The Isomorphism Conjecture for Constant Depth Reductions

Manindra Agrawal*
IIT Kanpur

March 6, 2010

Abstract

For any class $\mathcal{C}$ closed under TC^0 reductions, and for any measure u of uniformity containing Dlogtime, it is shown that all sets complete for $\mathcal{C}$ under u-uniform AC^0 reductions are isomorphic under u-uniform AC^0-computable isomorphisms.

1 Introduction

One of the long-standing conjectures about the structure of complete sets is the isomorphism conjecture (proposed in [BH77]) stating that all sets complete for NP under polynomial-time reductions are polynomial time isomorphic. As the conjecture cannot be resolved either way unless we discover non-relativizable techniques (see [KMR88, KMR95, FFK96] for more details), efforts have been made to prove the conjecture in restricted settings by restricting the power of reductions (see for example [Agr96, AAR98]). One of the most natural definitions of restricted reductions is that of functions computed by *constant-depth* (or AC^0) circuits (first studied in [CSV84]). These reductions provide the right notion of completeness for small complexity classes (logspace and below). Also, it has been observed that *natural* complete problems for various complexity classes remain complete under such reductions [IL95, Imm87]. Although the class of AC^0 functions is much smaller than the class of polynomial-time functions, it is interesting to note that there are very few known examples of an NP-complete set that is not complete under uniform AC^0 reductions ([AAI+01] provides one such example).

The notion of uniformity to be used with AC^0 circuits is widely accepted to be that of *Dlogtime-uniformity* (see Section 3 for a definition). Under this uniformity condition, these circuits admit a number of different characterizations [BIS90, AG91]: functions computed by first-order logic formulae [Lin92],

*N Rama Rao chair professor, Department of Computer Science, IIT Kanpur, Kanpur 208016, India, manindra@iitk.ac.in

$O(1)$-alternating log-time TMs [Sip83], logspace rudimentary predicates [Jon75] etc.

The isomorphism conjecture for complete sets for NP under AC^0 reductions has been studied before. Allender et. al. [ABI97] showed that all sets complete under *first-order projections* (these are very simple functions computed by uniform circuits with no gates [IL95]) are Dlogtime-uniform AC^0-isomorphic (i.e., the isomorphism between any two such sets is computable in both directions by Dlogtime-uniform AC^0 circuits). This was improved, at the cost of losing uniformity, in [AAR98] where it is shown that all sets complete under u-uniform (for any u) AC^0 reductions are non-uniform AC^0-isomorphic. Notice that this result proves the isomorphism conjecture for non-uniform AC^0 reductions but not for Dlogtime-uniform reductions. The uniformity condition for isomorphisms was improved in [AAI+01] to P-uniform. This still leaves open the conjecture for Dlogtime-uniform AC^0 reductions, which is, as observed above, the correct formulation of the isomorphism conjecture for constant depth reductions.

In this paper, we prove that all complete sets for NP under u-uniform AC^0 reductions are u-uniform AC^0-isomorphic for any uniformity u containing Dlogtime, thus proving the isomorphism conjecture for uniform constant depth reductions.[1] Since there are a number of alternative characterizations of Dlogtime-uniform AC^0 circuits, this theorem can be viewed in many interesting ways, e.g., *all sets complete under first-order reductions are first-order isomorphic* (first-order functions are computed by first-order formulae). The result, in fact, holds for any class closed under TC^0 reductions.

The next section provides an outline of our proof. Section 3 contains definitions, and the subsequent sections are devoted to proving the result.

2 Proof Outline

The overall structure of the proof remains as given in [AAR98]. The proof in [AAR98] proceeds in three steps:

Step 1 (Gap Theorem): This shows that all complete sets under u-uniform AC^0 reductions are also complete under *non-uniform* NC^0 reductions. This step is non-uniform.

Step 2 (Superprojection Theorem): This proves that all complete sets under u-uniform NC^0 reductions are also complete under (u+P)-uniform *superprojections*, where superprojections are functions similar to projections. This step is P-uniform.

Step 3 (Isomorphism Construction): This proves that all complete sets under u-uniform superprojections are isomorphic under (u+Dlogtime)-uniform AC^0 isomorphisms. This step is Dlogtime-uniform: starting with Dlogtime-uniform superprojections, one gets Dlogtime-uniform AC^0 isomorphisms.

[1] The results in this paper first appeared in [Agr01b] and [Agr01a].

The proof of the Gap Theorem uses the Switching Lemma of [FSS84] in the construction of NC^0 reductions and is the reason for its non-uniformity. In [AAI$^+$01] the lemma was derandomized using method of conditional probabilities making the stage P-uniform. The Superprojection Theorem of [AAR98] uses the Sunflower Lemma of [ER60] which is P-uniform.

Clearly, the uniformity of both these stages needs to be improved to obtain Dlogtime-uniformity. It is useful to note here that it is sufficient to make both the stages AC^0-uniform instead of Dlogtime-uniform as that makes the isomorphism constructed by Stage 3 also AC^0-uniform and then the AC^0 circuit used in uniformity can be incorporated in the AC^0 circuit for the isomorphism making the resulting AC^0 circuit Dlogtime-uniform. In fact this is the best that we can hope to do as it is known that the Gap Theorem *cannot* be made Dlogtime-uniform [AAR98].

We first consider the Gap Theorem. The method of conditional probability used to derandomize the Switching Lemma in [AAI$^+$01] appears inherently sequential. So to improve the uniformity, we need to find a different way of derandomizing the lemma. There does exist a different derandomization of the lemma in the literature [CSS97]: they obtain a *pseudorandom generator* against the Switching Lemma of [Hås86] that stretches a seed of length $(\log n)^{O(d)}$ to n bits and "fools" the lemma for depth d circuits. However, it does not serve our purpose since derandomizing the lemma using this generator would require superpolynomial sized circuits.

We construct a new pseudorandom generator against the Switching Lemma of [FSS84]. This generator stretches a seed of length $O(\log n)$ to n bits. We can thus derandomize the lemma by cycling through all the seed values. We show that the generator construction, and other related computations, can be performed by Dlogtime-uniform AC^0 circuits thus making the Gap Theorem AC^0-uniform.

Next, we consider the Superprojection Construction of [AAR98]. This uses the Sunflower Lemma which again appears inherently sequential. So we need a different construction here as well. We adopt the approach of the Gap Theorem and define a random construction that succeeds with high probability and then derandomize it using an appropriate pseudorandom generator. All the computations is this construction can also be performed by Dlogtime-uniform AC^0 circuits.

Combining the above constructions together with the Isomorphism Construction, we get Dlogtime-uniform AC^0-isomorphisms.

3 Basic Definitions and Preliminaries

We assume familiarity with the basic notions of many-one reducibility as presented, for example, in [BDG88].

