# Boolean complexity classes vs. their arithmetic analogs

Anna Gál* Avi Wigderson†

**Abstract**

This paper provides logspace and small circuit depth analogs of the result of Valiant and Vazirani, which is a randomized (or nonuniform) reduction from $NP$ to its arithmetic analog $\oplus P$. We show a similar randomized reduction between the Boolean classes $NL$ and semi-unbounded fan-in Boolean circuits and their arithmetic counterparts. These reductions are based on the Isolation Lemma of Mulmuley, Vazirani and Vazirani. Combinatorially our results can be viewed as simple (logspace) transformations of existential quantifiers into counting quantifiers in graphs and shallow circuits.

## 1 Introduction

Valiant and Vazirani [VV] gave a randomized reduction from $NP$ to $\oplus P$, using universal hash functions. $\oplus P$ is the counting version of $NP$, also defined by polynomial time nondeterministic Turing machnines. A language $F$ belongs to $\oplus P$ if there is a polynomial time nondeterministic Turing machine $M$ such that $F$ consists of exactly those inputs for which the number of accepting computation paths of $M$ is odd. ¿From the reduction of Valiant and Vazirani it follows that $NP/poly \subseteq \oplus P/poly$. (The notation $C/poly$ stands for the nonuniform version of the complexity class $C$.) This was the first result that related a nondeterministic and a counting complexity class.

*School of Math., Institute for Advanced Study, Princeton, NJ 08540, USA. Part of this work was done while visiting the Hebrew University in Jerusalem, and while at the Dept. of Computer Science at the University of Chicago. Research supported in part by NSF Grant DMS-9304580.

†Dept. of Computer Science, The Hebrew University, Jerusalem, 91906 ISRAEL. This work was partially supported by the Sloan Foundation, USA-Israel BSF grant 92-00106 and by a Wolfson research award administered by the Israeli Academy of Sciences.

We are interested in the relationship of the corresponding complexity classes $NL$ and $\oplus L$ defined by nondeterministic Turing machines bounded by logarithmic space rather than polynomial time. It was an open problem if a similar result holds in the logspace world, i.e does $NL/poly \subseteq \oplus L/poly$ ? This question may be best illustrated by comparing the graph reachability problems that are complete for these classes. A complete problem for $NL$ is asking whether or not a directed graph contains a path between two given vertices, and a complete problem for $\oplus L$ is whether or not the number of paths between two given vertices is odd or even. Asking what is the relationship between $NL$ and $\oplus L$ is related to asking which one of these two problems is easier, in the sense that whether we can reduce solving one of them to solving the other. Note that it is crucial in this case that we are allowed to use only logarithmic space, otherwise the reduction from deciding the existence of paths to counting the number of paths would be easy.

The method of Valiant and Vazirani does not seem to help in answering this question. One difficulty in applying the Valiant-Vazirani reduction, is that while the computations of hash functions could be easily embedded in CNF formulae (they used SAT as the $NP$-complete language for their reduction), it is not clear how to embed them in graph reachability or other $NL$-complete languages.

A similar problem exists for the circuit analog of these logspace classes, namely does $SAC^1 \subseteq \oplus SAC^1$ ? Recall that $SAC^1$, the circuit analog of $NL$, is the class of Boolean circuits of depth $O(\log n)$ over $\{\vee, \wedge, \neg\}$ with unbounded fan-in $\vee$ and bounded fan-in $\wedge$ gates. Negations are allowed only at the input level. Clearly $NL/poly \subseteq SAC^1$. The uniform version of $SAC^1$ is the same as the class $LOGCFL$ of languages logspace reducible to context-free languages [S], [V1]. Properties and characterizations of $LOGCFL$ are studied in [C], [SV]. Semi-unbounded fan-in circuits of larger depths correspond to extensions of context-free languages [C, I, Ru, V1]. An interesting property of classes defined by semi-unbounded fan-in circuits, proved by Borodin et al. [BCDRT], is that they are closed under complementation for all depths that are $\Omega(\log n)$. We consider the arithmetic analogs of the above complexity classes, defined by semi-unbounded fan-in arithmetic circuits. These classes have been studied for example in [AJ]. The class $\oplus SAC^1$, the circuit analog of $\oplus L$, is the class of arithmetic circuits of depth $O(\log n)$ over $\{\oplus, \times\}$, with unbounded fan-in $\oplus$ and bounded fan-in $\times$ gates. Clearly $\oplus L/poly \subseteq \oplus SAC^1$. (It is worth while to point out that for all bounded or all unbounded fan-in circuits the analogous inclusion of the Boolean in the arithmetic class is trivial.)

