

# Derandomizing RP if Boolean Circuits are not Learnable

Johannes Köbler and Wolfgang Lindner and Rainer Schuler

June 8, 1999

## Abstract

We show that every language in  $\mathcal{RP}$  has subexponential-time approximations for infinitely many input lengths if boolean circuits are not polynomial-time pac-learnable with membership queries under the uniform distribution.

## 1 Introduction

How to derandomize probabilistic computations, that is, how to efficiently simulate randomized computations by means of deterministic ones is an important and active research area in complexity theory. A central open question in this area regards the power of  $\mathcal{BPP}$ , the class of languages decidable in probabilistic polynomial time with small error. Obviously,  $\mathcal{BPP} \subseteq \mathcal{ECP}$ , but it is not known whether  $\mathcal{BPP}$  is in fact equal to  $\mathcal{ECP}$ . However, starting with the seminal work of Yao on pseudo-random generators [Yao82], there have been advances indicating that  $\mathcal{BPP}$  algorithms can be simulated significantly faster than by browsing through the whole underlying probability space. These results assume the existence of cryptographically secure one-way functions [Yao82, BH89], the hardness of problems in  $\mathcal{ECP}$  [BM84, NW94, BFNW93, IW97], or the existence of hitting set generators [ACR98], among others.

In this paper we build on yet another hypothesis regarding the learnability of boolean circuits, and show that  $\mathcal{RP}$ , the one-sided error version of  $\mathcal{BPP}$ , has

subexponential-time approximations if boolean circuits are not polynomial-time pac-learnable with membership queries under the uniform distribution. This hypothesis is known to follow from the existence of polynomially secure pseudorandom generators [GGM86], and has  $\mathcal{RP} \neq \mathcal{NP}$  as a consequence [BEHW87].

In the proof we use the well-known construction of a pseudorandom generator based on a hard function due to Nisan and Wigderson [NW94]. This construction is applied in a similar fashion as done by Impagliazzo and Wigderson [IW98] to obtain subexponential-time approximations for  $\mathcal{BPP}$ , based on the assumption  $\mathcal{EXP} \not\subseteq \mathcal{BPP}$ . The main departure from the arguments given in [IW98] is that here we have to deal with a whole concept class rather than a single language. We further make use of the equivalence of weak and strong learning under the uniform distribution as shown by Boneh and Lipton [BL93].

## 2 Preliminaries

**Probability.** We follow the notation used in the book [Lub97]. In particular,  $f : \{0, 1\}^{k(n)} \rightarrow \{0, 1\}^{\ell(n)}$  denotes a *function ensemble*, that is, for each fixed  $n$ ,  $f_n$  is a mapping from  $\{0, 1\}^{k(n)}$  to  $\{0, 1\}^{\ell(n)}$ .

We let  $D : \{0, 1\}^n$  denote a *probability ensemble*, where for each fixed  $n$ ,  $D_n$  is a probability distribution on  $\{0, 1\}^n$ . Throughout the paper, the uniform distribution is denoted by  $\mathcal{U}$ . We write  $X \in_D \{0, 1\}^n$  to indicate that  $X$  is a random variable on  $\{0, 1\}^n$  that is distributed according to  $D_n$ . A probability ensemble  $D : \{0, 1\}^n$  is *polynomial-time samplable* if there is a function ensemble  $f : \{0, 1\}^{r(n)} \rightarrow \{0, 1\}^n$  such that  $f$  is computable in time polynomial in  $n$ , and for  $X \in_{\mathcal{U}} \{0, 1\}^{r(n)}$ ,  $f(X)$  is distributed according to  $D_n$ .

