2012 IEEE 53rd Annual Symposium on Foundations of Computer Science

Pseudorandomness from Shrinkage

Russell Impagliazzo
University of California, San Diego and
Institute for Advanced Study

Raghu Meka*
Institute for Advanced Study

David Zuckerman†
University of Texas at Austin and
Institute for Advanced Study

***Abstract*—One powerful theme in complexity theory and pseudorandomness in the past few decades has been the use lower bounds to give pseudorandom generators (PRGs). However, the general results using this hardness vs. randomness paradigm suffer a quantitative loss in parameters, and hence do not give nontrivial implications for models where we don't know super-polynomial lower bounds but do know lower bounds of a fixed polynomial. We show that when such lower bounds are proved using random restrictions, we can construct PRGs which are essentially best possible without in turn improving the lower bounds.**

More specifically, say that a circuit family has shrinkage exponent Γ if a random restriction leaving a p fraction of variables unset shrinks the size of any circuit in the family by a factor of $p^{\Gamma+o(1)}$. Our PRG uses a seed of length $s^{1/(\Gamma+1)+o(1)}$ to fool circuits in the family of size s. By using this generic construction, we get PRGs with polynomially small error for the following classes of circuits of size s and with the following seed lengths:

1) **For de Morgan formulas, seed length $s^{1/3+o(1)}$;**
2) **For formulas over an arbitrary basis, seed length $s^{1/2+o(1)}$;**
3) **For read-once de Morgan formulas, seed length $s^{.234\ldots}$;**
4) **For branching programs of size s, seed length $s^{1/2+o(1)}$.**

The previous best PRGs known for these classes used seeds of length bigger than $n/2$ to output n bits, and worked only when the size $s = O(n)$ [1].

***Keywords*-pseudorandomness; shrinkage; average-case lower-bounds; random restrictions**

I. Introduction

Two of the most important general challenges for complexity are to prove constructive lower bounds for non-uniform measures of computational complexity such as circuit size, and to show that randomized algorithms have efficient deterministic simulations. The "Hardness vs. Randomness" paradigm ([2], [3], [4], [5], [6], [7]) shows that these questions are linked. More precisely, these results show how to use any problem that is hard for a class of circuits to create a pseudorandom generator (PRG) for the same class of circuits. This PRG can then be used to construct a relatively efficient deterministic version of any probabilistic algorithm with a corresponding complexity. This has been used both to create unconditional PRGs for circuit classes with known lower bounds, such as AC_0, and conditional results, implications between the existence of hard problems and derandomization for classes where no strong lower bounds are known. In the converse direction, it is easy to see that any PRG for a circuit class immediately gives a corresponding lower bound for the class. Somewhat more surprisingly, it has been shown that any efficient deterministic simulation of some probabilistic algorithms would yield circuit lower bounds ([8], [9], [10]). This hardness vs. randomness connection is one of the most important tools in computational complexity. It formalizes the intuition that efficient algorithms for "meta-computational problems", where the input is a computational device from a certain class, is linked to our ability to prove lower bounds for that class.

However, being so general comes at a quantitative price. Ideally, the stretch of a PRG (the output length as a function of the input length) equals the known lower bound. However, in the hardness to randomness constructions, there are a number of stages that each lose a large polynomial factor. In particular, this means, that, for example, a quadratic or cubic circuit lower bound for a class does not immediately give any nontrivial PRG. For completely generic, "black-box", reductions between a hard problem and a PRG, some of these costs are inherent ([11], [12], [13], [14]). In particular, this is an issue for those models where super-linear but not super-polynomial bounds are known, such as Boolean formulas.

In this work, we show a general method for obtaining *tight* "hardness to randomness" results from the *proofs* of lower bounds, rather than as a black-box consequence of the lower bounds. In particular, our methods apply to lower bound proofs that involve restricting some of the inputs to the circuit. Our construction goes in two stages. We start with a lower bound proved by the following kind of *shrinkage* argument: if we restrict a size s circuit leaving a p fraction of variables unset, the expected size of the restricted circuit is $O(p^\Gamma s)$. The best Γ for which this holds is known as the "shrinkage exponent" of the circuit class. The first stage of our construction is to *derandomize* the shrinkage argument, showing that there is a distribution with similar shrinkage properties that can be sampled with few bits. This stage of our argument is general, but not totally generic. While the same general construction and analysis ideas work in a variety of models, the details depend on the model. Then we show how to go from such a distribution on restrictions to a PRG. This part is generic, being identical for all models, and is very related to the generator from [15]. The total number of bits r used by the generator is roughly $s^{1/(1+\Gamma)}$ times the number of bits needed to sample from the distribution on restrictions.

Every generator using r bits to fool tests of size $s = s(r)$ immediately gives a problem requiring size $\Omega(s(r))$ to com-

*Supported in part by NSF grant DMS-0835373.
†Supported in part by NSF Grants CCF-0916160 and DMS-0835373.

0272-5428/12 $26.00 © 2012 IEEE
DOI 10.1109/FOCS.2012.78

pute in the model. So, if our function $s(r)$ is close to the known lower bounds, this shows that we have essentially converted all of the "hardness" in the lower bound to "randomness". This is indeed the case for a variety of natural models of computation. For Boolean formulas over the de Morgan basis, we give a generator with $s(r) = r^{3-o(1)}$, almost matching the known lower bound of $s(n) = \Omega(n^3/\log^2 n)$ due to Håstad ([16], based on earlier work by [17], [18], [19], [20]). To avoid technicalities, we assume that the size s is at least the number of input variables n in the following statements.

Theorem I.1. *For any constant $c > 0$, there is an explicit PRG using a seed of length $s^{1/3} \cdot 2^{O(\log^{2/3} s)} = s^{1/3+o(1)}$ random bits that s^{-c}-fools formulas of size s over the de Morgan basis.*

For Boolean formulas over an arbitrary basis, our generator has stretch $s(r) = r^{2-o(1)}$, almost matching the Khrapchenko bound of $s(n) = \Omega(n^2)$ ([21]).

Theorem I.2. *For any constant $c > 0$, there is an explicit PRG using a seed of length $s^{1/2} \cdot 2^{O(\log^{1/2} s)} = s^{1/2+o(1)}$ random bits that s^{-c}-fools formulas of size s over an arbitrary basis.*

For branching programs, with size being the total number of edges, we get a similar bound.

Theorem I.3. *For any constant $c > 0$, there is an explicit PRG using a seed of length $s^{1/2} \cdot 2^{O(\log^{1/2} s)} = s^{1/2+o(1)}$ random bits that s^{-c}-fools branching programs of size at most s.*

We also consider the case of read-once formulas over the DeMorgan basis. Here, there is no sensible notion of lower bound, since all functions computable in the model have size exactly n, but the notion of shrinkage is defined. The optimal shrinkage exponent for such read-once DeMorgan formulas was shown by [20], [22] to be $\Gamma = \log 2/\log(\sqrt{5}-1) = 3.27...$; using this result, we get a PRG with stretch $s(r) = \Omega(r^{4.27...})$.

Theorem I.4. *For any constant $c > 0$, there is an explicit PRG using a seed of length $s^{1/(\Gamma+1)} \cdot 2^{O(\log^{2/3} s)} = s^{1/(\Gamma+1)+o(1)}$ random bits that s^{-c}-fools read-once formulas of size s over the de Morgan basis, where $\Gamma = \log 2/\log(\sqrt{5}-1) = 3.27...$.*

Any substantial improvement in our PRGs would thus yield better lower bounds than what is currently known.

Our results dramatically improve previous work. The only directly comparable PRG was by Bogdanov, Papakonstantinou, and Wan [1], who constructed a PRG using a $(1-\Omega(1))n$ bit seed to output n bits that fool read-once formulas and read-once branching programs, where the order of the bits is unknown beforehand. There has been significant work on read-once branching programs where the order of the bits is known in advance (e.g., [6], [23], [15]), but that is a much simpler model and the generators of [6], [23] are known to fail if the order of bits is unknown [1].