In this paper we prove both inclusions, namely $NL/poly \subseteq \oplus L/poly$ and $SAC^1 \subseteq \oplus SAC^1$. We observe that the "right" technique to use is applying the Isolation Lemma of Mulmuley-Vazirani-Vazirani [MVV]. We note that [MVV] showed how the Isolation Lemma can be used to rederive the Valiant-Vazirani result for the clique function. The Isolation Lemma can be embedded in graph reachability, generating from an arbitrary graph which has an *st*-path another graph in which such a path is unique. From this the $NL/poly \subseteq \oplus L/poly$ easily follows. Similarly, we use an extension of the Isolation Lemma to generate from an arbitrary $SAC^1$ circuit which outputs 1, another circuit with the same output, which has a unique "certificate subcircuit". Again, the simulation by arithmetic circuits is then easy. We obtain analogous results for the corresponding classes defined by semi-unbounded fan-in circuits of larger depths as well.

Both results extend to any finite field, not just $GF(2)$. Moreover, both results improve some earlier circuit complexity results.

Our result on $NL$ has an interesting application to the relationship between arithmetic and Boolean circuit depth for (arithmetic and Boolean, respectively) matrix powering. Borodin [Bo] proved that, given a depth $d(n)$ arithmetic circuit for computing the $n$th power of an $n \times n$ 0/1 matrix, it can be converted to a depth $d(n) \log d(n)$ Boolean circuit for the transitive closure of that matrix. Moreover, the size blows up only polynomially. The conversion uses arithmetic over small primes and the Chinese Remainder Theorem. The loss in depth comes from a Boolean simulation of the arithmetic operations over the small fields, and it was not clear if this loss is necessary. Our reduction to graphs with unique paths immediately implies that any depth $d(n)$ arithmetic circuit family as above yields a depth $O(d(n))$ Boolean circuit family for transitive closure. Again, this entails no more than polynomial blow up of size.

Our circuit result improves an earlier result of Borodin [Bo2] about the general relation of Boolean and arithmetic circuits. Razborov [Ra] showed that an $m$-input $\vee$ can be well approximated by degree $\log m$ polynomials over finite fields. Using these polynomials and amplifying the approximation it is possible to get $O(d \log\log n + \log n)$ depth and polynomial size semi-unbounded arithmetic circuits that simulate semi-unbounded (and even completely unbounded fan-in) depth $d$ Boolean circuits [Bo2]. However we cannot hope for further improvement if we try to simulate each $\vee$ gate separately. Our "global" reduction gives depth $O(d + \log n)$ simulation in the semi-unbounded fan-in case.

All our simulations (as well as Valiant-Vazirani's) are non-uniform (or at least randomized). A very interesting question that remains open is whether one can remove the nonuniformity from any of these results. The only result in this direction we are aware of is the observation in [KW] that $SL \subseteq \oplus L$, where $SL$ stands for symmetric logspace (the class of languages logspace reducible to undirected s-t connectivity).

# 2 Preliminaries and Results

## 2.1 The Isolation Lemma

Let $E$ be a finite set, and let $w : E \to \mathcal{R}$ be an arbitrary (weight) function. Extend $w$ to subsets of $E$ by $w(S) = \sum_{e \in S} w(e)$. Let $\mathcal{F}$ be a family of subsets of $E$. For a fixed weight function $w$, denote by $min(\mathcal{F}, w)$ the weight of the lightest set in $\mathcal{F}$ under $w$, and $MIN(\mathcal{F}, w)$ those subsets in $\mathcal{F}$ whose weight is minimum. Clearly $|MIN(\mathcal{F}, w)| \geq 1$ for every $\mathcal{F}$ and $w$. Now the Isolation Lemma of Mulmuley, Vazirani and Vazirani states that choosing $w$ at random appropriately, there will be a unique minimum weight subset in $\mathcal{F}$ with high probability.

**Theorem 1** *[MVV] Fix an integer $k$. Pick $\mathbf{w}$ at random as follows: for every $e \in E$, $\mathbf{w}(e)$ is chosen uniformly from the integers in $[1, k|E|]$, independently of all other elements in $E$. Then for every $\mathcal{F}$ we have*

$$Pr[|MIN(\mathcal{F}, \mathbf{w})| > 1] \leq 1/k$$

In this paper we need a version of the Isolation Lemma that holds for multisets as well, i.e. for sets possibly containing some elements with multiplicities. The *support* of a multiset $F$ is the set of elements occuring in $F$.

For multisets the Isolation Lemma does not hold in its original form. However the proof of the Isolation Lemma in [MVV] yields the following.

**Theorem 2** *[MVV] Let $\mathcal{F}$ be a family of multisets of the elements of $E$. Let us assign integer weights to the elements of $E$ uniformly and independently from $[1, k|E|]$. Then with probability $\geq 1 - 1/k$, all minimum weight sets in $\mathcal{F}$ have the same support.*

We note that Nisan [N] proved that the Isolation Lemma holds for multisets as well if we allow larger weights depending on the maximum multiplicity.

## 2.2 Graphs, Languages and Complexity

All graphs and all paths mentioned in this paper are directed. $G = G(V, E)$ will denote a graph with vertices $V$ and directed edges $E$, and $s, t$ will refer to distinct vertices in $V$.

