjecture is true.

6.3 Reducing Depth

Logspace reductions can be computed by (unbounded fan-in) *circuits* of logarithmic depth.[3] Therefore, another type of restricted reducibility is obtained by further reducing the depth of the circuit family computing the reduction. Before proceeding further, let us define the basic notions of circuit model.

[3] For a detailed discussion on the circuit model of computation, see [52].

Definition 6.11. *A* circuit family *is a set $\{C_n : n \in \mathbb{N}\}$ where each C_n is an acyclic circuit with n Boolean inputs $x_1, \dots, x_n$ (as well as the constants 0 and 1 allowed as inputs) and some number of output gates $y_1, \dots, y_r$. $\{C_n\}$ has* size $s(n)$ *if each circuit C_n has at most $s(n)$ gates; it has* depth $d(n)$ *if the length of the longest path from input to output in C_n is at most $d(n)$.*

A circuit family has a notion of uniformity associated with it:

Definition 6.12. *A family $\mathcal{C} = \{C_n\}$ is* uniform *if the function $n \mapsto C_n$ is easy to compute in some sense. This can also be defined using the complexity of the* connection set *of the family:*

$$conn(\mathcal{C}) = \{(n, i, j, T_i, T_j) \mid \text{ the output of gate } i \text{ of type } T_i \text{ is input to gate } j \text{ of type } T_j \text{ in } C_n\}.$$

Here, gate type T_i can be Input, Output, or some Boolean operator.

Family $\mathcal{C}$ is Dlogtime-uniform *[18] if conn($\mathcal{C}$) is accepted by a linear-time TM. It is* p-uniform *[15] if conn($\mathcal{C}$) is accepted by a exponential-time TM (equivalently, by a TM running in time bounded by a polynomial in the circuit size). If we assume nothing about the complexity of conn($\mathcal{C}$), then we say that the family is* non-uniform*.*

An important restriction of logspace functions is to functions computed by *constant depth* circuits.

Definition 6.13. *Function f is a u-*uniform AC^0-function *if there is a u-uniform circuit family $\{C_n\}$ of size $n^{O(1)}$ and depth $O(1)$ consisting of unbounded fan-in AND and OR and NOT gates such that for each input x of length n, the output of C_n on input x is $f(x)$.*

Note that with this definition, an AC^0-function cannot map strings of equal size to strings of different sizes. To allow this freedom, we adopt the following convention: each C_n will have $n^k + k\log(n)$ output bits (for some k). The last $k \log n$ output bits will be viewed as a binary number r, and the output produced by the circuit will be the binary string contained in the first r output bits.

It is worth noting that, with this definition, the class of Dlogtime-uniform AC^0-functions admits many alternative characterizations, including *expressibility in first-order logic with* $\{+, \times, \leq\}$ [47, 18], the *logspace-rudimentary reductions* [39, 17], *logarithmic-time alternating Turing machines with $O(1)$ alternations* [18] etc. Moreover, almost all known NP-complete sets are also complete under Dlogtime-uniform AC^0-reductions (an exception is provided by [7]). We will refer to Dlogtime-uniform AC^0-functions also as *first-order-functions*.

AC^0-reducibility is important for our purposes too since the complete sets under the reductions of the previous two subsections are also complete under AC^0-reductions (with uniformity being Dlogtime- or p-uniform). This follows from the fact that these sets are also complete under some appropriate notion of forgetful reductions. Therefore, the class of AC^0-complete sets for NP is larger than all of the previous classes of this section.

The first result for depth-restricted functions was proved by Allender, Balcázar, and Immerman [16]:

Theorem 6.14 (Allender-Balcázar-Immerman)**.** *Complete sets for NP under first-order projections are first-order-isomorphic to each other.*

First-order projections are computed by a very restricted kind of Dlogtime-uniform AC^0 family in which no circuit has AND and OR gates. This result was generalized by the author and Allender [6] to NC^0*-functions*, which are functions computed by AC^0 family in which the fan-in of every gate of every circuit is at most two.

