Graph Isomorphism is not AC^0 reducible to Group Isomorphism

Arkadev Chattopadhyay[1] Jacobo Torán[2] Fabian Wagner[2]

[1]Department of Computer Science,
University of Toronto, ON M5S 3G4 Canada,
arkadev@cs.toronto.edu,

[2]Institut für Theoretische Informatik,
Universität Ulm, 89069 Ulm, Germany,
{jacobo.toran, fabian.wagner}@uni-ulm.de

July 22, 2010

Topics: Complexity, Algorithms, Isomorphism Problems, Circuit Complexity.

Abstract

We give a new upper bound for the Group and Quasigroup Isomorphism problems when the input structures are given explicitly by multiplication tables. We show that these problems can be computed by polynomial size nondeterministic circuits of unbounded fan-in with $O(\log\log n)$ depth and $O(\log^2 n)$ nondeterministic bits, where n is the number of group elements. This improves the existing upper bound from [Wol94] for the problems. In the previous upper bound the circuits have bounded fan-in but depth $O(\log^2 n)$ and also $O(\log^2 n)$ nondeterministic bits. We then prove that the kind of circuits from our upper bound cannot compute the Parity function. Since Parity is AC^0 reducible to Graph Isomorphism, this implies that Graph Isomorphism is strictly harder than Group or Quasigroup Isomorphism under the ordering defined by AC^0 reductions.

1 Introduction.

The input of the Group Isomorphism problem GroupIso consists in two groups G_1 and G_2 of order n given by multiplication tables ($n \times n$ matrices of integers between 1 and n) and it is asked whether the groups are isomorphic, that is, whether there is a bijection φ between the elements of both groups satisfying for every pair of elements i, j, $\varphi(ij) = \varphi(i)\varphi(j)$[1].

[1] For convenience we represent in both quasigroups the group operation by concatenation.

A quasigroup is an algebraic structure $(\Omega, \cdot)$ where the set Ω is closed under a binary operation $\cdot$ that has the following property: for each pair of elements a, b, there exists a unique element c_L (and c_R) such that $c_L \cdot a = b$ (and $a \cdot c_R = b$). In contrast to groups, a quasigroup is not necessarily associative and does not need to have an identity. The Quasigroup Isomorphism problem QGroupIso is defined as GroupIso but the input structures are multiplication tables of quasigroups, also called Latin squares. GroupIso is trivially reducible to QGroupIso but a reduction in the other direction is not known. The complexity of both problem has been studied for more than three decades. Groups and quasigroups of order n have generator sets of size bounded by $\log n$. Because of this fact an isomorphism algorithm for GroupIso or QGroupIso running in time $n^{\log n + O(1)}$ can be obtained by computing a generator set of size $\log n$ in G_1, mapping this set in every possible way to a set of elements in G_2 and checking that by extending the mapping to all the (quasi)group elements following the multiplication tables of G_1 and G_2, an isomorphism is defined. This algorithm is attributed to Tarjan in [Mil78]. A stronger result showing that GroupIso can be solved in space $O(\log^2 n)$ was given in [LSZ76]. The same result for the case of quasigroups was obtained later by Wolf in [Wol94].

In spite of these facts, no deterministic polynomial time algorithm for these problems is known although they seem far from being NP-complete[2]. The status of the problems is similar to that of the better known Graph Isomorphism problem (GI). It is known that QGroupIso is AC^0 reducible to GI [Mil78], but the second one seems to be a harder problem. In this paper we prove this intuition by showing without assumptions that an AC^0 reduction in the other direction is not possible. This is done in two steps: first we improve the existing upper bound for QGroupIso to a class of polynomial size nondeterministic circuits of $O(\log \log n)$ depth (Section 3). Then in Section 4 we show that this circuit class cannot compute the Parity function. It follows that GroupIso and QGroupIso cannot be hard under AC^0 reductions for any class that is powerful enough to compute Parity, like NC^1 or L. This contrasts with the hardness properties of GI [Tor04, Tor10]. It also implies that GI cannot be AC^0 reducible to GroupIso or to QGroupIso .