A. Outline of Constructions

Our techniques build upon those of [24], [23], [15]. The intuition behind all of these PRGs is to exploit *communication bottlenecks* in the computation. If the random inputs to a computation can be partitioned into two subsets X and Y, and the computation can be simulated with k bits of communication between these two subsets, then given the communication pattern, the two sets of bits have high entropy conditioned on each other. Then, instead of using independent bits for the two sides, we can use a *randomness extractor* to convert the conditional entropy of X given the communication into random bits to be used in place of Y.

Our construction follows the same basic intuition. The key insight is that shrinkage under random restrictions is a form of communication bottleneck, between X, the set of variables with values specified by the restriction ρ, and Y, the set of variables left unrestricted (and chosen later). Consider a one-way protocol where the player knowing X has to send a message allowing the Y-player to compute the function f. What this message exactly needs to specify is the restricted function f_ρ. If the circuit size of f_ρ is small, much smaller than the size of X, the message can be the circuit computing the restricted function, showing low communication.

Most of the previous constructions were for computational models like read-once branching programs, where one had an explicit description of which sets X and Y had a communication bottleneck, and there was a hierarchical structure available on such sets so that the construction could be defined recursively. Here, we do not have either one, but we know the bottleneck occurs for most sets X and their complements. Instead of explicitly partitioning the variables into blocks with low communication, we randomly sample sets that exhibit this bottleneck until all variables are covered. So far, we are not able to utilize recursion, which blocks us from making the seed size sub-polynomial (and hence proving super-polynomial lower bounds).

More concretely, consider the case of read-once width w branching programs, where the bits may be read in any order (as opposed to some fixed order, which is the setting of Nisan [24]). In this arbitrary-order case, we show that the Nisan-Zuckerman PRG [15], without recursion, gives a PRG with seed length $\tilde{O}(\sqrt{n})$. Recall that this PRG uses an extractor $E : \{0,1\}^s \times \{0,1\}^d \to \{0,1\}^m$ and is defined by $G(x, y_1, \ldots, y_t) = E(x, y_1)E(x, y)2) \ldots E(x, y_t)$, where $x \in \{0,1\}^s$ and $y_i \in \{0,1\}^d$.

To see that this works, suppose a branching program accepts a uniform input with significantly different probability than the output of G. By a hybrid argument, changing some $Z_i = E(X, Y_i)$ to uniform must change the probability significantly. However, if we fix all bits except the m bits corresponding to Z_i, we are left with a read-once branching program on these m bits. There are at most w^{wm} such branching programs on m bits. Thus, if we condition on a typical such branching program for these m bits, X still has min-entropy at least $s - 2mw \log w$. As long as this exceeds the min-entropy requirement of the extractor, the extractor output is close to uniform, contradicting the assumption of significantly different acceptance probabilities. We can set $t = m = \sqrt{n}$, $s = 4mw \log w$, and $d = O(\log n)$.

For general branching programs, we need to handle variables that are read many times, which we can do by pseudorandomly permuting the output of the above generator. However, for general formulas and to get a general reduction, we need to extend the above generator. We do this by combining the extractor outputs with pseudorandom restrictions that shrink with high probability (and leave every bit unfixed with the same probability). Specifically, for a restriction ρ that leaves m bits unfixed, we can define the random variable $V_\rho \in \{0,1\}^n$ that takes the values of ρ for the fixed bits and the values of $E(X, Y_\rho)$ for the unfixed bits. We do this for enough independent pseudorandom restrictions that with high probability every coordinate has some ρ which leaves that coordinate unfixed (via a coupon collector bound). The PRG output is the xor of all these V_ρ.

In fact, the above achieves the desired bounds only when the shrinkage $\Gamma = 1$. For larger shrinkage, we must also apply a k-wise independent distribution to $E(X, Y_\rho)$.

a) Derandomized Shrinkage Bounds.: To use our main generator construction, we need a family of random restrictions that can be sampled with few random bits, and still causes the circuits to shrink. For branching programs and formulas over an arbitrary basis (shrinkage exponent $\Gamma = 1$) these are not too hard to get by taking $O(\log n)$-wise independent random restrictions. For formulas over the de Morgan basis and read-once formulas getting such restrictions is far trickier.

The first difficulty we face is that Håstad's original proof [16] only shows shrinkage in expectation and does not give a high probability bound for the formulas to shrink. We get around this difficulty as follows. Let f be a formula over the DeMorgan basis. We first show shrinkage under restrictions for which the probability of being unset $p = n^{-\alpha}$ for some $\alpha = o(1)$ and have $k = n^{o(1)}$-wise independence. By repeating this process independently, we get shrinkage for all values of p (both in the known lower bounds and in our PRG construction we need $p \sim n^{1/(\Gamma+1)}$). To do this, we decompose the target formula f into $O(n/\ell)$ subformulas g_i of size at most ℓ, for a suitable $\ell < k$. Since each g_i now has size at most k, the behavior of g_i under restrictions should be the same under k-wise independent restrictions or truly random restrictions. Thus, we can roughly expect each g_i to shrink by p^Γ in expectation.

For read-once formulas, the events that the different g_i shrink are independent, and hence by a Chernoff bound with high probability the total shrinkage is as promised. For the read-t case (each variable appears at most t times in the formula), we partition the subformulas into $t\ell+1$ color classes, such that within a color class the subformulas are on disjoint sets of variables. We can then proceed as in the read-once case. For the general case, we condition on heavy variables (the ones that appear *many* times) in a subtle way and reduce to the read-t case.

B. Related Work

Independently and concucurrently to this work, Komargodski and Raz [25] showed an average-case lower bound for de Morgan formulas nearly matching Andreev's [18] worst-case lower bound: [25] give an explicit function on n variables such that any de Morgan formula of size $n^{2.499}$ agrees with the function on at most $1/2 + \varepsilon$ fraction of the inputs, where $\varepsilon = \exp(n^{-\Omega(1)})$ is exponentially small. In the course of showing their result, Komargodski and Raz also show that shrinkage happens with high probability as opposed to in expectation which compares to Lemmas IV.2 and IV.8 in this work. However, the corresponding results in [25] work with truly random restrictions and achieve an exponent of 1.5 for de Morgan formulas.

II. Preliminaries

We start with some definitions and notations.

- For a restriction $\rho \in \{0,1,*\}^n$, let the set of *active* variables be $A(\rho) = \{i : \rho_i = *\}$.
- For $\rho \in \{0,1,*\}^n$ and $f : \{0,1\}^n \to \{0,1\}$, define the ρ-restricted function $f\restriction_\rho : \{0,1\}^{A(\rho)} \to \{0,1\}$ by $f\restriction_\rho(y) = f(x)$, where $x \in \{0,1\}^n$ satisfies $x_i = y_i, i \in A(\rho)$ and $x_i = \rho_i$ otherwise. When there is no ambiguity about subscripts, we sometimes use f_ρ to denote $f\restriction_\rho$.
- Call a distribution $\mathcal{D}$ on $\{0,1,*\}^n$ p-regular if for every $i \in [n]$, $\mathsf{Pr}_{\rho \leftarrow \mathcal{D}}[\rho_i = *] = p$. We say $\mathcal{D}$ is k-wise independent if any k coordinates of $\mathcal{D}$ are independent. There exist explicit k-wise independent distributions samplable with $O\left(k(\log n)\log(1/p)\right)$ random bits [26].
- For a class of functions $\mathcal{F}$ on n variables we say $s : \mathcal{F} \to \mathbb{N} \setminus [n]$ is a *size function* if $|\{f \in \mathcal{F} : s(f) \leq m\}| \leq m^{O(m)}$ for $m \geq n$. By default, we shall assume that $\mathcal{F}$ is closed under negating the input variables and for any $f \in \mathcal{F}$, $s(g) = O(s(f))$ if g is obtained from f by negating some input variables. We also assume that any $f \in \mathcal{F}$ depends on at most $s(f)$ variables.
- We say two distributions $\mathcal{D}, \mathcal{D}'$ (on the same universe) are ε-close if the statistical distance between $\mathcal{D}$ and $\mathcal{D}'$ is at most ε.
- We say a generator $G : \{0,1\}^r \to \{0,1\}^n$ δ-fools a function $f : \{0,1\}^n \to \{0,1\}$ if
$$\left| \mathop{\mathsf{Pr}}_{x \in_u \{0,1\}^n}[f(x) = 1] - \mathop{\mathsf{Pr}}_{y \in_u \{0,1\}^r}[f(G(y)) = 1] \right| \leq \delta.$$
Similarly, we say G δ-fools a class of functions $\mathcal{F}$ if G δ-fools all functions in $\mathcal{F}$. We refer to the parameter r as the seed-length of the generator G and say G is explicit if G can be computed in $\mathrm{poly}(n, 1/\delta)$ time.
- Throughout, we use upper case letters for random variables and lower case for constants.

