Nisan's PRG (lecture notes)

Course: Derandomizing Space-Bounded Computation, Winter 2025, University of Chicago
Instructor: William Hoza (williamhoza@uchicago.edu)

In these lecture notes, we study a PRG for space-bounded computation called "Nisan's PRG" [Nis92]. Nisan's PRG is similar to the INW PRG, which we studied previously in this course. In terms of parameters, Nisan's PRG is not any better than the INW PRG. However, we will see in upcoming classes that the specific *structure* of Nisan's PRG is useful for certain applications. In particular, we will use Nisan's PRG to prove $\mathsf{BPL} \subseteq \mathsf{SC}$ [Nis94] and $\mathsf{BPL} \subseteq \mathsf{DSPACE}((\log n)^{3/2})$ [SZ99]. For today, we focus on Nisan's PRG itself.

1 The Pairwise Uniformity Mixing Lemma

Recall that the INW PRG is based on expander graphs and the Expander Mixing Lemma. In a similar fashion, Nisan's PRG is based on *pairwise uniform hash functions* and the *Pairwise Uniformity Mixing Lemma*.

Definition 1.1 (Pairwise uniform hash functions)**.** Let Σ be a finite set, and let $\mathcal{H}$ be a family of hash functions $h \colon \Sigma \to \Sigma$. We say that $\mathcal{H}$ is *pairwise uniform* if, for every two distinct $x, y \in \Sigma$, when we sample $h \sim \mathcal{H}$, the pair of outputs $(h(x), h(y))$ is distributed uniformly over Σ^2.

Fact 1.2. *For every $k \in \mathbb{N}$, there exists an explicit pairwise uniform family $\mathcal{H}$ of hash functions $h \colon \{0,1\}^k \to \{0,1\}^k$ such that sampling $h \sim \mathcal{H}$ costs $O(k)$ truly random bits.*

Pairwise uniform families of hash functions are also called *strongly universal* families. For any hash function $h \colon \Sigma \to \Sigma$, we will use the notation $G_h \colon \Sigma \to \Sigma^2$ to denote the PRG

$$G_h(x) = (x, h(x)).$$

Lemma 1.3 (Pairwise Uniformity Mixing Lemma)**.** *Let $\mathcal{H}$ be a pairwise uniform family of hash functions $h \colon \Sigma \to \Sigma$, and let $f \colon \Sigma^2 \to \{0,1\}$ be a two-dimensional combinatorial rectangle, i.e., $f(x,y) = f_L(x) \cdot f_R(y)$ for some $f_L, f_R \colon \Sigma \to \{0,1\}$. Then for every $\varepsilon \in (0,1)$, except with probability $\mathbb{E}[f]/(\varepsilon^2 \cdot |\Sigma|)$ over the choice of $h \sim \mathcal{H}$, the PRG G_h fools f with error ε.*

Proof. For any fixed $x \in \Sigma$, when we choose $h \sim \mathcal{H}$, the value $f_R(h(x))$ is a Bernoulli random variable, with expectation $\mathbb{E}_h[f_R(h(x))] = \mathbb{E}[f_R]$ and variance $\mathrm{Var}_h[f_R(h(x))] = \mathbb{E}[f_R] \cdot (1 - \mathbb{E}[f_R]) \leq \mathbb{E}[f_R]$. Now let Z be the random variable

$$Z = \sum_{x \in f_L^{-1}(1)} f_R(h(x)).$$

By linearity of expectation, we have $\mathbb{E}[Z] = |f_L^{-1}(1)| \cdot \mathbb{E}[f_R] = |\Sigma| \cdot \mathbb{E}[f]$. Furthermore, the variance of a sum of *pairwise independent* variables is the sum of the variances, so $\mathrm{Var}[Z] \leq |\Sigma| \cdot \mathbb{E}[f]$. Therefore,

$$\begin{aligned}
\Pr_h[G_h \text{ does not fool } f \text{ with error } \varepsilon] &= \Pr_h\left[|Z - \mathbb{E}[Z]| > \varepsilon \cdot |\Sigma|\right] \\
&\leq \frac{\mathrm{Var}[Z]}{\varepsilon^2 |\Sigma|^2} && \text{(Chebyshev's inequality)} \\
&\leq \frac{\mathbb{E}[f]}{\varepsilon^2 \cdot |\Sigma|}.
\end{aligned}$$