We will consider three logspace classes (and their nonuniform versions): $NL, \oplus L, UL$. The first two are captured respectively by their complete (under deterministic logspace reductions) graph reachability languages *STCONN*, *ODD-STCONN* defined below.

- $(G, s, t) \in$ *STCONN* iff $G$ contains an $s - t$ path.
- $(G, s, t) \in$ *ODD-STCONN* iff $G$ contains an odd number of $s-t$ paths.

The class *UL* is not known to have such complete problems. Nevertheless for our purposes it is captured by any language (which we generically call *UNIQUE-STCONN*) satisfying the following two properties:

1. $(G, s, t)$ has no $s - t$ path implies $(G, s, t) \notin$ *UNIQUE-STCONN*.
2. $(G, s, t)$ has a unique $s-t$ path implies $(G, s, t) \in$ *UNIQUE-STCONN*.

We shall use two notions of reducibilities, performed respectively by functions computable in *RL* (probabilistic logspace machines) and *L/poly* (nonuniform logspace). Let $T, T'$ be two languages.

- $T \leq_{L/poly} T'$ if there is a machine $M \in L/poly$ such that for all inputs $x$, $x \in T \iff M(x) \in T'$.
- $T \leq_{RL}^{\epsilon} T'$ if there is a machine $M \in RL$ such that:

$$x \notin T \Rightarrow M(x) \notin T'$$

$$x \in T \Rightarrow Pr[M(x) \in T'] \geq \epsilon$$

## 2.3 Arithmetic vs. Boolean circuits

We consider arithmetic circuits with gates from the basis $\{+, -, \times\}$ (we disallow division!) over a field $F$. All constants of the field may be used. Boolean circuits have the standard Boolean basis $\{\wedge, \vee, \neg\}$. For more information on arithmetic circuits we refer to [Gat], for Boolean circuits see [We].

Let $d_F(p)$ (resp. $d(f)$) denote the smallest depth of a polynomial size arithmetic (resp. Boolean) circuit for the polynomial $p$ (resp. Boolean function $f$) using only constant fan-in gates. *Semi-unbounded fan-in* circuits have constant fan-in $\times$ (resp. $\wedge$) gates and unbounded fan-in $+$ (resp. $\vee$) gates. Semi-unbounded fan-in Boolean circuits may have negations only at the input level. We denote by $\hat{d}_F(p)$ (resp. $\hat{d}(f)$) the smallest depth of polynomial size semi-unbounded fan-in arithmetic (resp. Boolean) circuits for $p$ (resp. $f$). We note that in the above definitions formally we consider the infimum of the depth over all fixed polynomial size bounds.

$SAC^k$ denotes the class of languages accepted by polynomial size, depth $O((\log n)^k)$ semi-unbounded fan-in Boolean circuits. $\oplus SAC^k$ denotes the class of polynomials over $GF(2)$ computed by polynomial size, depth $O((\log n)^k)$ semi-unbounded fan-in arithmetic circuits over $GF(2)$.

As usual, we will be interested in circuit families for families of polynomials and functions.

On input an $n$ vertex graph $G$, presented by a 0/1 matrix, we consider the polynomial $\#STCONN$ computing the number of $s-t$ paths in $G$. This is the arithmetic analog of the Boolean function $STCONN$.

## 2.4 Main results

**Theorem 3** *STCONN* $\leq_{RL}^{1/n^3}$ *UNIQUE-STCONN*

**Theorem 4** *STCONN* $\leq_{L/poly}$ *ODD-STCONN*

A direct corollary of the second theorem is the following:

**Corollary 1** $NL/poly \subseteq \oplus L/poly$

In fact, there is clearly nothing special about counting modulo 2. The same result holds for the $Mod_kL$ classes defined in [BDHM].

**Corollary 2** $NL/poly \subseteq Mod_kL/poly$

Another corollary of Theorem 4 shows that the depth of polynomial size Boolean circuits for $STCONN$ is never worse than that of arithmetic circuits for $\#STCONN$.

**Corollary 3** $d(STCONN) = O(d_Q(\#STCONN))$*, where Q denotes the field of rational numbers.*

We extend the techniques used for proving the above theorems to circuits.

**Theorem 5** *For every Boolean function $f$ on $n$ variables and every finite field $F$, $\hat{d}_F(f) = O(\hat{d}(f) + \log n)$*

**Corollary 4** *$LOGCFL/poly \subseteq \oplus SAC^1$*

**Corollary 5** *$SAC^k \subseteq \oplus SAC^k$*

**Theorem 6** *Every Boolean function $f$ on $n$ variables can be approximated, i.e. computed on $\geq (1 - 2^{-k})$ fraction of the inputs, by semi-unbounded fan-in arithmetic circuits over any fixed finite field in polynomial size and depth $O(\hat{d}(f) + \log k)$, for arbitrary $k > 0$.*

# 3 Proofs for logspace classes

## 3.1 Proof of Theorem 3

We first need some simple definitions and lemmas. Let $G(V, E)$ be a directed graph, and $w$ a weight (distance) function on $E$. We let $d_w(a, b)$ denote the length of the shortest path between two nodes $a, b \in V$ in this weighted graph. The following lemma follows immediately from the Isolation Lemma, taking the family of subsets $\mathcal{F}$ to be all $s$–$t$ paths in $G$.