**Learning.** A *concept*  $c$  over a predefined *instance space*  $U$  is a subset  $c \subseteq U$ . A *concept class* over  $U$  is a collection of concepts over  $U$ . We identify a concept  $c \subseteq U$  with its characteristic function  $c : U \rightarrow \{0, 1\}$ . A *representation class* is a quadruple

$$\mathcal{R} = (\Sigma, \Delta, R, \Phi),$$

where  $\Sigma$  and  $\Delta$  are finite alphabets,  $R \subseteq \Delta^*$  is the set of *representations*, and  $\Phi$  is a mapping from  $R$  to subsets of  $\Sigma^*$ . The concept class  $\mathcal{C}$  *represented* by  $\mathcal{R}$  is the set of concepts  $\Phi(r) \subseteq \Sigma^*$  for  $r \in R$ . The *size* of a representation  $r \in R$  is just its

length  $|r|$ . The *size* of a concept  $c \in \mathcal{C}$  is  $|c| = \min_{\Phi(r)=c} |r|$ , i.e., the size of the smallest representation of  $c$ . Concepts  $c \notin \mathcal{C}$  are defined to have infinite size.

In this paper we will only consider *boolean concepts*  $c$ . This means that for some positive integer  $n$ ,  $c$  is a subset of the finite instance space  $\{0, 1\}^n$ . A *boolean concept class* consists only of boolean concepts. A *boolean representation class*  $\mathcal{R}$  is a representation class representing a boolean concept class  $\mathcal{C}$ . We use  $\mathcal{C}_n$  to denote the set of concepts  $c : \{0, 1\}^n \rightarrow \{0, 1\}$  in  $\mathcal{C}$ , and we use  $\mathcal{C}_{n,s}$  to denote all concepts  $c \in \mathcal{C}_n$  of size at most  $s$ .

Let  $\mathcal{R}$  be a boolean representation class, and let  $D : \{0, 1\}^n$  be a probability ensemble. In the pac-learning model [Val84], a learning algorithm attempts to determine an unknown *target concept*  $\hat{c}$  from the boolean concept class  $\mathcal{C}$  represented by  $\mathcal{R}$ . The learning algorithm may make calls to an oracle  $EX(\hat{c}, D)$  which in unit time returns a *labeled example*  $(x, \hat{c}(x))$ , where  $x$  is drawn randomly and independently according to  $D$ . The goal of the learning algorithm is to output a representation of a concept that approximates the target well, where the quality of the approximation is measured w.r.t.  $D$ . The boolean representation class  $\mathcal{R}$  is *polynomial-time pac-learnable on the distribution  $D$*  if there exists a probabilistic algorithm  $A$  with the following property: for all integers  $n$  and  $s$ , for every target concept  $\hat{c} \in \mathcal{C}_{n,s}$ , for all rationals  $\epsilon > 0$  and  $\delta > 0$ ,  $A$  runs in time polynomial in  $n$ ,  $s$ ,  $1/\epsilon$  and  $1/\delta$ , and if  $A$  is given inputs  $n$ ,  $s$ ,  $\epsilon$ ,  $\delta$  and access to  $EX(\hat{c}, D)$ , then with probability at least  $1 - \delta$ ,  $A$  outputs a *hypothesis*  $h \in \mathcal{R}$  satisfying

$$\Pr(h(X) = \hat{c}(X)) \geq 1 - \epsilon,$$

where  $X \in_D \{0, 1\}^n$ . We refer to the algorithm  $A$  as the *learning algorithm* for  $\mathcal{R}$ . Further we refer to the input  $\epsilon$  as the *error parameter*, and to the input  $\delta$  as the *confidence parameter*.

The representation class  $\mathcal{R}$  is polynomial-time pac-learnable with *membership queries* on the distribution  $D$  if the learning algorithm for  $\mathcal{R}$  has additionally access to the oracle  $\hat{c}$ .

Kearns and Valiant [KV94] studied the *weak* variant of pac-learning where the hypothesis produced by the learning algorithm is required to perform only slightly better than a random guess. The boolean representation class  $\mathcal{R}$  is *weakly polynomial-time pac-learnable* on the distribution  $D$  if there exists a probabilistic algorithm  $A$  and a polynomial  $p$  such that for all integers  $n$  and  $s$ , for every target concept  $\hat{c} \in \mathcal{C}_{n,s}$ , and for all rationals  $\delta > 0$ ,  $A$  runs in time polynomial in  $n$ ,  $s$  and  $1/\delta$ , and if  $A$  is given inputs  $n$ ,  $s$ ,  $\delta$  and access to  $EX(\hat{c}, D)$ , then with probability

at least  $1 - \gamma$ ,  $A$  outputs a hypothesis  $h \in R$  satisfying

$$\Pr(h(X) = \hat{c}(X)) \geq \frac{1}{2} + \frac{1}{p(n, s)},$$

where  $X \in_D \{0, 1\}^n$ . Weak polynomial-time pac-learnability *with membership queries* is defined analogously.