A *circuit family* is a set $\{C_n : n \in \mathbf{N}\}$ where each C_n is an acyclic circuit with n Boolean inputs $x_1, \ldots, x_n$ (as well as the constants 0 and 1 allowed as inputs) and some number of output gates $y_1, \ldots, y_r$. $\{C_n\}$ 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 $d(n)$.

For a circuit family $\{C_n\}$, the *connection set* of the family is defined as:

$$\text{Conn}_C = \{\langle n, t, i, j\rangle \mid \text{ gate } i \text{ in } C_n \text{ is of type } t \text{ and takes input from gate } j\}.$$

The connection set can be used to give a *binary encoding* of circuit C_n: bit $\langle t, i, j\rangle$ of the encoding is 1 iff $(n, t, i, j) \in \text{Conn}_C$.

A family $\{C_n\}$ is *u-uniform* if the connection set can be computed by a machine (or circuit) with a resource bound of u. In this paper, we will primarily use two notions of uniformity: Dlogtime-uniformity [BIS90] and AC^0-uniformity. In the first, the connection set is computed by a TM with random access tapes working in $O(\log n)$ time (which is linear time as a function of input size), and in the second, the connection set is computed by an AC^0 circuit of polynomial size (which is exponential size in terms of input size). We will follow the standard convention that whenever the connection set is computed by a circuit family, the circuit family is assumed to be Dlogtime-uniform. So, for example, AC^0-uniform means that the set can be computed by a Dlogtime-uniform AC^0 family of circuits.

A function f is said to be in AC^0 if there is a 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 $f(x)$. We will adopt the following specific convention for interpreting the output of such a circuit: 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 binary string contained in the first r output bits. It is easy to verify that this convention is AC^0-equivalent to any other reasonable convention that allows for variable sized output, and for us it has the advantage that only $O(\log n)$ output bits are used to encode the length.

With this definition, the class of Dlogtime-uniform AC^0-computable functions admits many alternative characterizations, including expressibility in first-order with $\{+, \times, \leq\}$, [Lin92, BIS90] the logspace-rudimentary reductions of Jones [Jon75, AG91], logarithmic-time alternating Turing machines with $O(1)$ alternations [BIS90] and others. This lends additional weight to our choice of this definition.

NC^0 is the class of functions computed in this way by circuit families of size $n^{O(1)}$ and depth $O(1)$, consisting of fan-in two AND and OR and NOT gates. Note that for any NC^0 circuit family, there is some constant c such that each output bit depends on at most c different input bits. An NC^0 function is a *projection* if its circuit family contains no AND or OR gates. For the sake of simplicity, we assume that NC^0 and projection functions *do not have variable sized output*. This restricts the class of these functions, however, all NC^0 and projection functions that we use will be of this kind.

For a complexity class $\mathcal{C}$, a $\mathcal{C}$-isomorphism is a bijection f such that both f and f^{-1} are in $\mathcal{C}$. Since only many-one reductions are considered in this paper, a "$\mathcal{C}$-reduction" is simply a function in $\mathcal{C}$.

A *language* is in a complexity class $\mathcal{C}$ if its characteristic function is in $\mathcal{C}$. This convention allows us to avoid introducing additional notation such as FAC^0, FNC^1, etc. to distinguish between classes of languages and classes of functions.

4 Derandomizing the Switching Lemma

A derandomization of the Switching Lemma of [FSS84] gives a deterministic way of assigning values to certain input bits in a manner that transforms a given AC^0 circuit to an NC^0 circuit. We will obtain a derandomization that requires a seed of size $O(\log n)$ and is independent of the given AC^0 circuit. We first go through the proof of the Switching Lemma as in [FSS84] and then show how each randomized step of the construction can be derandomized[2]. We will follow a simplification of the original proof of [FSS84]. This proof has been sketched at several places (see, e.g., [AAI$^+$01]), we will sketch it once more with the required parameter values.

Let C be a circuit with n input bits. A *random restriction* of the inputs to C is a random assignment of values to a random subset of inputs.

In this section, we will denote, by $\mathrm{AC}(d, s, n)$ the class of circuits with AND, OR, and NOT gates (AND and OR gates having unbounded fanin) of depth d and size s on n input bits. We now state the lemma in the form that we need:

Lemma 4.1 *There exists a constant γ (depending on d and k only) such that for large enough n and for any circuit C in $\mathrm{AC}(d, n^k, n)$, when a sequence of random restrictions is applied to C with appropriate parameters, C reduces, with probability at least $1 - \frac{1}{n^2}$, to a depth two circuit having at least $n^{1/\gamma}$ unset bits, with the property that the output of the circuit depends on at most γ of the unset bits.*

Proof Sketch. Let $C \in \mathrm{AC}(d, n^k, n)$ be an AC^0 circuit of depth d and size n^k on n input bits. We can assume, without loss of generality, that C is arranged into d alternating levels of ANDs and ORs on $n^{\delta_0} = n$ unset bits with its leaves being depth $c_0 = 1$ decision trees. The proof proceeds in d stages. After stage i, the circuit reduces to a depth $d-i$ circuit of size n^k on n^{δ_i} unset bits with leaves being decision trees of depth at most c_i. Stage i has at most c_{i-1} substages. After substage j of stage i, the circuit reduces to a depth $d - i + 1$ circuit of size n^k on $n^{\delta_{i-1}/2^j}$ unset bits whose bottom layer is made up of decision trees of depth at most $c_{i,j}$ (with $c_{i,0} = 0$) with leaves that are ANDs (or ORs) of decision trees of depth at most $c_{i-1} - j$. We now describe a single substage j of stage i.

After the substage $j - 1$ of stage i, the bottom layer of the circuit consists of decision trees of depth at most $c_{i,j-1}$ with leaves that are ANDs (or ORs) of decision trees of depth at most $c_{i-1} - j + 1$. Assume it is ANDs of decision trees (the proof for ORs is identical). Therefore, each AND gate of the bottom layer

[2]It is interesting to note that the stronger Switching Lemma of [Hås86] does not admit such a construction.

can be expressed as an AND of ORs of fanin at most $c_{i-1} - j + 1$. Denote these ANDs by Q_1, Q_2, ..., $Q_{2^{c_{i,j-1}} n^k}$ (there will be at most $2^{c_{i,j-1}} n^k$ such ANDs since the size is n^k and each decision tree above the AND gates in the bottom later has depth at most $c_{i,j-1}$). Represent the unset input bits in the circuit as distinct boolean variables. For each Q_m, define set Maxset(Q_m) to be the lex-first maximal set of clauses in Q_m that are variable disjoint. If these are more than $\alpha \log n$ for $\alpha = (k+5)2^{c_{i-1}-j+1}$, then redefine Q_m to be the lex-first $\alpha \log n$ of these clauses. So each Q_m contains at most $(c_{i-1} - j + 1)\alpha \log n$ variables.

