# Pseudorandomness for read-once formulas

Andrej Bogdanov
*Dept. of Computer Science and Eng. and ITCSC, Chinese University of Hong Kong*

Periklis A. Papakonstantinou
*ITCS at IIIS, Tsinghua University*

Andrew Wan
*ITCS at IIIS, Tsinghua University*

**Abstract**— We give an explicit construction of a pseudorandom generator for read-once formulas whose inputs can be read in arbitrary order. For formulas in $n$ inputs and arbitrary gates of fan-in at most $d = O(n/\log n)$, the pseudorandom generator uses $(1-\Omega(1))n$ bits of randomness and produces an output that looks $2^{-\Omega(n)}$-pseudorandom to all such formulas.

Our analysis is based on the following lemma. Let $\mathrm{P} = Mz+e$, where $M$ is the parity-check matrix of a sufficiently good binary error-correcting code of constant rate, $z$ is a random string, $e$ is a small-bias distribution, and all operations are modulo 2. Then for every pair of functions $f, g\colon \{0,1\}^{n/2}\to\{0,1\}$ and every equipartition $(I, J)$ of $[n]$, the distribution P is pseudorandom for the pair $(f(x|_I), g(x|_J))$, where $x|_I$ and $x|_J$ denote the restriction of $x$ to the coordinates in $I$ and $J$, respectively.

More generally, our result applies to read-once branching programs of bounded width with arbitrary ordering of the inputs. We show that such branching programs are more powerful distinguishers than those that read their inputs in sequential order: There exist (explicit) pseudorandom distributions that separate these two types of branching programs.

## 1. Introduction

We consider the problem of obtaining an explicit pseudorandom distribution for the class of boolean read-once formulas. A read-once formula on $n$ inputs with fan-in $d$ is a rooted binary tree whose internal nodes called *gates* are labeled by boolean functions on at most $d$ bits. The tree has $n$ leaves that are labeled by the inputs $1, 2, \ldots, n$ so that every input is used exactly once. The labeling of the leaves by inputs can be done in arbitrary order and we do not impose any restriction on the depth of the formula.

A family of distributions $\mathrm{P}\colon \{0,1\}^{s(n)}\to\{0,1\}^n$ is *pseudorandom* with seed length $s(n) < n$ and bias $\epsilon(n)$ for read-once formulas if for every read-once formula $f$ in $n$ inputs,

$$\left|\mathbf{E}_{\mathrm{P}}[f(\mathrm{P})] - \mathbf{E}_{\mathrm{U}}[f(\mathrm{U})]\right| \le \epsilon(n).$$

where $u$ is the uniform distribution on $n$ bits.

The construction of explicit pseudorandom distributions for various models of computation is a central research program in computational complexity. Celebrated examples include generators that fool circuits of constant depth [1], [2], [14], [7] and logarithmic space (more generally space-bounded) machines that read their input once and in a fixed, left-to-right order [13], [11].

Read-once boolean formulas are one of the simplest natural classes of computations for which no efficient construction of pseudorandom generators was previously known. Read-once formulas formulas can compute parities, which are hard for constant-depth circuits. This makes it unlikely that known techniques for constructing pseudorandom generators for constant-depth circuits can be easily extended to handle read-once formulas.

The relation between read-once formulas and read-once branching programs (the nonuniform variant of space restricted computations) is more subtle. As we show in Lemma 1, a read-once formula of constant fan-in can be converted to a read-once branching program of polynomial width, which is the non-uniform analogue of read-once logspace. So why shouldn't Nisan's pseudorandom generator for logarithmic space also apply to read-once formulas? The answer has to do with the ordering of the inputs. Nisan's pseudorandom generator fools branching programs that read their inputs obliviously and in a fixed order. It is not known whether Nisan's pseudorandom generator fools read-once formulas. In fact, Nisan's analysis crucially relies on the fact that the inputs are read in this order. The intuition behind Nisan's analysis is that after reading the inputs in the first half, owing to the space restriction the computation must "forget" most information about these inputs, and these can be recycled by applying an appropriate extractor (this intuition can be made precise [16]). In contrast, our pseudorandom generator fools read-once branching programs that work under an arbitrary ordering of the inputs. The construction of pseudorandom generators against more complex models of computation (which include read-once formulas) has applications in the derandomization of polynomial time with limited access to randomness [10].

If instead of pseudorandomness we consider worst-case hardness then there is a long line of research in lower bounds, including generalizations of read-once branching programs. For read-once formulas such a lower bound can be obtained directly through standard communication complexity reductions to functions that are hard in the best-

A.B. is partially supported by RGC GRF grant CUHK410309. P.A.P. and A.W. are partially supported by the National Basic Research Program of China Grant 2007CB807900, 2007CB807901, the National Natural Science Foundation of China Grant 61050110452, 61150110163, 61033001, 61061130540, 61073174.

case partition, two-party model [15].