As mentioned in the introduction, our generator is motivated by the pseudorandom generator for small space machines of Nisan and Zuckerman [15]. As in their paper, our construction will make use of extractors for linear min-entropy sources[1].

Definition II.1 (Extractor)**.** *We say $E : \{0,1\}^N \times \{0,1\}^d \to \{0,1\}^m$ is a (k, ε)-extractor if for every random variable X*

[1]The min-entropy of a variable X is defined by $H_\infty(X) \equiv -\max_x \left(\log_2 \left(\mathsf{Pr}[X = x]\right)\right)$.

over $\{0,1\}^N$ *with* $H_\infty(X) \geq k$*, and* $Y \in_u \{0,1\}^d$*,* $E(X,Y)$ *is* ε*-close to the uniform distribution on* $\{0,1\}^m$*. We say* $E(\ )$ *is explicit if it can be computed in time* $\mathrm{poly}(N,d)$*.*

We use the following explicit extractor as given by the work of Zuckerman [27].

Theorem II.2 ([27])**.** *There exists an explicit function* $E : \{0,1\}^N \times \{0,1\}^d \to \{0,1\}^m$ *that is an* $(N/2,\varepsilon)$*-extractor with* $m = N/4$ *and* $d = O(\log(N/\varepsilon))$*.*

We also use the following large-deviation bounds for k-wise independent random variables. The first is an easy corollary of Theorem 4 in [28].

Lemma II.3. *Let* $a_1,\ldots,a_n \in \mathbb{R}_+$ *with* $\max_i a_i = m$*, and suppose* $X_1,\ldots,X_n \in \{0,1\}$ *are* k*-wise independent indicator random variables with* $\Pr[X_i = 1] = p$*. Let* $X = \sum_i a_i X_i$ *and* $\mu = \mathbb{E}[X] = p\sum_i a_i$*. Then,* $\mathsf{Pr}[\, X \geq 2k(m+\mu)\,] \leq 2^{-k}$*.*

When the expectation is small, the following simple lemma sometimes gives better bounds.

Lemma II.4. *Suppose* $X_1,\ldots,X_n \in \{0,1\}$ *are* k*-wise independent indicator random variables with* $\Pr[X_i = 1] = p$*. Let* $X = \sum_i X_i$ *and* $\mu = \mathbb{E}[X] = np$*. Then,* $\mathsf{Pr}[X \geq k] \leq \mu^k/k!$*.*

Proof: This probability is at most $\binom{n}{k}p^k \leq (np)^k/k!$. ∎

III. Pseudorandom Generators from Shrinkage

We now describe our main construction which allows us to use classical lowerbound arguments based on random restrictions to get pseudorandom generators (PRGs). Our main result will apply to any class of functions with nontrivial "shrinkage exponent". We next define this central notion.

Definition III.1. *Let* $\mathcal{F}$ *be a class of functions with an associated size function* $s : \mathcal{F} \to \mathbb{R}_+$ *and let* $\mathcal{D}$ *be a* p*-regular distribution on* $\{0,1,*\}^n$*. We say* $\mathcal{F}$ *has shrinkage exponent* Γ *with respect to* $\mathcal{D}$ *if for all* $f \in \mathcal{F}$*,*

$$\mathop{\mathbb{E}}_{\rho\leftarrow\mathcal{D}}[s(f_\rho)] = O\left(p^\Gamma \cdot s(f) + 1\right).$$

We say $\mathcal{F}$ *has* ε*-shrinkage exponent* Γ *w.r.t* $\mathcal{D}$ *if, there exists a constant* c *such that for all* $f \in \mathcal{F}$*,*

$$\mathop{\mathsf{Pr}}_{\rho\leftarrow\mathcal{D}}\left[\, s(f_\rho) > c\,(p^\Gamma s(f) + 1)\cdot\log(1/\varepsilon)\,\right] \leq \varepsilon.$$

The shrinkage exponent is a classical concept in complexity theory with its origins going back to the very first lowerbounds of Subbotovskaya (1961) [17]. The best lowerbounds we know for several important complexity classes such as read-once formulas, de Morgan formulas are based on estimating the shrinkage exponent of the associated class. This connection can be summarized in the following informal statement:

Theorem III.2 ([18])**.** *If a class* $\mathcal{F}$ *has shrinkage exponent* Γ*, then there is an explicit Boolean function* $h : \{0,1\}^n \to \{0,1\}$ *that cannot be computed by functions in* $\mathcal{F}$ *of size at most* $n^{\Gamma+1}/\mathrm{poly}(\log n)$*.*

Our main result shows that with some additional guarantees on the behavior of $\mathcal{F}$ under random restrictions, one can actually get very strong average-case lowerbounds, PRGs, for $\mathcal{F}$. Our construction and its analysis are quite different from that of Andreev, and give the first pseudorandom generators with $o(n)$ seed-length for several well-studied classes of functions like read-once formulas, de Morgan formulas, branching programs of linear size.

Theorem III.3. *Fix* $\varepsilon > 0$ *and let* $\mathcal{F} : \{0,1\}^n \to \{0,1\}$ *be a class of functions with an associated size function* $s : \mathcal{F} \to \mathbb{N}$*. Fix* $s > 0$ *and let* $p = 1/s^{\Gamma+1}$*. Let* $\mathcal{D}_p$ *be a* p*-regular distribution on* $\{0,1,*\}^n$ *such that* $\mathcal{F}$ *has* ε*-shrinkage exponent* Γ *w.r.t* $\mathcal{D}$*. Then, there exists an explicit pseudorandom generator* $G : \{0,1\}^r \to \{0,1\}^n$ *that* δ*-fools all functions of size at most* s *in* $\mathcal{F}$ *for* $\delta = O(\varepsilon\cdot r)$ *and has seed-length*

$$r = O\left((R(s) + \log(s/\varepsilon))\cdot\log(n/\varepsilon)\cdot s^{1/(\Gamma+1)}\right),$$

where $R(s)$ *denotes the number of bits needed to efficiently sample from* $\mathcal{D}_p$*.*

Proof of Theorem III.3: Here is a high-level description of the generator. We use the restriction family $\mathcal{D}_p$ to sample t restrictions $\rho_1,\ldots,\rho_t$ so that together, the set of active variables in them covers $[n]$ (with high probability). We next have to choose the assignments for the active $(*)$ variables in the restrictions. Instead of choosing these variables independently (which would lead to no gains in the number of bits used), we use a single string X and independent seeds $Y_1,\ldots,Y_t$ (which are much shorter) to set the values for the unassigned variables in the restrictions according to $G_k(E(X,Y_1)), G_k(E(X,Y_2)),\ldots,G_k(E(X,Y_t))$, where $E(\ )$ is an explicit extractor as given by Theorem II.2.