Let the  $m$ -fold xor of a boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be the function  $f^{\oplus(m)} : \{0, 1\}^{mn} \rightarrow \{0, 1\}$  defined as

$$f^{\oplus(m)}(x_0, \dots, x_{m-1}) = \bigoplus_{i=0}^{m-1} f(x_i),$$

where  $x_0, \dots, x_{m-1} \in \{0, 1\}^n$ . We say that a boolean representation class  $\mathcal{R}$  is *polynomially closed under  $\oplus$*  if there exists a polynomial  $p$  such that for all integers  $m$  and for all  $c$  in the concept class  $\mathcal{C}$  represented by  $\mathcal{R}$ , the concept  $c^{\oplus(m)}$  has size at most  $p(|c|, m)$ .

**Theorem 1 ([BL93]).** *Let  $\mathcal{R}$  be a boolean representation class which is polynomially closed under  $\oplus$ . Then the following are equivalent:*

1.  $\mathcal{R}$  is weakly polynomial-time learnable under the uniform distribution.
2.  $\mathcal{R}$  is polynomial-time learnable under the uniform distribution.

*This equivalence also holds in the presence of membership queries.*

### Subexponential-time approximations.

**Definition (cf. [IW98]).** A language  $L$  has *subexponential-time approximations* if for all  $\gamma > 0$ , there exists a  $2^{n^\gamma}$ -time bounded deterministic Turing machine  $M$  such that for all polynomial-time samplable probability ensembles  $D$ , for all polynomials  $p$ , for almost all  $n$ , and for  $X$  randomly chosen according to  $D_n$ ,

$$\Pr(L(X) \neq M(X)) < \frac{1}{p(n)}.$$

If this holds only for infinitely many  $n$ , then  $L$  is said to have *weak* subexponential-time approximations.

### 3 Derandomization of $\mathcal{RP}$

In this section, we prove the following theorem.

**Theorem 2.** *Suppose that boolean circuits are not weakly polynomial-time learnable with membership queries under the uniform distribution. Then  $\mathcal{RP}$  admits weak subexponential-time approximations.*

We first recall some notation from [NW94].

**Definition.** A  $(\ell, m, n, k)$ -design is a collection  $\mathcal{D} = (D_0, \dots, D_{\ell-1})$  of sets  $D_i \subseteq \{0, \dots, m-1\}$ , each of which has cardinality  $n$ , such that for all  $i \neq j$ ,  $|D_i \cap D_j| \leq k$ . Given a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , the *nearly disjoint sets generator* (based on  $f$  and  $\mathcal{D}$ ),  $f^{\mathcal{D}} : \{0, 1\}^m \rightarrow \{0, 1\}^{\ell}$ , is for every seed  $x = x_0 \cdots x_{m-1}$  of length  $m$  defined by

$$f^{\mathcal{D}}(x) = f(x[D_0]) \cdots f(x[D_{\ell-1}]),$$

where  $\mathcal{D} = \{D_0, \dots, D_{\ell-1}\}$ , and  $x[D_i]$ , for  $0 \leq i \leq \ell - 1$ , denotes the restriction of  $x$  to  $D_i = \{i_0 < \cdots < i_{n-1}\}$  defined as  $x[D_i] = x_{i_0} \cdots x_{i_{n-1}}$ .

We also need the following lemma.

**Lemma 3 ([NW94]).** *For all integers  $n$  and  $\ell$  with  $\ell \leq 2^n$ , there exists a  $(\ell, 4n^2, n, \lceil \log \ell \rceil)$ -design  $\mathcal{D}$ . Moreover, there is an algorithm which for every  $n$  and  $\ell$  computes  $\mathcal{D}$  in time polynomial in  $n$  and  $\ell$ .*

*Remark 1.* In the following, we will refer to the design  $\mathcal{D}$  computed by the algorithm in the previous lemma as the *generic*  $(\ell, 4n^2, n, \lceil \log \ell \rceil)$ -design.