We now use a random restriction that first picks a random subset of size $n^{\delta_{i-1}/2^j}$ from the $n^{\delta_{i-1}/2^{j-1}}$ unset variables and then sets the remaining variables to 0 and 1 with equal probability. A simple calculation (based on Chernoff bounds on tail distribution) shows that the probability that a Q_m has more than c' unset variables is less than $(\frac{ec_{i-1}\alpha \log n}{c' n^{\delta_{i-1}/2^j}})^{c'}$. Choosing $c' = \frac{2^j(k+5)}{\delta_{i-1}}$, this probability becomes less than $\frac{1}{n^{k+4}}$ for large enough n. Summing over all m's, the probability that any Maxset(Q_m) has more than c' unset variables is less than $\frac{1}{n^3}$ for large enough n.

Consider those Q_m's for which $|\text{Maxset}(Q_m)| = \alpha \log n$. By the above calculation, most of the restrictions will leave at most c' unset variables in it. We consider such restrictions only. Drop the (at most) c' ORs that have an unset variable from the set Maxset(Q_m). Because those input input variables that are set take the values 0 and 1 with equal probability, the probability that a particular OR in Maxset(Q_m) will have the value 1 is at most $1 - \frac{1}{2^{c_{i-1}-j+1}}$. And since the ORs in the set are disjoint, the probability that *all* of them will have value 1 is at most $(1 - \frac{1}{2^{c_{i-1}-j+1}})^{\alpha \log n - c'} \leq \frac{1}{n^{k+4}}$ (substituting the value of α) for large enough n. Summing over all Q_m's, the probability that some Q_m with $|\text{Maxset}(Q_m)| = \alpha \log n$ *survives* the random restriction (i.e., does not become zero) is less than $\frac{1}{n^3}$ for large enough n.

Consider now those Q_m's for which $|\text{Maxset}(Q_m)| < \alpha \log n$. Replace every such Q_m with a decision tree of depth at most c' by querying the c' unset variables in Maxset(Q_m). Since variables in Maxset(Q_m) intersect every clause of Q_m, the leaves of this decision tree will be ANDs of ORs of fanin at most $c_{i-1} - j$. Thus the bottom layer becomes a decision tree of depth at most $c_{i,j} = c_{i,j-1} + c'$ whose leaves are ANDs of decision trees of depth at most $c_{i-1} - j$. This finishes substage j of stage i. Repeating this at most c_{i-1} times will result in a depth $d - i$ circuit of the kind mentioned above with suitable values of c_i and δ_i. Further, this will happen with probability at least $1 - O(\frac{1}{n^3})$ for large enough n. After d steps, the circuit will be simply a decision tree of depth at most c_d thus depending on at most 2^{c_d} unset variables out of n^{δ_d} for large enough n. Moreover, this event will occur with probability at least $1 - O(\frac{1}{n^3}) \geq 1 - \frac{1}{n^2}$ for large enough n. Choosing $\gamma = \max\{2^{c_d}, \frac{1}{\delta_d}\}$ completes the proof. ■

We now proceed with the derandomization. It will be convenient to assume that $n = 2^{2^t}$ for some $t \geq 0$ for the subsequent arguments.

Notice the following three crucial points about any particular substage of

the above proof:

1. In any substage, we have argued about properties of sets of input variables of size at most $\hat{c}\log n$ where $\hat{c} = c_d(k+5)2^{c_{d-1}}$.
2. We use two properties of the random restriction. The first one is: given any subset of size at most $\hat{c}\log n$ of a set of $m \geq n^{1/\gamma}$ variables, the probability that a random subset of size $m^{1/2}$ intersects the given subset with cardinality more than $c' \leq \frac{k+5}{\delta_d}$ is at most $\frac{1}{n^{k+4}}$.
3. The second property we use is: given any AND of disjoint ORs, with AND of fanin at least $\alpha \log n - c_d$ and ORs of fanin at most c_d, the probability that a random assignment to the input variables makes the AND output a 1 is at most $\frac{1}{n^{k+4}}$.

Therefore, for any random restriction satisfying the above two properties, the proof will remain valid. We can easily derandomize the construction of such random restrictions using known constructions. We now describe these derandomizations.

4.1 Setting input variables

This is straightforward: we can use any $(\hat{c}\log n)$-wise independent source. However, such sources have seed size of $\Omega(\log^2 n)$ which does not give a complete derandomization. So, instead, we use a $(\hat{c}\log n)$-wise independent $\frac{1}{n^{k+4}}$-biased source [NN93]. Efficient constructions of such sources are known [NN93, AGHP92]. We describe one of these (given in [AGHP92]).

Let F_p be the field of p elements for some prime $p = n^{O(1)}$. The seed for the source is a random element r of the field F_p. Given r, the i^{th} bit of the source, $i \leq n$, is 1 iff the number $r + i$ is a quadratic non-residue in F_p. Let G_I denote this source.

4.2 Choosing subsets of variables

Here we use a source based on designs defined in [NW94]:

Let $m = \frac{\log n}{2}$ and $\hat{c} = \frac{k+5}{\delta_d}$. Let $\bar{a} = (a_0, \ldots, a_{\hat{c}-1})$ with $a_i \in F_{2^m}$, the field of 2^m elements. For polynomial $P_{\bar{a}}(x) = \sum_{i=0,\hat{c}-1} a_i \cdot x^i$ let

$$S_{\bar{a}} = \{xP_{\bar{a}}(x) \mid x \in F_{2^m}\}.$$

Our source will have seed $\bar{a}$, and will output the set $S_{\bar{a}}$. This source provides $n^{\hat{c}/2}$ subsets of size $n^{1/2}$. Let $G_{D,n}$ denote this source.

The following lemma shows that this source satisfies the required property of subsets:

Lemma 4.2 *Let X be any subset of $\{1, 2, \ldots, n\}$ such that $|X| = O(\log n)$. Then for large enough n,*

$$\Pr_{\bar{a}}[|G_{D,n}(\bar{a}) \cap X| \geq \hat{c}] \leq \frac{1}{n^{\hat{c}-1}}.$$

Proof. Fix any subset Y of X of size $\hat{c}$. Imposing the condition that $G_{D,n}(\bar{a})$ contains all of Y gives rise to a system of $\hat{c}$ linearly independent equations in a_0, ..., $a_{\hat{c}-1}$, and hence has exactly one solution. Therefore, for exactly one seed, $Y \subseteq G_{D,n}(\bar{a})$. Since there are $\binom{|X|}{\hat{c}}$ ways of choosing Y, the number of seeds for which $|G_{D,n}(\bar{a}) \cap X| \geq \hat{c}$ is at most $\binom{|X|}{\hat{c}}$. The lemma follows. ■

4.3 Constructing a hybrid source

It is now clear how to derandomize the Switching Lemma: The proof of the lemma has a constant number of substages, and each substage uses a random restriction on n^δ unset input bits to leave $n^{\delta/2}$ bits unset for some δ. For this substage, we use G_{D,n^δ} to pick the subset and set the remaining bits using the source G_I.

