Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 116 (2012)

A Derandomized Switching Lemma and an Improved Derandomization of AC0

Luca Trevisan* TongKe Xue†

September 12, 2012

Abstract

We describe a new pseudorandom generator for AC0. Our generator ε-fools circuits of depth d and size M and uses a seed of length $\tilde{O}(\log^{d+4} M/\varepsilon)$. The previous best construction for $d \geq 3$ was due to Nisan, and had seed length $O(\log^{2d+6} M/\varepsilon)$. A seed length of $O(\log^{2d+\Omega(1)} M)$ is best possible given Nisan-type generators and the current state of circuit lower bounds; Seed length $\Omega(\log^d M/\varepsilon)$ is a barrier for any pseudorandom generator construction given the current state of circuit lower bounds. For $d = 2$, a pseudorandom generator of seed length $\tilde{O}(\log^2 M/\varepsilon)$ was known.

Our generator is based on a "pseudorandom restriction" generator which outputs restrictions that satisfy the conclusions of the Håstad Switching Lemma and that uses a seed of polylogarithmic length.

1 Introduction

Bounded-depth circuits are one of the few general models of computation for which unconditional constructions of pseudorandom generators have been known with sub-linear seed length, beginning with the work of Ajtai and Wigderson [AW85].

Although logarithmic seed length and polynomial time derandomization are open problems even for depth-2 circuits, the Nisan generator [Nis91] provides a quasi-polynomial derandomization of AC0. Nisan's generator, when instantiated to be pseudorandom with accuracy ε against circuits of depth d and size M has seed length $O(\log^{2d+6} M/\varepsilon)$. In the depth-2 case, Bazzi's Theorem [Baz07] provides a generator of seed length $O(\log^2 M/\varepsilon \cdot \log n)$; by working with small-bias distributions instead of bounded-independence distributions, and by adapting Razborov's proof of Bazzi's Theorem [Raz09], De et al. [DETT10] devise a generator of seed length $\tilde{O}(\log^2 M/\varepsilon)$. Braverman [Bra09] extended Bazzi's Theorem to any depth, showing that bounded-independence distributions are pseudorandom for AC0, but, unlike the $d = 2$ case, Braverman's result does not improve the seed length of Nisan's generator.

We devise a pseudorandom generator of seed length $\tilde{O}(\log^{d+4} M/\varepsilon)$.

*trevisan@stanford.edu. Computer Science Department, Stanford University. This material is based upon work supported by the National Science Foundation under grant No. CCF 1161812 and by the US-Israel Binational Science Foundation under grant no. 2010451.

†tkxue@tkxue.org. Computer Science Department, Stanford University. This material is based upon work supported by the National Science Foundation under grant No. CCF 1161812.

ISSN 1433-8092

Our Proof

Our result follows by the construction of a *pseudorandom restriction* generator that uses a seed of length $\tilde{O}(\log^4 M/\varepsilon)$ to assign values to a $p = 1/O(\log^{d-1} M/\varepsilon)$ fraction of the bits of the n-bit string that we want to generate. The generator is such that, for every circuit C of size M and depth d, fixing a subset of the inputs of the circuit according to the pseudorandom restriction changes the acceptance probability of the circuit, on average, by at most ε. By picking $O(p^{-1} \log n/\varepsilon)$ independent pseudorandom restrictions, we will (with probability at least $1-\varepsilon$) have an assignment to all the variables; such an assignment is generated using $\tilde{O}(\log^{d+4} M/\varepsilon)$ random bits and $\varepsilon \cdot \log^{d-2} M/\varepsilon$-fools the circuit.

Håstad proved that a truly random restriction in which one fixes a $1-p$ fraction of variables, where $p = 1/O(\log^{d-1} M/\varepsilon)$, has a high probability of turning the circuit into a decision tree of depth $\log M$. We prove that the same is true with a restriction that can be generated using a seed of length $\tilde{O}(\log^4 M/\varepsilon)$. Now, it seems that we could get a pseudorandom generator of seed length $\tilde{O}(\log^4 M/\varepsilon)$ simply by fixing a $1-p$ fraction of variables according to our pseudorandom restriction and then the remaining variables according to a small-bias distribution, which is known to fool small decision trees. Unfortunately, our pseudorandom restriction does simplify the circuit, as a truly random restriction does, but we are not able to show that the restriction also preserves the acceptance probability of the circuit; in fact, some instantiations of our pseudorandom restriction generator can be shown to severely distort the acceptance probability of some circuits.