Theorem 6.15 (Agrawal-Allender)**.** *Let A be a non-uniform NC^0-complete set for NP. Then for any $B \in \mathsf{NP}$, B non-uniform NC^0-reduces to A via a reduction that is 1-1, size-increasing, and non-uniform AC^0-invertible. Further, all non-uniform NC^0-complete sets for NP are non-uniform AC^0-isomorphic to each other where these isomorphisms can be computed and inverted by depth three non-uniform AC^0 circuits.*

Proof Sketch. The proof we describe below is the one given in [3]. Let $B \in \mathsf{NP}$, and define $\hat{B}$ to be the set accepted by the following algorithm:

> On input y, let $y = 1^k 0 z$. If k does not divide $|z|$, then reject. Otherwise, break z into blocks of k consecutive bits each. Let these be $u_1 u_2 u_3 \ldots u_p$. Accept if there is an i, $1 \le i \le p$, such that $u_i = 1^k$. Otherwise, reject if there is an i, $1 \le i \le p$, such that $u_i = 0^k$. Otherwise, for each i, $1 \le i \le p$, label u_i as *null* if the number of ones in it is 2 modulo 3; as *zero* if the number of ones in it is 0 modulo 3; and as *one* otherwise. Let $v_i = \epsilon$ if u_i is null, 0 if u_i is zero, and 1 otherwise. Let $x = v_1 v_2 \cdots v_p$, and accept iff $x \in B$.

Clearly, $\hat{B} \in \mathsf{NP}$. Let $\{C_n\}$ be the NC^0 circuit family computing a reduction of $\hat{B}$ to A. Fix size n and consider circuit C_{k+1+n} for $k = 4\lceil \log n \rceil$. Let C be the circuit that results from setting the first input $k+1$ bits of C_{k+1+n} to $1^k 0$. Randomly set each of the n input bits of C in the following way: with probability $\frac{1}{2}$, leave it unset; with probability $\frac{1}{4}$ each, set it to 0 and 1 respectively. The probability that any block of k bits is completely set is at most $\frac{1}{n^4}$. Similarly, the probability that there is a block that has at most three unset bits is at most $\frac{1}{n}$, and therefore, with high probability, every block has at least four unset bits.

Say that an output bit is *good* if, after the random assignment to the input bits described above is completed, the value of the output bit depends on exactly one unset input bit. Consider an output bit. Since C is an NC^0 circuit, the value of this bit depends on at most a constant, say c, number of input bits. Therefore, the probability that this bit is good after the assignment is at least $\frac{1}{2} \cdot \frac{1}{4^{c-1}}$. Therefore, the expected number of good output bits is at least $\frac{m}{4^c}$, where m is the number of output bits of C whose value depends on some input bit. Using the definition of set $\hat{B}$, it can be argued that $\Omega(n)$ output bits depend on some input bit, and hence $\Omega(n)$ output bits are expected to be good after the assignment. Fix any assignment that does this, as well as leaves at least four unset bits in each block. Now set some more input bits so that each block that is completely set is null, each block that has exactly two unset bits has number of ones equal to 0 modulo 3, and there are no blocks with one, three, or more unset bits. Further, for at least one unset input bit in a block, there is a good output bit that depends on the bit, and there are $\Omega(\frac{n}{\log n})$ unset input bits. It is easy to see that all these conditions can be met.

Now define a reduction of B to $\hat{B}$ as: on input x, $|x| = p$, consider C_{k+1+n} such that the number of unset input bits in C_{k+1+n} after doing the above process is at least p. Now map the ith bit of x to the unset bit in a block that influences a good output bit and set the other unset input bit in the block to zero. This reduction can be computed by an NC^0 circuit (in fact, the circuit does not need any AND or OR gate).

Define a reduction of B to A given by the composition of the above two reductions. This reduction is a *superprojection*: it is computed by circuit family $\{D_p\}$ with each D_p being an NC^0 circuit such that for every input bit to D_p, there is an output bit that depends exactly on this input bit. A superprojection has the input written in certain bit positions of the output. Therefore, it is 1-1 and size-increasing. Inverting the function is also easy: given string y, identify the locations where the input is written, and check if the circuit D_p (p = number of locations) on this input outputs y. This checking can be done by a depth two AC^0 circuit.

This gives a 1-1, size-increasing, AC^0-invertible, NC^0-reduction of B to A. The circuit family is non-uniform because it is not clear how to deterministically compute the settings of the input bits. Exploiting the fact that the input is present in the output of the reductions, an AC^0-isomorphism, computed by depth three circuits, can be constructed between two complete sets following [21] (see [8] for details). □

Soon after, the author, Allender, and Rudich [8] extended it to all AC^0-functions, proving the Isomorphism Conjecture for non-uniform AC^0-functions.

