THEORY OF COMPUTING, Volume 8 (2012), pp. 231–238
www.theoryofcomputing.org

NOTE

Monotone Circuits: One-Way Functions versus Pseudorandom Generators

Oded Goldreich* Rani Izsak†

Received: January 24, 2012; published: June 3, 2012.

Abstract: We study the computability of one-way functions and pseudorandom generators by monotone circuits, showing a substantial gap between the two: On one hand, there exist one-way functions that are computable by (uniform) polynomial-size monotone functions, provided (of course) that one-way functions exist at all. On the other hand, no monotone function can be a pseudorandom generator.

ACM Classification: F.1.1, F.1.3

AMS Classification: 68Q17, 03D15

Key words and phrases: monotone circuits, one-way functions, pseudorandom generators, cryptography

1 Introduction

One-way functions and pseudorandom generators play a central role in computational complexity theory and cryptography. Loosely speaking, one-way functions (OWFs) are functions that are easy to compute but hard to invert (in the average-case sense). Pseudorandom generators (PRGs) are efficient algorithms that stretch short random seeds into longer (pseudorandom) sequences that are computationally indistinguishable from truly random sequences. (Indeed, we refer to the standard definitions, which are recalled in Section 2; for further discussion, the interested reader is referred to [5, 6].)

*Partially supported by the Israel Science Foundation (grant No. 1041/08).
†Partially supported by the Israel Science Foundation (grant No. 1041/08).

DOI: 10.4086/toc.2012.v008a010

A fundamental result in this area asserts that one-way functions exist if any only if pseudorandom generators exist [7] (see also [5, Sec. 3.5]). A relatively recent result of Applebaum, Ishai, and Kushilevitz [2] indicates that (under some widely believed conjectures) both OWFs and (sublinear-stretch) PRGs can be computed by very simple circuits; specifically, by circuits in which each output bit depends only on a constant number of input bits (i. e., $\mathcal{NC}^0$).

The latter result raises the natural question of whether OWFs and PRGs can be computed by other restricted families of circuits. Recalling that PRGs constitute OWFs (see [5, Sec. 3.5]), it is natural to first ask whether OWFs can be computed by polynomial-size monotone circuits, and then to ask the same regarding PRGs. We show that the answer to the first question is positive (assuming, of course, that OWFs exist at all), while the answer to the second question is negative. That is:

Theorem 1.1. *If there exist one-way functions, then there exist one-way functions that are computable by uniform families of polynomial-size* monotone *circuits.*

We stress that not only are these one-way functions monotone, but also their *monotone circuit complexity* is polynomial.

Theorem 1.2. *No* monotone *function is a pseudorandom generator. Furthermore, for any* monotone *function $f:\{0,1\}^n \to \{0,1\}^{n+1}$, there exists a* (monotone) *circuit D in $\mathcal{NC}^0$ such that*

$$|\Pr[D(U_{n+1})=1] - \Pr[D(f(U_n))=1]| = \Omega\left(1/n^2\right),$$

where U_m denotes a random variable uniformly distributed over $\{0,1\}^m$.

We stress that Theorem 1.2 makes no reference to the monotone (or even the general) complexity of f. We also note that the distinguishers witnessing this failure are very simple (and monotone).

Indeed, these two results indicate that in the "monotone world" there is a fundamental gap between one-way functions and pseudorandom generators; thus, the "hardness-vs-randomness" paradigm [4, 11, 9] fails in the monotone setting.

Organization Theorems 1.1 and 1.2 are proved in Sections 3 and 4, respectively. But before turning to these proofs, we recall (in Section 2) the standard definitions.

2 Preliminaries

We recall the standard definitions (adapted from [5], where the interested reader may find further discussions). A function $f:\mathbb{N}\to\mathbb{R}$ is called *negligible* if it decreases faster than the reciprocal of any positive polynomial (i. e., for every positive polynomial p and all sufficiently large n it holds that $f(n) < 1/p(n)$). A function $f:\mathbb{N}\to\mathbb{R}$ is called *noticeable* if it decreases slower than the reciprocal of some positive polynomial (i. e., there exists a positive polynomial p such that for all sufficiently large n it holds that $f(n) > 1/p(n)$). We say that a family of circuits $\{C_n\}$ is *polynomial-size* if there exists a polynomial p such that for all n it holds that $\mathrm{size}(C_n) \le p(n)$, while the number of input bits to the circuit C_n is not necessarily n.