Fix $p = 1/s^{1/(1+\Gamma)}$ and $t = \lceil \log(n/\varepsilon)/p \rceil$. For k to be chosen later, let $G_k : \{0,1\}^{r_k} \to \{0,1\}^n$ be an explicit function generating a k-wise independent distribution on $\{0,1\}^n$. Let $N \geq 4r_k$ and let $E : \{0,1\}^N \times \{0,1\}^d \to \{0,1\}^{r_k}$ be an explicit extractor that works for entropy rate at least $1/2$ sources with $d = O(\log(N/\varepsilon))$ and error at most ε as given by Theorem II.2.

We now describe our PRG by giving a randomized algorithm to compute the output of the generator.

1) Sample t independent restrictions $\rho_1,\rho_2,\ldots,\rho_t$ from $\mathcal{D}_p$.
2) Sample $X \in_u \{0,1\}^N$ and $Y_1,\ldots,Y_t \in_u \{0,1\}^d$ independently.
3) For $1 \leq i \leq t$, let $Z_i = G_k(E(X,Y_i))$.
4) For $1 \leq i \leq t$, define $V_i \in \{0,1\}^n$ by $(V_i)_j = (Z_i)_j$ if $j \in A(\rho_i)$ and $(\rho_i)_j$ otherwise.
5) Output $V \equiv G(\rho_1,\ldots,\rho_t,X,Y_1,\ldots,Y_t) = V_1 \oplus V_2 \oplus \cdots \oplus V_t$, where $\oplus$ denotes bit-wise xor.

We will show that for $N = \tilde{O}(p^\Gamma s)$ sufficiently large, functions in $\mathcal{F}$ of size at most s cannot distinguish V from a truly random string. We will do so by a *hybrid argument*. To this end, let Z_i' be independent uniformly random strings in $\{0,1\}^n$ and with $\rho_1,\ldots,\rho_t$ as in the definition of V, define

$U_i \in \{0,1\}^n$, $1 \le i \le t$, by $(U_i)_j = (Z'_i)_j$ if $j \in A(\rho_i)$ and $(\rho_i)_j$ otherwise.

For $0 \le i \le t$, let $W_i = U_1 \oplus \cdots \oplus U_i \oplus V_{i+1} \oplus V_{i+2} \oplus \cdots \oplus V_n$. Then, $W_0 \equiv V$ and $W_t = U_1 \oplus \cdots \oplus U_t$. We first observe that W_t is ε-close to the uniform distribution on $\{0,1\}^n$.

Claim III.4. *The distribution of W_t is ε-close to the uniform distribution on $\{0,1\}^n$.*

Proof: Observe that if $\cup_{i=1}^t A(\rho_i) = [n]$, then W_t is exactly the uniform distribution on $\{0,1\}^n$. Thus, it suffices to bound the probability that $\cup_{i=1}^t A(\rho_i) \subsetneq [n]$. Now, as $\mathcal{D}_p$ is p-regular, for every $i \in [t]$, $j \in [n]$, $\mathsf{Pr}[j \in A(\rho_i)] = p$ and these events are independent for different $i \in [t]$. Thus, $\mathsf{Pr}[j \notin \cup_{i=1}^t A(\rho_i)] = (1-p)^t \le \varepsilon/n$. The claim now follows by a union bound. ∎

From the above claim, it suffices to show that $\mathcal{F}$ cannot distinguish W_0 from W_t. Fix a $f \in \mathcal{F}$ with $s(f) \le s$ and $i \ge 1$. We will show that f cannot distinguish between W_{i-1} and W_i.

Claim III.5. *For $i \ge 1$, $|\mathsf{Pr}[f(W_{i-1}) = 1] - \mathsf{Pr}[f(W_i) = 1]| \le 5\varepsilon$.*

Proof: Let $W = U_1 \oplus \cdots \oplus U_{i-1} \oplus V_{i+1} \oplus \cdots \oplus V_n$. Then, $W_{i-1} \equiv W \oplus V_i$ and $W_i = W \oplus U_i$. The intuition behind our argument is as follows. Note that W does not depend on ρ_i, Y_i. Let $f_W : \{0,1\}^n \to \{0,1\}$ be given as $f_W(x) = f(x \oplus W)$. Now, by our assumption about the shrinkage exponent of $\mathcal{F}$, for any fixing of W, with high probability over the choice of ρ_i, $s(f_W\lceil_{\rho_i}) \le cp^\Gamma s \log(1/\varepsilon) = s_0$. Let $\mathcal{E}$ denote this event. Observe that conditioned on $\mathcal{E}$, the restricted function $f_W\lceil_{\rho_i}$ can be described with roughly $O(s_0 \log s_0)$ bits (as it has size at most s_0). We then argue that under conditioning on $\mathcal{E}$ (which is independent of Y_i), X has min-entropy at least $N - O(s_0 \log s_0)$ even when given the function $g = f_W\lceil_{\rho_i}$. Therefore, for N sufficiently large, $E(X, Y_i)$ is ε-close to a uniformly random string on $\{0,1\}^{r_k}$ so that $G_k(E(X,Y_i))$ is ε-close to a k-wise independent distribution on $\{0,1\}^n$. Finally, as g depends only on at most $s(f_W\lceil_{\rho_i}) \le s_0$ variables, we get that g cannot distinguish V_i from a truly random string if $k \ge s_0$. We now formalize this intuition.

For brevity, let H denote the random variable $f_W\lceil_{\rho_i} : \{0,1\}^{A(\rho_i)} \to \{0,1\}$. Observe that H is independent of Y_i and the domain of H is independent of X. We abstract the two essential properties of the random restriction family $\mathcal{D}_p$ that we shall exploit.

Fact III.6. *With probability at least $1 - \varepsilon$, $s(H) \le s_0$. In particular, with probability at least $1 - \varepsilon$: (1) H can be described by $O(s_0 \log s_0)$ bits and (2) H is fooled by s_0-wise independent distributions.*

Proof: By our assumption about $\mathcal{D}_p$, for every W, $\mathsf{Pr}_{\rho_i}[s(f_W\lceil_{\rho_i}) \ge s_0] < \varepsilon$. The claim now follows as the number of functions in $\mathcal{F}$ of size at most s_0 is $s_0^{O(s_0)}$ and any function of size s_0 has at most s_0 relevant variables. ∎

Let $\mathcal{F}'_i$ denote the set of all possible values for H obtained under arbitrary settings of W and ρ_i. Let $\mathcal{F}_i = \{g \in \mathcal{F}'_i : \mathsf{Pr}_{W,\rho_i}[H = g] \ge \varepsilon / s_0^{cs_0}\}$. Let $\mathcal{E}'$ denote the event that the conclusions of Fact III.6 hold and let $\mathcal{E}$ be the event that $H \in \mathcal{F}_i$ and $\mathcal{E}'$. Note that conditioned on $\mathcal{E}'$, the number of possibilities for H is at most $s_0^{O(s_0)}$ as H is described completely by $O(s_0 \log s_0)$ bits. Therefore,

$$1 - \mathsf{Pr}[\mathcal{E}] \le \mathsf{Pr}[\neg\mathcal{E}'] + \mathsf{Pr}[(H \notin \mathcal{F}_i) \wedge \mathcal{E}'] \le \varepsilon + s_0^{O(s_0)} \cdot \varepsilon / s_0^{cs_0} \le 2\varepsilon, \quad \text{(III.1)}$$

for c a sufficiently large constant.