**Lemma 1** *Let $G$ be a graph with an $s$–$t$ path. Let $\mathbf{w}$ be chosen at random with each weight independently taken uniformly from $[1, 2|E|]$. Then*

$$Pr[\exists a \text{ unique } s\text{-}t \text{ path of distance } d_{\mathbf{w}}(s,t)] \geq 1/2$$

**Comment:** Observe that increasing the range of $\mathbf{w}$ to $|V|^2|E|$ would yield a graph in which (with high probability) the shortest distance between *every* pair of nodes is achieved by a unique path. We see no application of this observation.

Now given a graph $G(V, E)$, a weight function $w$ and integer $l$ define the (unweighted, layered) graph $G^l_w(U, F)$ as follows. For every vertex $a \in V$ and every integer $0 \leq i \leq l$ put the vertex $< a, i >$ in $U$ (i.e. $l+1$ copies of $V$, arranged in layers). For every edge $e = (a, b) \in E$ and every $0 \leq i \leq l - w(e)$ put an edge $(< a, i >, < b, i + w(e) >)$ in $F$.

**Lemma 2** *$G^l_w$ can be constructed from input $G, w, l$ in deterministic logspace.*

**Lemma 3** • *If $G$ has no $s$–$t$ path, then for every $w$ and $l$, $G^l_w$ has no $< s, 0 >$–$< t, l >$ path.*

- *If $G$ has an $s$-$t$ path and $l = d_w(s,t)$ then $G^l_w$ has an $< s, 0 >$–$< t, l >$ path. Moreover the later path is unique if the shortest weighted $s$–$t$ path in $G$ is unique.*

We conclude Theorem 3 from the above lemmas as follows: Given $G$ we construct $G^{\mathbf{l}}_{\mathbf{w}}$ with $\mathbf{w}$ picked at random as in Lemma 1, and $\mathbf{l}$ chosen uniformly from $[1, 2|V||E|]$. Note that $Pr[\mathbf{l} = d_{\mathbf{w}}(s,t)] \geq 1/|V|^3$. Also, it is clear that this reduction can be performed in $RL$, if we are allowed to print the edges of the new graph in the "right" order (i.e in groups of edges related to each weight).

**Comment:** We caution that the fact that this reduction can be carried in $RL$ does not mean that we can use it as a subroutine in a logspace computation. If any algorithm needs to query some edge of the output graph of this reduction more than once or in the "wrong" order, we need to remember the random bits used for the weights. This will not hurt us later, as we'll move into nonuniform reductions.

## 3.2 Proof of Theorem 4

The random reductions of the previous subsection, together with a standard counting argument yield the following.

**Lemma 4** *Fix $m = n^{10}$. There exists $m$ pairs $\{(w_j, l_j) | 1 \leq j \leq m\}$ satisfying the following for every graph $G$ on $n$ vertices.*

- *For every $j$, $l_j$ and the weights in $w_j$ are integers below $n^3$.*
- *If $G$ has no $s$–$t$ path, then for every $j$, $G^{l_j}_{w_j}$ has no $(s,0) - (t, l_j)$ path.*
- *If $G$ has an $s$–$t$ path, then there exists $j$ such that $G^{l_j}_{w_j}$ has a unique $(s,0) - (t, l_j)$ path.*

We abbreviate by $(G_j, s_j, t_j)$ the triple $(G^{l_j}_{w_j}, (s,0), (t, l_j))$.

Now assume we get as an advice (of polynomial length) a set of $m$ pairs satisfying Lemma 4. On input $(G, s, t)$, we will construct $(G', s', t')$ such that $G'$ has an odd number of $s$–$t$ paths iff $G$ had an $s$–$t$ path. This uses a standard idea of adding direct $s$–$t$ edges to change the parity of the number of $s$–$t$ paths.

Construct the graphs $(G_j, s_j, t_j)$ for $j \in [m]$. To each of them add the direct edge $(s_j, t_j)$. Identify $t_j$ and $s_{j+1}$ for every $j < m$. Change the name of $s_1$ to $s'$, and of $t_m$ to $t'$. Finally, add the direct edge $(s', t')$ and call the new graph $G'$. It is easy to verify that $(G', s', t')$ satisfies the requirement above, and we are done.

### 3.3 Proof of Corollary 3

First notice that the construction of the gragh $G'$ from the input graph $G$ in the proof of Theorem 4 can be easily carried out by an $NC^1$ circuit, i.e. a Boolean circuit $D$ of polynomial size and $O(\log n)$ depth.