Definition 2.1 (One-Way Functions – OWF)**.** Let $h : \mathbb{N} \to [0,1]$. A function $F : \{0,1\}^* \to \{0,1\}^*$ is called **h-hard one-way** if it satisfies the following two conditions.

- *Easy to compute:* There exists a polynomial-time algorithm that on input x outputs $F(x)$.
- *h-hard to invert:* For every family of *(uniform)*[1] polynomial-size circuits $\{I_n\}_{n\in\mathbb{N}}$ it holds that

$$\Pr\left[I_n(F(U_n)) \notin F^{-1}(F(U_n))\right] \geq h(n)\,.$$

If h is noticeable, then F is called a **weak one-way function**, whereas if $1-h$ is negligible then F is called a **strong one-way function** (or just a **one-way function**).

Note that the above definitional framework has two versions, one referring to uniform polynomial-size circuits and one referring to all (including non-uniform) polynomial-size circuits. Our results refer to both versions. The same applies also to the following definition.

Definition 2.2 (Pseudorandom Generator – PRGs)**.** A function $G : \{0,1\}^* \to \{0,1\}^*$ is called a **pseudorandom generator** if it satisfies the following three conditions:

- *Stretch:* For every s it holds that $|G(s)| > |s|$.
- *Easy to compute:* There exists a polynomial-time algorithm that on input s outputs $G(s)$.
- *Pseudorandomness:* For every family of *(uniform)* polynomial-size circuits $\{D_n\}_{n\in\mathbb{N}}$ it holds that the function Δ defined by $\Delta(n) = |\Pr[D_n(G(U_n)) = 1] - \Pr[D_n(U_{|G(1^n)|}) = 1]|$ is negligible.

Monotone functions and circuits A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is called *monotone* if for every $x \prec y$ it holds that $f(x) \leq f(y)$, where $\prec$ denotes the standard partial order on (fixed length) bit strings (i. e., $x_1x_2\cdots x_n \prec y_1y_2\cdots y_n$ if for every i it holds that $x_i \leq y_i$ and for some i it holds that $0 = x_i < y_i = 1$). A function $f : \{0,1\}^n \to \{0,1\}^m$ is called *monotone* if, for every $i \in [m]$, the projection of f on its i^{th} output bit (i. e., $f_i(x) \stackrel{\text{def}}{=} f(x)_i$) yields a monotone Boolean function. This notion extends naturally to length regular functions defined over $\{0,1\}^*$ (i. e., functions $f : \{0,1\}^* \to \{0,1\}^*$ such that for every $|x| = |y|$ it holds that $|f(x)| = |f(y)|$). Throughout this paper, we shall consider only length-regular functions.

3 OWFs computable by monotone circuits

In this section, we prove Theorem 1.1. We focus on proving that the existence of OWFs implies the existence of weak-OWFs that are computable by small (uniform) monotone circuits. We derive standard OWFs (so computable) by observing that the standard amplification of one-way functions (cf., e. g., [5, Sec. 2.3]) applies to the current (monotone) setting.

The basic idea is to transform a standard OWF into a weak monotone OWF by restricting its "actual action" to the middle slice, and modifying it on all others slices so as to obtain a monotone function.

[1] As usual, in the uniform case, we consider probabilistic circuits.

Recall that the *k-slice* of a Boolean function $b:\{0,1\}^n \to \{0,1\}$ is defined as $\{x \in \{0,1\}^n : \mathrm{wt}(x)=k\}$, where $\mathrm{wt}(x) \stackrel{\mathrm{def}}{=} |\{i : x_i = 1\}|$ denotes the Hamming weight of x. Now, given a OWF F, we define F' such that $F'(x)=F(x)$ if $\mathrm{wt}(x)=\lfloor |x|/2 \rfloor$ (i. e., x is in the middle slice) and $F'(x)=\sigma^{|F(x)|}$ otherwise, where $\sigma = 1$ if $\mathrm{wt}(x) > \lfloor |x|/2 \rfloor$ and $\sigma=0$ if $\mathrm{wt}(x) < \lfloor |x|/2 \rfloor$.