In this work we give an explicit construction of a pseudorandom distribution for read-once formulas. Our savings in randomness are rather modest, although our construction allows almost-linear fan-in and gives exponentially small error. As the model of read-once formulas brings out new issues in the design of pseudorandom generators, we believe that the real contribution of our work lies in the techniques used in our construction and analysis rather than in the result itself.

**Theorem 1.** *There exists constants $\rho, K > 0$ and an explicit pseudorandom distribution family $\mathrm{P}\colon \{0,1\}^{(1-\rho)n} \to \{0,1\}^n$ so that for every $n$,*

$$\left|\mathbf{E}_{\mathrm{P}}[f(\mathrm{P})] - \mathbf{E}_{\mathrm{U}}[f(\mathrm{U})]\right| = 2^{-\Omega(n)}$$

*for every read-once formula $f$ over $n$ inputs of fan-in at most $n/K \log n$.*

Our pseudorandom distribution is quite simple: It has the form $\mathrm{P} = Mz + e$, where $M$ is the parity check matrix of a binary linear list-decodable code of linear rate and list-decoding radius bounded away from $1/4$, $z$ is a uniformly random string, and $e$ is a small-bias distribution [12]. By "explicit" we mean that there is an algorithm that takes as inputs $1^n$ and a random seed of length $(1-\rho)n$ and outputs a sample from the distribution P in time polynomial in $n$.

We also prove that, in general, allowing arbitrary ordering of the inputs makes branching programs into more powerful distinguishers: We give examples of pseudorandom distributions that fool all branching programs that read their input bits sequentially, but not those that have read-once access to their input in arbitrary order. Recently, we've become aware of independent, work which appears in Yoav Tzur's MSc thesis [18], showing that Nisan's generator does not work under arbitrary ordering of the inputs.

*Techniques:* To give some intuition for the proof of Theorem 1, let's assume $n$ is even and we have a read-once formula $F$ with fan-in $d = 2$ which is a full binary tree of depth $\log_d n$. Consider the root gate of this formula and let $f$ and $g$ be the functions computed by the left and right subtrees, respectively. Then a sufficient condition for the output of $F$ to be pseudorandom is the following one:

> For every pair of read-once formulas $f, g\colon \{0,1\}^{n/2} \to \{0,1\}$ and every equipartition $(I, J)$ of $[n]$, the joint distribution $(f(\mathrm{P}|_I), g(\mathrm{P}|_J))$ is close (in statistical distance) to the joint distribution $(f(\mathrm{U}|_I), g(\mathrm{U}|_J))$.

Our analysis proceeds by forgetting that $f$ and $g$ are themselves read-once formulas and treating them as arbitrary functions. At this level of generality for the choice of $f$ and $g$, it is clear that the pseudorandom distribution P cannot have much fewer than $n/2$ bits of entropy.

To understand the elements that go into our construction, we make two observations. Our first observation is that in order to fool the joint distribution $(f(x|_I), g(x|_J))$, for every equipartition $(I, J)$ of $[n]$, it is at least necessary to fool the marginal distribution $f(x|_I)$ for every subset $I$ of $[n]$ of size $n/2$. This suggests choosing P to be an $(n/2)$-wise independent distribution. One way to obtain such a distribution is to set $\mathrm{P} = Mz$, where $M$ is an $n \times k$ ($k < n$) matrix whose every $(n/2) \times k$ minor has full rank, and $z \in \{0,1\}^k$ is uniformly random.

Our second observation is that since the collection of functions $f(x|_I) + g(x|_J)$ in particular includes all parities over subsets of $[n]$ (in fact, all such parities can be computed by read-once formulas), the distribution P must be a small-bias distribution.

Taken together, these two observations suggest the pseudorandom distribution $\mathrm{P} = Mz + e$, where $M$ and $z$ are as above and $e$ is a small-bias distribution. Using Fourier analysis, it is not difficult to show this distribution is indeed pseudorandom.

One issue with this construction is that it may not provide any savings in randomness at all: To obtain a matrix $M$ such that every $(n/2) \times k$ minor has full rank, by the Plotkin bound one needs $k = n - O(1)$. However, it turns out that the analysis still works if every $(n/2) \times k$ minor of $M$ has *almost* full rank; for a precise statement see Lemma 2. Under this relaxed property, it is possible to obtain an explicit matrix $M$ of dimensions $n \times (1 - \Omega(1))n$ using known constructions of binary error-correcting codes; see Proposition 1.