So, the derandomization of the Switching Lemma for circuits in $\mathrm{AC}(d, n^k, n)$ is obtained by a hybrid source $\mathcal{H}$ that uses $\tau \leq \log \gamma$ pairs of sources—one for each substage—with the i^{th} pair being $(G_{D,n^{1/2^{i-1}}}, G_I)$.

Given a seed $((\bar{a}_0, r_0), \ldots, (\bar{a}_{\tau-1}, r_{\tau-1}))$ of the hybrid source $\mathcal{H}$, bit j of the output can be calculated as follows:

> Let $j = j_0 j_1 \cdots j_{\tau-1} j_\tau$ where $|j_i| = 2^{t-i-1}$ for $0 \leq i < \tau$ and $|j_\tau| = 2^{t-\tau}$ (recall that we have assumed $n = 2^{2^t}$). Let i be the smallest index for which $G_{D,n^{1/2^i}}(\bar{a}_i)$ does not contain the number $k = j_i j_{i+1} \cdots j_{\tau-1} j_\tau$. Set bit j of the source to the bit k of $G_I(r_i)$. If there is no such i, bit j remains unset.

By the arguments above, the derandomization of the Switching Lemma follows:

Lemma 4.3 *There exists a constant $\gamma \geq 2$ (depending on d and k only) such that for large enough n, and for any circuit C in $\mathrm{AC}(d, n^k, n)$, when the input to C is set using the restriction output by the source $\mathcal{H}$, C reduces, with probability at least $1 - \frac{1}{n^2}$, to a depth two circuit having at least $n^{1/\gamma}$ unset bits, with the property that the output of the circuit depends on at most γ of the unset bits.*

An interesting feature of the source $\mathcal{H}$ is that every restriction output by the source has exactly one unset bit in every block of $n^{1-\frac{1}{2^\tau}}$ bits:

Lemma 4.4 *For any seed $((\bar{a}_0, r_0), \ldots, (\bar{a}_{\tau-1}, r_{\tau-1}))$, the restriction $\mathcal{H}((\bar{a}_0, r_0), \ldots, (\bar{a}_{\tau-1}, r_{\tau-1}))$ has exactly one unset bit in every block of $n^{1-\frac{1}{2^\tau}}$ bits.*

Proof. The locations of unset bits are determined by $G_{D,n^{1/2^i}}(\bar{a}_i)$ for $0 \leq i < \tau$. For index j, $0 \leq j < n$, let $j = j_0 j_1 \cdots j_{\tau-1} j_\tau$ where $|j_i| = 2^{t-i-1}$ for $0 \leq i < \tau$ and $|j_\tau| = 2^{t-\tau}$. Bit j remains unset if for every i, $0 \leq i < \tau$, the number $j_i j_{i+1} \cdots j_{\tau-1} j_\tau$ occurs in the set $G_{D,n^{1/2^i}}$. Recall that

$$G_{D,n^{1/2^i}} = \{x P_{\bar{a}_i}(x) \mid x \in F_{2^{t-i-1}}\}$$

with $|P_{\bar{a}_i}(x)| = |x| = 2^{t-i-1}$. Since $|j_i| = |j_{i+1} \cdots j_\tau| = 2^{t-i-1}$, we get that for every possible value of j_i, there is exactly one value of $j_{i+1} \cdots j_\tau$ such that $j_i j_{i+1} \cdots j_\tau$ is in $G_{D,n^{1/2^i}}$.

Therefore, for every possible value of $j_0 j_1 \cdots j_{\tau-1}$, there is exactly one value of j_τ for which the bit $j_0 j_1 \cdots j_{\tau-1} j_\tau$ remains unset. The lemma follows. ■

This feature will be useful in our uniform construction later.

4.4 The complexity of derandomization

We now calculate the resources required to achieve the derandomization in Lemma 4.3. First observe that:

Lemma 4.5 *The function $\mathcal{H}$ can be computed by a Dlogtime-uniform* AC^0 *circuit of size $n^{O(1)}$.*

Proof. The source $\mathcal{H}$ uses several copies of G_I and G_{D,n^δ}. We consider computation of these two sources first. For G_I, the computations required are:

- compute a prime $p = n^{O(1)}$,
- test if there exists an $s \in F_p$ such that $s^2 = r + i$ in F_p.

Both can be done in Dlogtime-uniform AC^0 as the field size is small (see [BIS90]). For G_{D,n^δ}, the computations required are:

- compute field F_{2^m} where $m = \frac{\delta}{2} \log n$,
- test if $i = \sum_{j=0}^{\hat{c}-1} a_j k^j$ in F_{2^m}.

Again, since the field size is small, both the computations can be done by a Dlogtime-uniform AC^0 circuit [BIS90].

Computing bit j of $\mathcal{H}$, $j = j_0 j_1 \cdots j_{\tau-1} j_\tau$, requires finding the smallest i for which $k = j_i j_{i+1} \cdots j_{\tau-1} j_\tau$ is not in $G_{D,n^{1/2^i}}$ and then using the output bit number k of the $(i+1)^{st}$ copy of G_I. This is clearly a Dlogtime-uniform AC^0 computation. ■

Now we show that finding a seed of $\mathcal{H}$ that works in Lemma 4.3 can also be done in Dlogtime-uniform AC^0.

Lemma 4.6 *There is a Dlogtime-uniform* AC^0 *circuit that, given as input a seed s of $\mathcal{H}$ and a binary encoding of circuit C in $\mathrm{AC}(d, n^k, n)$, tests if C reduces, on input $\mathcal{H}(s)$, to a depth-2 circuit depending on at most γ unset bits (the constant γ is the same as in Lemma 4.3) and outputs the binary encoding of the reduced circuit if the test is positive.*

Proof. The AC^0 circuit that we desire is constructed in substages, one for each substage in the proof of Lemma 4.1. The substage j of stage i will take as input the part of the seed of $\mathcal{H}$ meant for this substage, and the binary encoding of the circuit resulting after the restrictions of previous substages have been applied. Assuming that all the previous restrictions have been good, the bottom layer of the input circuit to this substage consists of decision trees of depth at most $c_{i,j-1}$ with leaves that are ANDs (or ORs) of decision trees of depth at most $c_{i-1} - j + 1$. Assuming it is ANDs of decision trees without loss of generality, each AND gate of the bottom layer is an AND of ORs of fanin at most $c_{i-1} - j + 1$. The restriction of substage j is good if, after applying it, the resulting circuit has bottom layer consisting of decision trees of depth at most $c_{i,j}$ with leaves that are ANDs of ORs of fanin at most $c_{i-1} - j$.