Theorem 6.16 (Agrawal-Allender-Rudich)**.** *Non-uniform AC^0-complete sets for NP are non-uniform AC^0-isomorphic to each other. Further, these isomorphisms can be computed and inverted by depth three non-uniform AC^0 circuits.*

Proof Sketch. The proof shows that complete sets for NP under AC^0-reductions are also complete under NC^0-reductions and invokes the above theorem for the rest. Let A be a complete set for NP under AC^0-reductions. Let $B \in \mathsf{NP}$. Define set $\hat{B}$ exactly as in the previous proof. Fix an AC^0-reduction of $\hat{B}$ to A given by family $\{C_n\}$. Fix size n, and consider C_{k+1+n} for $k = n^{1-\epsilon}$ for a suitable $\epsilon > 0$ to be fixed later. Let D be the circuit that results from setting the first $k+1$ input bits of C_{k+1+n} to $1^k 0$.

Set each input bit of D to 0 and 1 with probability $\frac{1}{2} - \frac{1}{2n^{1-2\epsilon}}$ each and leave it unset with probability $\frac{1}{n^{1-2\epsilon}}$. By the Switching Lemma of Furst, Saxe, and Sipser [28], the circuit D will reduce, with high probability, to an NC^0 circuit on the unset input bits for a suitable choice of $\epsilon > 0$. In each block of k bits, the expected number of unset bits will be n^ϵ, and therefore, with high probability, each block has at least three unset bits. Fix any settings satisfying both of the above.

Now define a reduction of B to $\hat{B}$ that, on input x, $|x| = p$, identifies n for which the circuit D has at least p blocks, and then maps ith bit of input x to an unset bit of the ith block of the input to D, setting the remaining bits of the block so that the sum of ones in the block is zero modulo 3. Unset bits in all remaining blocks are set so that the sum of ones in the block equals two modulo 3.

The composition of the reduction of B to $\hat{B}$ and $\hat{B}$ to A is an NC^0-reduction of B to A. Again, it is non-uniform due to the problem of finding the right settings of the input bits. □

The focus then turned towards removing the non-uniformity in the above two reductions. In the proof of theorem 6.15 given in [6], the uniformity condition is p-uniform. In [7], the uniformity of 6.16 was improved to p-uniform by giving a polynomial-time algorithm that computes the correct settings of input bits. Both the conditions were further improved to logspace-uniform in [3] by constructing a more efficient derandomization of the random assignments. And finally, in [2], the author obtained very efficient derandomizations to prove that:

Theorem 6.17 (Agrawal)**.** *First-order-complete sets for NP are first-order-isomorphic.*

The isomorphisms in the theorem above are no longer computable by depth three circuits; instead, their depth is a function of the depth of the circuits computing reductions between the two complete sets.

6.4 Discussion

At first glance, the results for weak reducibilities above seem to provide equal support to both the conjectures: the Isomorphism Conjecture is true for 1-NL and AC^0-reductions for any reasonable notion of uniformity, while the Encrypted Complete Set Conjecture is true for 1-L and 2-DFA reductions. However, on a closer look a pattern begins to emerge. First of all, we list a common feature of all the results above:

Corollary 6.18. *For $r \in \{$1-L, 1-NL, 2-DFA, NC^0, $\mathsf{AC}^0\}$, r-complete sets for NP are also complete under 1-1, size-increasing, r-reductions.*

The differences arise in resources required to invert the reductions and to construct the isomorphism. Some of the classes of reductions that we consider are so weak, that for a given function f in the class, there is no function in the class that can check, on input x and y, whether $f(x) = y$. For example, suppose f is an NC^0-function and one needs to construct a circuit that, on input x and y, outputs 1 if $y = f(x)$, and outputs 0 otherwise. Given x and y, an NC^0 circuit can compute $f(x)$, and can check if the bits of $f(x)$ are equal to the corresponding bits of y; however, it cannot output 1 if $f(x) = y$, since this requires taking an AND of $|y|$ bits. Similarly, some of the reductions are too weak to construct the isomorphism between two sets given two 1-1, size-increasing, and invertible reductions between them. Theorems 6.3 and 6.4 show this for 1-L-reductions, and the same can be shown for NC^0-reductions too. Observe that p-reductions do not suffer from either of these two drawbacks. Hence we cannot read too much into the failure of the Isomorphism Conjecture for r-reductions. We now formulate another conjecture that seems better suited to get around the above drawbacks of some of the weak reducibilities. This conjecture was made in [1].