To extend our analysis to general read-once formulas (with larger fan-in and without restrictions on the shape of the formula tree), in Lemma 1 we show that a read-once formula with fan-in $d$ can be converted into a read-once branching program of width $n^{O(d)}$. The celebrated result of Barrington [4] shows that any polynomial size formula can be represented by polynomial size width-5 branching programs (not read-once), and later results [8], [9], [17] reduced the total size of the branching program to $s^{1.2}$ if $s$ is the size of the formula. In Lemma 1, we do not attempt to optimize the width or size of the program, but we need to maintain the property that each variable is read only once. Lemma 3 shows that such branching programs are indeed fooled by the pseudorandom distribution P.

## 2. Read-once formulas and branching programs

A boolean *read-once formula* on $n$ inputs with fan-in $d$ is a rooted binary tree with branching factor at most $d$ and exactly $n$ leaves. The leaves of this tree are labeled by the inputs $1, 2, \ldots, n$, where each input is used exactly once, and each internal node $g$ with $d_g$ children is labeled by a boolean function $g\colon \{0,1\}^{d_g} \to \{0,1\}$ called a *gate*. A read-once formula represents a boolean function $f\colon \{0,1\}^n \to \{0,1\}$ in the natural manner: The inputs of this function are placed at the leaves of the tree with corresponding labels and each gate is evaluated inductively. The output of $f$ is the value at the root.

A boolean *read-once branching program* on $n$ inputs and width $w$ is a layered sequence of $n$ directed bipartite graphs whose left and right vertices come from the set of states $\{1,\ldots,w\}$. In each of these graphs, each left vertex has exactly two outgoing edges, one labeled by 0 and the other labeled by 1. Each layer in the sequence is labeled by a unique input bit coming from the set $\{1,\ldots,n\}$. There is a special vertex $s$ in the initial layer called the start vertex and a special vertex $t$ in the last layer called the final vertex. A read-once branching program computes a boolean function $f\colon \{0,1\}^n \to \{0,1\}$ as follows: Starting at $s$, follow the unique path along the sequence of graphs that is indexed by the values of the input bits. If this path reaches $t$ output 1, otherwise output 0.

**Lemma 1.** *Suppose $f$ is computable by a read-once formula on $n$ inputs with fan-in $d$. Then $f$ is computable by a read-once branching program on $n$ inputs of width $n^{O(d)}$.*

We stress that the order in which the inputs are read in the branching program may differ from the order in which the inputs are read when the formula is evaluated in the natural way – depth-first, left-to-right.

*Proof:* Let $s_d(n)$ be the maximum width of a read-once branching program that computes a read-once formula on $n$ inputs with fan-in $d$, and let $F$ be any read-once formula on $n$ inputs with fan-in $d$. Starting at the root node $v$, we iteratively select the sub-formula rooted at one of the children of $v$ with the largest number of leaves, until we end up with a sub-formula $F'$ of size at most $dn/(d+1)$. Let $F_x$ be the formula obtained by replacing the subformula $F'$ from $F$ by a fresh new input $x$.

We argue that both formulas $F_x$ and $F'$ have at most $\lceil dn/(d+1)\rceil$ leaves, and hence each may be computed by a read-once branching program of width $s_d(\lceil dn/(d+1)\rceil)$. The bound on $F'$ follows by construction. On the other hand, we also know that $F'$ as the largest child must have more than $n/(d+1)$ leaves, otherwise its parent would have at most $dn/(d+1)$ leaves and would have been selected by the procedure. Then $F_x$ has fewer than $n - n/(d+1) + 1 = dn/(d+1) + 1$ leaves (the extra leaf comes from the new input $x$).

Now consider the following read-once branching program for computing $F$: First, run the branching program for $F'$. Then run two parallel copies of the branching program for $F_x$, corresponding to the cases when $F'$ evaluates to 0 and $F'$ evaluates to 1, respectively. When the input $x$ in these branching programs is to be read, replace the value of $x$ by the corresponding output of $F'$. If $F'$ evaluates to true on its input, the input proceeds along the copy $F_{x=1}$, otherwise it proceeds along the copy $F_{x=0}$.

The width of the branching program for $F$ is the larger of the width of the branching program for $F'$ and twice the width of the branching program for $F_x$. Since $F$ was arbitrary, we obtain the recursive relation

$$s_d(n) \le 2 s_d(\lceil dn/(d+1)\rceil).$$

which solves to $s_d(n) = n^{O(d)}$. ∎

## 3. The pseudorandom generator

We say a linear code $C$ over $\{0,1\}^n$ is $(\delta,\ell)$ *list-decodable* if for every $x \in \{0,1\}^n$, the number of codewords of $C$ within hamming distance $\delta n$ of $x$ is at most $\ell$. A *parity check matrix* $M$ for $C$ is a $GF(2)$ matrix such that $c^T M = 0$ if and only if $c$ is a codeword of $C$.