The proof above (of Lemma 4.1) uses Maxset(Q_m) for each AND Q_m of bottom layer. It is not clear how to construct Maxset(Q_m) in AC^0, hence we adopt a different strategy: the AC^0 circuit *directly* checks the desired property of the resulting circuit. This requires checking, for each AND gate Q_m that one of the following two conditions hold:

- One of the OR gates gets all the inputs set to 0 under the restriction.
- There is a subset of at most c' inputs that remain unset by the restriction and every OR gate has at least one of these inputs.

The first condition can be easily checked by a Dlogtime-uniform AC^0 circuit: using the binary encoding of the input circuit to the substage, identify the bottom layer of AND gates and for each such gate first transform its leaves from decision trees to ANDs of ORs (these are constant sized and so can be done trivially); then check if there is an OR whose inputs are all set to 0. For the second condition, the circuit we construct tries out all possible subsets of size $\leq c'$ of the inputs and checks if (1) it remains unset, and (2) it intersects with the input set of every OR gate. Therefore, this is also done in Dlogtime-uniform AC^0. After making these checks, our circuit outputs the binary encoding of the resulting circuit in which all except the bottom layers are copied from the input circuit and for the last layer, each AND gate is replaced either by 0 or by a decision tree of depth at most c' whose leaves are ANDs of ORs of fanin at most $c_{i-1} - j$ depending on which of the two conditions hold.

Putting all the substages one on top of other, we get a Dlogtime-uniform AC^0 circuit that checks the goodness of the restriction given by $\mathcal{H}(s)$ and outputs the reduced depth-2 circuit. ■

5 AC^0-Uniform Gap Theorem

In this section, we prove the AC^0-uniform version of the Gap Theorem of [AAR98]:

Theorem 5.1 *For any class $\mathcal{C}$ closed under TC^0 reductions, all complete sets for $\mathcal{C}$ under u-uniform AC^0 reductions are also complete under ($u + AC^0$)-uniform NC^0 reductions.*

Proof. We begin by outlining the proof in [AAR98].

Fix a set A in $\mathcal{C}$ that is complete under u-uniform AC^0 reductions and let $B \in \mathcal{C}$ be an arbitrary set. We need to show that B reduces to A via a $(u+\mathrm{AC}^0)$-uniform NC^0 reduction. We first define a set $\hat{B}$, which is a highly redundant version of B, as accepted by the following procedure:

> On input y, reject if y contains no zeros. Otherwise, let $y = 1^k 0 z$. Reject if $k = 0$, or $|z| = 0$, or k does not divide $|z|$. Otherwise, break z into q blocks of k consecutive bits each, $|z| = kq$. Let these blocks be $u_1 u_2 u_3 \cdots u_q$. For each i, $1 \leq i \leq q$, let v_i be the parity of the bits in u_i. Reject if every v_i is 1. Otherwise, let $v_1 v_2 \cdots v_q = 1^\ell 0 v$. Accept iff $v \in B$.

As one can readily observe, corresponding to each string in B there are infinitely many strings in $\hat{B}$. Also, $\hat{B}$ reduces to B via a TC^0 reduction and so $\hat{B} \in \mathcal{C}$. Fix a reduction of $\hat{B}$ to A given by u-uniform AC^0 circuit family $\{C_n\}$ of depth d and size n^k. Now define a reduction of B to $\hat{B}$ as follows (it would be useful to keep the above definition of $\hat{B}$ in mind while reading this definition):

> Given an input v, let $x = 1^\ell 0 v$ such that $|x| = n = 2^{2^t} \geq c$ for an appropriate constant c to be fixed later. Let $m = n^s$ for $s = 2\gamma$. Consider the circuit $C_{\frac{m}{n}+1+m}$ with the first $\frac{m}{n}+1$ bits set to $1^{\frac{m}{n}}0$ resulting in circuit C'_m, say. Apply the derandomized Switching Lemma 4.3 on C'_m to obtain a setting of all but n^2 input bits such that the circuit reduces to an NC^0 circuit and in addition, all the n blocks of $\frac{m}{n} = n^{s-1}$ consecutive bits in the input have exactly n unset bits (follows from Lemma 4.4). Now set to 0 all those unset bits that influence at least one of the last $O(\log n)$ bits of the output that encode the length of the output as per our convention. This sets $O(\log n)$ additional unset bits. Since each block has n unset bits to begin with, each block would still have at least two unset bits for large enough n (ensured by appropriate choice of constant c). Now for each of the n blocks, set all but one bit of the block to ensure that the number of ones in the block is 0 modulo 2 (this can always be done using one of the two unset bits available in each block). This sets all the m bits of input to C'_m except for n bits and on these n unset bits the circuit C'_m becomes an NC^0 circuit. Now map x to a string of length $\frac{m}{n}+1+m$ whose first $\frac{m}{n}+1$ bits are set to $1^{\frac{m}{n}}0$ and the remaining bits are set according to the above procedure and the i^{th} remaining unset bit is given the value of i^{th} bit of x.

It is easy to verify that the mapping constructed above is indeed a reduction of B to $\hat{B}$. Notice that this reduction is simply a *projection*: each input bit is mapped to some output bit directly and there are no gates in the circuit computing the reduction. It is also clear that a composition of this reduction with the reduction of $\hat{B}$ to A is a reduction of B to A that can be computed by an NC^0 circuit family. The uniformity machine (or circuit) for this NC^0 circuit

family is required to do the following tasks, apart from generating the circuit C'_m itself:

1. Identity the settings of input bits to circuit C'_m that make the circuit an NC^0 circuit,
2. Given such a setting, transform the circuit C'_m to the equivalent NC^0 circuit, and
3. Set some of the unset bits as outlined above to leave only one unset bit in each block (in which string x would be placed).

The first two tasks can be done by a Dlogtime-uniform AC^0 circuit as shown in Lemma 4.6 and by observing that a good setting can be identified by checking all the seeds of source $\mathcal{H}$ in parallel.

For the third task, a Dlogtime-uniform AC^0 circuit can identify which unset bits influence the output bits coding length of the output and set them all to 0, however, to set the second-to-last unset bit in a block appropriately (so that number of ones is 0 modulo 2), one requires computing parity of $n^{s-1} - 2$ bits. This *cannot* be done by even non-uniform AC^0 circuits!

