Typically-Correct Derandomization of BPP

Jan Strzeszynski, Jon Rosario

May 16, 2024

Abstract

In the quest for derandomization, two approaches are explored: Shaltiel's conditional derandomization of **BPP** and Kinne et al.'s conditional derandomization of **BPP**. Both derandomizations are weak in the sense that deterministic simulations of randomized algorithms achieve correctness for most inputs of each length. These derandomization efforts intertwine with circuit lower bounds, offering insights into the complexities involved. Shaltiel's method relies on the construction of extractors for recognizable distributions, leveraging circuit lower bound hypotheses to establish the typically-correct derandomization. Conversely, Kinne et al. introduce seed-extending pseudorandom generators and exploit XOR hardness amplification to construct a typically-correct derandomization algorithm for **BPP**. Both approaches underscore the intricate interplay between randomness, computational complexity, and typically-correct behavior, shedding light on the role of randomness in algorithmic computations. In this paper, we provide a comprehensive summary of these two significant approaches.

1 Introduction

Complexity theorists have long been intrigued by the pursuit of efficient deterministic alternatives to randomized algorithms. While every randomized algorithm inherently possesses a deterministic counterpart, the challenge lies in devising a deterministic approach that replicates the probabilistic success of its randomized counterpart. For instance, given a decision problem and a randomized algorithm A that has high success rates, a straightforward deterministic strategy involves simulating A on all potential pathways and selecting the majority outcome. However, this particular deterministic simulation necessarily comes at the cost of increased runtime complexity. Consequently, this naturally introduces randomness as a fundamental metric of computational complexity, alongside time and space considerations.

At the same time, most researchers have come to the belief that randomness is not nearly as important as time or space is. There is a long line of research that contribute to this idea—specifically the idea that all randomized algorithms have a unrandomized counterpart which is not too much worse in terms of time and space. One of the most important results in this direction was by Nisan and Zuckerman in 1998 [NZ96]:

Theorem 1. *(Nisan-Zuckerman) Suppose* $L \in \mathbf{BPTISP}[poly(S(n)), S(n)]$ *where* $S(n) \geq n$. *Then* $L \in \mathbf{DSPACE}[S(n)]$.

Many researchers believe that the Nisan-Zuckerman can be strengthen. Unfortunately, achieving a stronger derandomization result than the Nisan-Zuckerman Theorem has proven elusive. This roadblock has led some researchers to explore alternative approaches, focusing on weaker forms of derandomization, in hopes of making progress in this area. One such approach has been creating deterministic simulations that only need to be correct on certain inputs. A classic result was given by [BFNW91] in 1993:

Theorem 2. *If* $\mathbf{EXP} \not\subset \mathbf{P}/poly$, *then* $\mathbf{BPP} \subseteq i.o. - \mathbf{SUBEXP}$.

In simpler terms, if we accept a widely believed circuit lower bound, it implies that problems solvable with random polynomial-time algorithms also have deterministic sub-exponential time solutions for infinitely many input lengths.

This leads us to another similar line of research, which is the focus of this paper: "typically-correct derandomization". This concept, introduced by Goldreich and Widgerson in [GW02], allows a deterministic simulation of a randomized algorithm to be wrong on a few inputs of each length.

Such simulations have been of interest for researchers for several reasons. Firstly, there are notable connections between circuit lower bounds and typically-correct derandomizations, similarly to the results given in [BFNW91]. Secondly, it's worth noting that most typically-correct derandomizations hinge on hardness assumptions, suggesting that establishing them unconditionally might pose significant challenges. Lastly, the pursuit of good unconditional typically-correct derandomizations holds promise as they have the potential to distinguish between complexity classes such as **EXP** and **BPP** according to the work done in [IW01]. In this paper, we focus on typically-correct derandomizations for **BPP**.

2 Preliminaries

We now formalize the notion of "typically-correct derandomization" and several related notions. We employ the language used by [KMS10].

2.1 Probability

We denote the random variable representing the uniform distribution over $\{0,1\}^m$ as U_m.