□

For comparison, recall that the Expander Mixing Lemma says that if H is a λ-spectral expander graph, then the PRG $G_H(x,y) := (x, H[x,y])$ fools two-dimensional combinatorial rectangles with error λ. The Pairwise Uniformity Mixing Lemma is "cheaper" in one respect, namely, the seed length of G_h is smaller than

the seed length of G_H. However, the Pairwise Uniformity Mixing Lemma is "more expensive" in another respect, namely, we have to use some additional random bits to sample the hash function $h \sim \mathcal{H}$. This extra expense is not so bad, because h is "good" with high probability. In contrast, the Expander Mixing Lemma does not have any concept of "goodness with high probability."

2 Fooling finite automata

Our analysis of the INW generator was based on decomposing an ROBP as a sum of two-dimensional combinatorial rectangles. For today's purposes, instead of ROBPs, it is more convenient to work with the closely-related *finite automaton* model. Recall that a w-state automaton over the alphabet Σ is defined by its transition function $M\colon [w] \times \Sigma \to [w]$. We will use the square bracket notation $M[u, x]$ to denote the output of this transition function, in keeping with the notation we used when we were thinking about connectivity algorithms. Running an automaton for n steps is equivalent to running the *n-th power automaton*, denoted $M^n\colon [w] \times \Sigma^n \to [w]$, for one step. Recall that the n-th power automaton is defined recursively by the formula

$$M^{i+1}[u, x_1 x_2 \dots x_{i+1}] = M[M^i[u, x_1 \dots x_i], x_{i+1}].$$

We will often identify a finite automaton $M\colon [w]\times\Sigma \to [w]$ with its transition probability matrix $M \in [0,1]^{w\times w}$.

Definition 2.1. Let $M\colon [w] \times \Sigma \to [w]$, let X be a distribution over Σ^n, and let $M'_{u,v} = \Pr[M^n[u, X] = v]$. We say that X *fools* M *with* ℓ_1 *error* ε if $\|M^n - M'\|_1 \le \varepsilon$, where[1]

$$\|E\|_1 := \max_u \sum_v |E_{u,v}|.$$

This is equivalent to saying that for every start state u, the distribution over final states $M^n[u, X]$ is ε-close in ℓ_1 distance to the distribution $M^n[u, Y]$, where Y is sampled uniformly at random from Σ^n.

Lemma 2.2 (The Pairwise Uniformity Mixing Lemma, applied to finite automata)**.** *Let $M\colon [w] \times \Sigma \to [w]$ be a finite automaton. For every $\varepsilon \in (0, 1)$, if we sample $h \sim \mathcal{H}$ where $\mathcal{H}$ is pairwise uniform, then except with probability $\frac{w^5}{\varepsilon^2|\Sigma|}$, the PRG G_h fools M with ℓ_1 error at most ε.*

Proof. For any three states $u, m, v \in [w]$, we can define a two-dimensional combinatorial rectangle $f_{u,m,v}\colon \Sigma^2 \to \{0, 1\}$ by the formula

$$f_{u,m,v}(x, y) = 1 \iff M[u, x] = m \text{ and } M[m, y] = v.$$

Then

$$M^2_{u,v} = \sum_m \mathbb{E}[f_{u,m,v}],$$

and if we let $M'_{u,v} = \Pr_x[M^2[u, G_h(x)] = v]$, then

$$M'_{u,v} = \sum_m \mathop{\mathbb{E}}_{x\in\Sigma}[f_{u,m,v}(x, h(x))].$$

By the Pairwise Uniformity Mixing Lemma, combined with the union bound over $m \in [w]$ and the triangle inequality, except with probability $M_{u,v}/(\delta^2|\Sigma|)$ over the choice of $h \sim \mathcal{H}$, we have $|M^2_{u,v} - M'_{u,v}| \le w\delta$. Now take another union bound over all $u, v \in [w]$: Except with probability $w/(\delta^2|\Sigma|)$, we have $\|M^2 - M'\|_1 \le w^2\delta$. Choosing $\delta = \varepsilon/w^2$ completes the proof. □