**Proposition 1.** *Let $C$ be a $(\frac{1}{4},\ell)$ list-decodable code over $\{0,1\}^n$, where $n$ is even and $\gamma > 0$. Let $M$ be the parity check matrix of $C$. Then every subset of $n/2$ rows of $M$ has dimension at least $n/2 - \log_2(2\ell)$ (as a vector space over $GF(2)$).*

*Proof:* Let $r = \log_2(2\ell)$. Assume for contradiction that there exists a subset $S$ of $n/2$ rows of dimension less than $n/2 - r$. Then there exist more than $2^r$ vectors $V$ such that for every $v \in V$, $v^T M = 0$ and $v_i = 0$ for every $i \notin S$, i.e., the vectors in $V$ are in $C$ and all have distance at most $n/2$ from each other.

It is easy to see that there is a vector within Hamming distance $n/4$ on at least $|V|/2$ of them. Let $\overline{v}$ be 0 on the coordinates outside of $S$ and uniformly random on the coordinates in $S$. Then the distance between any $v \in V$ and $\overline{v}$ is the sum of $n/2$ independent unbiased Bernoulli random variables, and so it exceeds $n/4$ with probability at most $1/2$. Therefore the expected number of $v \in V$ whose distance to $\overline{v}$ exceeds $n/4$ is at most $|V|/2$. So there must exist a choice of $\overline{v}$ that is within distance $n/4$ from at least half the vectors in $V$.

Since $|V|/2 > \ell$ and $V \subseteq C$, this contradicts the fact that $C$ is $(\frac{1}{4},\ell)$ list-decodable. ∎

We now consider the following construction:

1) Let $M$ be a $n \times k$ matrix over $GF(2)$ such that every subspace spanned by $n/2$ rows has dimension $n/2 - r$, and let $z \sim \{0,1\}^k$ be a uniformly random vector.
2) Let $e \in \{0,1\}^m$ be chosen independently of $z$ from an $\epsilon$-biased distribution. We say that a distribution $D$ over $\{0,1\}^n$ is $\epsilon$-biased if for every $S \subseteq [n], S \neq \emptyset$,
$$\left|\mathbf{E}_{e\sim D}\left[(-1)^{\langle s,e\rangle}\right]\right| \le \epsilon.$$
3) Let $\mathrm{P}$ be the (pseudorandom) distribution over $\{0,1\}^n$ defined by
$$\mathrm{P} = Mz + e \tag{1}$$
where the addition is modulo 2.

**Lemma 2.** *Assume $n$ is even. Then for every partition $I, J$ of $[n]$ with $|I| = |J| = n/2$ (where $I$ and $J$ are ordered sets) and every pair of functions $f, g\colon \{0,1\}^{n/2} \to [-1,1]$,*

$$\left|\mathbf{E}_{\mathrm{P}}[f(\mathrm{P}|_I) g(\mathrm{P}|_J)] - \mathbf{E}_{\mathrm{U}}[f(\mathrm{U}|_I) g(\mathrm{U}|_J)]\right| \le 2^r \epsilon$$

*where $u$ is the uniform distribution over $\{0,1\}^n$, $p$ is defined as above, and $x|_I, x|_J$ denote the projections of $x$ on the sets $I$ and $J$, respectively.*

In particular, when $g = 1$, $|\mathbf{E}_{\mathrm{P}}[f(\mathrm{P}|_I)] - \mathbf{E}_{\mathrm{U}}[f(\mathrm{U}|_I)]| \leq 2^r\epsilon$, so the pseudorandom distribution also preserves the marginal probabilities of events (within $2^r\epsilon$) over all subsets of size $n/2$.

*Proof:* For any partition $(I, J)$ of $[n]$ with $|I| = |J| = n/2$, we have

$$\begin{aligned}
&\mathbf{E}_{\mathrm{P}}[f(\mathrm{P}|_I)g(\mathrm{P}|_J)] = \mathbf{E}_{z,e}[f((Mz+e)|_I)g((Mz+e)|_J)] \\
&= \sum_{\substack{S\subseteq I\\ T\subseteq J}} \hat{f}(S)\hat{g}(T)\mathbf{E}_{z,e}[\chi_S(Mz|_I)\chi_S(e|_I)\chi_T(Mz|_J)\chi_T(e|_J)] \\
&= \sum_{S\subseteq I, T\subseteq J} \hat{f}(S)\hat{g}(T)\mathbf{E}_z[\chi_S(Mz|_I)\chi_T(Mz|_J)]\mathbf{E}_e[\chi_{S\cup T}(e)].
\end{aligned}$$