Definition 1. *(min-entropy) Let X be a random variable and $\text{supp}(X)$ be the support of a random variable. That is, the set of values of X that occur with non-zero probability. Then the **min-entropy** of X is*

$$H_\infty(X) = \min_{x \in \text{supp}(X)} \log 1/p(x).$$

It is also not hard to see that we can think of the min-entropy as the largest non-negative real number $k \in \mathcal{R}_{\geq 0}$ such that $P[X = x] \leq 2^{-k}$ for every x in the support of X.

We also need to formalize the meaning of "close to the uniform distribution" in order describe most derandomizations.

Definition 2. *(Variation Distance) For random variables X and Y taking values in $\mathcal{U} = \text{supp}(X) \cup \text{supp}(Y)$, the **variation distance** is $\Delta(X,Y) = \max_{E \subset \mathcal{U}} |P_X[E] - P_Y[E]|$. We say X and Y are ϵ-close if $\Delta(X,Y) \leq \epsilon$.*

2.2 Extractors and PRGs

All the derandomizations that will be described in this paper make use of either *pseudorandom generators* (PRGs) or *randomness extractors*. We describe them briefly here, and then in more detail in later sections.

Definition 3. *Let $\mathcal{X}$ be a family of random variables with $\text{supp}(X) = \{0,1\}^n$ for $X \in \mathcal{X}$. We say a function $Ext : \{0,1\}^n \to \{0,1\}^m$ is a **seedless** (k,ϵ)-**extractor** for $\mathcal{X}$ if for all $X \in \mathcal{X}$, $H_\infty(X) \geq k$ and,*

$$\Delta(Ext(X), U_m) \leq \epsilon,$$

where U_m is the uniform distribution on m bits. We sometimes refer to ϵ as the error rate for the extractor.

Pseudorandom generators can be defined in a variety of ways, but for our purposes a *test based* definition will be most appropriate.

Definition 4. *We say that a function $G : \{0,1\}^n \to \{0,1\}^m$ is ε-pseudorandom for a test $T : \{0,1\}^m \to \{0,1\}$ if*

$$|P_{r \leftarrow U_m}[T(r) = 1] - P_{x \leftarrow U_n}[T(G(x)) = 1]| \leq \varepsilon.$$

2.3 Typically-Correct Behavior

We now describe what we mean by typically-correct derandomization.

Definition 5 (Typically-Correct Behavior)**.** *Let $L : \{0,1\}^n \to \{0,1\}$ be a function (language). We say a function (algorithm) $B : \{0,1\}^n \to \{0,1\}$ computes L within a distance δ if $\forall n$:*

$$P_{x \leftarrow U_n}[B(x) \neq L(x)] \leq \delta.$$

Alternatively, we say that B is within δ of L.

Definition 6 (Class Distance)**.** *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two classes of languages. We say that $\mathcal{C}_1$ is within distance δ of $\mathcal{C}_2$ if for every language $L_1 \in \mathcal{C}_1$, there is a language $L_2 \in \mathcal{C}_2$ that is within δ of L_1.*

3 Shaltiel's conditional derandomization of BPP

We now describe the approach taken by Ronen Shaltiel in [Sha11]. This approach is motivated by a desire to find a explicit form of Yao's Lemma. We state it explicitly:

Lemma 1. *Let $f : \{0,1\}^n \to \{0,1\}$ be a function. Let $A : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$ represent a randomized algorithm computing f with with error p. In other words, for all fixed $n \in \mathbf{N}$, we have that for all $x \in \{0,1\}^n$:*

$$P_{r \leftarrow U_m}[A(x,r) \neq f(x)] \leq \rho.$$

Then, there exists an $r^* \in \{0,1\}^m$ such that:

$$P_{x \leftarrow U_n}[A(x,r^*) \neq f(x)] \leq \rho.$$

Yao's lemma is proved using an averaging argument, meaning that it does not provide the value of r^* without requiring us to do expensive computation. However, it is clear that finding such an r^* would provide an algorithm $B(x) = A(x, r^*)$ computing f within a distance ρ.

Shaltiel's approach at a high level uses a seedless extractor to extract randomness from the input. The paper introduces a new notion of extractor called an extractor for recognizable distributions, and shows that any randomized complexity class $\mathcal{C}$ can be derandomized in the framework of typically-correct derandomization if one can construct an extractor for distributions recognizable by that class $\mathcal{C}$.