Nisan and Wigderson showed that if the function  $f$  is hard to approximate by polynomial-size circuits, then the generator  $f^{\mathcal{D}}$  has polynomial non-uniform security. This means that if there is a polynomial-size test  $T$  with sufficiently large distinguishing probability for  $f^{\mathcal{D}}$ , then there is a polynomial-size circuit  $C$  approximating  $f$ . Impagliazzo and Wigderson [IW98] showed that  $C$  can be uniformly obtained from  $T$  with polynomially many membership queries to  $f$ .

**Lemma 4 (cf. [IW98]).** *There is a probabilistic oracle algorithm  $A$  with the following property: For all integers  $n$  and  $\ell \leq 2^n$ , for every probabilistic circuit  $C$  with input length  $\ell$ , and for every function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , for all rationals*

$\epsilon > 0$ ,  $\gamma > 0$ , if  $A$  gets inputs  $n$ ,  $\ell$ ,  $\epsilon$ ,  $\gamma$ ,  $C$  and oracle  $f$ , then  $A$  runs in time polynomial in  $n$ ,  $\ell$ ,  $|C|$ ,  $1/\epsilon$ , and  $\log(1/\gamma)$ , and with probability at least  $1 - \gamma$ ,  $A$  outputs a deterministic circuit  $D$  which for  $Z \in_{\mathcal{U}} \{0, 1\}^n$  satisfies

$$\Pr(D(Z) = f(Z)) \geq \frac{1}{2} + \delta/\ell - \epsilon,$$

where for  $X \in_{\mathcal{U}} \{0, 1\}^{4n^2}$  and  $Y \in_{\mathcal{U}} \{0, 1\}^{\ell}$ ,

$$\delta = |\Pr(C(f^{\mathcal{D}}(X)) = 1) - \Pr(C(Y) = 1)|$$

and  $\mathcal{D}$  is the generic  $(\ell, 4n^2, n, \lceil \log \ell \rceil)$ -design.

For the proof of our theorem we also need the following two lemmas.

**Lemma 5.** For functions  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and  $g : \{0, 1\}^n \times \{0, 1\}^r \rightarrow \{0, 1\}$ , and for  $y \in \{0, 1\}^r$  and  $X \in_{\mathcal{U}} \{0, 1\}^n$ , let

$$\sigma(y) = \Pr(g(X, y) = f(X)).$$

and let  $\sigma$  be the expected value of  $\sigma(Y)$ , where  $Y \in_{\mathcal{U}} \{0, 1\}^r$ . Furthermore, for an integer  $q$ , for  $x_0, \dots, x_{q-1} \in \{0, 1\}^n$  and  $y_0, \dots, y_{q-1} \in \{0, 1\}^r$ , define  $h(x_0, \dots, x_{q-1}, y_0, \dots, y_{q-1})$  to be the smallest index  $j \in \{0, \dots, q-1\}$  such that the cardinality

$$|\{i \in \{0, \dots, q-1\} : g(x_i, y_j) = f(x_i)\}|$$

is maximal. Then there exists a polynomial  $p$  such that for all functions  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and  $g : \{0, 1\}^n \times \{0, 1\}^r \rightarrow \{0, 1\}$ , for all rationals  $\epsilon > 0$ ,  $\gamma > 0$ , for  $q = p(1/\epsilon, \log(1/\gamma))$  and for independently chosen  $X_0, \dots, X_{q-1} \in_{\mathcal{U}} \{0, 1\}^n$  and  $Y_0, \dots, Y_{q-1} \in_{\mathcal{U}} \{0, 1\}^r$ , it holds that

$$\sigma(Y_{h(X_0, \dots, X_{q-1}, Y_0, \dots, Y_{q-1})}) \geq \sigma - \epsilon,$$

with probability at least  $1 - \gamma$ .