The term $S = T = \varnothing$ contributes $\hat{f}(\varnothing)\hat{g}(\varnothing) = \mathbf{E}_u[f(u|_I)g(u|_J)]$ to the summation. Since $e$ is an $\epsilon$-biased distribution, whenever $S\cup T \neq \varnothing$ we have $|\mathbf{E}_e[\chi_{S\cup T}(e)]| \leq \epsilon$. Therefore

$$\begin{aligned}
&\left|\mathbf{E}_{\mathrm{P}}[f(\mathrm{P}|_I)g(\mathrm{P}|_J)] - \mathbf{E}_{\mathrm{U}}[f(\mathrm{U}|_I)g(\mathrm{U}|_J)]\right| = \\
&\quad\left|\sum_{S\subseteq I, T\subseteq J, S\cup T\neq\varnothing} \hat{f}(S)\hat{g}(T)\mathbf{E}_z[\chi_S(Mz|_I)\chi_T(Mz|_J)]\right. \\
&\qquad\qquad\left.\mathbf{E}_e[\chi_{S\cup T}(e)]\right| \\
&\leq \sum_{S\subseteq I, T\subseteq J} \epsilon\cdot|\hat{f}(S)||\hat{g}(T)|\left|\mathbf{E}_z[\chi_S(Mz|_I)\chi_T(Mz|_J)]\right|.
\end{aligned}$$

Let $G$ be a bipartite graph over vertices (subsets of $I$) $\cup$ (subsets of $J$), with an edge $(S,T)$ present whenever $\mathbf{E}_z[\chi_S(Mz|_I)\chi_T(Mz|_J)] \neq 0$. We will shortly argue that $G$ has maximum degree $2^r$. Assuming this, we can upper bound the last expression by

$$\begin{aligned}
\epsilon\cdot\sum_{\substack{\text{edge}\\(S,T)}}|\hat{f}(S)||\hat{g}(T)| &\leq \epsilon\cdot\sqrt{\sum_{\substack{\text{edge}\\(S,T)}}\hat{f}(S)^2}\sqrt{\sum_{\substack{\text{edge}\\(S,T)}}\hat{g}(T)^2} \\
&\leq \epsilon\cdot\sqrt{2^r\sum_{S\subseteq I}\hat{f}(S)^2}\sqrt{2^r\sum_{T\subseteq J}\hat{g}(T)^2} \\
&\leq \epsilon\cdot 2^r,
\end{aligned}$$

where the first line follows from the Cauchy-Schwarz inequality, the second line follows from the fact that $G$ has maximum degree $2^r$, and the third line is an application of Parseval's identity.

It remains to argue that $G$ has maximum degree $2^r$. Let $s \in \{0,1\}^I, t \in \{0,1\}^J$ be indicator vectors for the sets $S$ and $T$, respectively, and $s\circ t \in \{0,1\}^n$ be the vector obtained by concatenating $s$ and $t$ according to the partition $(I, J)$ of $[m]$. Then

$$\begin{aligned}
\mathbf{E}_z[\chi_S(Mz|_I)\chi_T(Mz|_J)] &= \mathbf{E}[(-1)^{(s\circ t)^T Mz}] = \\
&= \begin{cases} 1, & \text{if } (s\circ t)^T M = 0 \\ 0, & \text{otherwise.}\end{cases}
\end{aligned}$$

Now $(s\circ t)^T M = 0$ if and only if $s^T M|_I = t^T M|_J$, where $M|_I$ and $M|_J$ are the minors of $M$ restricted to the rows indexed by $I$ and $J$, respectively. Since (by assumption) the matrix $M|_I$ has rank at least $n/2 - r$, for every $t \in \{0,1\}^J$, there can be at most $2^r$ vectors $s \in \{0,1\}^I$ such that $s^T M|_I = t^T M|_J$. By an analogous argument, for every $s \in \{0,1\}^I$, there can be at most $2^r$ vectors $t \in \{0,1\}^J$ such that $s^T M|_I = t^T M|_J$. ∎

*Parameters:* Combining Proposition 1 and Lemma 2 , we have that if $C$ is a $(1/4, 2n)$ list-decodable code with parity check matrix $M$, then P is a $4n\epsilon$ pseudorandom distribution in the sense of Lemma 2. By the Johnson bound, any $C$ with minimum distance $3/8$ is $(1/4, 2n)$ list-decodable.

Using standard constructions, for example [3], for some constant $\rho > 0$ and sufficiently large $n$, we can obtain explicit families of linear codes over $GF(2)$ of rate $2\rho n$ and minimum distance $3/8$. If $M$ is the parity check matrix of such a code, then P has seed length $(1-2\rho)n + sl(n,\epsilon)$, where $sl(n,\epsilon)$ is the seed length of the $\epsilon$-biased distribution $e$. With the construction from [12] we can achieve $sl(n,\epsilon) = O(\log(n/\epsilon))$. Choosing $\epsilon$ so that $sl(n,\epsilon) = \rho n$ (we can use the construction in [3] for a better constant), we obtain an explicit pseudorandom generator P of seed length $(1-\rho)n$ so that for for every partition $I, J$ of $[n]$ with $|I| = |J| = n/2$ and every pair of functions $f, g\colon \{0,1\}^{n/2}\to\{-1,1\}$,

$$\left|\mathbf{E}_{\mathrm{P}}[f(\mathrm{P}|_I)g(\mathrm{P}|_J)] - \mathbf{E}_{\mathrm{U}}[f(\mathrm{U}|_I)g(\mathrm{U}|_J)]\right| = 2^{-\Omega(n)}.$$