3.1 Recognizable Distributions

To maintain clarity, we focus solely on random variables that are uniformly distributed over sets, thereby restricting the definitions in this section to uniform distributions.

Definition 7. *Let $L : \{0,1\}^n \to \{0,1\}$ be a function. Then, the uniform distribution over the set of binary strings where $L(x) = 1$ is denoted as U_L. We say U_L is the **distribution recognized** by L. Moreover, for a collection of (boolean) functions $\mathcal{C}$, a uniform distribution Y is **recognized** by $\mathcal{C}$ if there exists a function $L' \in \mathcal{C}$ such that Y is the distribution recognized by L'.*

We also specialize the definition of seedless extractor to work for the distributions recognized by collections $\mathcal{C}$.

Definition 8. *A function $E : \{0,1\}^n \to \{0,1\}^m$ is a (seedless) (k,ϵ)-**extractor for distributions recognized** by $\mathcal{C}$ if, for all distributions Y recognized by $\mathcal{C}$ with min-entropy $H_\infty(Y) \geq k$, the variation distance between $E(Y)$ and the uniform distribution U_m is bounded by ϵ.*

For clarity's sake, we expand upon a motivating example from [Sha09].

Example. For any randomized algorithm $A : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$, consider the family of functions C_A indexed by $r_1, r_2 \in \{0,1\}^m$:

$$C_A = \{A_{r_1,r_2} : r_1, r_2 \in \{0,1\}^m\},$$

where $A_{r_1,r_2}(x) = 1$ if and only if $A(x,r_1) = A(x,r_2)$

Theorem 3. *(Improved from [Sha09]) Let $A : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$ be a function, $3 \leq c \leq n/m$, and $k = n - c \cdot m$ and $\epsilon = 2^{-c \cdot m}$. Let $E : \{0,1\}^n \to \{0,1\}^m$ be a (k, ϵ)-extractor for distributions recognizable by C_A. Let $\rho \leq 1/3$ and $f : \{0,1\}^n \to \{0,1\}$ be a function satisfying the following:*

$$P_{X \leftarrow U_n, R \leftarrow U_m}[A(X,R) = f(X)] \geq 1 - \rho.$$

Then the algorithm $B(x) = A(x, E(X))$ computes f within distance $3\rho - 2^{-(c-2)m}$. In other-words:

$$P_{X \leftarrow U_n}[A(X, E(X)) = f(X)] \geq 1 - 3\rho - 2^{-(c-2)m}.$$

Proof. We reproduce the proof given in [Sha09], except with optimized parameter choice. We consider the joint probability space over which X and R are independent random variables. X is uniformly distributed over $\{0,1\}^n$ and R is uniformly distributed over $\{0,1\}^m$. Define:

$$\begin{aligned}
\alpha &= P_{X \leftarrow U_n, R \leftarrow U_m}[A(X,R) = f(X)] \geq 1 - \rho \text{ (by assumption)} \\
\beta &= P_{X \leftarrow U_n}[A(X, E(X)) = f(X)]
\end{aligned}$$

Note that there must exist $r^* \in \{0,1\}^m$ such that $P_{X \leftarrow U_n}[A(X, r^*) = f(X) \geq 1 - \rho$ by an averaging argument similar to Yao's Lemma. We can replace α and β by the following:

$$\begin{aligned}
\alpha' &= P_{X \leftarrow U_n, R \leftarrow U_m}[A(X,R) = A(X, r^*)] \\
\beta' &= P_{X \leftarrow U_n}[A(X, E(X)) = A(X, r^*)]
\end{aligned}$$

It follows directly that $|\alpha - \alpha'| \leq \rho$ and $|\beta - \beta'| \leq \rho$. Thus, if we can show that $\beta' \geq \alpha' - 2^{-(c-2)m}$, then this immediately shows that $\beta \geq 1 - 3\rho - 2^{-(c-2)m}$. Consider the following set T:

$$T = \{r \in \{0,1\}^m : P_{X \leftarrow U_n}[A_{r,r^*}(X) = 1] < 2^{k-n}\}.$$

We know that the strings in T do not contribute much to the probability $A(X,R) = A(X, r^*)$:

$$\sum_{r \in T} P_{X \leftarrow U_n}[A_{r,r^*}(X) = 1] \leq |T| \cdot 2^{k-n} \leq 2^m 2^{k-n} = 2^{m - c \cdot m}.$$