Consider a 1-1, size-increasing r-function f for a resource bound r. Consider the problem of accepting the set range(f). A TM accepting this set will typically need to guess an x and then verify whether $f(x) = y$. It is, therefore, a non-deterministic TM with resource bound at least r. Let $r^{range} \geq r$ be the resource bound required by this TM. For a circuit accepting range(f), the non-determinism is provided as additional "guess bits" and its output is 1 if the circuit evaluates to 1 on some settings of the guess bits. We can similarly define r^{range} to be the resource bound required by such a *non-deterministic* circuit to accept range(f).

r-Complete Degree Conjecture. *r-Complete sets for NP are also complete under 1-1, size-increasing, r-reductions that are r^{range}-invertible.*

Notice that the invertibility condition in the conjecture *does not allow non-determinism.* For p-reductions,

Proposition 6.19. *The p-Complete Degree Conjecture is equivalent to the Isomorphism Conjecture.*

Proof. Follows from the observation that $p^{range} = p$ as range of a p-function can be accepted in non-deterministic polynomial-time, and from Theorem 3.2. □

Moreover, for the weaker reducibilities that we have considered, one can show that,

Theorem 6.20. *For $r \in \{$1-L, 1-NL, 2-DFA, NC^0, $\mathsf{AC}^0\}$, the r-Complete Degree Conjecture is true.*

Proof. It is an easy observation that for $r \in \{$1-L, 1-NL, $\mathsf{AC}^0\}$, $r^{range} = r$. The conjecture follows from Theorems 6.3, 6.6, and 6.17.

Accepting range of a 2-DFA-function requires verifying the output of 2-DFA TM on each of its constant number of passes on the input. The minimum resources required for this are to have multiple heads stationed at the beginning of the output of each pass, guess the input bit-by-bit, and verify the outputs on this bit for each pass simultaneously. Thus, the TM is a non-deterministic TM with no space and multiple heads, each moving in one direction only. So Theorem 6.9 proves the conjecture.

Accepting range of an NC^0-function requires a non-deterministic AC^0 circuit. Therefore, Theorems 6.15 and 6.17 prove the conjecture for $r = \mathsf{NC}^0$. □

In addition to the reducibilities in the above theorem, the r-Complete Degree Conjecture was proven for some more reducibilities in [1].

These results provide evidence that r-Complete Degree Conjecture is true for all reasonable resource bounds; in fact, *there is no known example of a reasonable reducibility for which the conjecture is false.*

The results above also raise doubts about the intuition behind the Encrypted Complete Set Conjecture as we shall argue now. Consider AC^0-reductions. There exist functions computable by depth d, Dlogtime-uniform AC^0 circuits that cannot be inverted on most of the strings by depth three, non-uniform AC^0 circuits [35]. However, by Theorem 6.16, AC^0-complete sets are also complete under AC^0-reductions that are invertible by depth two, non-uniform AC^0 circuits and the isomorphisms between all such sets are computable and invertible by depth three, non-uniform AC^0 circuits. So, for every 1-1, size-increasing, AC^0-function, it is possible to efficiently find a dense subset on which the function is invertible by depth two AC^0 circuits.

Therefore, the results for weak reducibilities provide evidence that the Isomorphism Conjecture is true.

7 A New Conjecture

In this section, we revert back to the conjectures in their original form. The investigations for weak reducibilities provide some clues about the structure of NP-complete sets. They strongly suggest that all NP-complete sets should also be complete under 1-1, size-increasing p-reductions. Proving this, of course, is hard as it implies $\mathsf{P} \neq \mathsf{NP}$ (Proposition 3.5). Can we prove this under a reasonable assumption? This question was addressed and partially answered by the author in [5], and subsequently improved by the author and Watanabe [12]:

Theorem 7.1 (Agrawal-Watanabe)**.** *If there exists a 1-1, 2^{n^ϵ}-secure one-way function for some $\epsilon > 0$, then all NP-complete sets are also complete under 1-1, and size-increasing, $\mathsf{P/poly}$-reductions.*

In the above theorem, $\mathsf{P/poly}$-*functions* are those computed by polynomial-size, non-uniform circuit families.

Proof Sketch. Let A be an NP-complete set and let $B \in \mathsf{NP}$. Let f_0 be a 1-1, 2^{n^ϵ}-secure one-way function. Recall that we have assumed that $|f_0(y)|$ is determined by $|y|$ for all y. Håstad et. al. [36] showed how to construct a *pseudorandom generator* using any one-way function. Pseudorandom generators are size-increasing functions whose output cannot be distinguished from random strings by polynomial-time probabilistic TMs. Let G be the pseudorandom generator constructed from f_0.

Without loss of generality, we can assume that $|G(y)| = 2|y| + 1$ for all y. We also modify f_0 to f as: $f(y, r) = f_0(y)rb$ where $|r| = |y|$ and $b = y \cdot r$, the inner product of strings y and r. It is known that the bit b is a *hard-core bit*, i.e., it cannot be predicted by polynomial-time probabilistic TMs on input $f_0(y)r$ [32].