*Proof.* For  $Y \in_{\mathcal{U}} \{0, 1\}^r$ ,  $\sigma(Y)$  is a random variable that takes only values in the interval  $[0, 1]$ . Since the expectation of  $\sigma(Y)$  is  $\sigma$ , this implies that  $\sigma(Y) < \sigma - \epsilon/3$  with probability at most  $1 - \epsilon/3$ . Hence, for  $t \geq 3/\epsilon \ln(2/\gamma)$  independently chosen  $Y_0, \dots, Y_{t-1} \in_{\mathcal{U}} \{0, 1\}^r$ , it holds that  $\sigma(Y_j) < \sigma - \epsilon/3$  for all  $j \in \{0, \dots, t-1\}$  with probability at most

$$(1 - \epsilon/3)^t \leq e^{-t\epsilon/3} \leq \gamma/2.$$

For  $x_0, \dots, x_{s-1} \in \{0, 1\}^n$  and  $y \in \{0, 1\}^r$  define

$$\tilde{\sigma}(x_0, \dots, x_{s-1}, y) = \frac{|\{i \in \{0, \dots, s-1\} : g(x_i, y) = f(x_i)\}|}{s}.$$

For every  $y \in \{0, 1\}^r$  and for  $X_0, \dots, X_{s-1} \in_{\mathcal{U}} \{0, 1\}^n$ , the expected value of  $\tilde{\sigma}(X_0, \dots, X_{s-1}, y)$  is  $\sigma(y)$ . Applying Chernoff Bounds, it is possible to choose  $s$  polynomial in  $1/\epsilon$  and  $\log(t/\gamma)$  such that for every  $y$ ,

$$|\tilde{\sigma}(X_0, \dots, X_{s-1}, y) - \sigma(y)| > \epsilon/3$$

holds with probability at most  $\gamma/(2t)$ . Hence, for  $Y_0, \dots, Y_{t-1} \in_{\mathcal{U}} \{0, 1\}^r$ , the probability that

- there exists some  $j \in \{0, \dots, t-1\}$  with  $\sigma(Y_j) \geq \sigma - \epsilon/3$ , and
- for all  $j \in \{0, \dots, t-1\}$ ,  $|\tilde{\sigma}(X_0, \dots, X_{s-1}, Y_j) - \sigma(Y_j)| \leq \epsilon/3$

is at least  $1 - \gamma$ .

In the case that there exists some  $j \in \{0, \dots, t-1\}$  with  $\sigma(y_j) \geq \sigma - \epsilon/3$  and that  $|\tilde{\sigma}(x_0, \dots, x_{s-1}, y_j) - \sigma(y_j)| \leq \epsilon/3$  holds for all  $i \in \{0, \dots, s-1\}$ , we have

$$\tilde{\sigma}(x_0, \dots, x_{s-1}, y_{h(x_0, \dots, x_{s-1}, y_0, \dots, y_{t-1})}) \geq \sigma - 2\epsilon/3,$$

implying that

$$\sigma(y_{h(x_0, \dots, x_{s-1}, y_0, \dots, y_{t-1})}) \geq \sigma - \epsilon.$$

Hence it follows that

$$\sigma(Y_{h(X_0, \dots, X_{t-1}, Y_0, \dots, Y_{s-1})}) \geq \sigma - \epsilon$$

holds with probability at least  $1 - \gamma$ . Now the lemma follows by choosing  $q = s \geq t$ .  $\square$

**Lemma 6.** *If boolean circuits of size at most  $2n$  are weakly polynomial-time pac-learnable under the uniform distribution, then boolean circuits of arbitrary size are weakly polynomial-time pac-learnable under the uniform distribution. This also holds in the presence of membership queries.*

*Proof.* Let  $A$  be a weak polynomial-time learning algorithm for boolean circuits of size at most  $2n$ , i.e., for some polynomial  $p$ , any circuit  $\hat{c} : \{0, 1\}^n \rightarrow \{0, 1\}$  of size at most  $2n$ ,  $A$  on input  $n$ ,  $\delta$  outputs with probability at least  $1 - \delta$  a circuit  $c$  satisfying

$$\Pr(c(X) = \hat{c}(X)) \geq \frac{1}{2} + \frac{1}{p(n)},$$

where  $X \in_{\mathcal{U}} \{0, 1\}^n$ . We describe the learning algorithm  $A'$  for boolean circuits of arbitrary size in two steps. In the first step, it uses  $A$  to compute a circuit  $C$  as follows.