The upper bound is based on the bounded nondeterminism properties of the problems. Observe that Tarjan's algorithm can in fact be converted into a polynomial time nondeterministic procedure for QGroupIso that uses only $\log^2 n$ nondeterministic bits, by guessing the mapping from the generator set in G_1 to G_2 instead of testing all possible 1-1 mappings, and then extend this partial map to the whole quasigroup. This observation is mentioned explicitly in [PY96, Wol94]. Papadimitriou and Yannakakis [PY96] show that the quasigroup isomorphism problem is in β_2P, a restricted version of NP, where on input of length n, a polynomial time bounded Turing machine has access to $O(\log^2 n)$ non-deterministic bits (more detail is given in the preliminaries section). In [AT06] some evidence is given indicating that QGroupIso is probably not complete for β_2P. Wolf [Wol94] improved the nondeterministic complexity of the problem by showing that QGroupIso $\in \beta_2\mathrm{NC}^2$ the class of problems computed by NC^2 circuits having additionally $O(\log^2 n)$ non-deterministic bits

[2] In fact, as we show in this paper, GroupIso and QGroupIso cannot be NP-complete under AC^0 reductions.

on inputs of size n. As in the β_2P upper bound, the circuit can guess the generators of both quasigroups as well as a bijection between both generator sets. Wolf shows that checking whether this partial bijection can be extended to an isomorphism, can be done by an NC^2 circuit. We improve this upper bound to β_2FOLL, the class of problems computable by (uniform) families of polynomial size unbounded fan-in circuits with $O(\log \log n)$ depth and $O(\log^2 n)$ nondeterministic bits, where n is the number of quasigroup elements. The proof of this result is based on a special kind of generating sequences for the quasigroups called cube generating sequences. The cube generating sequences provide a representation for the structures that allow very quick isomorphism tests. Erdős and Rényi showed that groups have many generating sequences of this kind. We extend in Section 3 their result to the more general case of quasigroups.

The lower bound for β_2FOLL circuits for the Parity function is proved, in Section 4, by first showing that computation by a few non-deterministic bits implies the existence of a polynomial size deterministic circuit of depth $O(\log \log n)$ that approximates Parity non-trivially. The argument is completed by a routine application of the decision-tree version of the Switching Lemma due to Razborov [Raz93] that rules out such approximations of Parity.

2 Preliminaries

2.1 Quasigroups

Given a set of elements from a quasigroup, a parenthesization specifies the sequence in which to multiply the elements. A parenthesization can be represented as a binary tree with the quasigroup elements at the leaves. The depth of a parenthesization is the depth of the binary tree representing it. For a quasigroup G and a set of elements $g_1, \ldots, g_l \in G^l$ and a parenthesization P we denote by $P(g_1, \ldots, g_l)$ the result of the multiplication of the elements according to P. For our results we need the following elementary fact.

Fact 2.1 *Let P be any correct parenthesization for the multiplication of l elements in G. Then for every $i \in \{1, \ldots, l\}$ for every $b \in G$ and every fixed choice of elements $g_1, \ldots, g_{i-i}, g_{i+1}, \ldots, g_l, b \in G^l$ there is a unique element $g_i \in G$ such that $P(g_1, \ldots, g_{i-i}, g_i, g_{i+1}, \ldots, g_l) = b$.*

Proof: By induction on l. For the base case $l = 2$ the quasigroup axioms imply the result. For $l > 2$ we consider the binary tree representing the parenthesization. We search for a value of g_i such that the equation $P(g_1, \ldots, g_{i-i}, g_i, g_{i+1}, \ldots, g_l) = b$ holds. The value of one of the successors of the root is determined by the values of $g_1, \ldots, g_{i-i}, g_{i+1}, \ldots, g_l$. W.l.o.g. let this be the left successor and denote its multiplication value by c. The value of the other successor must then be equal to d, the unique element in G with $c \cdot d = b$. By induction hypothesis there is a unique value for q_i such that the multiplication of the right subtree equals d. □

2.2 Complexity Classes

For the standard complexity classes used in this paper, like L or the circuit classes AC^i or NC^i we refer the reader to the standard books in complexity theory.

The complexity class FOLL, or $\mathsf{FO}(\log\log n)$ was introduced in [BKLM00] in order to characterize the complexity of the group membership problem. FOLL is the class of problems solvable by uniform polynomial size circuit families of unbounded fan-in and depth $O(\log\log n)$. Since the Parity function is not in FOLL, no problem in FOLL can be complete under AC^0-reductions for any class containing Parity, such as NC^1 or L. Currently AC^1 is the best upper bound for FOLL and the class is not known to be contained even in NL.