Define B_1 to be the set:

$$B_1 = \{(x, w) \mid x \in B \ \wedge \ |w| = |x|^{2/\epsilon}\} \ \cup \ \text{range}(G),$$

and B_2 to be the set:

$$B_2 = \{f(z) \mid z \in B_1\}.$$

Both the sets are in NP. Let B_2 reduce to A via polynomial-time reduction g. Since f is 1-1, $h = g \circ f$ is a reduction of B_1 to A. We now show that h rarely maps a large number of strings to a single string. For an odd n, let

$$p_n = \Pr_{z,z' \in_U \Sigma^n}[h(z) = h(z')].$$

In other words, p_n is the collision probability of the function h for strings of length n. Define function $\bar{f}(y, r) = f_0(y)r\bar{b}$ where $\bar{b}$ is the complement of the inner product value $y \cdot r$. Since f_0 is 1-1, range(f) and range($\bar{f}$) are disjoint and therefore, range($\bar{F}$) is a subset of $\bar{B}_2$. Let

$$\bar{p}_n = \Pr_{z,z' \in_U \Sigma^n}[h(z) = g(\bar{f}(z'))].$$

Define a probabilistic TM M^+ that on input u, $|u| = |f(z)|$ for $|z| = n$, randomly picks $z' \in \Sigma^n$ and accepts iff $g(u) = h(z')$. The probability, over random $z \in \Sigma^n$, that M^+ accepts $f(z)$ is exactly p_n. The probability, over random $y, r \in \Sigma^{\frac{n-1}{2}}$ and $b \in \Sigma$, that M^+ accepts $u = f_0(y)rb$ is exactly $\frac{1}{2}p_n + \frac{1}{2}\bar{p}_n$ (since b is either $y \cdot r$ or its complement with probability $\frac{1}{2}$ each). Hence the gap between the two probabilities is exactly $|\frac{1}{2}p_n - \frac{1}{2}\bar{p}_n|$. If this is large, then M^+ can be used to predict the hard-core bit of f with high probability, which is not possible. Therefore, the difference of p_n and $\bar{p}_n$ is small.

To show that $\bar{p}_n$ is small, define another TM M^- that on input z, $|z| = n$, randomly picks $z' \in \Sigma^n$ and accepts iff $h(z) = g(\bar{f}(z'))$. On a random $z \in \Sigma^n$, the probability that M^- accepts is exactly $\bar{p}_n$. On input $G(x)$ when x is randomly chosen from $\Sigma^{\frac{n-1}{2}}$, the probability that M^- accepts is zero since range(G) is contained in B_1 and range($\bar{f}$) is contained in $\bar{B}_2$. Hence the difference between the two probabilities is exactly $\bar{p}_n$. This cannot be large as otherwise it violates the pseudorandomness of G. Therefore, p_n is small.

Now define function t as follows. For every n, randomly choose a w_n, $|w_n| = n^{2/\epsilon}$; let $t(x) = (x, w_{|x|})$. Note that t is a probabilistic function. It can be argued that with high probability (over the choices of w_n), (1) range(t) does not intersect with range(G), and so t is a reduction of B to B_1, and (2) $h \circ t$ is 1-1, and size-increasing. Non-uniformly fixing a choice of w_n for every n, we get that $h \circ t$ is a 1-1, size-increasing, non-uniform polynomial-time reduction of B to A. □

On hindsight, the above theorem is not surprising since the analogous result for EXP was shown using diagonalization [20] and one-way functions provide a strong form of diagonalization that works within NP in contrast to standard diagonalization techniques. It is a little unsatisfactory though, since it only shows completeness under *non-uniform* 1-1, size-increasing reductions. It is, however, sufficient to conclude that:

Corollary 7.2. *If there exists a 1-1, 2^{n^ϵ}-secure one-way function for some $\epsilon > 0$, then all* NP-*complete sets are dense.*

Proof. By the above theorem, all NP-complete sets are also complete under 1-1, size-increasing, P/poly-reductions. It is an easy observation that if A is dense and reduces to B via a 1-1 reduction then B is also dense. The corollary follows from the fact that SAT is dense. □

Another suggestion from the previous section is that one-way functions may have easily identifiable dense subsets on which they are p-invertible. This was investigated in [12], where the *easy cylinder* property was defined.