In the remaining argument, we condition on the event $\mathcal{E}$. From the above equation, this will only effect our error estimate by an additive 3ε. Fix an element $g \in \mathcal{F}_i$. Then, as the random function H equals g with probability at least $\varepsilon / s_0^{cs_0}$, conditioning on this event cannot change the min-entropy of X much:

$$H_\infty(X | H = g) \ge N - \log(1/\varepsilon) - cs_0 \log s_0.$$

Recall that H is independent of Y_i. Thus, by the definition of the extractor, for $N \ge 2cs_0 \log s_0 + 2\log(1/\varepsilon)$, $E(X, Y_i)$ is ε-close to the uniform distribution on $\{0,1\}^{r_k}$ even conditioned on $H = g$. In particular, $Z_i = G_k(E(X,Y_i))$ is ε-close to a k-wise independent distribution. Therefore, even conditioned on $H = g$, $(V_i)_{A(\rho_i)} = (Z_i)_{A(\rho_i)}$ is ε-close to k-wise independent. Finally, note that $f(W_{i-1}) = (f_W\lceil_{\rho_i})((V_i)_{\rho_i}) = H((V_i)_{\rho_i})$ and similarly, $f(W_i) = H((U_i)_{\rho_i})$. Thus, for $k \ge s_0$,

$$\begin{aligned}
\mathbb{E}[f(W_{i-1}) \,|\, \mathcal{E}, H = g] &= \mathbb{E}\left[(f_W)_{\rho_i}((V_i)_{\rho_i}) \,|\, \mathcal{E}, H = g\right] \\
&= \mathbb{E}\left[g((V_i)_{A(\rho_i)}) \,|\, \mathcal{E}, H = g\right] \\
&= \mathbb{E}\left[g((U_i)_{A(\rho_i)}) \,|\, \mathcal{E}, H = g\right] \pm \varepsilon \qquad \text{(Fact III.6)} \\
&= \mathbb{E}[f(W_i) | \mathcal{E}, H = g] \pm \varepsilon.
\end{aligned}$$

As the above is true for every $g \in \mathcal{F}_i$, it follows that

$$|\mathbb{E}[f(W_{i-1}) - f(W_i) \,|\, \mathcal{E}]| \le \varepsilon.$$

Combining Equation III.1 and the above equation, we get

$$\begin{aligned}
|\,\mathbb{E}[f(W_{i-1})] - \mathbb{E}[f(W_i)]| = \\
\mathsf{Pr}[\mathcal{E}] \cdot |\mathbb{E}[f(W_{i-1}) - f(W_i) \,|\, \mathcal{E}]| + \\
\mathsf{Pr}[\neg\mathcal{E}] \cdot |\mathbb{E}[f(W_{i-1}) - f(W_i) \,|\, \neg\mathcal{E}]| \\
\le \varepsilon + 2\,\mathsf{Pr}[\neg\mathcal{E}] \le 5\varepsilon.
\end{aligned}$$

The claim now follows. ∎

Combining Claims III.4, III.5 and summing from $1 \le i \le t$, we get that

$$|\,\mathbb{E}[f(V)] - \mathop{\mathbb{E}}_{U \in_u \{0,1\}^n}[f(U)]| \le 5\varepsilon t + \varepsilon.$$

Let us now estimate the seed-length of the generator. To generate V we need to sample $\rho_1, \ldots, \rho_t$ and $X, Y_1, \ldots, Y_t$ for a total of

$$r = (R(s)+d)t + N = O\left(R(s) + \log(s_0/\varepsilon)\right) \cdot (\log(n/\varepsilon))/p + O(s_0 \log s_0).$$

Substituting $s_0 = cp^\Gamma s\log(1/\varepsilon)$, in the above equation gives us the theorem. The above calculation explains our choice of $p = 1/s^{(\Gamma+1)}$: we want to balance out $1/p$ and $p^\Gamma s$. ∎

We next use Theorem III.3 to get PRGs for specific classes of functions.

IV. PRGs for Formulas

A formula is a tree where each leaf is labeled by a literal (either a variable or its negation) and each internal node by an operation of constant arity. Any such tree naturally defines a function f. Let $L(f)$ denote the formula size (number of leaves) of f. We assume without loss of generality that $L(f) = n$, the number of variables, since we can always add dummy variables otherwise. A de Morgan formula is a binary tree with the set of allowed operations being $\{\wedge, \vee\}$.

The simplest case for our arguments is formulas over an arbitrary basis, since these have shrinkage 1. More challenging are de Morgan formulas. It has been known for many years that shrinkage for general such formulas is 2 [16] and for read-once formulas (no variable appears more than once) is $\log 2/\log(\sqrt{5}-1) = 3.27...$ [22]. In this section, we show that even pseudorandom restrictions using $n^{o(1)}$ random bits achieve essentially the same shrinkage with high probability. This will be shown in Lemmas IV.2, IV.8. We then use Theorem III.3 to get Theorems I.2, I.1, I.4.

In our arguments, we will often have to handle "heavy" variables – variables that appear in many leaves. The following lemma shows that any s variables can only increase the formula size by a factor of about 2^s.

Lemma IV.1. *Let f be any formula, and let H denote any subset of the variables. For each $h \in \{0,1\}^H$, let ρ_h denote the restriction formed by setting variables in H to h, leaving all other variables unfixed. Then $L(f) \leq \sum_{h\in H}(L(f\lceil_{\rho_h}) + |H|) \leq 2^{|H|}\left(\max_{h\in H} L(f\lceil_{\rho_h}) + |H|\right)$.*

Proof: Let $x(H) = h$ denote the formula of size $|H|$ which is true iff all variables in H are set to h. Then $f = \bigvee_{h\in\{0,1\}^H}((x(H) = h) \wedge (f\lceil_{\rho_h}))$. ∎

We begin with formulas over an arbitrary basis.

A. Arbitrary Basis

Here Lemma IV.1 and concentration bounds imply shrinkage with $\Gamma = 1$ for $O(\log n)$-wise independent restrictions.

Lemma IV.2. *For any constant c and formula f with $L(f) = n$, a $(p = 1/\sqrt{n})$-regular $c\log n$-wise independent random restriction ρ yields*

$$\Pr[L(f\lceil_\rho) \geq 2^{3\sqrt{c\log n}}pn] \leq 2n^{-c}.$$

Proof: Let $k = c\log n$. The formula f depends on at most n variables $x_1, \ldots, x_n$. Let variable x_i appear as a leaf n_i times, so $n = L(f) = \sum_i n_i$. For α to be chosen later, call x_i *heavy* if $n_i \geq p^{1-\alpha}n$ and *light* otherwise. Then for H the set of heavy variables, $|H| \leq p^{\alpha-1}$. Let $H(\rho)$ denote the heavy variables set to * by ρ, and $h(\rho) = |H(\rho)|$. Define a new restriction ρ' with $\rho'(x) = \rho(x)$ for $x \notin H(\rho)$, and adversarially chosen in $\{0,1\}$ otherwise. Lemma IV.1 implies that $L(f\lceil_\rho) \leq 2^{1+h(\rho)}L(f\lceil_{\rho'})$. We bound

$$\Pr[L(f\lceil_\rho) \geq 2^{h+3}k\cdot p^{1-\alpha}n] \leq \Pr[h(\rho) \geq h] + \Pr[L(f\lceil_{\rho'}) \geq 4k\,p^{1-\alpha}n].$$

Define random variables $X_i = 1$ if $\rho(x_i) = *$, and 0 otherwise. We bound the first term with Lemma II.4. Here, we have $\mu = |H|p \leq p^\alpha$. Thus as long as $h\alpha \geq 2c$ this term will contribute at most n^{-c}.

For the second term, we need only consider light variables L and apply Lemma II.3. Now the coefficients are the n_i. Note that $\mu \leq pn$ and $m = \max_{x_i\in L} n_i < p^{\alpha-1}n$, so $m + \mu \leq 2p^{1-\alpha}n$. Hence Lemma II.3 bounds the second term by $2^{-k} \leq n^{-c}$.