We can, however, make the following observations. First, the conclusion of our derandomized Switching Lemma holds also if we select the variables to restrict according to our generator, and then assign truly random variables to them. This means that the following distribution fools the circuit: (1) select a set S of approximately $(1-p) \cdot n$ variables using our pseudorandom restriction generator; (2) assign values to the variables in S randomly; (3) assign values to the variables not in S according to a small-bias generator. This also means that if we (1) select a set S of approximately $(1-p) \cdot n$ variables using our pseudorandom restriction generator; (2) assign values to the variables not in S according to a small-bias generator, then we pseudorandomly restrict approximately pn variables in a way that preserves the acceptance probability of the circuit, as desired.

In order to devise a pseudorandom projection generator that turns a bounded-depth circuit into a small decision tree, it is sufficient to construct a pseudorandom projection generator that turns a depth-2 circuit into a small decision tree, as in Håstad's Switching Lemma, and then apply the generator independently a constant number of times. Toward the goal of building a pseudorandom restriction generator that satisfies the Håstad Switching Lemma, it is sufficient to use a pseudorandom generator that fools a class of statistical test powerful enough to test whether a given restriction does turn a given CNF into a bounded-depth decision tree. This is by itself a very difficult task, but Håstad's proof shows that, with high probability, not only the CNF is turned into a small decision tree by the restriction, but also that a small decision tree (which we call a *canonical* decision tree) computing the restricted formula can be computed efficiently by a relatively simple algorithm. Furthermore, in order to check if the canonical decision tree has bounded depth, we can perform a case-analysis over all possible long computational paths, and check if the path is part of the canonical decision tree. The latter property can be checked by a small depth-2 circuit, and so we able to show that any pseudorandom generator that fools depth-2 circuits (in particular, the generator of [DETT10], of seed $\tilde{O}(\log^2 M/\varepsilon)$) can be used to construct pseudorandom restrictions that match the Håstad Switching Lemma.

Barriers to Further Progress

The best known lower bound for AC0 circuit remains Håstad's lower bound, establishing that the parity function requires depth-d circuits of size $2^{\Omega(n^{\frac{1}{d-1}})}$. Without improving Håstad's result, every pseudorandom generator construction for AC0 must have seed length $\Omega(\log^d M/\varepsilon)$. This is due to the following well-known reduction: suppose that $G : \{0,1\}^\ell \to \{0,1\}^{2\ell}$ is $1/2s$-pseudorandom for circuits of size M and depth d. Then the problem of deciding if a given string in $\{0,1\}^{2\ell}$ is a possible output of the generator cannot be decided by circuits of size M and depth $d+1$. (If there were a circuit C of size M and depth $d+1$ deciding such a problem, then the output of the circuit is either the AND or the OR of $\leq M$ wires, each being the result of the computation of a circuit of size $\leq M$ and of depth $\leq d$; because of the pseudo randomness of G, each of these wires has approximately the same probability of carrying a 1 if the circuit is given a random string or a random output of the generator; the circuit, however, accepts with very low probability in the former case and with probability one in the latter case, which causes a contradiction.)

Without new circuit lower bounds, seed length $\log^{2d+\Omega(1)} M$ is a barrier for pseudorandom generator constructions in the framework of Nisan and Wigderson. In such a framework, when constructing a pseudorandom generator for circuits of size M and depth d, one starts from an exponential-time computable function $f : \{0,1\}^t \to \{0,1\}$ which is hard on average for circuits of size about M^2 and depth $d+1$; then one uses the function to devise a pseudorandom generator of seed length about $t^2/\log M$. The seed length $t^2/\log M$ is determined by the use of *combinatorial designs* in the construction, and it is best possible given known lower bounds on combinatorial designs. With the current circuit lower bound results, we have $t = \log^{d+\Omega(1)} M$ and so the seed length is $\log^{2d+\Omega(1)} M$.