For a circuit class C, $\beta_k C$ is the class of languages recognized by a (uniform) family of C circuits with n input bits and $O(\log^k n)$ nondeterministic bits. We say that such a nondeterministic circuit accepts an string x if for some choice of the nondeterministic bits the circuit with input x outputs a one. Classes of bounded nondeterminism have appeared in different forms in the literature [KF84, DT90, PY96, GLM96]. As we show in this paper, the circuit setting is well suited to argue about these classes.

3 Nondeterministic Circuit Complexity of QGroupIso

We show in this section that GroupIso can be solved by a uniform family of nondeterministic FOLL circuits with $O(\log^n)$ nondeterministic bits: QGroupIso $\in \beta_2$FOLL. This result improves a series of upper bounds of this kind for the problem: Papadimitriou and Yannakakis showed in [PY96] that QGroupIso $\in \beta_2$P this was improved to β_2NC^{2}by Wolf [Wol94] and more recently by Wagner to β_2SAC^1 [Wag10].

In our proof the nondeterministic bits of the circuits are used in order to guess a special kind of generator sequence for both quasigroups. We call these generators cube generating sequences.

Definition 3.1 *A sequence of group elements $g = (g_0, g_1, \ldots, g_k)$ together with a parenthesization P for k elements is a* cube generating sequence *for quasigroup G if*

$$G = \{P(g_0, g_1^{\epsilon_1}, \ldots, g_k^{\epsilon_k}) \mid \epsilon_i \in \{0,1\}\}$$

The set $\{P(g_0, g_1^{\epsilon_1}, \ldots, g_k^{\epsilon_k}) \mid \epsilon_i \in \{0,1\}\}$ is the cube $Cube(g, P)$ *generated by the sequence g and the parenthesization P.*

In a cube generating sequence, the generators are given in a fixed order. Erdős and Renyi [ER65] proved that every group with n elements has cube generating sequences of size $O(\log n)$. As a matter of fact there are many such short sequences. In the case of groups we do not need to talk about parenthesizations since the operation is associative.

Theorem 3.2 *[ER65] Let G be a finite group with n elements. For any $\delta > 0$ the probability that a sequence of group elements of size $k \geq \log n + 2\log\frac{1}{\delta} + \log\log n + 5$ selected uniformly at random is a cube generating sequence for G, is $> 1 - \delta$.*

This result can be adapted to work also for quasigroups. For our purposes a simpler existential version of the result suffices. However we need to make sure that the multiplications of the generators can be be performed very fast in parallel and therefore we need a a short cube generation sequence with shallow parentisation.

Theorem 3.3 *For a finite quasigroup G with n elements, there exists a cube generating sequence g for G, together with a parenthesization P such that g has $O(\log n)$ elements and P has depth $O(\log\log n)$.*

Proof: Let G be a quasigroup with n elements and for $k > 0$ let P be any fixed parenthesization of $k+1$ elements. Let $g_0, \ldots, g_k$ be $k+1$ elements chosen in G uniformly at random and independently of each other. For $b \in G$ let $V_k(b)$ be the number of representations of b of the form $b = P(g_0, g_1^{\epsilon_1}, \ldots, g_k^{\epsilon_k})$ with $\epsilon_i \in \{0,1\}$. For succinctness for $\epsilon = (\epsilon_1, \ldots, \epsilon_k) \in \{0,1\}^k$ and $g = (g_0, \ldots, g_k) \in G^{k+1}$ we represent $P(g_0, g_1^{\epsilon_1}, \ldots, g_k^{\epsilon_k})$ by $P(g^\epsilon)$ (or even g^ϵ when the parenthesization is clear).