We solve this problem by modifying the source G_I in the definition of $\mathcal{H}$ slightly. Each copy of G_I in $\mathcal{H}$ is required to be a $(\hat{c} \log n)$-wise independent $\frac{1}{n^{k+4}}$-biased source for an appropriate constant $\hat{c}$. Let $\tilde{c} > \hat{c}$ be a power of two. Change G_I by setting every $(\tilde{c} \log n)$-th bit to be the parity of the previous $\tilde{c} \log n - 1$ bits. Since $\tilde{c} > \hat{c}$, the modified source remains $(\hat{c} \log n)$-wise independent with a similar bias. The modified G_I now has the property that, splitting it into blocks of size $\tilde{c} \log n$ bits, the parity of each block is zero.

Observe that during any substage of the transformation of C'_m to an NC^0 circuit, the number of unset bits in each block is a power of n. This implies that the number of bits set during any substage in every block is a multiple of n. Also, since $n = 2^{2^t}$ and $\tilde{c}$ is a power of two, $\tilde{c} \log n$ divides n for large enough n (ensured by appropriate choice of constant c). Therefore, the parity of bits contributed by each copy of G_I to every block will always be zero, and the third task takes care of itself automatically!

This completes the proof of the theorem. ■

6 AC^0-Uniform Superprojection Theorem

We start with the definition of a superprojection [AAR98].

Definition 6.1 An NC^0 reduction $\{C_n\}$ is a *superprojection* if the circuit that results by deleting zero or more of the output bits in each C_n is a projection wherein each input bit (or its negation) is mapped to some output.

Now we prove the AC^0-uniform Superprojection Theorem:

Theorem 6.2 *For any class $\mathcal{C}$ closed under* TC^0 *reductions, all complete sets for $\mathcal{C}$ under u-uniform* NC^0 *reductions are also complete under (u + AC^0)-uniform superprojections.*

Proof. Fix a set A in $\mathcal{C}$ that is complete under u-uniform NC^0 reductions and let $B \in \mathcal{C}$ be an arbitrary set. We need to show that B reduces to A via a (u + AC^0)-uniform superprojection. We first define, as before, a set $\hat{B}$ as accepted by the following procedure:

> On input y, let $y = z'11z$ such that $z' \in \{00, 01, 10\}^*$ (reject if y is not of this form). Break z' into pairs of bits. Ignoring all the 00 pairs, consider the first $\lceil \log |z| \rceil$ pairs. If there are fewer than $\lceil \log |z| \rceil$ such pairs, then reject. Define number k by setting the i^{th} bit of k to 1 if the i^{th} of the above $\lceil \log |z| \rceil$ pairs is 10, to 0 otherwise. Reject if $k = 0$, or $|z| = 0$, or k does not divide $|z|$. Else, break z into blocks of k consecutive bits each. Reject if the number of blocks is not a multiple of four. Else, let $z = u_1u_2u_3 \cdots u_{4q}$ with $|u_i| = k$. Let v_i be the parity of bits in u_i. Let $w_i = v_{4i-3}v_{4i-2}v_{4i-1}v_{4i}$ for $1 \leq i \leq q$ (so each w_i is a four bit string). If $w_i = 1111$ for any $1 \leq i \leq q$, accept. Else if some w_i has exactly three ones, reject. Else, for each i, $1 \leq i \leq q$, let $b_i = 1$ if w_i has exactly two ones, $b_i = 0$ if w_i has exactly one one, $b_i = \epsilon$ otherwise. Reject if no b_i is zero. Otherwise, let $b_1b_2 \cdots b_q = 1^p0v$. Accept iff $v \in B$.

The definition of set $\hat{B}$ is more complicated that the previous one. Even the block size ($= k$) is coded in the string in a non-straightforward way. We refer to the bits of z' of any instance y of $\hat{B}$ as *length encoder bits* and to the bits of z as *string encoder bits*. It is easy to see that $\hat{B}$ reduces to B via a TC^0 reduction and so $\hat{B} \in \mathcal{C}$. Fix a reduction of $\hat{B}$ to A given by a u-uniform NC^0 circuit family $\{C_n\}$. Let each output bit of any circuit C_n depend on at most c input bits.

As before, we now define a reduction of B to $\hat{B}$. The idea is same: for an appropriate m and ℓ, consider the circuit $C_{\ell+2+m}$. Set some of the input bits of $C_{\ell+2+m}$ so that the circuit on the remaining unset bits is a superprojection. Now set some more bits (including all of length encoder bits) to satisfy all the conditions in the definition of set $\hat{B}$ and finally map string x to the remaining unset bit positions.

We first discuss a simple idea that does not work directly. Say that an input bit *influences* an output bit if the value of the output bit is a non-trivial function of the value of the input bit. In other words, there is a setting of all other input bits under which the output bit value changes on changing the value of the input bit.

> Consider circuit $C_{\ell+2+m}$ with first $\ell+2$ bits set to $s11$ where s codes the length $\frac{m}{n}$ for a suitable n (thus, the block size is $\frac{m}{n}$). Randomly set every unset input bit of the circuit to 0 or 1 with probability $\frac{1}{4}$ each, and leave it unset with probability $\frac{1}{2}$. Say that an input bit

in the string encoder part is *good* if it remains unset and there is at least one output bit that now depends *only on this bit*. For any input bit that influences some output bit in $C_{\ell+2+m}$, the probability that this bit is good is at least $\frac{1}{2} \cdot (\frac{1}{4})^{c-1} > \frac{1}{4^c}$. Therefore, the expected number of good input bits is $\Omega(m')$ where m' is the number of input bits in the string encoder part of $C_{\ell+2+m}$ that influence at least one output bit. Identify all the good bits and set all the other unset input bits appropriately. This makes the circuit $C_{\ell+2+m}$ on the remaining unset bits a superprojection.

The above construction yields $\Omega(m)$ good bits provided we can ensure that nearly all the input bits influence the output (part of the complexity in definition of $\hat{B}$ is due to this requirement). The construction can easily be derandomized by using a $2c$-wise independent source for selecting unset bits and setting the remaining bits. However, this does not guarantee that in every block (of $\frac{m}{n}$ bits) at least one good bit is present (because the events that two bits are good are not independent of each other). This makes the mapping of bits of x difficult as we need to use threshold gates to find the i^{th} unset bit.

We solve this problem in a similar fashion to the handling of ANDs of bounded fanin ORs in the proof of Lemma 4.1: either every block will have a good bit with high probability or we can reduce the number of input bits that influence an output bit by one with high probability.

We now expand this idea in a way that the entire transformation can be done by a Dlogtime-uniform AC^0 circuit. Let v be an instance of B, and $x = 1^p 0 v$ such that $|x| = n = 2^{2^t} > c_0$ for a suitable constant c_0. Let $m = (4n^2)^c$. Consider the circuit $C_{2c^2 4^{c+1} \log m + 2 + m}$. To begin with, set the bit numbers $2c^2 4^{c+1} \log m + 1$ and $2c^2 4^{c+1} \log m + 2$ of the input to $C_{2c^2 4^{c+1} \log m + 2 + m}$ to 1 identifying the first $2c^2 4^{c+1} \log m$ bits as length encoder and the last m bits as string encoder bits. We will consider length encoder bits in pairs; so there are $c^2 4^{c+1} \log m$ pairs of such bits. Let C be the resulting circuit.