Setting $h = \lceil 2c/\alpha\rceil$, we minimize $2^h(1/p)^\alpha$ by setting $\alpha = 2\sqrt{c/\log n}$. ∎

Combining this with Theorem III.3 with $\Gamma = 1$, yields Theorem I.2.

B. de Morgan Basis

We follow the high-level intuition described in the introduction. One subtle issue we face in carrying out the approach is that the subformulas g_i in our decomposition will have some overlapping nodes, which in turn forces some additional constraints on these nodes. We next show that these additional constraints can be removed with only a minor loss. Throughout this section we assume that Γ denotes the shrinkage exponent for the class of formulas under consideration — $\Gamma = 2$ for general formulas and $\Gamma = 3.27...$ for read-once formulas.

Lemma IV.3. *For any positive ℓ and any formula f on a set of variables X with $L(f) \geq \ell$, there exists at most $6n/\ell$ formulas g_i with $L(g_i) \leq \ell$, where the g_i may depend on variables outside X, such that the following holds. For any restriction ρ, $L(f\lceil_\rho) \leq \sum_i L(g_i\lceil_{\rho'})$, where $\rho'(x) = \rho(x)$ for $x \in X$, and $\rho'(x) = *$ otherwise. Moreover, each g_i depends on at most 2 variables outside X (called* special variables*).*

This follows from the following claim.

Claim IV.4. *Any binary tree of size $n \geq \ell$ can be decomposed into at most $6n/\ell$ subtrees of size at most ℓ, such that each subtree has at most two other subtree children. Here subtree T_1 is a child of subtree T_2 if there exists nodes $t_1 \in T_1$, $t_2 \in T_2$, such that t_1 is a child of t_2.*

Proof: Proceed inductively, using the well-known fact that any binary tree of size s can be divided into two edge-disjoint subtrees, each of size between $s/3$ and $2s/3$. This results in subtrees of size between $\ell/3$ and ℓ, and hence there are at most $3n/\ell$ of them. For each subtree T with more than two subtree children, find a subtree T' of T with exactly two subtree children, and divide T into T' and $T \setminus T'$. Note that $T \setminus T'$ now has one fewer subtree children. Continue doing this until all subtrees have at most two subtree children. This process can continue at most the original number of subtrees

steps, and hence the total number of such subtrees is as desired. ∎

Proof of Lemma IV.3: View the formula f as a tree. By Claim IV.4, we can decompose f into subformulas g_i, where each input to g_i is either an ordinary variable in X or a special input: the output of some other g_j. In each g_i, replace these special inputs with distinct, new variables not in X. The total number of new variables is at most the number of subformulas. ∎

We'd now like to show that restricting by ρ' is not much worse than restricting by ρ, i.e., requiring a few variables to be * does not hurt the restricted size too much. We want to show this simply using results about restrictions by ρ as a black box. For general formulas, this follows from Lemma IV.1. However, for read-once formulas we need a different method. This method involves replacing these special variables by relatively short formulas which are unlikely to get fixed. We show that such read-once formulas exist using a result of Valiant [29] on computing the majority function by monotone formulae.

Lemma IV.5. *For any $0 < p, \varepsilon < 1$, there exists a read-once formula h of size at most $(\log(1/\varepsilon)/p)^4$ such that a p-regular (truly) random restriction fixes h with probability less than ε.*

Proof: We shall use Valiant's result on computing the majority function using monotone formulas [29]. His main result is a probabilistic way to construct monotone formulas for majority. However, the formulas he constructs come from a distribution on read once formulas of size $\mathrm{poly}(1/p)$ so that, if the inputs have bias $1/2 + p$ (of being 1), they almost always output 1, and if they have bias $1/2 - p$, they almost always output 0. He then boosts the error probability to be exponentially small in n. We don't need to do that last step.

The point is that if a monotone formula has the above property, then it is resistant to restrictions leaving p fraction of bits unrestricted. Because, if we go back and set the unset bits to 1, we get random bits biased towards 1 as inputs, and if we set them to 0, we get random inputs biased towards 0. Since the output has to change with high probability, the circuit cannot be constant after the restriction.

The precise bound one gets from Valiant's arguments is $O\left((\log(1/\varepsilon))^2/p^{3.27\cdots}\right) < (\log(1/\varepsilon)/p)^4$. ∎

Lemma IV.6. *Suppose that for all formulas f of size ℓ_0, and a p-regular random restriction ρ, $\mathbb{E}[L(f\lceil_\rho)] \le p^\Gamma L(f)$. Suppose g is a formula with w special variables with $L(g) \ge \ell_0$. Let ρ' be a p-regular restriction with the constraint that the special variables in g be unrestricted. Then,*

$$\mathbb{E}\left[L(g\lceil_{\rho'})\right] \le p^\Gamma L(g) + O\left(w \cdot p^{\Gamma-4} \cdot \log^4\left(w\, p^\Gamma\, L(g)\right)\right).$$

Proof: Construct a new formula g' by replacing each special variable in g by the formula h given in Lemma IV.5 for $\varepsilon = 1/w\, p^\Gamma L(g)$, on disjoint sets of variables. Let A denote the event that none of these formulas h is fixed. The key observation is that conditioned on A, we have $L(g'\lceil_\rho) \ge L(g\lceil_{\rho'})$. Therefore,

$$\mathbb{E}\left[L(g'\lceil_\rho)\right] \ge \Pr[A]\cdot\mathbb{E}\left[L(g'\lceil_\rho) \mid A\right] \ge (1-w\varepsilon)\,\mathbb{E}\left[L(g\lceil_{\rho'}) \mid A\right].$$

Thus, for $r = (\log(1/\varepsilon)/p)^4$,

$$\mathbb{E}[L(g\lceil_{\rho'})|A] \le (1+2w\varepsilon)\,\mathbb{E}[L(g'\lceil_\rho)] \le (1+2w\varepsilon)\left(p^\Gamma(L(g)+2\,w\,r)\right).$$

The lemma follows. ∎

We next show that k-wise independent restrictions shrink formulas in which no variable is read too many times with high probability.

Lemma IV.7. *There is a large enough constant c, such that for any $\varepsilon > 0$, any $p \ge n^{-1/(4\Gamma)}$, any $t \le p^{9\Gamma} n/(c\log(n/\varepsilon))$, and any read-$t$ formula f on n variables with $L(f) = n$, a p-regular $\left(k = c\log(n/\varepsilon)/p^\Gamma\right)$-wise independent restriction ρ yields $\Pr\left[L(f\lceil_\rho) \ge 60\, p^\Gamma n\right] \le \varepsilon$.*

Proof: Set $\ell = 1/p^{4\Gamma}$. Use Lemma IV.3 to get the formulas g_i. By Lemma IV.1 and Lemma IV.6, for any g_i, we have $\mathbb{E}[L(g_i\lceil_{\rho'}] \le 5\, p^\Gamma \ell$.

Form a graph where the vertices are the g_i, with an edge between g_i and g_j if they share a variable. This graph has m vertices, where $n/\ell \le m \le 6n/\ell$, and degree at most ℓt. Hence we can divide the vertices into independent sets of size at least $s = \lfloor m/(\ell t + 1)\rfloor \ge c\log(n/\varepsilon)/(2p^\Gamma)$.

For any such independent set I, note that $Y_i = L(g_i\lceil_{\rho'})/\ell$ are independent random variables in the range $[0,1]$. Hence we can apply Lemma II.3 for large enough c to show that

$$\Pr\left[\sum_{i\in I} L(g_i\lceil_{\rho'}) \ge 2\,\mathbb{E}\left[\sum_{i\in I} L(g_i\lceil_{\rho'})\right]\right] \le \varepsilon/n.$$

Thus, by a union bound, with probability at least $1-\varepsilon$ no such event occurs. The lemma follows because

$$\sum_I \mathbb{E}\left[\sum_{i\in I} L(g_i\lceil_{\rho'})\right] \le 5p^\Gamma\ell \cdot \sum_I |I| = 5p^\Gamma \ell m \le 30\, p^\Gamma n.$$

∎

We now remove the assumption that the formula is read t, leading to our final derandomized shrinkage bound.