For each $b \in G$, $V_k(b)$ is a random variable. We estimate its expectation and its variance. For a random sequence $g = (g_0, \ldots, g_k) \in G^{k+1}$ consider the indicator variable

$$X_\epsilon(b) = \begin{cases} 1 & \text{if } g^\epsilon = p \\ 0 & \text{otherwise.} \end{cases}$$

For random g, $\Pr[X_\epsilon(b) = 1] = \frac{1}{n}$. This is because $X_\epsilon(b) = 1$ if and only if $g^\epsilon = b$ and this is true exactly when g_0 is equal to the unique element $x \in G$ satisfying the equation $b = P(x, g_1^{\epsilon_1}, \ldots, g_k^{\epsilon_k})$ (Fact 2.1). Since g_0 is chosen uniformly at random this probability is $\frac{1}{n}$. It follows:

$$E[V_k(b)] = E[\sum_{\epsilon \in \{0,1\}^k} X_\epsilon(b)] = \sum_{\epsilon \in \{0,1\}^k} E[X_\epsilon(b)] = \frac{2^k}{n}.$$

For the variance we need to estimate the probability $\Pr[g^\epsilon = g^{\epsilon'} = b]$ for a random $g \in G^{k+1}$ and fixed $b \in G$ and $\epsilon, \epsilon' \in \{0,1\}^k$. In the case when $\epsilon = \epsilon'$ the probability is equal to $\frac{1}{n}$ as already explained. For the case $\epsilon \neq \epsilon'$ we can suppose there is a position i with $\epsilon_i = 1$ and $\epsilon'_i = 0$. $g^{\epsilon'} = b$ if and only if g_0 is equal to the unique element $x \in G$ satisfying the equation $b = P(x, g_1^{\epsilon'_1}, \ldots, g_k^{\epsilon'_k}$. If this holds then $g^\epsilon = b$ if and only if g_i is equal to the unique element $y \in G$ satisfying $b = P(x, g_1^{\epsilon_1}, \ldots, g_{i-1}^{\epsilon_{i-1}}, y, g_{i+1}^{\epsilon_{i+1}}, \ldots, g_k^{\epsilon'_k})$. Since g_0 and g_i are chosen independently, the probability that these two facts hold is then $\frac{1}{n^2}$. Now we can estimate the variance of $V_k(b)$.

$$\begin{aligned}
Var[V_k(b)] &= E[V_k^2(b)] - E[V_k(b)]^2 = \\
&= E[(\sum_{\epsilon\in\{0,1\}^k} X_\epsilon(b))(\sum_{\epsilon\in\{0,1\}^k} X_\epsilon(b))] - \frac{2^{2k}}{n^2} = \\
&= E[\sum_{\epsilon,\epsilon'\in\{0,1\}^k} X_\epsilon(b)X'_\epsilon(b)] - \frac{2^{2k}}{n^2} = \\
&= \sum_{\epsilon\in\{0,1\}^k} E[X_\epsilon(b)] + \sum_{\epsilon\neq\epsilon'\in\{0,1\}^k} E[X_\epsilon(b)X'_\epsilon(b)] - \frac{2^{2k}}{n^2} = \\
&= \frac{2^k}{n} + 2^k(2^k-1)\Pr[g^\epsilon = g^{\epsilon'} = b] - \frac{2^{2k}}{n^2} = \\
&= \frac{2^k}{n} + \frac{2^k(2^k-1)}{n^2} - \frac{2^{2k}}{n^2} \leq \frac{2^k}{n}
\end{aligned}$$

Let N_k be the number of elements in G not having any representation in the cube generated by a random sequence g of size k+1. We show next that $E[N_k] \leq \frac{n^2}{2^k}$. For this we need to estimate the probability that for an element $b \in G$, $V_k(b) = 0$.

$$\begin{aligned}
\Pr[V_k(b) = 0] &\leq \Pr[\ |V_k(b) - \frac{2^k}{n}| \geq \frac{2^k}{n}] \\
&\leq \frac{\mathrm{Var}[V_k(b)]n^2}{2^{2k}} = \frac{n}{2^k}
\end{aligned}$$

The third step follows by Chebyshev's inequality. We can now estimate the expectation for N_k.

$$E[N_k] = \sum_{b\in G} \Pr[V_k(b) = 0] \leq \frac{n^2}{2^k}.$$

Considering $k = \lceil 2\log n\rceil + 1$ we have $E[N_k] < 1$, which means that there must be a sequence g of size $k+1$ that represents all the elements in G. Since this works for any parenthesization we can fix P to be a balanced binary tree with $k+1$ leaves and therefore depth $O(\log\log n)$.