For given inputs  $n$ , size  $s$ , confidence parameter  $\delta$ , and with respect to a target  $\hat{c} : \{0, 1\}^n \rightarrow \{0, 1\}$  computable by a circuit of size  $s$ , simulate  $A$  with parameters  $s$  for the domain of the target concept,  $2s$  for the size and confidence parameter  $\delta/2$ . Whenever  $A$  requests a random labeled example, request a labeled example  $(x, \hat{c}(x))$ , choose  $y \in_{\mathcal{U}} \{0, 1\}^{s-n}$ , and provide  $A$  with  $(xy, \hat{c}(x))$ . In case  $A$  makes a membership query  $z$  of length  $s$ , then make a membership query  $x$ , where  $x$  consists of the first  $n$  bits of  $z$ , and provide  $A$  with the answer  $\hat{c}(x)$ . Let  $C$  be the circuit produced by  $A$ .

In other words,  $A$  is used by  $A'$  to compute a hypothesis  $C$  for the target  $\tilde{c} : \{0, 1\}^s \rightarrow \{0, 1\}$  defined as  $\tilde{c}(xy) = \hat{c}(x)$  for all  $x \in \{0, 1\}^n$  and all  $y \in \{0, 1\}^{s-n}$ . Since the size of  $\tilde{c}$  is at most  $s + s - n \leq 2s$ , it follows that with probability at least  $1 - \delta/2$ , the circuit  $C$  satisfies

$$\Pr(C(X, Y) = \tilde{c}(XY)) \geq \frac{1}{2} + \frac{1}{p(s)},$$

where  $X \in_{\mathcal{U}} \{0, 1\}^n$  and  $Y \in_{\mathcal{U}} \{0, 1\}^{s-n}$ . Now let  $q$  and  $h$  be as in Lemma 5 with respect to the functions  $C$  and  $\hat{c}$ , and parameters  $\epsilon = \frac{1}{2p(s)}$  and  $\gamma = \delta/2$  and let the algorithm  $A'$  continue as follows.

Request  $q$  random labeled examples  $(x_0, \hat{c}(x_0)), \dots, (x_{q-1}, \hat{c}(x_{q-1}))$ .  
 Choose  $y_0, \dots, y_{q-1} \in_{\mathcal{U}} \{0, 1\}^{s-n}$ , compute  $j_0 = h(x_0, \dots, x_{q-1}, y_0, \dots, y_{q-1})$ , and output the circuit  $C'$  that computes  $C'(x) = C(x, y_{j_0})$  for all  $x \in \{0, 1\}^n$ .

By Lemma 5,  $\Pr(C(X, Y_{h(X_0, \dots, X_{q-1}, Y_0, \dots, Y_{q-1})}) = \hat{c}(X)) \geq \frac{1}{2} + \frac{1}{p(s)} - \frac{1}{2p(s)} = \frac{1}{2} + \frac{1}{2p(s)}$  holds with probability at least  $1 - \delta/2$ , where  $X, X_0, \dots, X_{q-1} \in_{\mathcal{U}} \{0, 1\}^n$



and  $Y_0, \dots, Y_{q-1} \in_{\mathcal{U}} \{0, 1\}^{s-n}$ , implying that  $C'$  satisfies

$$\Pr(C'(X) = \hat{c}(X)) \geq \frac{1}{2} + \frac{1}{2p(s)}$$

with probability at least  $1 - \delta$ . □

Now we are ready to prove our main result.

*Proof of Theorem 2.* Let  $L$  be a language in  $\mathcal{RP}$ . Then, for some polynomial  $r$  there is a polynomial-time function ensemble  $R : \{0, 1\}^n \times \{0, 1\}^{r(n)} \rightarrow \{0, 1\}$  such that for all strings  $x \in \{0, 1\}^n$  and for  $Y \in_{\mathcal{U}} \{0, 1\}^{r(n)}$ ,

1.  $x \in L \implies \Pr(R(x, Y) = 1) \geq 2/3$ , and
2.  $x \notin L \implies \Pr(R(x, Y) = 1) = 0$ .