Now assume $d_Q(\#STCONN) = d(n)$ (note that $\log n \leq d(n) \leq (\log n)^2$). Let $C'$ be the arithmetic circuit for $\#STCONN$ for $n^{10}$ vertex graphs. Thus $C'$ has polynomial size and depth $d(n^{10}) = O(d(n))$ (using the upper bound on $d$ and assuming $d$ is a reasonably smooth function). Convert $C'$ into a Boolean circuit $C$ simply by thinking of the gates as performing the arithmetic operations over $GF(2)$, and then simulating them with Boolean gates. Thus $C$ also has polynomial size and depth $O(d(n))$.

The Boolean circuit for $STCONN$ we promised is obtained simply by identifying the outputs of the circuit $D$ with the inputs of the circuit $C$. By Theorem 4 it correctly computes $STCONN$ for $n$ vertex graphs, and by the discussion above it has depth $O(d(n) + \log n) = O(d(n))$ and polynomial size.

## 4 Circuits and certificates

To obtain an analog of Theorem 3 for circuits, we need an analog of isolating unique paths for circuits.

Let us consider an arbitrary Boolean circuit with $\vee$, $\wedge$ gates, provided with the $2n$ input literals $x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n$. Fixing a 0-1 assignment to the variables $x_1, \ldots, x_n$ determines the values computed by each gate of the circuit. We refer to the wires of the circuit as edges. The edges are oriented from the inputs of a gate $g$ to $g$. For a fixed input assignment we label each edge of the circuit with the value computed by the gate at the starting node of the edge. For each gate that outputs 1, there is a set of edges all labelled 1 that forces the given gate to output 1. We call the graphs formed by these edges *certificates*. A certificate depends on the gate it belongs to and on the particular assignment to the input variables. There may be several certificates for the same gate on the same input. Gates that output 0 on a given input do not have certificates on that input.

Let us now give a formal definition of certificates. We denote the set of gates that are inputs to a given gate $g$ by $I(g)$.

**Definition 1** *The circuit $Z$ is called a partial circuit of $C$ if it satisfies the following conditions:*

- *the set of gates of $Z$ is a subset of the set of gates of $C$*
- *the output gate of $Z$ is the output gate of $C$,*
- *for every $\wedge$ gate $g$ of $Z$, $I_Z(g) = I(g)$,*
- *for every $\vee$ gate $g$ of $Z$, $\emptyset \neq I_Z(g) \subseteq I(g)$,*

*where $I_Z(g)$ stands for the set of input gates to $g$ in the circuit $Z$.*

**Definition 2** *A partial circuit $Z$ is minimal, if for every $\vee$ gate $g$ of $Z$, $|I_Z(g)| = 1$.*

Let $\alpha$ be a fixed assignment to the input variables. Let $\epsilon \in \{0,1\}$. We say that $g(\alpha) = \epsilon$ if the gate $g$ outputs the value $\epsilon$ on the input assignment $\alpha$. We say that $C(\alpha) = \epsilon$ if the circuit $C$ outputs the value $\epsilon$ on the input assignment $\alpha$.

**Observation 1** *If $Z$ is a partial circuit of $C$ then given any input assignment $\alpha$, $Z(\alpha) \leq C(\alpha)$.*

**Definition 3** *A certificate for $C(\alpha) = 1$ is a partial circuit $Z$ of $C$ that satisfies the condition that all the gates of $Z$ output 1 on the assignment $\alpha$.*

We note that an equivalent definition is to require that all the literals participating in $Z$ are set to 1 by the assignment $\alpha$.

**Definition 4** *For a given gate $g$ of a circuit $C$ the subcircuit $C_g$ is defined as follows:*

- *the set of gates of $C_g$ is a subset of the set of gates of $C$,*
- *the output gate of $C_g$ is the gate $g$,*
- *for every gate $h$ of $C_g$, $I_{C_g}(h) = I(h)$.*

**Definition 5** *A certificate for $C_g(\alpha) = 1$ is called a certificate for $g(\alpha) = 1$.*

We note that a certificate for $g(\alpha) = 1$ exists if and only if $g(\alpha) = 1$.

**Definition 6** *A certificate is minimal if the corresponding partial circuit is minimal.*

**Observation 2** *If there is a unique certificate for $g(\alpha) = 1$, it has to be a minimal certificate.*

Let $G$ be a partial circuit of the circuit $C$, and suppose that the edges of $C$ have been assigned weights. Let $E$ be the set of edges of $G$. For a given partial circuit $G$ with edges $E$ we define a multiset $\tilde{E}(G)$ with support $E$ as follows. We expand $G$ into a tree $\tilde{G}$ by taking the output of $G$ to be the root and by splitting the nodes of $G$ that have outdegree $\geq 2$ into several copies. We define $\tilde{E}(G)$ to be the multiset of edges of $G$, taking each edge with multiplicity according to $\tilde{G}$. We assign the weight of each edge of $G$ to all of its copies in $\tilde{E}(G)$.