Definition 7.3. *Let f be a 1-1, size-increasing,* P/poly-*function. The function f has an* easy cylinder *if there exist*

- *polynomials $q(\cdot)$, $q'(\cdot)$, and $\ell(\cdot)$ with $\ell(n) \geq 2q(q'(n) + n + \lceil\log(q'(n) + n)\rceil)$, and*
- *a* P/poly *embedding function e, computable by circuits of size $\leq q(|e(y)|)$ on input y,*

such that for every n and for every string u of length $\ell(n)$, there exists a polynomial size circuit C_u, and string s_u, $|s_u| \leq q'(n)$, such that $C_u(f(u, e(s_u, x))) = x$ for all $x \in \Sigma^n$.

Intuitively, a function f has an easy cylinder if there exists a parametrized (on u) dense subset in its domain on which it is easy to invert, and the dense subset depends on the parameter in a simple way (via the string s_u). Note that the circuit C_u can be chosen depending on f as well as u but the embedding function e must be independent of u.

Define set K as:

$$K = \{(p, y) \mid p \text{ is a code of an NTM } M_p \text{ such that } M_p \text{ accepts } y \text{ in at most } |py|^2 \text{ steps}\}.$$

K is easily seen to be NP-complete. The author and Watanabe [12] showed that:

Theorem 7.4 (Agrawal-Watanabe)**.** *Suppose K reduces to A via f and f is a 1-1, size-increasing,* P/poly-*reduction with an easy cylinder. Then K is* P/poly-*isomorphic to A.*

Proof Sketch. Suppose f has an easy cylinder with embedding function e. We define a P/poly-reduction h from K to K such that f is easy to invert on the range of h. Fix any n, and consider a nondeterministic Turing machine M that executes as follows:

> Input (u, y). Guess x, s, $|x| = n$, $|s| \leq q'(n)$, and check whether $e(s, x)$ equals y; if not, reject; if yes, accept if and only if x is in K.

Here we note that the advice of size $q(q'(n) + n + \lceil\log(q'(n) + n)\rceil)$ for computing e on $\Sigma^{q'(n)+n+\lceil\log(q'(n)+n)\rceil}$ is hardwired in M. Further, from the complexity of e, $M(y)$ halts within $2q(q'(n) + n + \lceil\log(q'(n) + n)\rceil)$ steps. Thus, by letting p_n be a code of this machine M that is (with some padding) of size $\ell(n) \geq 2q(q'(n) + n + \lceil\log(q'(n) + n)\rceil)$, we have that M_{p_n} halts and accepts $(p_n, e(s, x))$ in $|p_n e(s, x)|^2$ steps iff M accepts $(p_n, e(s, x))$ iff $x \in K$ for all $x \in \Sigma^n$.

With these machine codes p_n for all n, the reduction h of K to itself is defined as follows for each n and each $x \in \Sigma^n$.

$$h(x) = (p_n, e(s_{p_n}, x)).$$

It follows from the above argument that h is a reduction of K to K. Furthermore, h is $\mathsf{P/poly}$-function.

Let $g = f \circ h$. Function g is clearly a 1-1, size-increasing $\mathsf{P/poly}$-reduction of K to A. We show that g is also $\mathsf{P/poly}$-invertible. This follows from the existence of circuit C_{p_n} such that $x = C_{p_n}(f(p_n, e(s_{p_n}, x)))$ for all $x \in \Sigma^n$. □

Finally, [12] showed that many of the candidate one-way functions do have easy cylinders. For example, the function Mult defined above:

> Mult has two inputs numbers x and y. Fix polynomials $q'(n) = 0$, $q(n) = n$, and $\ell(n) = 2(n + \lceil \log n \rceil)$. Fix $s_u = \epsilon$ and the embedding function $e(s_u, z) = (s_u, z) = zt$ where $|t| = \lceil \log |z| \rceil$ and t equals the number $|z|$ in binary. Therefore, $\text{Mult}(u, e(s_u, z)) = \text{Mult}(u, zt)$. Since $|u| \geq |zt|$, fixing u fixes the first number x and z determines the second number y. Therefore, given u, it is trivial to invert $\text{Mult}(u, zt)$.

The function MMult also has an easy cylinder: use u to fix all but the second string of the last pair. It is also proved in [12] that all 1-1, size-increasing, AC^0-functions have easy cylinders. The notion of easy cylinders is a formalization of the property of AC^0 functions identified at the end of the last section. As already observed, many well-known candidate one-way functions do have easy cylinders. Based on this, [12] conjectured that:

Easy Cylinder Conjecture. *All 1-1, size-increasing,* $\mathsf{P/poly}$*-functions have an easy cylinder.*

The following corollary follows from the above two theorems.