□

Observe that for a quasigroup G, a fixed k and a fixed parenthesization P, the family of functions obtained by choosing a sequence g of $k+1$ elements in G uniformly at random and mapping $\epsilon \in \{0,1\}^k$ to $g^\epsilon \in G$ (with parenthesization P) is in fact a 2-universal family of hash functions. As it can be seen in our previous proof, the argument does not need fully independence while choosing the elements in G, but just pairwise independence. As a consequence it is possible to obtain small cube generating sets for G deterministically. However this would not bring any advantage to our nondeterministic algorithm, since $O(\log^2 n)$ nondeterministic bits are needed to guess the cube generating set of the second input structure in a way that the isomorphism can be extended to all the elements in the canonical way.

We can now prove our upper bound for QGroupIso .

Theorem 3.4 *The Quasiroup Isomorphism problem is in β_2FOLL.*

Proof: Let G, H be two quasigroups given as multiplication tables let $g = (g_1, \dots, g_k)$ and $h = (h_1, \dots, h_k)$ be generating sequences of the same length, and P be a balanced parenthesization with $G = \text{Cube}(g, P)$ and $H = \text{Cube}(h, P)$.

If we can prove that the function that maps g_i to h_i for $i \in \{1, \dots, k\}$ can be extended to an isomorphism between G and G' then clearly both quasigroups are isomorphic. This is true if and only if for every $\epsilon, \epsilon', \epsilon''\{0,1\}^k$ $g^\epsilon = g^{\epsilon'}g^{\epsilon''}$ if and only if $h^\epsilon = h^{\epsilon'}h^{\epsilon''}$. On the other hand if the quasigroups are not isomorphic, the function mapping g_i to h_i would not pass the mentioned isomorphism test.

This is the basis for the upper bound. $O(\log^2 n)$ nondeterministic bits in the circuit circuit are used for guessing the cube generating sequences for G and H in the right order. The isomorphism tests can be done then in the depth of the multiplications which is the depth of the parenthesization P, $O(\log\log n)$.

input: Quasigroups G, H on elements in $\{1, \dots, n\}$ given as multiplication tables, cube generating sequences $g = (g_0, g_1, \dots, g_k)$ for G and $h = (h_0, h_1, \dots, h_k)$ for H with balanced parenthesization P.

```
{ test $G = Cube(g, P)$ and $H = Cube(h, P)$}
for all $a, b \in \{1, \dots, n\}$
    for all $(\epsilon_1, \dots, \epsilon_k) \in \{0,1\}^k$
        check whether $a = g_0 g_1^{\epsilon_1} \dots g_k^{\epsilon_k}$ and $b = h_0 h_1^{\epsilon_1} \dots h_k^{\epsilon_k}$
    if $a$ or $b$ was not generated any $\epsilon$ then reject and halt.
{ isomorphism test }
for all $(\epsilon_1, \dots, \epsilon_k) \in \{0,1\}^k$
    for all $(\eta_1, \dots, \eta_k) \in \{0,1\}^k$
        $c \leftarrow g_0 g_1^{\epsilon_1} \dots g_k^{\epsilon_k}, d \leftarrow g_0 g_1^{\eta_1} \dots g_k^{\eta_k}$
        $c' \leftarrow h_0 h_1^{\epsilon_1} \dots h_k^{\epsilon_k}, d' \leftarrow h_0 h_1^{\eta_1} \dots h_k^{\eta_k}$
        for all $(\nu_1, \dots, \nu_k) \in \{0,1\}^k$
            if $cd = g_0 g_1^{\nu_1} \dots g_k^{\nu_k} \leftrightarrow c' \cdot d' \neq h_0 h_1^{\nu_1} \cdot \dots \cdot h_k^{\nu_k}$ then halt and reject.
halt and accept.
```

Since $k \in O(\log n)$, the number of performed ϵ-tests is bounded by a polynomial. Because of the parenthesization P, every multiplication $g = g_1^{\epsilon_1} \dots g_k^{\epsilon_k}$ can be computed by a sub-circuit of depth $O(\log\log n)$ with unbounded fan-in. Each sub-circuit is organized as a pyramid. At the bottom level it uses the multiplication tables to multiply pairs of elements $g_i^{\epsilon_i} g_{i+1}^{\epsilon_{i+1}}$. At the next level it multiplies pairs of results of the previous level, and so on. The depth of the sub-circuits is bounded by $O(\log\log n)$ since $k \in O(\log n)$. □

The upper bound that we get for groups is the same one. For concrete group families it is possible to get better bounds. We include as example the case of Abelian groups.