We use a stagewise construction such that each stage sets some more bits of input to C and simplifies it. In the last stage of the construction, we obtain a reduction. At the beginning of the $(k+1)^{st}$ stage, when the k^{th} stage was not the final stage, the circuit C has the following properties:

- There are exactly $(c-k)^2 4^{c+1} \log m$ unset pairs of length encoder bits, and those length encoder bit pairs that have already been set are set to 00.
- The $(4n^2)^c$ string encoder bits are divided into $(4n^2)^k$ blocks, each consisting of $(4n^2)^{c-k}$ consecutive bits. One of these blocks has all bits unset, and all other string encoder bits are set to 0.
- Every output bit of circuit C depends on at most $c-k$ unset input bits.

For $k = 0$, this is trivially true.

In the $(k+1)^{st}$ stage, split the unset string encoder bits of the input to C into $4n$ blocks of equal size $(= n \cdot (4n^2)^{c-k-1})$. Firstly, notice that *every bit in*

every block must influence some output bit. Suppose not. Let such a bit belong to the $(4i+j)^{th}$ block, $0 \leq i < n$, $1 \leq j \leq 4$. Set all the bits in all the blocks, except for block numbers $4i+1$ through $4i+4$, to 0. Set bits in blocks $4i+1$ through $4i+4$ except those in block $4i+j$ such that each of these blocks has an odd number of bits that are set to 1. Set all the bits in the block $4i+j$ except the bit that does not influence any output bit to 0. Set the unset length encoder bits such that the block size is $n \cdot (4n^2)^{c-k-1}$. This fixes the output of circuit C. However, the value of the lone unset bit decides whether the input string belongs to the set $\hat{B}$ or not, contradicting the fact that family $\{C_n\}$ computes a reduction of $\hat{B}$ to A.

Consider the j^{th} block. Let $o_1, \ldots, o_p$ be all the output bits of the circuit that are influenced by some bit in the block. For output bit o_i, let I_i be the set of input bits that influence o_i. Clearly, $|I_i| \leq c-k$. Let Maxset_j be any *maximal* set of disjoint I_is. Now there are two cases.

Case I. There is a j_0 such that $|\text{Maxset}_{j_0}| < 4^{c+1} \log m$. Set all the unset bits in all other blocks to 0. Split this block into n subblocks of size $(4n^2)^{c-k-1}$ each. For large enough n, one of these subblocks will not intersect any of the sets in Maxset_{j_0} since the total number of bits in Maxset_{j_0} is at most $(c-k)4^{c+1} \log m < n$. Fix this subblock and set bits in all other subblocks to 0. Set all the length encoder pairs that have one or both of their bits present in Maxset_{j_0} to 00. This sets at most $(c-k)4^{c+1} \log m$ pairs leaving at least $(c-k-1)^2 4^{c+1} \log m$ unset pairs. Set some more of these pairs to 00 to leave exactly $(c-k-1)^2 4^{c+1} \log m$ unset pairs. This sets all the bits in Maxset_{j_0} besides setting all other blocks. Hence, every output bit will now be influenced by at most $c-k-1$ bits. Go to Stage $k+2$.

Case II. $|\text{Maxset}_j| \geq 4^{c+1} \log m$ for every $1 \leq j \leq 4n$. This is the last stage. For each j, remove those I_i's from Maxset_j that contain any bit from the first $\log m$ unset length encoder pairs. This will still leave at least $4^{c+1} \log m - 2 \log m \geq 4^c \log m$ sets in Maxset_j.

Now apply a random restriction on the input of C in the following way. Randomly set all but first $\log m$ unset pairs of length encoder bits using a truly random source. Use two $\frac{1}{2n^2}$-biased, $(c4^c \log m)$-wise independent sources G_I^0 and G_I^1 with independent seeds to generate two sequences of random bits. Set the i^{th} unset string encoder bit to the i^{th} bit of G_I^0 if the i^{th} bit of G_I^1 is 0; leave the bit unset otherwise. Thus, the i^{th} string encoder bit is left unset with probability close to $\frac{1}{2}$ and is set to 0 or 1 with probabilities close to $\frac{1}{4}$ each. Also observe that settings to any collection of $c4^c \log m$ string encoder bits are almost independent: This follows from the fact that the corresponding $c4^c \log m$ bits of both the sources, viewed as a collection of $2c4^c \log m$ bits, are independent with a bias of at most $\frac{1}{n^2}$.

Consider the set Maxset_j for some j. Drop some I_i's from Maxset_j to retain exactly $4^c \log m$ sets. Each of these sets contains at least one bit

from the j^{th} block. The probability that this bit becomes good under the above assignment restriction is at least $\frac{1}{24^{c-k-1}} - \frac{1}{n^2} \geq \frac{1}{4^{c-k}}$. Hence, by independence of the source, the probability that none of I_i's has a good bit is at most $(1 - \frac{1}{4^c})^{4^c \log m} + \frac{1}{n^2} \leq \frac{1}{m} + \frac{1}{n^2}$. Therefore, the probability that for some j, none of the j^{th} block has a good bit is at most $\frac{4n}{m} + \frac{4}{n} < \frac{1}{2}$.

So there exists a random restriction that (1) leaves the first $\log m$ pairs of length encoder bits unset, and (2) leaves at least one good bit in each of the $4n$ blocks. Now use the length encoder pairs to code the block size $n \cdot (4n^2)^{c-k-1}$. Set all except the first good bit in each block as well as all the remaining unset bits to the corresponding bit value of G_I^0. Group the $4n$ blocks into n groups of 4 blocks each. For each group, set the unset good bit in each of the last two blocks so that the parity of bits in those blocks is even, and set the unset bit in the second block so that the parity of that block is odd. This leaves exactly n unset good bits; one in each group. Map the j^{th} bit of x to the unset bit of the j^{th} group.

This defines a projection reduction of B to $\hat{B}$, and on this output the circuit C is a superprojection. Hence their composition is a superprojection reduction of B to A.

We need to show that (1) Case II eventually occurs, and (2) the entire construction can be done by a Dlogtime-uniform AC^0 circuit. For (1), observe that after $c-1$ stages, the circuit C has $4^{c+1} \log m \geq \log m$ unset pairs of length encoder bits, $4n^2$ string encoder bits, and every output bit of C depends on at most one unset input bit. Hence, $|\text{Maxset}_j| = n$ for every block of size n. Therefore, after at most $c-1$ stages, Case II occurs.