Other Applications of Pseudorandom Restrictions

The recent work of Impagliazzo, Meka and Zuckerman [IMZ12] and of Gopalan et al. [GMR+12] also relies on pseudorandom restrictions. The work of Gopalan et al. [GMR+12] applies pseudorandom restrictions to read-once CNF formulas, and the restrictions both simplify the formula (by reducing the number of clauses) and preserve the acceptance probability – we are able to construct pseudorandom restriction generators that satisfy either requirement, but not both at the same time. The use of pseudorandom restrictions in [IMZ12] is more similar to ours, although the restrictions are composed in a different way.

2 Preliminaries

In this section we state Håstad's switching and provide some related definitions.

Definition 1 (CNFs) *An m-clause t-CNF is a boolean formula $F = \bigwedge_{i=1}^m C_i$, where each clause C_i is a disjunction of at most t literals, and a literal is either a variable or a negated variable.*

Definition 2 (Restriction, Selection, Assignment) *A* restriction *to a set of variables $x_1, ..., x_n$ is a string $\rho \in \{0,1,*\}^n$.*

If $\rho_i = *$ then we refer to x_i as being unrestricted*, otherwise we refer to x_i as being* assigned*.*

If F is a boolean formula over $x_1, ..., x_n$ then the restricted *formula $F_{/\rho}$ is the formula over the* unrestricted *variables obtained by* assigning *$x_i \leftarrow \rho_i$ for all $\rho_i \neq *$.*

It will be convenient to specify a restriction ρ *as a pair* (θ, β)*, where* $\theta \in \{*, \square\}^n$ *is a* selection *and* $\beta \in \{0,1\}^n$ *is an* assignment*, and* $\rho_i = *$ *if* $\theta_i = *$ *and* $\rho_i = \beta_i$ *if* $\theta_i = \square$*.*

Definition 3 (Random Selections) *We denote by* D_p *the* p*-biased probability distribution over selections* θ*, such that* $\theta_i = *$ *with probability* p *independently for each* i*. When* p *is a power of* $1/2$*, we will think of the process of sampling a random element of* D_p *as the process of sampling a uniformly random bit string* $\tau \in \{0,1\}^{n \log_2 1/p}$ *and then assigning* $\theta_i = *$ *if* $\tau_j = 1$ *for all* $(i-1)\log_2 1/p < j \leq i \log_2 1/p$*, and* $\theta_i = \square$ *otherwise.*

If ρ_1 and ρ_2 are two restrictions, then we denote by $\rho_1 \circ \rho_2$ the application of ρ_1 followed by the application of ρ_2. That is, if $\rho := \rho_1 \circ \rho_2$, then $\rho(x_i) = \rho_1(x_i)$ if $\rho_1(x_i) \neq *$, and $\rho(x_i) = \rho_2(x_i)$ otherwise.

A *decision tree* is a tree in which each internal node is labeled by a variable and each leaf is labeled by a boolean value. A decision tree defines the computation of a boolean function by traversing down the tree, picking the branch by examining the variable and the value assigned to the variable. Every boolean function can be computed by a decision tree. The *canonical* decision tree of a restricted boolean formula is the decision tree constructed via a simple greedy procedure as described below.

Definition 4 *We denote by* $\textsc{cdtree}(F, \rho)$ *the* canonical decision tree *of a CNF* $F = C_1 \wedge ... \wedge C_m$ *with respect to a restriction* ρ*, defined as follows:*

- *if* F *is empty, return* $\textsc{leaf}(1)$
- *if* $C_{1/\rho} \equiv 0$*, return* $\textsc{leaf}(0)$
- *if* $C_{1/\rho} \equiv 1$*, return* $\textsc{cdtree}((\wedge_{i=2}^{m} C_i)_{/\rho})$
- *otherwise let* S *be the set of* $*$*-ed variables in* $C_{1/\rho}$
  - *create a depth* $|S|$ *decision tree by considering all assignments* σ *to* F
  - *for each* σ*, attach* $\textsc{cdtree}(C_{/\rho\sigma})$ *at the leaf corresponding to* σ