Our desired result, $\beta' \geq \alpha' - 2^{-(c-2)m}$, follows from these two observations:

$$\alpha' \leq 2^{-m} \cdot \sum_{r \notin T} P_{X \leftarrow U_n}[A_{r,r^*}(X) = 1] + 2^{m - c \cdot m}. \tag{1}$$

$$\beta' \geq 2^{-m} \cdot \sum_{r \notin T} P_{X \leftarrow U_n}[A_{r,r^*}(X) = 1] - 2^{m - c \cdot m}. \tag{2}$$

We prove equation 2, and refer the reader to Shaltiel's paper for the proof of 1:

$$\begin{aligned}
\beta' &= P_{X \leftarrow U_n}[A(X, E(X)) = A(X, r^*)] \\
&= \sum_{r \in \{0,1\}^m} P_{X \leftarrow U_n}[A_{r,r^*}(X) = 1] \cdot P_{X \leftarrow U_n}[E(X) = r | A_{r,r^*}(X) = 1] \\
&\geq \sum_{r \notin T} P_{X \leftarrow U_n}[A_{r,r^*}(X) = 1] \cdot P_{X \leftarrow U_n}[E(X) = r | A_{r,r^*}(X) = 1].
\end{aligned}$$

Here, we use the fact that for $r \notin T$:

$$(X | A_{r,r^*}(X) = 1) = U_{A_{r,r^*}},$$

which of course is recognizable by C_A. We also have that $H_\infty(U_{A_{r,r^*}}) = \log_2 |\operatorname{supp}(U_{A_{r,r^*}})| \geq k$ by the fact that $r \notin T$. This allows us to say:

$$\begin{aligned}
\beta' &\geq \sum_{r \notin T} P_{X \leftarrow U_n}[A_{r,r^*}(X) = 1] \cdot P_{X \leftarrow U_n}[E(X) = r | A_{r,r^*}(X) = 1] \\
&\geq \sum_{r \notin T} P_{X \leftarrow U_n}[A_{r,r^*}(X) = 1] \cdot (2^{-m} - 2^{-c \cdot m}) \\
&= \sum_{r \notin T} P_{X \leftarrow U_n}[A_{r,r^*}(X) = 1] \cdot 2^{-m} - \sum_{r \notin T} P_{X \leftarrow U_n}[A_{r,r^*}(X) = 1] \cdot 2^{-c \cdot m} \\
&\geq 2^{-m} \cdot \sum_{r \notin T} P_{X \leftarrow U_n}[A_{r,r^*}(X) = 1] - 2^{m - c \cdot m},
\end{aligned}$$

which proves our result. This gives a much tighter relationship between the constants than the constants originally presented. □

3.2 Circuit Lower Bounds Imply Good Extractors

Theorem 3 is the main technical tool used in [Sha09]. In the context of typically-correct derandomization **BPP**, what remains to show is that good extractors exist that satisfy the hypothesis of the theorem. Shaltiel's results are conditioned on a hypothesis that is not known to imply **P** = **BPP**:

Hypothesis 1. There exists a constant $\gamma > 0$ such that for every constant c, there exists a constant d and family of functions $h = \{h_n : \{0,1\}^n \to \{0,1\}\}$, such that:

- h is computable in time n^d,
- but for sufficiently large n and every circuit $C \in_{\mathbf{SIZE}} [n^c]$, $\mathrm{P}_{X \to U_n}[C(X) = h_n(X)] \leq 1/2 + 2^{-n^\gamma}$.

A subsequent theorem, based on hypothesis 1, asserts the existence of a suitable extractor E.

Theorem 4. *Assume hypothesis 1. For large n, fix $l = polylog(n)$, and let $m = n/l$. There is an extractor $E : \{0,1\}^n \to \{0,1\}^m$ for distributions recognized by polynomial time algorithms. Furthermore, this extractor is computable in polynomial time.*

Finally, given this theorem, the conditional typically-correct derandomization result of Shaltiel follows by applying Theorem 3 using the aforementioned extractor.