The function F' is monotone, since for every $x \prec y$ it holds that at most one of these strings belongs to the middle slice while the values of F' on all other slices conform with any value given to the strings on the middle slice. However, this does not mean that F' can be computed by polynomial-size monotone circuit. Nevertheless, the latter fact is a direct corollary of Berkowitz's theorem [3]:

Theorem 3.1. *Let $b:\{0,1\}^n \to \{0,1\}$ be a Boolean function and let C be a circuit computing it. Then, for every $k \in \{1,\dots,n\}$ there exists a monotone circuit C_M of size* $\mathrm{poly}(n)\cdot \mathrm{size}(C)$ *that computes the k-slice function of b* (i. e., the function that agrees with b on the k-slice, is zero on lower slices and one on higher slices). *Moreover, C_M is polynomial-time constructible, given C as an input.*

Specifically, let $T_k:\{0,1\}^n \to \{0,1\}$ denote the k^{th} threshold function (i. e., $T_k(x)=1$ iff $\mathrm{wt}(x) \geq i$) and recall that $\mathrm{size}(T_k)=\widetilde{O}(n)$ (cf. [1]). Let $C':\{0,1\}^{2n} \to \{0,1\}$ be the monotone circuit obtained from C by pushing all negations to the bottom level and replacing negated variables by auxiliary variables; that is, $C(x)=C'(x,\overline{x})$, where $\overline{x}_i = \neg x_i$. The crucial observation is that for any x such that $\mathrm{wt}(x)=k$, it holds that $\neg x_i = T_k(x \wedge 1^{i-1}01^{n-i})$ (since in that case $T_k(x\wedge 1^{i-1}01^{n-i})=1$ iff $x_i=0$). Letting $N(x) = (T_k(x\wedge 01^{n-1}),\dots,T_k(x\wedge 1^{n-1}0))$, we get $C_M(x)=(T_k(x)\wedge C'(x,N(x)))\vee T_{k+1}(x)$, which is a monotone circuit computing the k-slice function of b.

It follows that F' has (uniform) monotone polynomial-size circuits, and it is left to show that F' is a weak OWF.

Proposition 3.2. *Let F be a* (strong) *one-way function and let F' be as above. Then, no polynomial-size circuits may invert F' on $F'(U_n)$ with success probability exceeding* $1-\Omega(1/\sqrt{n})$.

Proof. Intuitively, if the potential inverter has success probability exceeding $\Pr[\mathrm{wt}(U_n) \neq \lfloor n/2 \rfloor]$, then the excess must be due to preimages that reside in the middle slice. But since F' agrees with F on the middle slice, this excess translates to a success probability of inverting F.

The actual proof follows by using a standard reducibility argument. Specifically, suppose that algorithm A inverts $F'(U_n)$ with success probability at least $1-\rho(n)+\varepsilon(n)$, where $\rho(n)=\Pr[\mathrm{wt}(U_n) = \lfloor n/2 \rfloor] = \Omega(1/\sqrt{n})$. For simplicity, assume first that neither 0^n nor 1^n is in the image of F, which implies that for every $x \in \{0,1\}^n$ such that $\mathrm{wt}(x)=\lfloor n/2 \rfloor$ it holds that ${F'}^{-1}(y) \subseteq F^{-1}(y)$, where $y=F'(x)=F(x)$. Then, it must be that

$$\begin{aligned}
\Pr\left[A(F(U_n)) \in F^{-1}(F(U_n))\right] &\geq \Pr\left[A(F(U_n)) \in F^{-1}(F(U_n)) \wedge \mathrm{wt}(U_n) = \lfloor n/2 \rfloor\right] \\
&\geq \Pr\left[A(F'(U_n)) \in {F'}^{-1}(F'(U_n)) \wedge \mathrm{wt}(U_n) = \lfloor n/2 \rfloor\right] \\
&\geq \Pr\left[A(F'(U_n)) \in {F'}^{-1}(F'(U_n))\right] - \Pr\left[\mathrm{wt}(U_n) \neq \lfloor n/2 \rfloor\right] \\
&\geq \varepsilon(n)\,.
\end{aligned}$$