3.1 On the Complexity of Abelian Group Isomorphism

We consider here the easier case when the input structures are Abelian groups.

Clearly, testing the property whether G is Abelian can be done in AC^0 by simply testing whether $a \cdot b = b \cdot a$ holds for all elements a, b in parallel. The isomorphism test is based on the following well known fact.

Fact 3.5 *Two finite Abelian groups G and H with $|G| = |H| = n$ are isomorphic iff the number of elements of order m in G and H is the same, for all $1 \leq m \leq n$.*

A proof of this fact can be found for example in [Hal59]. The order of an element a is the smallest integer $i \geq 0$ such that $a^i = e$. Hence, an isomorphism test simply computes the orders for all elements using the power predicate. Barrington et.al. [BKLM00] considered the complexity of the power predicate on Abelian groups.

Lemma 3.6 *([BKLM00]) Let G be a finite group given by its multiplication table. For all elements a and b in G and all $i \leq n$, the predicate $b = a^i$ can be computed in* $\mathrm{FOLL} \cap \mathrm{L}$.

In the isomorphism test, an FOLL computes outputs the order of all group elements. This is a set of numbers in arbitrary order.

Given two multisets of numbers, the problem of pairwise comparing them is not in AC^0, since the Majority function reduces to this problem. It is known that the Sorting, i.e. arranging n n-bit numbers in ascending order, is in TC^0. This suffices for an isomorphism test. When given two sorted multisets of numbers, say $e_1 \leq \cdots \leq e_n$ and $e'_1 \leq \cdots \leq e'_n$, it can be tested in AC^0 whether they coincide. We conclude:

Theorem 3.7 *The Abelian Cayley-group isomorphism problem is in* TC^0*(*FOLL*), and in* L.

4 Computing parity by shallow circuits with limited non-determinism

We prove in this section that FOLL circuits (in fact polynomial size circuits of depth $O\big((\log\log n)^k\big)$) cannot compute the Parity function even with the help of polylogarithmic many nondeterministic bits.

Theorem 4.1 *Let C be a circuit of polynomial size and depth $O\big((\log\log n)^k\big)$, with access to $O\big((\log n)^\ell\big)$-many non-deterministic bits, where k, ℓ are arbitrary constant numbers. Then C cannot be computing the parity function.*

Proof:

Let C be computing parity and have depth d. Then for every possible setting of the non deterministic bits C outputs zero for inputs of even parity. On the other hand, by averaging, there exists at least one setting θ of the non-deterministic bits for which C outputs 1 on at least $\frac{2^{n-1}}{2^{(\log n)^\ell}}$ many inputs of odd parity. Thus, the deterministic circuit C_θ obtained from C by fixing its non-deterministic bits to θ approximates parity well, i.e.

$$\Pr_x \left[C_\theta(x) = Parity(x)\right] \geq \frac{1}{2} + \frac{1}{2 \cdot 2^{O((\log n)^\ell)}}.$$

However, C_θ has the same size and depth as C. The proof gets completed by showing below, via Theorem 4.4, that such approximations to parity is impossible.

□

In order to prove the desired inapproximability results, we use a version of the Switching Lemma. Switching Lemmas were developed in a series of works by [FSS81, Ajt83, Yao85, Cai86, Has87] for proving lower bounds on the size of constant-depth circuits computing parity. We recall the following decision-tree version, due to Razborov[Raz93]. Let R_n^m be the space of all restrictions on n variables that leaves precisely m of them free. For any restriction ρ, we denote by f_ρ the boolean function induced from f on variables left free by ρ.

Lemma 4.2 (Switching Lemma, Razborov) *Let f be a CNF (or DNF) formula with clause width t on n variables. Let ρ be a random restriction in R_n^m. Then, there exists a constant $\gamma > 0$ such that the probability of f_ρ not having a decision tree of height at most s is less than $\left(\frac{\gamma m t}{n}\right)^s$.*

An immediate consequence of this lemma is the following corollary:

Corollary 4.3 *Let f be a function computed by a circuit of size S and depth d. Let $m = n/\left((2\gamma)^d (n^{1/(2d)})^{d-1}\right)$. Then,*

$$\Pr_{\rho \in R_n^m} \left[h(f_\rho) \ > \ n^{1/2d}\right] \leq S \cdot \frac{1}{2^{\Omega\left(n^{1/2d}\right)}},$$

where, $h(f_\rho)$ denotes the height of the best decision tree for f_ρ.