Lemma IV.8. *For any constant $c \ge 11$, any $p \ge n^{-1/\Gamma}$, and any formula f on n variables with $L(f) = n$, there is a a p-regular distribution on restrictions ρ using a $2^{O(\log^{2/3} n)}$ bit seed such that*

$$\Pr\left[L(f\lceil_\rho) \ge 2^{3c\log^{2/3} n} \cdot p^\Gamma n\right] \le n^{-c}.$$

Before proving the lemma we first note that combining the lemma with Theorem III.3, and the shrinkage exponent estimates of [16], [22] directly implies Theorem I.1 and Theorem I.4 respectively.

We now prove Lemma IV.8. We will implement this p-regular restriction as a sequence of r q-regular restrictions, where $p = q^r$ and $q = n^{-\alpha}$ for some α only slightly sub-constant. For each of the r rounds of restriction, we will have a set of at most $n^{6\alpha}$ heavy variables, which can change in each round. We handle the heavy variables by conditioning on the values of the restrictions applied to the heavy variables for the current round and 6 rounds ahead, and then applying

Lemma II.4. We handle all other variables with Lemma IV.7. Note that the shrinkage exponents proved in [16], [22] have an extra polylogarithmic term. However, the extra factor when restricting by $q = n^{-\alpha}$ is $\mathrm{polylog}(n)$, so the total extra factor is $(\mathrm{polylog}(n))^r$, which can be absorbed into the $2^{O(\log^{2/3} n)}$ term. We now formalize this.

Proof: Set $q = p^{1/r}$ for an $r \geq 11$ such that $q = n^{-\alpha}$ for α to be chosen later. Let $k_0 = n^{6\alpha}$, and $k = rk_0$. Our pseudorandom p-regular restriction will be the composition of r independent restrictions ρ_i, where each ρ_i is a k-wise independent q-regular restriction.

The analysis proceeds in rounds. Let $X_0 = X$ denote the n variables for f. Let X_i denote the unfixed variables in round i, and let $n_i = |X_i|$. Let $f_0 = f$, and let f_i denote the restricted formula after i rounds. Call a variable x_j *heavy* in round i if x_j appears more than $t_i = L(f_i)/n^{10\alpha}$ times in f_i. Letting H_i denote the heavy variables in round i, we see that $|H_i| \leq n^{10\alpha}$. Let $H = \cup H_i$, and $Y_i = X_i \setminus H$. Let $p_i = p/q^i$.

We now condition on the heavy variables in a somewhat subtle way. In round 1, we condition on all of ρ_1, as well as the values of all ρ_i on the variables H_1. Now all the ρ_i on H_1 determine ρ on H_1. Since all ρ_i are k-wise independent, so is ρ. Since $k \geq |H_1|$, Lemma II.4 implies that

$$\Pr[\,\rho \text{ leaves at least } s \text{ variables from } H_1 \text{ unfixed}\,] \leq \left(n^{10\alpha} p\right)^s \leq n^{-\alpha s}. \quad \text{(IV.1)}$$

Now, conditioned on all ρ_i on H_1, ρ_1 remains k_0-wise independent on $X \setminus H_1$. Suppose ρ leaves $s_1 < s$ variables unfixed from H_1. Then for each setting τ of these s_1 variables, Lemma IV.7 implies that

$$\Pr\left[L\left(f\restriction_{\rho_1 \cup \rho(H) \cup \tau}\right) \geq 60\, q^\Gamma n\right] \leq \exp(n^{-\Omega(\alpha)}). \quad \text{(IV.2)}$$

Combining (IV.1) and (IV.2) with Lemma IV.1, we obtain

$$\Pr\left[L\left(f\restriction_{\rho_1 \cup \rho(H)}\right) \geq 2^{s+6} q^\Gamma n\right] \leq 2n^{-\alpha s}.$$

We continue in this manner, in round i fixing ρ_i as well as all ρ_j, $j \geq i$, on H_i. We do this for $r - 11$ rounds, as in (IV.1) the p becomes p_i and we need to ensure that $n^{10\alpha} p_i \leq n^{-\alpha}$. Thus,

$$\Pr[L(f\restriction_\rho) \geq (2^{s+6} q^\Gamma)^{r-11} \cdot n] \leq 2(r-11) n^{-\alpha s}.$$

Since $q^{\Gamma r} = p^\Gamma$, the extra factor we lose in the formula size beyond p^Γ is at most $2^{(s+6)(r-11)} n^{11\alpha}$. To make the error at most n^{-c}, we set $s = 2c/\alpha$. Since $p \geq 1/n$, we have $r \leq 1/\alpha$. Thus the extra factor is at most $2^{rs} n^{11\alpha} = 2^{2c/\alpha^2 + 11\alpha \log n}$. To minimize this exponent (up to constants), we set $\alpha = (\log n)^{-1/3}$. We restrict $c \geq 11$ in the lemma statement so that $2c + 11 \leq 3c$. ∎

V. PRGs for Branching Programs

We now apply our main generator construction to get PRGs for branching programs to get a PRG with seed-length $s^{1/2+o(1)}$ for branching programs of size s. We first formally define branching programs.

Definition V.1. *An n-variable branching program (BP) M is a directed acyclic graph with the following properties:*

- *There are three special vertices—start which has no incoming edges and two terminal vertices accept, reject which have no outgoing edges.*
- *Every vertex in the graph is labeled with a variable from $\{x_1, \ldots, x_n\}$.*
- *Every non-terminal vertex has two outgoing edges labeled $\{0, 1\}$.*

The size of M, $s(M)$, is defined as the number of vertices in M. The length of M is defined as the length of the longest path from start to either of the terminals. We say M is read-once if no two vertices in a path from start to the terminals have the same label.

A branching program M as above, naturally induces a function $M : \{0,1\}^n \to \{0,1\}$ that on input $x \in \{0,1\}^n$, traverses the graph according to x and outputs 1 if the computation reaches accept and 0 otherwise.

We shall construct an explicit pseudorandom generator for branching programs of size at most s with seed-length $s^{1/2+o(1)}$ and error $1/\mathrm{poly}(s)$. Previously, only PRGs with seed-length $\Omega(n)$ were known even for the special case of *layered* read-once branching programs[2] (ROBPs) with each layer constrained to have a constant number of nodes (constant *width*). For the more restricted class of *oblivious* ROBPs (these are ROBPs where the labeling of the layers is known apriori) of length at most T and width at most W, Nisan [24] and Impagliazzo, Nisan and Wigderson [23] gave PRGs with seed-length $O((\log T)(\log(T/\varepsilon) + \log W))$ to get error ε.

We now prove Theorem I.3. The arguments here are basically the same as those from Section IV-A. To this end, we first show an analogue of Lemma IV.1 for branching programs.

Lemma V.2. *Let f be any BP, and let H denote any subset of the variables. For each $h \in \{0,1\}^H$, let ρ_h denote the restriction formed by setting variables in H to h, leaving all other variables unfixed. Then $s(f) \leq 2^{|H|} \cdot (\max_{h \in H} s(f\restriction_{\rho_h}) + 2)$.*

Proof: Build a complete decision tree T of depth $|H|$ so that leaves correspond to specific assignments for the variables in h. We can now obtain a BP M_f for f, by appending a BP for $f\restriction_{\rho_h}$ to the leaf of T corresponding to the assignment h. Clearly, the resulting BP has size as stated. ∎

Lemma V.3. *For any constant c and BP f with $s(f) = n$, a $(p = 1/\sqrt{s})$-regular $c \log s$-wise independent random restriction ρ yields*

$$\Pr\left[s(f\restriction_\rho) \geq 2^{3\sqrt{c \log n}} \cdot p\, s\right] \leq 2\, s^{-c}.$$

Proof: Let $k = c \log s$. The BP f on at most s variables $x_1, \ldots, x_s$. For $i \leq s$, let the number of vertices labeled x_i be n_i. Then, $s = s(f) = \sum_i n_i$. We now repeat the calculations from Lemma IV.2 for this setting of n_i's.