Corollary 7.5. *If there exists a* 2^{n^ϵ}*-secure one-way function and the Easy Cylinder Conjecture is true, then all sets complete for* NP *under* $\mathsf{P/poly}$*-reductions are* $\mathsf{P/poly}$*-isomorphic to each other.*

It is not clear if the Easy Cylinder Conjecture is true. The only indication we have is that the conjecture is true when translated to AC^0 settings, and that many well-known candidate one-way functions have easy cylinders. Goldreich [31] argued against the conjecture by defining a candidate one-way function of the form f^n where f is a candidate one-way function in NC^0 based on expander graphs. He argued that it is not clear whether f^n has an easy cylinder, and conjectured that it does not.

8 Future Directions

The results of previous two sections suggest that the Isomorphism Conjecture is true. However, the evidence is far from overwhelming. Answers to the following questions should make the picture clearer:

- Can one prove the r-Complete Degree Conjecture for other reducibilities, for example, $\mathsf{AC}^0[2]$ (computed by constant depth circuits with AND and PARITY gates)?
- Does Goldreich's function have an easy cylinder? Can one prove it does not under a reasonable hypothesis?

- Even if the Easy Cylinder Conjecture is true and strong one-way functions exist, the Isomorphism Conjecture is true only for P/poly-reductions. Can one define alternative and plausible conjecture(s) from which the Isomorphism Conjecture for p-reductions follows?

Acknowledgement. The author wishes to thank the anonymous referee whose suggestions helped improve the the paper substantially.

References

[1] M. Agrawal. On the isomorphism problem for weak reducibilities. *J. Comput. Sys. Sci.*, 53(2):267–282, 1996.

[2] M. Agrawal. The first order isomorphism theorem. In *Proceedings of the FST&TCS*, pages 70–82. LNCS 2245, 2001.

[3] M. Agrawal. Towards uniform AC^0 isomorphisms. In *Proceedings of the Conference on Computational Complexity*, pages 13–20, 2001.

[4] M. Agrawal. For completeness, sublogarithmic space is no space. *Information Processing Letters*, 82(6):321–325, 2002.

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

[6] M. Agrawal and E. Allender. An isomorphism theorem for circuit complexity. In *Proc. 11th Conference on Computational Complexity*, pages 2–11, 1996.

[7] M. Agrawal, E. Allender, R. Impagliazzio, T. Pitassi, and S. Rudich. Reducing the complexity of reductions. *Computational Complexity*, 10(2):117–138, 2001.

[8] M. Agrawal, E. Allender, and S. Rudich. Reductions in circuit complexity: An isomorphism theorem and a gap theorem. *J. Comput. Sys. Sci.*, 57:127–143, 1998.

[9] M Agrawal and V. Arvind. Quasi-linear truth-table reductions to p-selective sets. *Theoretical Computer Science*, 158:361–370, 1996.

[10] M. Agrawal and S. Biswas. Polynomial-time isomorphism of 1-L-complete sets. *J. Comput. Sys. Sci.*, 53(2):155–160, 1996.

[11] M. Agrawal and S. Venkatesh. The isomorphism conjecture for 2-DFA reductions. *Intl. Jl. on Foundations of Computer Science*, 7(4):339–352, 1996.

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

[13] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of Mathematics*, 160(2):781–793, 2004.

[14] E. Allender. Isomorphisms and 1-L reductions. *J. Comput. Sys. Sci.*, 36(6):336–350, 1988.

[15] E. Allender. P-uniform circuit complexity. *J. ACM*, 36:912–928, 1989.

[16] E. Allender, J. Balcázar, and N. Immerman. A first-order isomorphism theorem. *SIAM Journal on Computing*, 26(2):557–567, 1997.

[17] E. Allender and V. Gore. Rudimentary reductions revisited. *Information Processing Letters*, 40:89–95, 1991.

[18] D. Barrington, N. Immerman, and H. Straubing. On uniformity within NC^1. *J. Comput. Sys. Sci.*, 74:274–306, 1990.

[19] R. Beigel, M. Kummer, and F. Stephan. Approximable sets. *Information and Computation*, 120(2):73–90, 1995.

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

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

[22] H. Buhrman and J. Hitchcock. NP-hard sets are exponentially dense unless coNP $\subseteq$ NP/poly. In *Proceedings of the Conference on Computational Complexity*, pages 1–7, 2008.

[23] H. Buhrman, S. Homer, and L. Torenvliet. Completeness for nondeterministic complexity classes. *Mathematical Systems Theory*, 24(1):179–200, 1991.

[24] R. Chang, B. Chor, O. Goldreich, J. Hartmanis, J. Håstad, D. Ranjan, and P. Rohatgi. The random oracle hypothesis is false. *J. Comput. Sys. Sci.*, 49(1):24–39, 1994.