For a given rational  $\gamma > 0$  and input length  $n$ , let  $k(n) = \lfloor n^{\gamma/2} \rfloor$  and let  $m(n) = 4k(n)^2$ . Consider a procedure that on input  $x$  of length  $n$  accepts if and only if there is a circuit  $C : \{0, 1\}^{k(n)} \rightarrow \{0, 1\}$  of size at most  $2k(n)$  and a seed  $z$  of length  $m(n)$  such that  $R(x, C^{\mathcal{D}}(z)) = 1$ , where  $\mathcal{D}$  is the generic  $(k(n), m(n), r(n), \lceil \log r(n) \rceil)$ -design provided by Lemma 4. Since  $\mathcal{D}$  can be computed in time polynomial in  $n$  and  $r(n)$ , and since  $m(n) = \mathcal{O}(n^\gamma)$ , the procedure runs in time  $2^{\mathcal{O}(n^\gamma)}$ .

We now assume that the procedure fails to weakly approximate  $L$ . Based on this assumption we give a learning algorithm for boolean circuits, contradicting the assumption of the theorem. So let  $p$  be a polynomial and let  $D : \{0, 1\}^n$  be a polynomial-time samplable probability ensemble such that for almost all  $n$ , the procedure disagrees with  $L$  with probability at least  $1/p(n)$ , if the input is chosen according to  $D_n$ . First we prove the following claim.

**Claim 1.** *For almost all  $n$ , and for all functions  $f : \{0, 1\}^{k(n)} \rightarrow \{0, 1\}$  computable by a circuit of size at most  $2k(n)$ ,*

$$|\Pr(R(X, f^{\mathcal{D}}(Z)) = 1) - \Pr(R(X, Y) = 1)| \geq \frac{2}{3p(n)},$$

where  $X \in_D \{0, 1\}^n$ ,  $Y \in_{\mathcal{U}} \{0, 1\}^{r(n)}$ ,  $Z \in_{\mathcal{U}} \{0, 1\}^{m(n)}$ , and  $\mathcal{D}$  is the generic  $(r(n), m(n), k(n), \lceil \log r(n) \rceil)$ -design.

*Proof.* The procedure can only disagree with  $L$  on a string  $x$  of length  $n$ , if  $x$  is in  $L$  but the procedure rejects. This means that  $\Pr(R(x, Y) = 1) \geq 2/3$ , but for all functions  $f : \{0, 1\}^{k(n)} \rightarrow \{0, 1\}$  computable by a circuit of size at most  $2k(n)$ , and for all seeds  $z$  of length  $m(n)$ ,  $R(x, f^{\mathcal{D}}(z)) = 0$ , implying that

$$|\Pr(R(x, f^{\mathcal{D}}(Z)) = 1) - \Pr(R(x, Y) = 1)| \geq \frac{2}{3},$$

where  $Z \in_{\mathcal{U}} \{0, 1\}^{m(n)}$  and  $Y \in_{\mathcal{U}} \{0, 1\}^{r(n)}$ . The claim follows, since the procedure disagrees with  $L$  on a randomly chosen string (according to  $D_n$ ) with probability at least  $1/p(n)$ .  $\square$

Let  $C_n$  be a probabilistic circuit that for  $y \in \{0, 1\}^{r(n)}$ , computes  $C(y) = R(X, y)$ , where  $X \in_D \{0, 1\}^n$ . Based on the claim we give an algorithm that weakly learns any target circuit  $\hat{c} : \{0, 1\}^k \rightarrow \{0, 1\}$  of size at most  $2k$ .

On input  $k$  and confidence parameter  $\delta$ , choose  $n$  to be the smallest integer such that  $k = k(n)$  and compute the generic  $(r(n), m(n), k, \lceil \log r(n) \rceil)$ -design  $\mathcal{D}$ . Run the algorithm of Lemma 4 with the circuit  $C_n$ , oracle  $\hat{c}$ , and parameters  $\epsilon = 1/(2r(n)p(n))$  and  $\gamma = \delta$ . Output the resulting circuit  $C''$ .