**Definition 7** *We define the weight of a partial circuit $G$ to be the weight of the multiset $\tilde{E}(G)$.*

*We define the weight of a certificate to be the weight of the corresponding partial circuit.*

**Lemma 5** *Let us assign integer weights to the edges of the circuit $C$ uniformly and independently from $[1, 2m]$, where $m$ is the number of edges of $C$. Then for every fixed input assignment $\alpha$ such that $C(\alpha) = 1$, with probability $\geq 1/2$, there is a unique minimum weight certificate for $C(\alpha) = 1$ .*

**Proof** The statement of the lemma follows from Theorem 2, since multisets with the same support belong to the same certificate.

Now we are ready to prove an analog of Theorem 3 for circuits.

**Lemma 6** *Let $C$ be a polynomial size, depth $d = O(\log n)$ circuit with unbounded fan-in $\vee$ gates and bounded fan-in $\wedge$ gates. Let $m$ be the number of edges of $C$, and let $c$ be the maximum fan-in of the $\wedge$ gates. One can construct a polynomial size, depth $\leq 2d$ circuit $C'$ with unbounded fan-in $\vee$ gates and bounded fan-in $\wedge$ gates, such that*

1. *if $C(\alpha) = 0$ then $C'(\alpha) = 0$*

2. *if $C(\alpha) = 1$ then with probability $\geq 1/c^d 4m$, there is a unique certificate for $C'(\alpha) = 1$.*

**Proof** For simplicity, we present the construction for circuits with $\wedge$ gates of fan-in 2. It can be generalized easily to any fan-in $c$. The size remains polynomial if $c$ is constant.

We assume that both $C$ and $C'$ are provided with the values of the $2n$ input literals $x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n$.

Let $C$ be a Boolean circuit with $\vee$ gates of unbounded fan-in and $\wedge$ gates of fan-in 2, that has polynomial size and depth $d = O(\log n)$. Let $m$ be the number of edges in the circuit. We assign a weight to each edge of the circuit by chosing a random integer uniformly and independently from $[1, 2m]$.

Next we choose a random integer $L$ uniformly from $[1, 2^d 2m]$. ($2^d 2m$ is the maximum possible weight of a minimal certificate.)

Let us denote by $\Gamma(C)$ the set of gates of the circuit $C$.

$\Gamma(C')$ will consist of two disjoint classes of gates: principal gates and auxiliary gates. We denote the set of principal gates by $\Phi$ and the set of auxiliary gates by $\Phi_{aux}$. Thus $\Gamma(C') = \Phi \dot{\cup} \Phi_{aux}$. All principal gates (except the ones for the literals) will be $\vee$ gates and all auxiliary gates will be $\wedge$ gates.

$\Phi$ will be a subset of $\Gamma(C) \times \{1, \ldots, L\}$. For a fixed $g \in \Gamma(C)$ the set $H(g) \subseteq \{g\} \times \{1, \ldots, L\}$ will denote the set of all principal gates of the form $(g, i)$, and $\Phi = \dot{\bigcup}_{g \in \Gamma(C)} H(g)$.

If $g$ is an $\wedge$ gate of $C$ then the gate $(g, i) \in H(g)$ will be associated with a set $A(g, i) \subseteq \Phi_{aux}$ of auxiliary gates where $0 \leq |A(g, i)| \leq L^2$.

We refer to the gates in the sets $H(g)$ and $A(g, i)$ as copies of $g$.

We construct the circuit $C'$ inductively. We say that a gate $g \in \Gamma(C)$ has been *processed* if we have created all its copies in $C'$.

We start by processing the literals. For each literal $x_i^\epsilon$ we create a gate labelled $(x_i^\epsilon, 0)$. This gate will output the value of the corresponding literal $x_i^\epsilon$. Each literal will have only one copy.

Let $g \in \Gamma(C)$ be an $\vee$ gate that has all its input gates processed. Let $w_h$ denote the weight of the edge from $h \in I(g)$ to $g$. For each gate $h \in I(g)$ we consider the set $H(h)$. For each $(h, i) \in H(h)$ we create an $\vee$ gate $(g, i + w_h)$ if $i + w_h \leq L$ and if $(g, i + w_h)$ has not been created yet. The input gates to the $\vee$ gate $(g, j)$ are the gates $(h, i)$ with $h \in I(g)$ satisfying $i + w_h = j$.

Let $g \in \Gamma(C)$ be an $\wedge$ gate with input gates $h_1$, $h_2$ such that $h_1$ and $h_2$ have been processed. Let $w_1$ and $w_2$ denote the weights of the corresponding edges from $h_1$ and $h_2$, resp., to $g$. For each pair of gates $(h_1, i) \in H(h_1)$,

$(h_2, j) \in H(h_2)$ such that $i + j + w_1 + w_2 \leq L$ we create an $\wedge$ gate in $A(g, i+j+w_1+w_2)$ with input gates $(h_1, i)$, $(h_2, j)$. For each $k \in \{1, \ldots, L\}$ such that $A(g, k) \neq \emptyset$ we create an $\vee$ gate $(g, k)$ with input set $I((g, k)) = A(g, k)$.