[1] This norm is more commonly denoted $\|E\|_\infty$ or $\|E^T\|_1$, but note that we think of E as a linear transformation based on *left* multiplication, $\pi \mapsto \pi E$.

3 Nisan's PRG

Recall that the INW generator applies the Expander Mixing Lemma recursively to construct a PRG with a good stretch. Similarly, Nisan's PRG is based on a recursive application of the Pairwise Uniformity Mixing Lemma. To be more specific, for a sequence of hash functions $h_1, \ldots, h_{\log n} \colon \Sigma \to \Sigma$, we define an associated PRG $G_{h_1,\ldots,h_{\log n}} \colon \Sigma \to \Sigma^n$ by the rules

$$\begin{aligned} G_{()}(x) &= x \\ G_{h_1,\ldots,h_{\log n}}(x) &= (G_{(h_1,\ldots,h_{\log n-1})}(x), G_{(h_1,\ldots,h_{\log n-1})}(h_{\log n}(x))). \end{aligned}$$

Lemma 3.1 (Efficiency of Nisan's PRG)**.** *Given the truth tables of $h_1, \ldots, h_{\log n} \colon \Sigma \to \Sigma$, and given $x \in \Sigma$, the output $G_{h_1,\ldots,h_{\log n}}(x)$ can be computed using $O(\log(|\Sigma| \cdot n))$ bits of space.*

Proof sketch. It is helpful to "unroll" the recursive definition of $G_{h_1,\ldots,h_{\log n}}$. As an example, we have

$$G_{h_1,h_2,h_3}(x) = (x, h_1(x), h_2(x), h_1(h_2(x)), h_3(x), h_1(h_3(x)), h_2(h_3(x)), h_1(h_2(h_3(x)))).$$

In general, the i-th output symbol of $G_{h_1,\ldots,h_{\log n}}(x)$ is a composition of some subset of $h_1, \ldots, h_{\log n}$ applied to x. The subset can be efficiently computed (it is essentially given by the binary expansion of the number $i-1$). The lemma follows. □

To analyze the correctness of Nisan's PRG, we introduce the following useful notation, which should be compared to the "derandomized square" notion that we used to prove that undirected s-t connectivity is in L.

Definition 3.2 (Hash-based derandomized square)**.** Let $M \colon [w] \times \Sigma \to [w]$ and let $h \colon \Sigma \to \Sigma$. We define $M_h \colon [w] \times \Sigma \to [w]$ by the rule

$$M_h[u, x] = M^2[u, G_h(x)] = M^2[u, (x, h(x))].$$

Furthermore, for a sequence of functions $h_1, \ldots, h_{\log n} \colon \Sigma \to \Sigma$, we define $M_{h_1,\ldots,h_{\log n}}$ by

$$M_{h_1,\ldots,h_{\log n}} = (\cdots (M_{h_1})_{h_2} \cdots)_{h_{\log n}}.$$

Observe that $M_{h_1,\ldots,h_{\log n}}[u, x] = M^n[u, G_{h_1,\ldots,h_{\log n}}(x)]$.