Because  $D : \{0, 1\}^n$  is polynomial-time samplable, the probabilistic circuit  $C_n$  can be obtained from (finite) descriptions of the Turing machines computing  $R$  and  $D$ , respectively. Since the target  $\hat{c}$  has size at most  $2k$ , it follows from the claim that the distinguishing probability of  $C_n$  for  $\hat{c}^{\mathcal{D}}$  is at least  $2/3p(n)$ , i.e., for  $Y \in_{\mathcal{U}} \{0, 1\}^{r(n)}$  and  $Z \in_{\mathcal{U}} \{0, 1\}^{m(n)}$ ,  $C_n$  satisfies

$$|\Pr(C_n(\hat{c}^{\mathcal{D}}(Z)) = 1) - \Pr(C_n(Y) = 1)| \geq \frac{2}{3p(n)}.$$

Hence, the algorithm of Lemma 4 produces with probability at least  $1 - \delta$  a circuit  $C''$  such that

$$\Pr(C''(W) = \hat{c}(W)) \geq \frac{1}{2} + \frac{1}{6r(n)p(n)},$$

where  $W \in_{\mathcal{U}} \{0, 1\}^k$ . Thus we have shown that the class of circuits  $c : \{0, 1\}^k \rightarrow \{0, 1\}$  of size  $2k$  is weakly polynomial-time learnable with membership queries under the uniform distribution, provided that there is some language  $L$  in  $\mathcal{RP}$  for which the procedure given above fails to weakly approximate  $L$ . Therefore, the theorem follows by applying Lemma 6.  $\square$

From Theorem 1 we immediately get the following corollary.

**Corollary 7.** *Suppose that boolean circuits are not polynomial-time learnable with membership queries under the uniform distribution. Then  $\mathcal{RP}$  admits weak subexponential-time approximations.*

Since the existence of weak subexponential-time approximations for a language class  $\mathcal{C}$  implies that  $\mathcal{C}$  has  $\mathcal{EXPTIME}$ -measure zero (in the sense of resource bounded measure as introduced by Lutz [Lut92]) we additionally get the following corollary.

**Corollary 8.** *Suppose that boolean circuits are not polynomial-time learnable with membership queries under the uniform distribution. Then  $\mathcal{RP}$  has  $\mathcal{EXPTIME}$ -measure zero.*

## References

- [ACR98] A. Andreev, A. Clementi, and J. Rolim. A new general derandomization method. *Journal of the ACM*, 45(1):179–213, 1998.
- [BEHW87] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam’s razor. *Information Processing Letters*, 24(6):377–380, 1987.
- [BFNW93] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3:307–318, 1993.
- [BH89] R. B. Boppana and R. Hirschfeld. Pseudorandom generators and complexity classes. In *Advances in Computing Research*, volume 5, pages 1–26. JAI Press Inc., 1989.
- [BL93] D. Boneh and R. J. Lipton. Amplification of weak learning under the uniform distribution. In *Proc. 6th ACM Conference on Computational Learning Theory*, pages 347–351, 1993.
- [BM84] M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudorandom bits. *SIAM Journal on Computing*, 13(4):850–864, 1984.

- [GGM86] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986.
- [IW97] R. Impagliazzo and A. Wigderson. P=BPP unless E has sub-exponential circuits: derandomizing the XOR lemma. In *Proc. 29rd ACM Symposium on Theory of Computing*, pages 220–229. ACM Press, 1997.
- [IW98] R. Impagliazzo and A. Wigderson. Randomness vs. time: Derandomization under a uniform assumption. In *Proc. 39th IEEE Symposium on the Foundations of Computer Science*, pages 734–743. IEEE Computer Society Press, 1998.
- [KV94] M. J. Kearns and L. G. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *Journal of the ACM*, pages 67–95, 1994.
- [Lub97] M. Luby. *Pseudorandomness and Cryptographic Applications*. Princeton University Press, 1997.
- [Lut92] J. H. Lutz. Almost everywhere high nonuniform complexity. *Journal of Computer and System Sciences*, 44:220–258, 1992.
- [NW94] N. Nisan and A. Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994.
- [Val84] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.
- [Yao82] A. C. Yao. Theory and applications of trapdoor functions. In *Proc. 23rd IEEE Symposium on the Foundations of Computer Science*, pages 80–91. IEEE Computer Society Press, 1982.