Proof:

This can be shown by a simple inductive argument using the Switching Lemma. Assume, as our inductive hypothesis, the following: let $i \geq 2$ and $n_i = n/\left((2\gamma)^i (n^{1/(2d)})^{i-1}\right)$. Let G_i be the set of gates in the ith layer of C and let S_i be the number. Further, let $S_{\leq i} = \sum_{j=1}^{i} S_j$. Our inductive hypothesis is the following:

$$\Pr_{\rho \in R_n^{n_i}} \left[\exists g \in G_i \ : \ h(f_\rho^g) \ > \ n^{1/2d}\right] \leq S_{\leq i} \cdot \frac{1}{2^{n^{1/2d}}},$$

where f^g is the function computed at gate g. Now, if the ith layer of the circuit has AND (OR) gates then one can assume w.l.o.g that $i+1$th layer has OR (AND) gates. In this case, assuming that each f^g_ρ has a decision tree of height at most $n^{1/2d}$, we represent f^g_ρ as a DNF of width at most $n^{1/2d}$ by using the small height decision tree. This collapses layers i and $i+1$ and hence the output of every gate at layer $i+1$ is a DNF of width $n^{1/2d}$ under the restriction ρ. We apply the Switching Lemma to each such DNF where $n = n_i$, $m = n_{i+1}$ and $t = n^{1/2d}$. Clearly, the probability that any fixed such DNF under the next round of restriction fails to have a decision tree of height at most $n^{1/2d}$ is at most $2^{-n^{1/2d}}$. Applying the union bound to S_{i+1} such DNF's (one for each gate at layer $i+1$) immediately completes the induction.

□

Applying the above, we get the following inapproximability result (which is possibly implicit in work of Cai[Cai86]):

Theorem 4.4 *Let C be any polynomial size circuit of depth d. Then,*

$$\Pr_x\big[C(x) = Parity(x)\big] \leq \frac{1}{2} + \frac{1}{2^{\Omega\left(n^{1/2d}\right)}}.$$

Proof:

Applying Corollary 4.3, we see that if we pick a random restriction that leaves m variables free, where $m = n/\big((2\gamma)^d(n^{1/(2d)})^{d-1}\big)$ with probability at least $1 - \mathrm{Size}(C) \cdot 2^{-n^{1/2d}}$, the circuit will have a decision tree of height at most $n^{1/2d}$. Hence, with that much probability the number of free variables m is more than the height of the decision tree. For each such restriction, the restricted circuit computes the right answer (which is either Parity or its complement, on the m free variables) with probability exactly a half. Hence, even assuming that for all other restrictions we get perfect correlation,

$$\Pr_x\big[C\big(x\big) = Parity(x)\big] \leq \frac{1}{2} + \Pr_{\rho \in R_n^m}\Big[h\big(C_\rho\big) > n^{1/2d}\Big] \leq \frac{1}{2} + \mathrm{Size}\big(C\big) \cdot \frac{1}{2^{n^{1/2d}}}.$$

The proof is completed by observing that the size of the circuit, denoted by $\mathrm{Size}\big(C\big)$, by assumption is polynomial.

□

5 Discussion

Although no polynomial time algorithms for GroupIso or QGroupIso are known, we have shown in this paper that the problems are not hard enough to encode the Parity function. Therefore these problems cannot be hard under AC^0 reductions for any complexity class containig Parity, like L or NC^1. This contrasts sharply with the hardness properties of other isomorphism problems like Graph Isomorphism. In fact, our research started originally trying to prove that QGroupIso is hard for NC^1. At first sight it looks as if the difficulty in encoding the Parity function comes from the very structured way in which the input information is presented in the the multiplication tables. The way of proving the result, however was to show that the computation of QGroupIso can be divided in two faces, a first bounded nondeterministic part and a very efficient checking part. We then gave an upper bound for the checking part in terms of circuits with very restricted depth and showed that these circuits cannot compute Parity even with the help of poly-log many nondeterministic bits. We observe that this proof technique does not have anything to do with isomorphism problems and can be applied to other problems whose computation have similar bounded guessing and checking parts. For example the classes $LOGNP_0$ and $LOGSNP_0$ from [PY96] would fall in $\beta_2 AC^0$ in our setting. The results in this paper imply that the problems in these classes cannot be AC^0 hard for Parity. Observe that for example the problem LOGCLIQUE, deciding if a given graph with n vertices has a clique of size at least $\log n$ falls into this category. We find this surprising. It would be interesting to study, maybe with other techniques, the existence of longer hierarchies of natural problems defining different AC^0 degrees.