Thus, if A is efficient then ε must be negligible, otherwise we reach a contradiction to the hypothesis that F is (strongly) one-way. The simplifying assumption (regarding the image of F) may be avoided by noting that for any one-way function F it holds that $\Pr[F(U_n) \in \{0^n, 1^n\}]$ is negligible.[2] The proposition follows. $\square$

Conclusion: It follows that F' is an $\Omega(1/\sqrt{n})$-hard OWF that is computable by (uniform) polynomial-size monotone circuits. Applying the standard hardness amplification process (i. e., letting

$$F''(z) = F'(z_{[1,m]})F'(z_{[m+1,2m]}) \cdots F'(z_{[(m-1)m+1,m^2]}),$$

where $|z| = m^2$ and $z_{[i,j]} = z_i \cdots z_j$ for $i < j$), completes the proof of Theorem 1.1. $\square$

4 No PRGs are monotone

In this section, we prove Theorem 1.2. Intuitively, we prove that any *monotone* function that stretches its input either has a biased output bit or has two output bits that are correlated in a noticeable way. In each of these two cases, we obtain a very simple circuit that distinguishes the output of the function from a random sequence of the same length.

4.1 Technical background

We start by defining the core concepts.

Definition 4.1 (ε-biased function)**.** Let $b : \{0,1\}^n \to \{0,1\}$ be a Boolean function. We say that b is *ε-biased* if

$$|\Pr_x[b(x) = 1] - \Pr_x[b(x) = 0]| \leq 2\varepsilon\,. \tag{1}$$

We say that b is **unbiased** if it is 0-biased.

Note that equation (1) can be written as $|\Pr_x[b(x) = 1] - 1/2| \leq \varepsilon$.

Definition 4.2 (influence [8])**.** Let $b : \{0,1\}^n \to \{0,1\}$ be a Boolean function. We define the **influence** of the i^{th} input bit on b as

$$I_b(i) = \Pr_x\big[b(x) \neq b(x \oplus 0^{i-1}10^{n-i})\big]\,.$$

We next recall two fundamental results regarding these concepts. The first is a theorem proved by Kahn, Kalai, and Linial [8].

Theorem 4.3. *Let $b : \{0,1\}^n \to \{0,1\}$ be an unbiased Boolean function. Then, there exists an integer $i \in [n]$ such that $I_b(i) = \Omega((\log n)/n)$.*

(Here as well as in the sequel, n is viewed as a variable.) An almost immediate corollary of Theorem 4.3 is the following:

Corollary 4.4. *Let $b : \{0,1\}^n \to \{0,1\}$ be an $o((\log n)/n)$-biased Boolean function. Then, there exists an integer $i \in [n]$ such that $I_b(i) = \Omega((\log n)/n)$.*

[2] Alternatively, it suffices to prove the proposition for functions F that satisfy the simplifying assumption.

Proof. Let $b' : \{0,1\}^n \to \{0,1\}$ be an unbiased Boolean function that is closest to b (i. e., for which $\Pr_x[b'(x) \neq b(x)]$ is minimal). Then, $\Pr_x[b(x) \neq b'(x)] = o((\log n)/n)$. By Theorem 4.3, there exists an integer i, such that $I_{b'}(i) = \Omega((\log n)/n)$. Using

$$\begin{aligned} I_b(i) &= \Pr_x[b(x) \neq b(x \oplus 0^{i-1}10^{n-i})] \\ &\geq \Pr_x[b'(x) \neq b'(x \oplus 0^{i-1}10^{n-i})] - 2 \cdot \Pr_x[b(x) \neq b'(x)] \\ &= I_{b'}(i) - 2 \cdot \Pr_x[b(x) \neq b'(x)], \end{aligned}$$

the claim follows. □

The following theorem was proved by Talagrand [10].