Lemma 5 (Hastad Switching Lemma) *Let* F *be a* t*-CNF,* $p > 0$*,* $s > 0$*, and* $\rho = (\theta, \beta)$ *be a random restriction where* θ *is sampled from* D_p *and* β *is chosen uniformly at random.*

Then

$$\Pr_p[\textsc{depth}(\textsc{cdtree}(F, \rho)) > s] \leq (5pt)^s \ .$$

In order to find a pseudorandom distribution of restrictions that has a high probability of reducing CNFs to small depth decision trees, we need to study the complexity of determining, for a fixed F and a given ρ, whether the canonical decision tree of $F_{/\rho}$ has small depth. A distribution that is able to fool tests which perform such a computation will satisfy the conclusion of the switching lemma. We start by describing a certificate of the fact that the canonical decision tree of $F_{/\rho}$ is not shallow, namely a long computational path which must be included in the canonical tree.

In general, a computational path in a decision tree is a sequence of pairs (x_i, b_i) where x_i is a variable name and b_i is a boolean value. In the canonical decision tree of a CNF formula, the variables are always read in a particular pattern: namely, the computation can be seen as a sequence of phases, where in each phase the decision tree queries a subset of the variables in one of the clauses of the formula. We represent (prefixes of) such paths in the following way.

Definition 6 (Segment, Path) *A* segment *is a triplet (a_i, S_i, σ_i), where a_i is the index of clause C_{a_i}, S_i is a subset of the variables in the clause C_{a_i}, $\sigma_i \in \{0,1\}^{|S_i|}$ is an assignment to the variables in S_i, the sets S_i are disjoint, and $a_1 < a_2 < \cdots < a_k$. (A segment represents a clause expanded in Håstad's construction of the canonical decision tree, with S_i being the $*$-ed variables in C_{a_i}). A* path *is a list of segments. The length of a path is the sum of $|S_i|$ over all its segments. Notice that every computational path of $\text{CDTREE}(F, \rho)$ can be represented as a path according to the above definition. Furthermore, if there is a path of length more than s in $\text{CDTREE}(F, \rho)$ then there is a prefix of that computational path of length $\geq s+1$ and $\leq s+t$ according to the above definition.*

3 A Derandomized Switching Lemma

Recall that if $\mathcal{F}$ is a set of functions $f : \{0,1\}^n \to \{0,1\}$ and D is a distribution over $\{0,1\}^n$, then we say that D ε-fools $\mathcal{F}$ if

$$\forall f \in \mathcal{F}. \left| \mathop{\mathbb{P}}_{x \sim D}[f(x) = 1] - \mathop{\mathbb{P}}_{x \sim U_n}[f(x) = 1] \right| \leq \varepsilon$$

where U_n is the uniform distribution. We show that every distribution that fools CNFs can be used to select pseudorandom restrictions that obey a version of Håstad's switching lemma.

Lemma 7 (Main Lemma) *Let F be a M-clause t-CNF over n variables, $p = 2^{-q}$ be a positive parameter, D be a distribution over $\{0,1\}^{(q+1)n}$ that ε_0 fools all $M2^{t\cdot(q+1)}$-clause CNFs. Then*

$$\Pr_{(\theta,\beta)\sim D}[\text{DEPTH}(\text{CDTREE}(F_{|\theta,\beta})) > s] < 2^{s+t+1}(5pt)^s + \varepsilon_0 \cdot 2^{(s+1)(2t+\log M)}$$

PROOF: We begin by proving the following fact.