[25] S. Cook. The complexity of theorem proving procedures. In *Proceedings of Annual ACM Symposium on the Theory of Computing*, pages 151–158, 1971.

[26] S. Cook. A hierarchy for nondeterministic time hierarchy. *J. Comput. Sys. Sci.*, 7(4):343–353, 1973.

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

[28] M. Furst, J. Saxe, and M. Sipser. Parity, circuits, and the polynomial hierarchy. *Mathematical Systems Theory*, 17:13–27, 1984.

[29] K. Ganesan and S. Homer. Complete problems and strong polynomial reducibilities. *SIAM Journal on Computing*, 21:733–742, 1992.

[30] O. Goldreich. *Foundation of Cryptography I: Basic Tools.* Cambridge University Press, 2001.

[31] O. Goldreich. A candidate counterexample for the easy cylinder conjecture. Technical Report TR09-028, Electronic Colloquium on Computational Complexity (http://www.eccc.uni-trier.de/eccc), 2009.

[32] O. Goldreich and L. A. Levin. A hardcore predicate for all one-way functions. In *Proceedings of Annual ACM Symposium on the Theory of Computing*, pages 25–32, 1989.

[33] J. Hartmanis and L. Hemchandra. One-way functions and the non-isomorphism of NP-complete sets. *Theoretical Computer Science*, 81(1):155–163, 1991.

[34] J. Hartmanis, N. Immerman, and S. Mahaney. One-way log-tape reductions. In *Proceedings of Annual IEEE Symposium on Foundations of Computer Science*, pages 65–72, 1978.

[35] J. Håstad. Almost optimal lower bounds for small depth circuits. In *Proceedings of Annual ACM Symposium on the Theory of Computing*, pages 6–20, 1986.

[36] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby. A pseudo-random generator from any one-way function. *SIAM Journal on Computing*, pages 221–243, 1998.

[37] L. A. Hemchandra and A. Hoene. Collapsing degrees via strong computation. *J. Comput. Sys. Sci.*, 46(3):363–380, 1993.

[38] S. Homer, S. Kurtz, and J. Royer. On 1-truth-table hard languages. *Theoretical Computer Science*, 155:383–389, 1993.

[39] N. Jones. Space-bounded reducibility among combinatorial problems. *J. Comput. Sys. Sci.*, 11:68–85, 1975.

[40] D. Joseph and P. Young. Some remarks on witness functions for nonpolynomial and noncomplete sets in NP. *Theoretical Computer Science*, 39:225–237, 1985.

[41] K. Ko, T. Long, and D. Du. A note on one-way functions and polynomial-time isomorphisms. *Theoretical Computer Science*, 47:263–276, 1987.

[42] S. Kurtz. A relativized failure of Berman-Hartmanis conjecture. Unpublished Manuscript, 1983.

[43] S. Kurtz, S. Mahaney, and J. Royer. Noncollapsing degrees. Technical Report 87-001, Department of Computer Science, University of Chicago, 1987.

[44] S. Kurtz, S. Mahaney, and J. Royer. Collapsing degrees. *J. Comput. Sys. Sci.*, 37:247–268, 1988.

[45] S. Kurtz, S. Mahaney, and J. Royer. The structure of complete degrees. In A. Selman, editor, *Complexity Theory Retrospective*, pages 108–146. Springer-Verlag, 1988.

[46] S. Kurtz, S. Mahaney, and J. Royer. The isomorphism conjecture fails relative to a random oracle. *J. ACM*, 42(2):401–420, 1995.

[47] S. Lindell. A purely logical characterization of circuit complexity. In *Proceedings of the Structure in Complexity Theory Conference*, pages 185–192, 1992.

[48] C. Lund, L. Fortnow, H. Karloff, and N. Nissan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992.

[49] S. Mahaney. Sparse complete sets for NP: Solution of a conjecture of Berman and Hartmanis. *J. Comput. Sys. Sci.*, 25(2):130–143, 1982.

[50] J. Myhill. Creative sets. *Z. Math. Logik Grundlag. Math.*, 1:97–108, 1955.

[51] M. Ogihara. Polynomial time membership comparable sets. *SIAM Journal on Computing*, 24(5):1068–1081, 1995.

[52] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1995.

[53] A. L. Selman. A survey of one-way functions in complexity theory. *Mathematical Systems Theory*, 25:203–221, 1992.

[54] A. Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992.

[55] O. Watanabe. A comparison of polynomial time completeness notions. *Theoretical Computer Science*, 54:249–265, 1987.