*Remarks:* Our proof can be extended to obtain the following more general statement: For every $\alpha > 0$ there exists a $\rho > 0$ and an explicit matrix $M$ of size $n\times(1-\rho)n$ so that for every pair of (possibly intersecting) sets $I, J$ with $|I|, |J| \leq (1-\alpha)n$ and for every pair of functions $f, g$, the conclusion of Lemma 2 holds.

We observe that when the sets $I$ and $J$ are fixed, the conclusion of Lemma 2 can be achieved by choosing $(\mathrm{P}|_I, \mathrm{P}|_J)$ to be the endpoints of a random edge in an expander graph on $n/2$ vertices with eigenvalue gap $1-\epsilon$. (Lemma 2 then becomes simply the expander mixing lemma.) The seed length of this construction is $n/2 + O(\log 1/\epsilon)$ (if a constant-degree expander family is used), which is essentially optimal. It may be interesting to investigate how closely these parameters can be matched in our more general setting.

## 4. Pseudorandomness for read-once formulas

We now show that the pseudorandom generator constructed in the previous section fools read-once formulas. First, we apply Lemma 1 to obtain a read-once branching

program having width $n^{O(d)}$. We then apply the following lemma, which shows that the generator fools $\mathrm{poly}(n)$-width read-once branching programs (where the inputs may be read in any order).

**Lemma 3.** *Assume $n$ is even and $F : \{0,1\}^n \to \{1,-1\}$ is computable by a width-$w$ read-once branching program on $n$ inputs. Let* P *be the distribution defined in (1). Then*

$$|\mathbf{E}_{\mathrm{P}}[F(\mathrm{P})] - \mathbf{E}_{\mathrm{U}}[F(\mathrm{U})]| \leq w \cdot 2^d \epsilon.$$

*Proof:* We view the computation of the branching program for $F$ in two stages; the first stage follows the input from the start state to the middle ($n/2$-nd) layer, and the second stage proceeds from the middle layer to the final layer.

Let $I$ be the ordered set of variables read from the first to the middle layer and $J$ be the ordered set of variables in remaining layers. Then we can define a family of functions $\{f_q\}$ over the bits in $I$ for each state $q$ of the middle layer. For $x \in \{0,1\}^n$ let $f_q(x|_I)$ be an indicator for the event that $x|_I$ leads from the start state to state $q$ and $g_q(x|_J)$ be an indicator for the event that $x|_J$ leads from state $q$ to the final state. Observe that $F(x)$ evaluates to 1 if and only if the input $x$ induces a path $s \to q \to t$ for some state $q$ in the middle layer. Therefore $F(x) = \sum_{q \text{ in middle layer}} f_q(x|_I) g_q(x|_J)$, and by Lemma 2 and the triangle inequality $|\mathbf{E}_{\mathrm{P}}[F(\mathrm{P})] - \mathbf{E}_{\mathrm{U}}[F(\mathrm{U})]| \leq w 2^d \epsilon$. ∎

We now prove our main theorem.

*Proof of Theorem 1:* Let $f$ be a read-once formula over $n$ inputs of fan-in $d$. By Lemma 1, $f$ can be computed by a read-once branching program of width $n^{O(d)}$. By Lemma 3, the pseudorandom distribution (1) fools this branching program with bias $2^r n^{O(d)} \epsilon$. Plugging in the parameters from Section 3 and optimizing for $d$, we derive the theorem. ∎

## 5. A SEPARATION BETWEEN SEQUENTIAL AND RANDOM ACCESS

We now give an example of a pseudorandom distribution that fools read-once branching programs under sequential, read-once access of input bits, but fails to fool such programs under arbitrary read-once access. An interesting aspect of this construction is its generality; we construct the pseudorandom generator starting from an arbitrary pseudorandom generator that fools a sequential read-once branching program (e.g. Nisan's generator).

Let P be any distribution over $n$ bits, where $n$ is a power of two. Define the following distribution $\overline{\mathrm{P}}$ over strings of length $n + \log n$: Independently obtain a sample $x$ from P and a number $i \in [n]$, replace the $i$th entry of $x$ by 1, and output the concatenated pair $(x, i)$.