Claim 8 *Let $z = ((C_{a_1}, S_1, \sigma_1), \ldots, (C_{a_k}, S_k, \sigma_k))$ be a path as in Definition 6. Then we claim that there is a CNF G_z with $\leq 2^{(q+1)\cdot t} M$ clauses such, that for every restriction $\rho \in \{0,1\}^{(q+1)\cdot n}$, $G(\rho)$ is satisfied if and only if z is a prefix of a computational path of the canonical decision tree of $F_{|\rho}$.*

PROOF: For $j = 1, \ldots, k$, define the restriction γ_j as the composition of the restrictions $S_1 \leftarrow \sigma_1$, $\ldots$, $S_j \leftarrow a_j$. To prove the claim, observe that z is a prefix of a computational path in $\text{CDTREE}(F_{|\rho})$ if and only if the following conditions are true:

- For each $i = 1, \ldots, a_1 - 1$, ρ satisfies the clause C_i, meaning that ρ assigns at least one literal in C_i to a value that makes the literal (and hence the clause C_i) true;
- The variables left unrestricted by ρ in C_{a_1} are precisely the variables in S_i; furthermore, the literals restricted by ρ in C_{a_1} are all given values that contradict the literals, and γ_1 satisfies C_{a_1};
- For each $i = a_1 + 1, \ldots, a_2 - 1$, $\rho \circ \gamma_1$ satisfies the clause C_i;
- The variables left unrestricted by $\rho \circ \gamma_1$ in C_{a_2} are precisely the variables in S_2; furthermore, the literals restricted by ρ in C_{a_2} are all given values that contradict the literals, and γ_2 satisfies C_{a_2};

  $\ldots$

- For each $i = a_{k-1}+1, \ldots, a_k - 1$, $\rho \circ \gamma_{k-1}$ satisfies the clause C_i;
- The variables left unrestricted by $\rho \circ \gamma_{k-1}$ in C_{a_k} are precisely the variables in S_k; furthermore, the literals restricted by $\rho \circ \gamma_{k-1}$ in C_{a_k} are all given values that contradict the literals.

That is, we have once condition for each of the $a_k \leq M$ clauses $C_1, \ldots, C_{a_k}$, and the condition on clause C_i is only a function of the values of ρ on the t variables appearing in C_i (recall that we are thinking of z as fixed); in the representation of ρ as a bit string that we use, the value of ρ on one variable is determined by $q+1$ bits, and so the condition on clause C_i is a function of $t \cdot (q+1)$ and it can be expressed as a CNF with $\leq 2^{t \cdot (q+1)}$ clauses. This concludes the proof of the claim. $\square$

For a CNF F and a restriction ρ, we have that $\textsc{cdtree}(F_{|\rho})$ has depth $> s$ if and only if there is a path z (in the sense of Definition 6) that is of length bigger than s (and, without loss of generality, at most $s+t$) which occurs as a prefix of a computational path of $\textsc{cdtree}(F_{|\rho})$. The number of syntactically correct minimal paths $z = ((C_{a_1}, S_1, \sigma_1), \ldots, (C_{a_k}, S_k, \sigma_k))$ of length between $s+1$ and $s+t$ (minimal, in this context, means that $((C_{a_1}, S_1, \sigma_1), \ldots, (C_{a_{k-1}}, S_{k-1}, \sigma_{k-1}))$ has length $\leq s$) is at most $M^{s+1} \cdot 3^{t \cdot (s+1)}$, because we have $k \leq s+1$ (the minimality condition and the fact that the sets $S_1, \ldots S_{k-1}$ are non-empty implies $k-1 \leq s$) and there are $\binom{M}{k} \leq M^k \leq M^{s+1}$ ways of choosing the clauses $C_{a_1}, \ldots, C_{a_k}$, and there are at most $3^k t \leq 3^{(s+1) \cdot t}$ ways of choosing the sets S_i and the assignments σ_i. Let $\mathcal{Z}$ be the set of all syntactically correct minimal paths of length between $s+1$ and $s+t$.