We now look at uniformity of the reduction. Since computing Maxset_j is difficult, we distinguish the two cases in a different way. The uniformity circuit will work in stages, and after the k^{th} stage, will output the binary encoding of the circuit C after the k^{th} stage. In the k^{th} stage, the circuit, given the binary encoding of C after the $(k-1)^{st}$ stage, checks if there exists a subblock (there are $4n^2$ subblocks each of length $(4n^2)^{c-k}$ bits) such that by setting all other subblocks to 0 and by setting some $(c-k)4^{c+1} \log m$ pairs of length encoder bits to 00, every output bit of C is influenced by at most $c-k$ unset input bits. This can be done by checking all $4n^2$ subblocks and $\binom{(c-k)^2 4^{c+1} \log m}{(c-k)4^{c+1} \log m} = m^{O(1)}$ possible choices of length encoder pairs in parallel, and then picking the first one for which it is true. Once a subblock and length encoder pairs are found, the circuit makes the appropriate settings and outputs the binary encoding of the resulting circuit.

If there is no such subblock, then there must exist a random restriction that leaves the first $\log m$ pairs of length encoder bits unset and leaves at least one good bit in each block. By checking all seeds of the two sources and all random settings to length encoder bits in parallel (there are at most $m^{O(1)}$ of these), the uniformity circuit can identify one such restriction. It then sets most of the remaining unset bits as described above and outputs the binary encoding of the resulting circuit. However, there is a problem in setting the remaining bits:

these must be set to ensure that the parity of all the set bits in each block is the desired value, which cannot be ensured in general. So we use the same idea as before: by modifying the source G_I^0, we can ensure that the parity of every block of suitable size, say $2^{\hat{b}}$, in the output of G_I^0 is zero. However, our problem is still not fully solved since the number of set bits in a block is exactly $(4n^2)^{c-k} - 1$ (for some k) which is *not* divisible by $2^{\hat{b}}$. We solve this by a simple trick: $2^{\hat{b}}$ divides the block size $(4n^2)^{c-k}$; so associate a sign with the bit remaining unset in each block, its value is given by the corresponding bit of the source G_I^0; when setting the value of each of these bits (to 0, 1, or a bit of x), apply the sign also in setting (for example, if the sign is 1 and the value to be set is b, set it to $\bar{b}$). This ensures that the parity of all bits in a block has the desired value.

All the above steps can be carried out by a Dlogtime-uniform AC^0 circuit, hence completing the proof. ■

7 Dlogtime-uniform Isomorphism Theorem

We are now ready to prove the main result of the paper:

Theorem 7.1 *For any class $\mathcal{C}$ closed under TC^0 reductions, all complete sets for $\mathcal{C}$ under u-uniform AC^0 reductions are isomorphic to each other under u-uniform AC^0-isomorphisms where u is any measure of uniformity containing Dlogtime.*

Proof. Let A and B be two complete sets for $\mathcal{C}$ under u-uniform AC^0 reductions. By the theorems above, A and B reduce to each other via $(u + \mathrm{AC}^0)$-uniform superprojections. It was shown in [AAR98] how to construct $(u + \mathrm{AC}^0)$-uniform AC^0-isomorphisms between A and B. Now the AC^0 circuit in the uniformity part can be combined with the AC^0 circuit computing the isomorphism to obtain another AC^0 circuit computing the isomorphism. This new AC^0 circuit will be u-uniform since u contains Dlogtime. ■

Corollary 7.2 *Complete sets for classes* DLOG, NLOG, NC^k *($k \geq 1$),* P, NP *under first-order reductions are first-order isomorphic to each other.*

Acknowledgement. The author wishes to thank the anonymous referee whose suggestions helped clarify confusing points at several places in an earlier version of the paper.

References

[AAI+01] M. Agrawal, E. Allender, R. Impagliazzio, T. Pitassi, and S. Rudich. Reducing the complexity of reductions. *Computational Complexity*, 10(2):117–138, 2001.

[AAR98] 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.

[ABI97] E. Allender, J. Balcázar, and N. Immerman. A first-order isomorphism theorem. *SIAM Journal on Computing*, 26(2):557–567, 1997.

[AG91] E. Allender and V. Gore. Rudimentary reductions revisited. *Information Processing Letters*, 40:89–95, 1991.

[AGHP92] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple constructions of almost k-wise independent random variables. *Random Struct. Algorithms*, 3(3):289–304, 1992.

[Agr96] M. Agrawal. On the isomorphism problem for weak reducibilities. *J. Comput. Sys. Sci.*, 53(2):267–282, 1996.

[Agr01a] M. Agrawal. The first order isomorphism theorem. In *Proceedings of the FST&TCS*, pages 70–82. LNCS 2245, 2001.

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

[BDG88] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I.* EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1988.

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

[BIS90] D. Barrington, N. Immerman, and H. Straubing. On uniformity within NC^1. *J. Comput. Sys. Sci.*, 74:274–306, 1990.

[CSS97] J. Cai, D. Sivakumar, and M. Strauss. Constant depth circuits and the Lutz Hypothesis. In *Proceedings of Annual IEEE Symposium on Foundations of Computer Science*, pages 595–604, 1997.

[CSV84] A. Chandra, L. Stockmeyer, and U. Vishkin. Constant depth reducibility. *SIAM Journal on Computing*, 13:423–439, 1984.

[ER60] P. Erdös and R. Rado. Intersection theorems for systems of sets. *J. London Math. Soc.*, 35:85–90, 1960.

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

[FSS84] M. Furst, J. Saxe, and M. Sipser. Parity, circuits, and the polynomial hierarchy. *Mathematical Systems Theory*, 17:13–27, 1984.

[Hås86] 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.

[IL95] N. Immerman and S. Landau. The complexity of iterated multiplication. *Information and Computation*, 116:103–116, 1995.

[Imm87] N. Immerman. Languages that capture complexity classes. *SIAM Journal on Computing*, 16:760–778, 1987.

[Jon75] N. Jones. Space-bounded reducibility among combinatorial problems. *J. Comput. Sys. Sci.*, 11:68–85, 1975.

[KMR88] 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.

[KMR95] S. Kurtz, S. Mahaney, and J. Royer. The isomorphism conjecture fails relative to a random oracle. *J. ACM*, 42(2):401–420, 1995.

[Lin92] S. Lindell. A purely logical characterization of circuit complexity. In *Proceedings of the Structure in Complexity Theory Conference*, pages 185–192, 1992.

[NN93] J. Naor and M. Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM Journal on Computing*, 22(4):838–856, 1993.

[NW94] N. Nisan and A. Wigderson. Hardness vs. randomness. *J. Comput. Sys. Sci.*, 49(2):149–167, 1994.

[Sip83] M. Sipser. Borel sets and circuit complexity. In *Proceedings of Annual ACM Symposium on the Theory of Computing*, pages 61–69, 1983.