No matter what P is, the distribution $\overline{\mathrm{P}}$ is clearly distinguishable from uniform by a branching program with arbitrary read-once access of width $O(n^2)$. On the other hand:

**Lemma 4.** *If* P *is $(O(\sqrt{(\log w)/n}), w)$-pseudorandom against sequential access read-once branching programs of width $w$, then $\overline{\mathrm{P}}$ is $(O(\sqrt{(\log w)/n}), w)$ pseudorandom against such branching programs.*

Here "$(\epsilon, w)$-pseudorandom" means that no sequential, read-once branching program of width $w$ has distniguishing advantage greater than $\epsilon$.

When $\log w = o(n/\log n)$, the output of Nisan's pseudorandom generator satisfies the hypothesis, giving an explicit example of a pseudorandom distribution that is $O(\sqrt{(\log w)/n})$ pseudorandom against fixed-order branching programs of width $w$ but not $1/4$-pseudorandom against arbitrary read-once access branching programs of the same width.

We now prove the lemma. Let U be the uniform distribution on $n$ bits and $\overline{\mathrm{U}}$ be the distribution on $n + \log n$ bits obtained by sampling $(x, i)$ uniformly and changing the $i$th bit of $x$ to equal 1. The lemma follows immediately from the next two claims.

**Claim 1.** *If* P *and* U *are $(\epsilon, w)$-indistinguishable, then $\overline{\mathrm{P}}$ and $\overline{\mathrm{U}}$ are $(\epsilon, w)$ indistinguishable.*

*Proof:* Let $\overline{D}$ be a distinguisher such that

$$\Pr_{(x,i)\sim\overline{\mathrm{P}}}[\overline{D}(x,i) = 1] - \Pr_{(x,i)\sim\overline{\mathrm{U}}}[\overline{D}(x,i) = 1] > \epsilon.$$

In particular, there must exist a value $i = a$ for which

$$\Pr_{(x,i)\sim\overline{\mathrm{P}}}[\overline{D}(x,i) = 1 \mid i = a] - \Pr_{(x,i)\sim\overline{\mathrm{U}}}[\overline{D}(x,i) = 1 \mid i = a] > \epsilon.$$

Now consider the distinguisher $D(x) = \overline{D}(x^a, a)$, where $x^a$ is $x$ with its $a$th bit replaced by 1. Clearly

$$\Pr_{x\sim pr}[D(x) = 1] = \Pr_{(x,i)\sim\overline{\mathrm{P}}}[\overline{D}(x,i) = 1 \mid i = a] \qquad \text{and}$$
$$\Pr_{x\sim u}[D(x) = 1] = \Pr_{(x,i)\sim\overline{\mathrm{U}}}[\overline{D}(x,i) = 1 \mid i = a].$$

Moreover, if $\overline{D}$ is a width $w$ branching program, then so is $D$, contradicting our assumption. ∎

**Claim 2.** *The distribution $\overline{\mathrm{U}}$ is $(O(\sqrt{(\log w)/n}), w)$-indistinguishable from uniform.*

*Proof:* Assume $D$ is a branching program of width $w$ that distinguishes $\overline{\mathrm{U}}$ from the uniform distribution with advantage $\epsilon$. Then $\Pr[D(x^a, i) = 1] - \Pr[D(x, i) = 1] > \epsilon$, where $x$ and $i$ are uniformly random. Manipulating this expression, we obtain that $\Pr[D(x, i) = x_i] > 1/2 + \epsilon$. Letting $S \in [w]$ denote the state of $D$ after reading $x$ and $E$ denote the event $D(x, i) = x_i$, we have

$$\mathrm{H}(E) \geq \mathrm{H}(E \mid S) = \mathrm{H}(x_I \mid S) = \mathbf{E}_{i\sim[n]}[H(x_I \mid S, I = i)]$$
$$\geq \frac{1}{n}\mathrm{H}(X \mid S) \geq \frac{n - \log w}{n}$$

where H is Shannon entropy. Since $\Pr[E] > 1/2 + \epsilon$, $\mathrm{H}(E) > 1 - \frac{2}{\ln 2}\epsilon^2$, and $\epsilon = O(\sqrt{(\log w)/n})$. ∎

Claim 2 follows from the general round-elimination lemma in communication complexity; we provide a self-contained proof for our special case. We remark that the bound in Lemma 4 is tight and can be matched by a distinguisher that divides $x$ into $\log w$ consecutive blocks of equal length and outputs the majority value of the $i$th block.

## 6. Conclusion

We gave a construction of an explicit pseudorandom distribution P of seed length $(1-\Omega(1))n$ that fools read-once formulas in $n$ variables of fan-in $d = o(n/\log n)$ with bias $2^{-\Omega(n)}$. A natural question is whether a smaller seed length can be achieved for read-once formulas of, say, constant fan-in and constant bias.

The technical lemma that powers our result says that P fools the distribution $(f(x|_I), g(x|_J))$ for every pair of functions $f, g\colon \{0,1\}^{n/2} \to \{0,1\}$ and every equipartition $(I, J)$ of $[n]$. In the case of a read-once formula, the functions $f$ and $g$ have additional structure – they are themselves read-once formulas or read-once branching programs of small width – which we do not take advantage of in our analysis.