Finally, let $\text{occur}(T, z)$ be 1 if the path z occurs as a prefix of a computational path in the tree T. Because of the claim that we proved above, and the pseudo randomness assumption that we have on D, we conclude that for every path $z \in \mathcal{Z}$,

$$\Pr_{\rho \sim D}[\text{occur}(\textsc{cdtree}(F_{|\rho}), z)] \leq \Pr_{\rho \sim U_{(q+1) \cdot n}}[\text{occur}(\textsc{cdtree}(F_{|\rho}), z)] + \varepsilon_0 \tag{1}$$

We are now ready to bound the probability that $\textsc{cdtree}(F_{|\rho})$ has depth bigger than s when ρ is chosen from the distribution D.

$$\begin{aligned}
&\Pr_{\rho \sim D}[\mathit{depth}(\textsc{cdtree}(F_{|\rho})) > s] \\
&= \Pr_{\rho \sim D}[\exists z \in \mathcal{Z}.\text{occur}(\textsc{cdtree}(F_{|\rho}), z) = 1] \\
&\leq \mathop{\mathbb{E}}_{\rho \sim D} \sum_{z \in \mathcal{Z}} \text{occur}(\textsc{cdtree}(F_{|\rho}), z) \\
&= \sum_{z \in \mathcal{Z}} \Pr_{\rho \sim D}[\text{occur}(\textsc{cdtree}(F_{|\rho}), z) = 1]
\end{aligned}$$

Using (1), we have

$$\begin{aligned}
&\sum_{z \in \mathcal{Z}} \Pr_{\rho \sim D}[\text{occur}(\textsc{cdtree}(F_{|\rho}), z) = 1] \\
&\leq \sum_{z \in \mathcal{Z}} \left(\Pr_{\rho \sim U_{(q+1) \cdot n}}[\text{occur}(\textsc{cdtree}(F_{|\rho}), z) = 1] + \varepsilon_0 \right) \\
&\leq \mathop{\mathbb{E}}_{\rho \sim U_{(q+1) \cdot n}} \left[\sum_{z \in \mathcal{Z}} \text{occur}(\textsc{cdtree}(F_{|\rho})) \right] + \varepsilon_0 \cdot |\mathcal{Z}|
\end{aligned}$$

$$\leq 2^{s+t} \cdot (5pt)^s + \varepsilon_0 \cdot 3^{(s+1)\cdot t} \cdot M^{s+1}$$

where the last inequality uses the bound of Håstad's switching lemma and the fact that, for each restriction ρ, if the tree $\textsc{cdtree}(F_\rho)$ has depth at most s then it contains zero paths from $\mathcal{Z}_¡$ and if it has depth more than s then it contains at most 2^{s+t} paths from $\mathcal{Z}$. □

In our applications, we will consider a distribution of random restrictions $\rho = (\theta, \beta)$ where the selection θ is sampled from a pseudorandom distribution, but the assignment β is sampled uniformly at random. Lemma 7 applies to such distributions as well because of the following observation.

Fact 9 *Let X be a distribution over $\{0,1\}^{n_1}$ that ε-fools m-clause CNFs. Consider the distribution D over $\{0,1\}^{n_1+n_2}$ obtained by sampling an n_1-bit string according to X and then concatenating an n_2-bit string chosen uniformly at random. Then D ε-fools m-clause CNFs.*

Proof: Let F be an m-clause CNF over $n_1 + n + 2$ variables. Then we have

$$\left| \Pr_{(a,b)\sim D}[F(a,b)=1] - \Pr_{(a,b)\sim U_{n_1+n_2}}[F(a,b)=1] \right|$$

$$= \left| \mathop{\mathbb{E}}_{b\sim U_{n_2}}[\Pr_{a\sim X}[F(a,b)=1]] - \mathop{\mathbb{E}}_{b\sim U_{n_2}}[\Pr_{a\sim U_{n_1}}[F(a,b)=1]] \right|$$

$$= \left| \mathop{\mathbb{E}}_{b\sim U_{n_2}}[\Pr_{a\sim X}[F(a,b)=1] - \Pr_{a\sim U_{n_1}}[F(a,b)=1]] \right|$$

$$\leq \mathop{\mathbb{E}}_{b\sim U_{n_2}} \left| \Pr_{a\sim X}[F(a,b)=1] - \Pr_{a\sim U_{n_1}}[F(a,b)=1] \right|$$

$$\leq \varepsilon$$

□

4 Derandomizing AC0

We will use the following pseudorandom generator construction.