Theorem 5. *Assuming hypothesis 1, there exists a constant $\mu > 0$ such that for any language $L \in$ **BPP**, there exists a function B computable in deterministic polynomial time, which computes L within a variation distance of 2^{-n^μ}.*

4 Kinne et al.'s conditional derandomization of BPP

In this section we shift our focus to the approach of Kinne et al. from [KMS10], also for typically-correct derandomization of **BPP**. The main difference from the approach of Shaltiel described above is that instead of using extractors, the input and the pseudorandomness are analyzed together in the form of seed-extending pseudorandom generators.

4.1 Seed-extending PRGs

Definition 9 (Seed-extending PRG)**.** *We say that a seed extending function $G(x) = (x, E(x))$ is ε-pseudorandom for a test T if*

$$|P_{x \leftarrow U_n, r \leftarrow U_m}[T(x,r) = 1] - P_{x \leftarrow U_n}[T(x, E(x)) = 1]| \leq \varepsilon.$$

This definition is nearly identical to regular PRGs, with the only difference being the input of the function prepended to the pseudorandomness. The definition based on tests is used instead of one based on circuits, because the notion of tests will be useful later in the proof.

Generally seed-extending PRGs are harder to find then regular PRGs, because of the additional condition of outputting the input, and in particular for example they do not exist when the test T is

allowed more time for execution then the PRG, because then T could execute the PRG since it knows the input.

In our case, however, a sufficient PRG can be constructed (based on some hardness assumptions). Specifically the Nisan-Wigderson PRG [NW94] can be adapted to a seed-extending PRG in the following way:

Definition 10 (Seed extending NW-PRG)**.** *Given a function $H : \{0,1\}^{\lfloor\sqrt{n}/2\rfloor} \to \{0,1\}$ which is $(\frac{1}{2} - \frac{\varepsilon}{m})$-hard for circuits of size $s + 2^{O(\log m/\log n)}$, the NW-PRG $G : \{0,1\}^n \to \{0,1\}^{n+m}$ is ε-pseudorandom for tests computable by circuits of size s.*

The construction for the this version of the NW-PRG is the same as for the original only with the input added in the front.

4.2 XOR Hardness Amplification

The result of Kinne et al requires amplifying the hardness of the assumed function. The exact theorem used in the paper is quite technical, but the general intuition of what it achieves is as follows: Given a hard function H we can trade larger input size for additional hardness by constructing a function $H'(x_1, x_2, ..., x_k) = H(x_1) \oplus H(x_2) \oplus ... \oplus H(x_k)$. This is used in the proof of Theorem 7.

4.3 Main Lemma

The following lemma provides a framework for most results in [KMS10]:

Theorem 6 (Main Lemma)**.** *Given a randomized algorithm $A(x, r)$ which computes a language L within distance ρ and a seed extending PRG G which is ε-pseudorandom for all tests of the form $T_{r'}(x, r) = A(x, r) \oplus A(x, r')$, then the algorithm $B(x) = A(G(x))$ is within distance $3\rho + \varepsilon$ from the language L.*

Proof. Firstly, take r' that minimizes the distance of $A(x, r')$ from L. Since $A(x, r)$ is within distance ρ of L, then also $A(x, r')$ is within distance ρ of L.

1. $A(x, r)$ and $A(x, r')$ are each within distance ρ from L, so they are within distance 2ρ from each other
2. Since G is ε-pseudorandom

   $$|P[A(G(x)) \neq A(x, r')] - P[A(x, r) \neq A(x, r')]| \leq \varepsilon$$

   This is derived by noticing that the described test with the $\oplus$ is equivalent to a test with a not equal sign instead. It is interesting to note here that now each of of the two probabilities in this expression resembles the expression for distance, which ties the two measures together. With that, and given the previous point, $A(G(x))$ and $A(x, r')$ are within distance $2\rho + \varepsilon$ from each other
3. Since $A(x, r')$ is within distance ρ of L, then $A(G(x))$ is within distance $3\rho + \varepsilon$ from L

□

4.4 Typically-Correct BPP Derandomization

The derandomization achieved by Kinne et al. is conditional and depends on the following hypothesis:

Hypothesis 2. For a given constant c there exists a constant d and a function $H \in P$ which is $\frac{1}{n^c}$-hard for size n^d circuits.