This observation suggests that better parameters may possibly be obtained via recursive composition, along the lines of Nisan's pseudorandom generator for small space [13]. However, we were unable to show that our construction can be composed in this manner. Even for two levels of composition, Lemma 2 does not appear to be useful for arguing pseudorandomness at the bottom level of the construction. Sequential composition methods (e.g. [5]) do not appear to yield constructions that can be readily analyzed either.

Another direction that may be interesting to pursue would be to construct a pseudorandom generator for space-bounded machines that read every input once, but in an adaptive fashion. That is, its decision about which bit to read at time $t$ can be computed as a logspace computable function of its configuration and the set of *indices* of bits that have already been read (both determined by what has been observed up to time $t-1$). We observe that the hard function for the best-case partition model of Papadimitriou and Sipser [15] becomes easy to compute by an adaptive read-once branching program (or syntactic read-once branching program) but other functions are known to be hard for read-$k$ branching programs [6].

## Acknowledgements

We thank Shachar Lovett for directing us toward related results ([6] and the unpublished work of Yoav Tzur) and for pointing out an improvement in Proposition 1.

## References

[1] M. Ajtai and M. Ben-Or, "A theorem on probabilistic constant depth computations," in *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, ser. STOC '84. New York, NY, USA: ACM, 1984, pp. 471–474. [Online]. Available: http://doi.acm.org/10.1145/800057.808715

[2] M. Ajtai and A. Wigderson, "Deterministic simulation of probabilistic constant depth circuits," in *26th Annual Symposium on Foundations of Computer Science*. IEEE, 1985, pp. 11–19.

[3] N. Alon, J. Bruck, J. Naor, M. Naor, and R. Roth, "Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs," *Information Theory, IEEE Transactions on*, vol. 38, no. 2, pp. 509–516, 1992.

[4] D. A. M. Barrington, "Bounded-width polynomial-size branching programs recognize exactly those languages in NC1," *Journal of Computer and System Sciences*, vol. 38, no. 1, pp. 150–164, 1989.

[5] M. Blum and S. Micali, "How to generate cryptographically strong sequences of pseudo-random bits," *SIAM Journal on Computing*, vol. 13, pp. 850–864, 1984.

[6] B. Bollig, M. Sauerhoff, and I. Wegener, "On the nonapproximability of boolean functions by obdds and read-k-times branching programs," *Information and Computation*, vol. 178, no. 1, pp. 263 – 278, 2002.

[7] M. Braverman, "Polylogarithmic independence fools $ac^0$ circuits," *Journal of the ACM (JACM)*, vol. 57, no. 5, pp. 1–10, 2010.

[8] J. Cai and R. Lipton, "Subquadratic simulations of circuits by branching programs," in *Foundations of Computer Science, 1989., 30th Annual Symposium on*. IEEE, 1989, pp. 568–573.

[9] R. Cleve, "Towards optimal simulations of formulas by bounded-width programs," in *Proceedings of the twenty-second annual ACM symposium on Theory of computing*. ACM, 1990, pp. 271–277.

[10] M. David, P. Nguyen, P. A. Papakonstantinou, and A. Sidiropoulos, "Computationally limited randomness," in *Innovations in Computer Science (ICS)*, Beijing, China, January 2011, pp. 522–535.

[11] R. Impagliazzo, N. Nisan, and A. Wigderson, "Pseudorandomness for network algorithms," in *Proceedings of the 26th Annual ACM Symposium on Theory of Computing, STOC'94 (Montréal, Québec, Canada, May 23-25, 1994)*. New York: ACM Press, 1994, pp. 356–364.

[12] J. Naor and M. Naor, "Small-bias probability spaces: efficient constructions and applications," *SIAM Journal of Computing (SICOMP)*, vol. 22(4), pp. 838–856, 1993, earlier version in STOC'90.

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

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

[15] C. H. Papadimitriou and M. Sipser, "Communication complexity," in *ACM Symposium on Theory of Computing (STOC '82)*. Baltimore, USA: ACM Press, May 1982, pp. 196–200.

[16] R. Raz and O. Reingold, "On recycling the randomness of states in space bounded computation," in *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing (STOC'99)*. New York: Association for Computing Machinery, May 1999, pp. 159–168.

[17] M. Sauerhoff, I. Wegener, and R. Werchner, "Relating branching program size and formula size over the full binary basis," in *Proceedings of the 16th annual conference on Theoretical aspects of computer science*. Springer-Verlag, 1999, pp. 57–67.

[18] Y. Tzur, "Notions of weak pseudorandomness and GF($2^n$)-polynomials," Master's thesis, Weizmann Institute of Science, Rehovot, Israel, 2009.