Theorem 10 ([DETT10]) *There is a polynomial time pseudorandom generator of seed length $\tilde{O}(\log^2 M/\varepsilon)$ that is ε-pseudorandom for M-clause CNFs.*

By repeated application of the Main Lemma as in Hastad's work we have:

Theorem 11 (Derandomized Switching Lemma for AC0) *Let C be a size M, depth d circuit, and $p = 2^{-q}$ a positive parameter. Then there exists a pseudorandom selection generator $\mathcal{G}_0$ of seed length $d \cdot \tilde{O}(q^2 \log^2 \frac{M}{\varepsilon_0})$ such that:*

- $\Pr_{\theta' \leftarrow \mathcal{G}_0, \beta \leftarrow \mathcal{U}}[\textsc{depth}(\textsc{cdtree}(C_{/\theta'\beta})) > s] < M \left(2^{s+\log M+1} \cdot (10p \log M)^s + \varepsilon_0 \cdot 2^{(s+1)\cdot 3\log M)} \right)$

- *Each variable has probability at least $p^{d-1}/40$ chance of being starred in θ'.*
- *Furthermore, a generator with the same seed length and properties exist which outputs both a selection and restriction.*

PROOF: [Sketch] As in Håstad's proof, we construct $\mathcal{G}_0$ as follows: we run d iterative pseudorandom selections using the generator of [DETT10] in each iteration to produce a qn-bit string that is ε_0-pseudorandom for CNFs of size $M \cdot 2^{(\log M)\cdot(q+1)}$. As proved in the previous section, the pair θ', β obtained by sampling a restriction θ' via such a generator and an assignment β uniformly at random is a $(q+1)n$-bit string that is also ε_0-pseudorandom for CNFs of size $M \cdot 2^{(\log M)\cdot(q+1)}$. For the first iteration, we use parameter $1/40$. With high probability, after the restriction, all the surviving bottom gates of the circuit have fan-in at most $\log M$. For the remaining $d-1$ iterations, we use parameter $2p$, and at each iteration the derandomized switching lemma, applied with $t = \log M$, implies that (with high probability) each of the gates one level up from the bottom perform a computation that can be also performed by a $\log M$-depth decision tree, and hence both by a $\log M$-CNF and a $\log M$-DNF, and we use this fact to switch AND gates to OR gates or viceversa, and to reduce by one the depth of the circuit. □

Theorem 12 *Let C be a size M, depth d circuit and $\varepsilon > 0$ be a positive parameter. There is a pseudorandom restriction generator $\mathcal{G}_1$ of seed length $d \cdot \tilde{O}(\log^4 \frac{M}{\varepsilon})$ such that*

- $|\Pr_{\rho \leftarrow \mathcal{G}_1, x \leftarrow U_n}[C_\rho(x) = 1] - \Pr_{y \leftarrow U_n}[C(y) = 1]| < \varepsilon$
- *Each variable has probability at least $p = 1/(40^{d+1} \cdot \log^d M)$ of being assigned by ρ.*

PROOF: We first make the following observation: if θ is a selection, θ^c is the complementary selection (that is $\theta(x_i) = * \Leftrightarrow \theta^c(x_i) = \square$) and if β and x are assignments, then

$$C_{(\theta,\beta)}(x) = C_{(\theta^c,x)}(\beta)$$

because in both cases we are assigning the variables starred in θ according to x and the remaining variables according to β.

Consider the generator $\mathcal{G}_0$ from Theorem 11 instantiated with parameters $p := \frac{1}{40 \log M}$, $s := 2 \log M / 4\varepsilon$ and ε_0 that we will fix later. Then we have

$$\Pr_{\theta \leftarrow \mathcal{G}_0, x \leftarrow U_n}[depth(\text{CDTREE}(C_{(\theta,x)})) > s] \leq M \cdot 2 \cdot 2^s \cdot M \cdot 4^{-s} + \varepsilon_0 \cdot M \cdot 2^{(s+1)\cdot 3 \cdot \log M} < \varepsilon$$

if we pick $\varepsilon_0 = 2^{-O(\log^2 M/\varepsilon)}$.