Theorem 4.5. *For some universal constant $c > 0$, we consider the function $\varphi(t) = c \cdot t/\log(e/t)$. Then, for all n and all* monotone *functions $f, g : \{0,1\}^n \to \{0,1\}$, it holds that*

$$\Pr_x[f(x) = 1 \wedge g(x) = 1] - \Pr_x[f(x) = 1] \cdot \Pr_x[g(x) = 1] \geq \varphi\left(\sum_{i \in [n]} I_f(i) \cdot I_g(i)\right). \tag{2}$$

It is crucial that the functions considered here are monotone; indeed, the claim fails for general functions (e. g., consider any pair of distinct linear functions).

4.2 Proof of Theorem 1.2

We now prove Theorem 1.2. Fix any n. Let $f_1, \ldots, f_{n+1}$ be the output bits of f. We shall look at this sequence as a sequence of $n+1$ monotone Boolean functions; that is, functions of the n input variables $x_1, \ldots, x_n$.

If there exists i such that f_i is not $1/n^2$-biased (i. e., $|\Pr[f_i(U_n) = 1] - 1/2| > 1/n^2$), then we consider the (monotone $\mathcal{NC}^0$) distinguisher $D_i : \{0,1\}^{n+1} \to \{0,1\}$ defined by $D_i(z) = z_i$ (i. e., the i^{th} bit of z), and observe that

$$\left|\Pr[D_i(U_{n+1}) = 1] - \Pr[D_i(f(U_n)) = 1]\right| = \left|\frac{1}{2} - \Pr[f_i(U_n) = 1]\right| > 1/n^2.$$

Otherwise (i. e., each f_i is $1/n^2$-biased), by Corollary 4.4, for each f_i, there exists a corresponding input variable x_j such that $I_{f_i}(j) = \Omega((\log n)/n)$. Then, there exist two output indexes $i_1, i_2 \in [n+1]$ and an input index $j \in [n]$ such that both $I_{f_{i_1}}(j) = \Omega((\log n)/n)$ and $I_{f_{i_2}}(j) = \Omega((\log n)/n)$. Therefore, by Theorem 4.5, we get:

$$\begin{aligned} &\Pr_x[f_{i_1}(x) = 1 \wedge f_{i_2}(x) = 1] - \Pr_x[f_{i_1}(x) = 1] \cdot \Pr_x[f_{i_2}(x) = 1] \\ &\geq \varphi\left(\sum_{k \in [n]} I_{f_{i_i}}(k) \cdot I_{f_{i_j}}(k)\right) \\ &\geq \varphi\left(I_{f_{i_1}}(j) \cdot I_{f_{i_2}}(j)\right) \\ &= \varphi\left(\Omega\left(\frac{\log n}{n}\right)^2\right) \end{aligned}$$

which is $\Omega((\log n)/n^2)$. Then, for the (monotone $\mathcal{NC}^0$) distinguisher $D_{i_1,i_2} : \{0,1\}^{n+1} \to \{0,1\}$ defined by $D_{i_1,i_2}(z) = z_{i_1} \wedge z_{i_2}$, we get

$$\begin{aligned}
&\left|\Pr\left[D_{i_1,i_2}(U_{n+1}) = 1\right] - \Pr\left[D_{i_1,i_2}(f(U_n)) = 1\right]\right| \\
&= \left|\frac{1}{4} - \Pr\left[f_{i_1}(U_n) = f_{i_2}(U_n) = 1\right]\right| \\
&\geq \left|\frac{1}{4} - \Pr_x[f_{i_1}(x) = 1] \cdot \Pr_x[f_{i_2}(x) = 1] - \Omega\left(\frac{\log n}{n^2}\right)\right| \\
&\geq \left|\frac{1}{4} - \left(\frac{1}{2} - \frac{1}{n^2}\right)^2 - \Omega\left(\frac{\log n}{n^2}\right)\right| \geq \Omega\left(\frac{\log n}{n^2}\right).
\end{aligned}$$