Let $g_{out}$ be the output gate of the circuit $C$. The output gate of the circuit $C'$ will be the gate $(g_{out}, L)$ if it exists. If we did not create such a gate, we make the output of $C'$ to be constant 0.

The circuit $C'$ we constructed has the following properties.

**Observation 3** *Let $g$ be any gate of $C$. Then $(g, w) \in \Gamma(C')$ if and only if the circuit $C_g$ has a minimal partial circuit with weight $w$.*

**Observation 4** *Let $\alpha$ be a fixed assignment to the input variables and $(g, w) \in \Gamma(C')$. Then $(g, w)(\alpha) = 1$ if and only if there is a minimal certificate for $g(\alpha) = 1$ with weight $w$.*

We note that these properties hold only if we define the weight of partial circuits and certificates as in Definition 7, and that these properties are crucial for proving Lemma 6.

Next we show that the circuit $C'$ constructed this way satisfies the requirements of Lemma 6.

The depth of $C'$ is $\leq 2d$ and the size of $C'$ is $\leq 2L^3 S = n^{O(1)}$, where $S$ is the size of $C$.

For any gate $g$ of $C$ and all gates $(g, i)$ of $C'$ we have $(g, i)(\alpha) \leq g(\alpha)$ on any assignment $\alpha$. Thus if $C(\alpha) = 0$ then $C'(\alpha) = 0$.

To prove that the second requirement of Lemma 6 is satisfied, we need the following lemma.

Let $\alpha$ be a fixed input assignment such that $C(\alpha) = 1$. Let $\mathcal{F}_\alpha \neq \emptyset$ be the family of all certificates for $C(\alpha) = 1$. Let $W$ denote a fixed assignment of weights to the edges of $C$. Let $\rho(W, \alpha)$ denote the weight of the minimum weight certificate in $\mathcal{F}_\alpha$.

**Lemma 7** *Suppose there is a unique minimum weight certificate in $\mathcal{F}_\alpha$ and suppose that $L = \rho(W, \alpha)$ . Then there is a unique certificate for $C'(\alpha) = 1$.*

**Proof** From the conditions of the lemma it follows that there is a minimal certificate of weight $L$ for $C(\alpha) = 1$. By Observations 3 and 4 this means that in the circuit $C'$ there is a gate $(g_{out}, L)$ (where $g_{out}$ is the output gate of $C$), and that $C'(\alpha) = (g_{out}, L)(\alpha) = 1$.

Thus, there is a certificate for $C'(\alpha) = 1$. If it is not unique, then there is more than one minimal certificate for $C(\alpha) = 1$ with weight $L$, and we get a contradiction. This proves Lemma 7.

The probability that there is a unique minimum weight cerificate in $\mathcal{F}_\alpha$ and $L = \rho(W, \alpha)$ is at least $1/2 \cdot 1/(2^d 2m)$ (by Lemma 5 and by choosing $L$ uniformly from $[1, 2^d 2m]$). Thus, we proved that if $C(\alpha) = 1$ then with probability $\geq 1/2^d 4m$ there is a unique certificate for $C'(\alpha) = 1$. This concludes the proof of Lemma 6.

# 5 Proof of Theorem 5 and Theorem 6

Let us first consider the case of depth $d = O(\log n)$ circuits. By a standard probabilistic argument it follows from Lemma 6 that there exist $T = c^d 8nm = n^{O(1)}$ circuits $C^1, \ldots, C^T$ such that the following is true for every input assignment $\alpha$

- if $C(\alpha) = 0$ then $\forall i \in \{1, \ldots, T\}$, $C^i(\alpha) = 0$,
- if $C(\alpha) = 1$ then $\exists j \in \{1, \ldots, T\}$ such that there is a unique certificate for $C^j(\alpha) = 1$.

Now we are ready to construct the simulating arithmetic circuit. In each circuit $C^i$ we replace each $\vee$ gate by a $+$ gate and each $\wedge$ gate by a $\times$ gate. We denote the new circuit by $\diamondsuit C^i$. We let the circuits $\diamondsuit C^1, \ldots, \diamondsuit C^T$ compute in parallel over a common input $x_1, \ldots, x_n, 1 - x_1, \ldots, 1 - x_n$. To their outputs we apply a transformation that turns any value that is different from 1 into 0, and keeps the values that are equal to 1 unchanged. Over a given finite field this takes a constant number of gates for each $\diamondsuit C^i$. To compute the $\vee$ of these values, recall that the $\vee$ of $T$ variables can be represented by a polynomial of degree $T$ over the given field. We compute this polynomial by a $\log T$ depth semi-unbounded fan-in circuit with $+$ and $\times$ gates. Let us denote the circuit obtained this way by $\diamondsuit C$.

The circuits $\diamondsuit C^1, \ldots, \diamondsuit C^T$ have the following property:

- if $C(\alpha) = 0$ then $\forall i \in \{1, \ldots, T\}$, $\diamondsuit C^i(\alpha) = 0$ ,
- if $C(\alpha) = 1$ then $\exists j \in \{1, \ldots, T\}$ such that $\diamondsuit C^j(\alpha) = 1$.