Now, whenever $C_{\theta,x}$ is computable by a depth-s decision tree, then we have that if β is sampled from an s-wise independent distribution B then

$$\Pr_{\beta \leftarrow B}[C_{\theta,x}(\beta) = 1] = \Pr_{y \leftarrow U_n}[C_{\theta,x}(y) = 1]$$

so we have

$$\left| \Pr_{\theta \leftarrow \mathcal{G}_0, x \leftarrow U_n, \beta \leftarrow B}[C_{\theta,x}(\beta) = 1] - \Pr_{\theta \leftarrow \mathcal{G}_0, x \leftarrow U_n, y \leftarrow U_n}[C_{\theta,x}(y) = 1] \right| \leq \varepsilon$$

but note that

$$\Pr_{\theta\leftarrow\mathcal{G}_0,x\leftarrow U_n,y\leftarrow U_n}[C_{\theta,x}(y)=1]=\Pr_{y\leftarrow U_n}[C(y)=1]$$

and that

$$\Pr_{\theta\leftarrow\mathcal{G}_0,x\leftarrow U_n,\beta\leftarrow B}[C_{\theta,x}(\beta)=1]=\Pr_{\theta\leftarrow\mathcal{G}_0,\beta\leftarrow B,x\leftarrow U_n,}[C_{\theta^c,\beta}(x)=1]$$

The theorem now follows by taking $\mathcal{G}_1$ to be the generator that select θ from $\mathcal{G}_0$ with the above described parameter and β from a $2\log M/4\varepsilon$-wise independent distribution, and outputs (θ^c,β). The seed length required to construct β is just $O(\log M/\varepsilon\cdot\log n)$, and the seed length required to construct θ is $\tilde{O}(\log^4 M/\varepsilon)$. □

Theorem 13 *For every M, d, ε, there is a polynomial time computable ε-pseudorandom generator for circuits of size M and depth d, whose seed length is $\tilde{O}(\log^{d+4} M/\varepsilon)$.*

The theorem follows by iteratively appling $T:=p^{-1}\log\frac{2n}{\varepsilon}$ independent pseudorandom restrictions from $\mathcal{G}_1$, each with parameter $\varepsilon/2T$. The probability there remains an unfixed variable is at most $\varepsilon/2$, and the overall error caused by the restrictions is at most $\varepsilon/2$.

Acknowledgements

We thank Or Meir for insightful discussions and Russell Impagliazzo for explaining the intuition behind the arguments in [IMZ12].

References

[AW85] Miklos Ajtai and Avi Wigderson. Deterministic simulation of probabilistic constant-depth circuits. In *Proceedings of the 26th IEEE Symposium on Foundations of Computer Science*, pages 11–19, 1985. 1

[Baz07] Louay Bazzi. Polylogarithmic independence can fool DNF formulas. In *Proceedings of the 48th IEEE Symposium on Foundations of Computer Science*, pages 63–73, 2007. 1

[Bra09] Mark Braverman. Poly-logarithmic independence fools AC0 circuits. Technical Report TR09-011, Electronic Colloquium on Computational Complexity, 2009. 1

[DETT10] Anindya De, Omid Etesami, Luca Trevisan, and Madhur Tulsiani. Improved pseudorandom generators for depth 2 circuits. In *APPROX-RANDOM*, pages 504–517, 2010. 1, 2, 7, 8

[GMR+12] Parikshit Gopalan, Raghu Meka, Omer Reingold, Luca Trevisan, and Salil Vadhan. Better pseudorandom generators from milder pseudorandom restrictions. In *Proceedings of the 53rd IEEE Symposium on Foundations of Computer Science*, 2012. 3

[IMZ12] Russell Impagliazzo, Raghu Meka, and David Zuckerman. Pseudorandomness from shrinkage. In *Proceedings of the 53rd IEEE Symposium on Foundations of Computer Science*, 2012. 3, 9

[Nis91] N. Nisan. Pseudorandom bits for constant depth circuits. *Combinatorica*, 12(4):63–70, 1991. 1

[Raz09] Alexander Razborov. A simple proof of bazzi's theorem. *ACM Trans. Comput. Theory*, 1(1):1–5, 2009. 1

ECCC ISSN 1433-8092

http://eccc.hpi-web.de