Thus, for each n either one of the $n+1$ first distinguishers (i. e., the D_i) or one of the $\binom{n+1}{2}$ latter distinguishers (i. e., the D_{i_1,i_2}) distinguishes the output of f from a truly random $n+1$-bit long string. The Theorem follows. □

References

[1] M. AJTAI, J. KOMLÓS, AND E. SZEMERÉDI: An $O(n \log n)$ sorting network. In *Proc. 15th STOC*, pp. 1–9. ACM Press, 1983. [doi:10.1145/800061.808726] 234

[2] BENNY APPLEBAUM, YUVAL ISHAI, AND EYAL KUSHILEVITZ: Cryptography in NC^0. *SIAM J. Comput.*, 36:845–888, 2006. [doi:10.1137/S0097539705446950] 232

[3] S. J. BERKOWITZ: On some relationships between monotone and non-monotone circuit complexity. Technical report, University of Toronto, 1982. 234

[4] MANUEL BLUM AND SILVIO MICALI: How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Comput.*, 13:850–864, 1984. Preliminary version in FOCS'82. [doi:10.1137/0213053] 232

[5] ODED GOLDREICH: *Foundations of Cryptography: Basic Tools*. Volume 1. Cambridge University Press, 2001. 231, 232, 233

[6] ODED GOLDREICH: *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008. 231

[7] JOHAN HÅSTAD, RUSSELL IMPAGLIAZZO, LEONID A. LEVIN, AND MICHAEL LUBY: A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28:1364–1396, 1999. [doi:10.1137/S0097539793244708] 232

[8] JEFF KAHN, GIL KALAI, AND NATHAN LINIAL: The influence of variables on Boolean functions. In *Proc. 29th FOCS*, pp. 68–80. IEEE Comp. Soc. Press, 1988. [doi:10.1109/SFCS.1988.21923] 235

[9] NOAM NISAN AND AVI WIGDERSON: Hardness vs. randomness. *J. Comput. System Sci.*, 49:149–167, 1994. Preliminary version in FOCS'88. [doi:10.1016/S0022-0000(05)80043-1] 232

[10] MICHEL TALAGRAND: How much are increasing sets positively correlated? *Combinatorica*, 16:243–258, 1996. [doi:10.1007/BF01844850] 236

[11] ANDREW C. YAO: Theory and application of trapdoor functions. In *Proc. 23rd FOCS*, pp. 80–91. IEEE Comp. Soc. Press, 1982. [doi:10.1109/SFCS.1982.95] 232

AUTHORS

Oded Goldreich
professor
Weizmann Institute of Science, Rehovot, Israel
oded.goldreich@weizmann.ac.il
http://www.wisdom.weizmann.ac.il/~oded/

Rani Izsak
Ph. D. student
Weizmann Institute of Science, Rehovot, Israel
ran.izsak@weizmann.ac.il

ABOUT THE AUTHORS

ODED GOLDREICH (b. 1957) is a professor of Computer Science at the Faculty of Mathematics and Computer Science of the Weizmann Institute of Science, Israel. His professional career was greatly influenced by his graduate studies advisor, Shimon Even, and by his post-doctoral period (1983-86) at the theory group of the Laboratory for Computer Science of MIT. Oded feels privileged being a faculty member at Weizmann since 1994, and has enjoyed visits at MIT (1995-98), the Miller Institute of UC-Berkeley (1996), the Radcliffe Institute of Harvard University (2003-4), and the Institute for Advanced Study (2011-12). His research interests include the interplay of randomness and computation, the foundations of cryptography, and computational complexity. He has contributed to the development of pseudorandomness, zero knowledge proofs, secure function evaluation, property testing, and other areas in cryptography and computational complexity. He has authored numerous surveys and several books, including the two-volume work "Foundations of Cryptography" (2001 and 2004) and "Computational Complexity: A Conceptual Perspective" (2008).

RANI IZSAK is a Ph. D. student, supervised by Uri Feige, at the Faculty of Mathematics and Computer Science of the Weizmann Institute of Science, Israel. His research interests include computational complexity theory and the design of algorithms.