So far all the upper bounds known for GroupIso hold also for QGroupIso . The question of whether the problems are equivalent under some reduction remains open.

References

[Ajt83] Miklós Ajtai. Σ_1^1-formulae on finite structures. In *Annals of Pure and Applied Logic*, 24:1–48, 1983.

[AT06] V. Arvind and Jacobo Torán. The complexity of quasigroup isomorphism and the minimum generating set problem. In Tetsuo Asano, editor, *International Symposium on Algorithms and Computation (ISAAC)*, volume 4288 of *Lecture Notes in Computer Science*, pages 233–242. Springer, 2006.

[BKLM00] David Mix Barrington, Peter Kadau, Klaus-Jörn Lange, and Pierre McKenzie. On the complexity of some problems on groups input as multiplication tables. In *Proceedings of the 15th Annual IEEE Conference on Computational Complexity (COCO)*, page 62, Washington, DC, USA, 2000. IEEE Computer Society.

[Cai86] Jin-yi Cai. With probability one, a random oracle separates PSPACE from the polynomial-time hierarchy. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing (STOC)* 38(1), 21–29, 1986.

[DT90] Josep Díaz and Jacobo Torán. Classes of Bounded Nondeterminism. *Mathematical Systems Theory* 23(1): 21–32, 1990.

[FSS81] Merrick L. Furst, James B. Saxe and Michael Sipser. Parity, circuits and the polynomial-time hierarchy. In *22nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 260–270, 1981.

[ER65] Paul Erdős and Alfred Rényi. Probabilistic methods in group theory. *Journal d'Analyse Mathématique*, 14:127–138, 1965.

[GLM96] Judy Goldsmith, Matthew A. Levy and Martin Mundhenk. Limited nondeterminism. *SIGACT News* 27(2): 20–29, 1996.

[Hal59] Marshall Hall. *The theory of groups.* Macmillan, New York, 1959.

[Has87] John Håstad *Computational limitations of small-depth circuits.* MIT Press, 1987.

[KF84] Chandra Kintala and Patrick Fisher. Refining nondeterminism in relativized complexity classes. *SIAM Journal on Computing* 13:329–337, 1984.

[LSZ76] Richard J. Lipton, Lawrence Snyder, and Yechezkel Zalcstein. The complexity of word and isomorphism problems for finite groups. Technical report, John Hopkins, 1976.

[Mil78] Gary L. Miller. On the $n^{log n}$ isomorphism technique. In *ACM Symposium on Theory of Computing (STOC)*, 1978.

[PY96] Christos H. Papadimitriou and Mihalis Yannakakis. On limited nondeterminism and the complexity of the VC dimension. *Journal of Computer and System Sciences*, 53:161–170, 1996.

[Raz93] Alexander A. Razborov. An equivalence between second order bounded domain bounded arithmetic and first order bounded arithmetic. In P. Clote and J. Krajíček, editors, *Arithmetic, Proof Theory and Computational Complexity*, Oxford University Press (1993), 247–277.

[Tor04] Jacobo Torán. On the hardness of Graph Isomorphism. *SIAM Journal on Computing* 33(5): 1093–1108, 2004.

[Tor10] Jacobo Torán. Reductions to Graph Isomorphism. *Theory of Computing Systems* 47(1): 288–299, 2010.

[Wag10] Fabian Wagner. On the complexity of isomorphism testing for restricted classes of graphs. Ph.D. Thesis. Technical Report VTS-ID/7264, Institutional Repository of University of Ulm, 2010.

[Wol94] Marty J. Wolf. Nondeterministic circuits, space complexity and quasigroups. *Theoretical Computer Science (TCS)*, 125:295–313, 1994.

[Yao85] Andrew C.C. Yao. Separating the polynomial hierarchy by oracles: Part I. In *26th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1–10, 1985.