With that hypothesis we can state the main theorem of this section:

Theorem 7 (Typically-Correct BPP Derandomization)**.** *For any constant c, assuming Hypothesis 2 for that c, for every algorithm $A(x, r) \in$ **BPP**, there is a polynomial time algorithm B which decides L up to $\frac{1}{n^c}$.*

Proof. This proof uses all the ingredients explained earlier in this section:

1. use XOR hardness amplification to construct a function H' with hardness $(\frac{1}{2} - \frac{1}{n^a})$ for size n^a circuits $(a = 2\max(c+b, 2b))$
2. construct a seed-extending NW-PRG G which is $\frac{1}{2n^c}$-pseudorandom for circuits of size $O(n^{2b})$ (where n^b is the size or r)
3. the tests of the form $T_{r'}(x, r) = A(x, r) \oplus A(x, r')$ can be expressed as circuits of size $O(n^{2b})$
4. use error reduction (by majority voting) to construct an algorithm A' which is within $\frac{1}{6n^c}$ distance of L
5. using the Main Lemma, the deterministic algorithm $B(x) = A(G(x))$ is within distance $\frac{1}{n^c}$ from L

First consider the first two points. The XOR amplification is used in this case to construct a function which is hard enough to produce a good enough NW-PRG. Specifically, according to Definition 10 we need a function with hardness $(\frac{1}{2} - \frac{\varepsilon}{m})$ where m is the size of randomness, so by definition $m = n^b$, and, for reasons made more clear in the later part of the proof, we are constructing a $\frac{1}{2n^c}$-pseudorandom generator, so $\varepsilon = \frac{1}{2n^c}$. Thus $(\frac{1}{2} - \frac{1}{n^a}) \geq (\frac{1}{2} - \frac{1}{2n^{b+c}})$, which is where the $2(b+c)$ bound for a comes from. The $4b$ bound is related to the size of the circuit. The goal is for the NW-PRG to be pseudorandom for circuits of size $O(n^{2b})$ and the Definition 10 requires additional $2^O(\log m / \log n) \subseteq O(n^b)$ added to that size, which will be satisfied by the $4b$ bound.

Point 3 uses a construction similar to the Cook-Levin Theorem to convert the test T to a circuit of size $O(n^{2b})$.

Lastly, using simple majority voting scheme with $O(\log n)$ traces the distance of A to L is reduced to $\frac{1}{6n^c}$. With this distance and the $\varepsilon = \frac{1}{2n^c}$ the Main Lemma 6 gives a derandomization with distance $3\frac{1}{6n^c} + \frac{1}{2n^c} = \frac{1}{n^c}$ from L, as stated in the Theorem 7.

□

5 Conclusion

We hope that the two approaches to the same problem showcased in this paper paint a good picture of some of the methods used in the field of typically-correct derandomization. Even though the first result is weaker then the second one (it relies on a stronger hardness assumption) it is worthwhile to survey different ideas to understand how the later developments were made and because they may still come back in future results.

References

[BFNW91] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. Bpp has subexponential time simulations unless exptime has publishable proofs. In *[1991] Proceedings of the Sixth Annual Structure in Complexity Theory Conference*, pages 213–219, 1991.

[GW02] Oded Goldreich and Avi Wigderson. Derandomization that is rarely wrong from short advice that is typically good. In José D. P. Rolim and Salil Vadhan, editors, *Randomization and Approximation Techniques in Computer Science*, pages 209–223, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[IW01] Russell Impagliazzo and Avi Wigderson. Randomness vs time: Derandomization under a uniform assumption. *Journal of Computer and System Sciences*, 63(4):672–688, 2001.

[KMS10] Jeff Kinne, Dieter Melkebeek, and Ronen Shaltiel. Pseudorandom generators, typically-correct derandomization, and circuit lower bounds. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:129, 01 2010.

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

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.

[Sha09] R. Shaltiel. Weak derandomization of weak algorithms: Explicit versions of yao's lemma. In *2012 IEEE 27th Conference on Computational Complexity*, pages 114–125, Los Alamitos, CA, USA, jul 2009. IEEE Computer Society.

[Sha11] Ronen Shaltiel. Weak derandomization of weak algorithms: Explicit versions of yao's lemma. *computational complexity*, 20(1):87–143, Mar 2011.