[2]Meaning, the vertices can be partitioned into *layers* so that edges always go between consecutive layers and all the vertices in a layer are labeled with the same variable.

For α to be chosen later, call x_i *heavy* if $n_i \geq p^{1-\alpha}s$ and *light* otherwise. Then for H the set of heavy variables, $|H| \leq p^{\alpha-1}$. Let $H(\rho)$ denote the heavy variables set to * by ρ, and $h(\rho) = |H(\rho)|$. Define a new restriction ρ' with $\rho'(x) = \rho(x)$ for $x \notin H(\rho)$, and adversarially chosen in $\{0,1\}$ otherwise. Lemma IV.1 implies that $s(f\lceil_\rho) \leq 2^{h(\rho)+1}s(f\lceil_{\rho'})$. We bound

$$\Pr\left[s(f\lceil_\rho) \geq 2^{h+3}k \cdot p^{1-\alpha}s\right] \leq \Pr[h(\rho) \geq h] + \Pr\left[s(f\lceil_{\rho'}) \geq 4k\, p^{1-\alpha}s\right].$$

Define random variables $X_i = 1$ if $\rho(x_i) = *$, and 0 otherwise. We bound the first term with Lemma II.4. Here, we have $\mu = |H|p \leq p^\alpha$. Thus as long as $h\alpha \geq 2c$ this term will contribute at most s^{-c}.

For the second term, we need only consider light variables L and apply Lemma II.3. Now the coefficients are the n_i. Note that $\mu \leq ps$ and $m = \max_{x_i \in L} n_i < p^{\alpha-1}n$, so $m + \mu \leq 2p^{1-\alpha}s$. Hence Lemma II.3 bounds the second term by $2^{-k} \leq s^{-c}$.

Setting $h = \lceil 2c/\alpha \rceil$, we minimize $2^h(1/p)^\alpha$ by setting $\alpha = 2\sqrt{c/\log s}$. ∎

Combining this with Theorem III.3 with $\Gamma = 1$, yields Theorem I.3.

ACKNOWLEDGMENTS

We are grateful to Avi Wigderson for useful discussions and suggestions.

REFERENCES

[1] A. Bogdanov, P. A. Papakonstantinou, and A. Wan, "Pseudorandomness for read-once formulas," in *FOCS*, 2011, pp. 240–246.

[2] M. Blum and S. Micali, "How to generate cryptographically strong sequences of pseudo-random bits," *SIAM J. Comput.*, vol. 13, no. 4, pp. 850–864, 1984.

[3] A. C.-C. Yao, "Theory and applications of trapdoor functions (extended abstract)," in *FOCS*, 1982, pp. 80–91.

[4] M. Ajtai and A. Wigderson, "Deterministic simulation of probabilistic constant depth circuits (preliminary version)," in *FOCS*, 1985, pp. 11–19.

[5] N. Nisan and A. Wigderson, "Hardness vs randomness," *J. Comput. Syst. Sci.*, vol. 49, no. 2, pp. 149–167, 1994.

[6] N. Nisan, "Pseudorandom bits for constant depth circuits," *Combinatorica*, vol. 11, no. 1, pp. 63–70, 1991.

[7] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson, "BPP has subexponential time simulations unless EXPTIME has publishable proofs," *Computational Complexity*, vol. 3, pp. 307–318, 1993.

[8] R. Impagliazzo, V. Kabanets, and A. Wigderson, "In search of an easy witness: exponential time vs. probabilistic polynomial time," *J. Comput. Syst. Sci.*, vol. 65, no. 4, pp. 672–694, 2002.

[9] V. Kabanets and R. Impagliazzo, "Derandomizing polynomial identity tests means proving circuit lower bounds," *Computational Complexity*, vol. 13, no. 1-2, pp. 1–46, 2004.

[10] R. Williams, "Improving exhaustive search implies superpolynomial lower bounds," in *STOC*, 2010, pp. 231–240.

[11] D. Gutfreund and S. P. Vadhan, "Limitations of hardness vs. randomness under uniform reductions," in *APPROX-RANDOM*, 2008, pp. 469–482.

[12] R. Shaltiel and E. Viola, "Hardness amplification proofs require majority," in *STOC*, 2008, pp. 589–598.

[13] T. Watson, "Query complexity in errorless hardness amplification," in *APPROX-RANDOM*, 2011, pp. 688–699.

[14] S. Artemenko and R. Shaltiel, "Lower bounds on the query complexity of non-uniform and adaptive reductions showing hardness amplification," in *APPROX-RANDOM*, 2011, pp. 377–388.

[15] N. Nisan and D. Zuckerman, "Randomness is linear in space," *J. Comput. Syst. Sci.*, vol. 52, no. 1, pp. 43–52, 1996.

[16] J. Håstad, "The shrinkage exponent of de Morgan formulas is 2," *SIAM J. Comput.*, vol. 27, no. 1, pp. 48–64, 1998.

[17] B. A. Subbotovskaya, "Realizations of linear functions by formulas using +, *, -," *Sov. Math. Dokl.*, vol. 2, pp. 110–112, 1961.

[18] A. Andreev, "On a method for obtaining more than quadratic effective lower bounds for the complexity of π-schemes." *Moscow Univ. Math. Bull.*, vol. 42, no. 1, pp. 63–66, 1987.

[19] R. Impagliazzo and N. Nisan, "The effect of random restrictions on formula size," *Random Struct. Algorithms*, vol. 4, no. 2, pp. 121–134, 1993.

[20] M. Paterson and U. Zwick, "Shrinkage of de Morgan formulae under restriction," *Random Struct. Algorithms*, vol. 4, no. 2, pp. 135–150, 1993.

[21] V. Khrapchenko, "Complexity of the realization of a linear function in the class of π-circuits," *Math. Notes Acad. Sciences USSR*, vol. 9, pp. 21–23, 1971.

[22] J. Håstad, A. A. Razborov, and A. C.-C. Yao, "On the shrinkage exponent for read-once formulae," *Theor. Comput. Sci.*, vol. 141, no. 1&2, pp. 269–282, 1995.

[23] R. Impagliazzo, N. Nisan, and A. Wigderson, "Pseudorandomness for network algorithms," in *STOC*, 1994, pp. 356–364.

[24] N. Nisan, "Pseudorandom generators for space-bounded computation," *Combinatorica*, vol. 12, no. 4, pp. 449–461, 1992.

[25] I. Komargodski and R. Raz, "Average-case lower bounds for formula size," *ECCC*, vol. 19, no. 62, 2012.

[26] N. Alon and J. Spencer, *The Probabilistic Method*, ser. Wiley Series in Discrete Mathematics and Optimization. John Wiley & Sons, 2011.

[27] D. Zuckerman, "Randomness-optimal oblivious sampling," *Random Struct. Algorithms*, vol. 11, no. 4, pp. 345–367, 1997.

[28] J. P. Schmidt, A. Siegel, and A. Srinivasan, "Chernoff-Hoeffding bounds for applications with limited independence," *SIAM J. Discrete Math.*, vol. 8, no. 2, pp. 223–250, 1995.

[29] L. G. Valiant, "Short monotone formulae for the majority function," *J. Algorithms*, vol. 5, no. 3, pp. 363–366, 1984.