This follows from the corresponding properties of the circuits $C^1, \ldots, C^T$ and from the fact that if there is a unique certificate for $C^j(\alpha) = 1$ then $\diamondsuit C^j(\alpha) = 1$.

We conclude that on any input assignment $\alpha$, $C(\alpha) = \Diamond C(\alpha)$. This proves Theorem 5 if $d = O(\log n)$.

For larger $d$ we divide the circuit $C$ into $r = \log n$ depth parts and perform the above simulation on each part. The total depth of the simulating circuit will be $\leq (d/r) \cdot (2r + O(\log n)) = O(d + \log n)$, which concludes the proof of Theorem 5.

For proving Theorem 6 we construct the circuits $\Diamond C^1, \ldots, \Diamond C^T$ and transform their outputs to Boolean values as above. To achieve the $O(d + \log k)$ depth for approximate simulations over a fixed finite field we use polynomials of degree $k$ that approximate the $\vee$ of these values. By [Ra] (cf. [Sm], Lemma 1) given any probability distribution on the $2^T$ inputs there exist polynomials of degree $k$ that compute the $\vee$ of $T$ variables with probability $\geq (1-2^{-k})$ over the input distribution. This concludes the proof of Theorem 6.


## Acknowledgements

Venkateswaran independently suggested a similar construction for the circuit result [V2].


## References


[AJ] E. Allender and J. Jiao, "Depth reduction for noncommutative arithmetic circuits," In *Proc. of the 25th STOC*, (1993), pp. 515-522.

[Bo] A. Borodin, "On relating time and space to size and depth," *SIAM J. Comput.*, 6, (1977), pp. 733-744.

[Bo2] A. Borodin, Personal communication.

[BCDRT] A. Borodin, S. A. Cook, P. W. Dymond, W. L. Ruzzo, M. Tompa, "Two applications of inductive counting for complementation problems," *SIAM J. Comput.*, Vol. 18, No. 3, (1989), pp. 559-578.

[BDHM] G. Buntrock, C. Damm, H. Hertrampf, and C. Meinel, "Structure and importance of the logspace-mod class," *Math. Systems Theory*, 25, (1992), pp. 223-237.

[C] S. A. Cook, "A taxonomy of problems with fast parallel algorithms," *Inform. and Control*, 64 (1985), pp. 2-22.

[G] A. Gál, "Semi-unbounded fan-in circuits: Boolean vs. arithmetic," In *Proceedings of the 10th Annual Symposium on Structure in Complexity Theory*, (1995), pp. 82-87.

[Gat] J. von zur Gathen, "Algebraic complexity theory," *Ann. Rev. Comput. Sci*, 3 (1988), pp. 317-347.

[I] N. Immerman, "Upper and lower bounds on first order expressibility," *J. Comput. System Sci.*, 25 (1982), pp. 76-98.

[KW] M. Karchmer and A. Wigderson, "On span programs," In *Proceedings of the 8th Annual Symposium on Structure in Complexity Theory*, (1993), pp. 102-111.

[MVV] K. Mulmuley, U. Vazirani and V. Vazirani, "Matching is as easy as matrix inversion," In *Proc. of the 19th STOC*, (1987), pp. 345-354.

[N] N. Nisan, Personal communication.

[Ra] A. A. Razborov, "Lower bounds for the size of circuits of bounded depth with basis $\{\wedge, \oplus\}$," *Math. notes of the Academy of Sciences of the USSR*, 41(4), (1987), pp. 333-338.

[Ru] W. L. Ruzzo, "Tree-size bounded alternation," *J. Comput. System Sci.*, 21, (1980), pp. 218-235.

[Sm] R. Smolensky, "Algebraic methods in the theory of lower bounds for Boolean circuit complexity," In *Proc. of the 19th STOC*, (1987), pp. 77-82.

[SV] S. Skyum and L. G. Valiant, "A complexity theory based on Boolean algebra," In *Proc. of the 22nd FOCS*, (1981), pp. 244-253.

[S] I. H. Sudborough, "On the tape complexity of deterministic context-free languages," *J. Assoc. Comput. Mach.*, 25 (1978), pp. 405-414.

[V1] H. Venkateswaran, "Properties that characterize LOGCFL," In *Proc. of the 19th STOC*, (1987), pp. 141-150.

[V2] H. Venkateswaran, Personal communication.

[VV] L. G. Valiant and V.V. Vazirani, "NP is as easy as detecting unique solutions," *Theoretical Computer Science*, 47 (1986), pp. 85-93.

[We] I. Wegener, "The complexity of Boolean functions," Wiley-Teubner 1987.

[W] A. Wigderson, "$NL/poly \subseteq \oplus L/poly$," *Proc. of 9th Conf. Structure in Complexity Theory*, (1994), pp. 59-62.