Lemma 3.3 (Accumulation of error)**.** *Let $M \colon [w] \times \Sigma \to [w]$ be a finite automaton, let n be a power of two, and let $h_1, \ldots, h_{\log n} \colon \Sigma \to \Sigma$. Assume that for every $i \in [\log n]$, the PRG G_{h_i} fools the automaton $M_{h_1,\ldots,h_{i-1}}$ with ℓ_1 error at most δ. Then the PRG $G_{h_1,\ldots,h_{\log n}}$ fools M with ℓ_1 error at most $\delta \cdot (n-1)$.*

Proof. We prove it by induction on n. Let $M' = M_{h_1,\ldots,h_{\log n-1}}$. Then

$$\begin{aligned} \|M'_{h_{\log n}} - M^n\|_1 &\le \|M'_{h_{\log n}} - (M')^2\|_1 + \|(M')^2 - M' \cdot M^{n/2}\|_1 + \|M' \cdot M^{n/2} - M^n\|_1 \\ &\qquad \text{(Triangle inequality)} \\ &\le \delta + \|M'\|_1 \cdot \|M' - M^{n/2}\|_1 + \|M' - M^{n/2}\|_1 \cdot \|M^{n/2}\|_1 \\ &\qquad \text{(Assumption; submultiplicativity)} \\ &\le \delta + 2\|M' - M^{n/2}\|_1 \\ &\qquad (M' \text{ and } M^{n/2} \text{ are stochastic matrices}) \\ &\le \delta + 2\delta \cdot (n/2 - 1) \\ &\qquad \text{(Induction)} \\ &= \delta \cdot (n-1). \end{aligned}$$

□

Theorem 3.4 (Nisan's PRG [Nis92])**.** *For every $w, n \in \mathbb{N}$ and $\varepsilon \in (0,1)$, there exists an explicit PRG $G \colon \{0,1\}^s \to \{0,1\}^n$ that fools automata $M \colon [w] \times \{0,1\} \to [w]$ with ℓ_1 error ε, where $s = O(\log(wn/\varepsilon) \cdot \log n)$.*

Proof sketch. Let $\mathcal{H}$ be a pairwise uniform family of hash functions $h\colon \Sigma \to \Sigma$, where $\Sigma = \{0,1\}^k$ for a suitable value $k = O(\log(wn/\varepsilon))$. The generator G:

1. Sample $h_1, h_2, \ldots, h_{\log n} \sim \mathcal{H}$.
2. Sample $x \in \Sigma$ uniformly at random.
3. Let $(x^{(1)}, \ldots, x^{(n)}) = G_{h_1,\ldots,h_{\log n}}(x)$.
4. Output the first bit of each symbol, i.e., output $(x_1^{(1)}, \ldots, x_1^{(n)})$.

To prove that this works, let $M'\colon [w] \times \Sigma \to [w]$ simulate M by ignoring all but the first bit of the symbol it sees. By Lemma 2.2 and the union bound, except with probability $\frac{w^5 \cdot \log n}{\delta^2 \cdot 2^k}$ over the choice of $h_1, \ldots, h_{\log n}$, the PRG G_{h_i} fools the automaton $M'_{h_1,\ldots,h_{i-1}}$ with ℓ_1 error at most δ. In this case, by Lemma 3.3, $G_{h_1,\ldots,h_{\log n}}$ fools M' with ℓ_1 error at most $\delta \cdot n$. In any case, the ℓ_1 error is at most 2. Consequently, G fools M with ℓ_1 error at most $\delta \cdot n + \frac{2w^5 \cdot \log n}{\delta^2 \cdot 2^k}$. To complete the correctness proof, choose $\delta = \varepsilon/(2n)$, and choose a large enough $k = O(\log(wn/\varepsilon))$. □

References

[Nis92] Noam Nisan. "Pseudorandom generators for space-bounded computation". In: *Combinatorica* 12.4 (1992), pp. 449–461. ISSN: 0209-9683. DOI: 10.1007/BF01305237.

[Nis94] Noam Nisan. "RL ⊆ SC". In: *Comput. Complexity* 4.1 (1994), pp. 1–11. ISSN: 1016-3328. DOI: 10.1007/BF01205052.

[SZ99] Michael Saks and Shiyu Zhou. "$\mathrm{BP_HSPACE}(S) \subseteq \mathrm{DSPACE}(S^{3/2})$". In: *J. Comput. System Sci.* 58.2 (1999), pp. 376–403. ISSN: 0022-0000. DOI: 10.1006/jcss.1998.1616.