Simple Optimal Hitting Sets for Small-Success RL

William M. Hoza\(^1\) David Zuckerman\(^2\)

The University of Texas at Austin

September 24, 2018
Dagstuhl Seminar 18391

---

\(^1\)Supported by the NSF GRFP under Grant DGE-1610403 and by a Harrington Fellowship from UT Austin

\(^2\)Supported by NSF Grant CCF-1526952, NSF Grant CCF-1705028, and a Simons Investigator Award (#409864)
Randomized log-space complexity classes

Let $L$ be a language
Randomized log-space complexity classes

- Let $L$ be a language
- $L \in \text{BPL}$ if there is a randomized log-space algorithm $A$ that always halts such that
  \[
  x \in L \implies \Pr[A(x) \text{ accepts}] \geq \frac{2}{3}
  \]
  \[
  x \notin L \implies \Pr[A(x) \text{ accepts}] \leq \frac{1}{3}.
  \]
Randomized log-space complexity classes

- Let $L$ be a language
- $L \in \textbf{BPL}$ if there is a randomized log-space algorithm $A$ that always halts such that
  \[ x \in L \implies \Pr[A(x) \text{ accepts}] \geq \frac{2}{3} \]
  \[ x \notin L \implies \Pr[A(x) \text{ accepts}] \leq \frac{1}{3}. \]
- $L \in \textbf{RL}$ if there is a randomized log-space algorithm $A$ that always halts such that
  \[ x \in L \implies \Pr[A(x) \text{ accepts}] \geq \frac{1}{2} \]
  \[ x \notin L \implies \Pr[A(x) \text{ accepts}] = 0. \]
The power of randomness for small-space algorithms

- $L \subseteq RL \subseteq BPL$
The power of randomness for small-space algorithms

- $L \subseteq RL \subseteq BPL$
- Conjecture: $L = RL = BPL$
The power of randomness for small-space algorithms

- \( L \subseteq RL \subseteq BPL \)
- Conjecture: \( L = RL = BPL \)
The power of randomness for small-space algorithms

- $L \subseteq RL \subseteq BPL$
- Conjecture: $L = RL = BPL$
Read-once branching programs

$n + 1$ layers

start

width $n$

acc
Read-once branching programs

Computes function $f: \{0, 1\}^n \to \{0, 1\}$

$n + 1$ layers

width $n$
Read-once branching programs

$\text{start}$

$n + 1 \text{ layers}$

$\text{width } n$

$\text{acc}$

$x =$

Computes function $f : \{0, 1 \}^n \rightarrow \{0, 1 \}$
Read-once branching programs

$x = 1$
Read-once branching programs

\[x = 1\ 0\]

\[\text{width } n\]

\[n + 1 \text{ layers}\]
Read-once branching programs

$\begin{array}{cccc}
1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 \\
\end{array}$

Computes function $f: \{0, 1\}^n \rightarrow \{0, 1\}$
Read-once branching programs

$\begin{array}{cccc}
0 & 1 & 1 & 1 \\
1 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 \\
\end{array}$

$x = 1 \quad 0 \quad 0 \quad 0 \quad 1$
Read-once branching programs

\[ x = 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \]

\[ n + 1 \text{ layers} \]

width \( n \)

Computes function \( f: \{0, 1\}^n \rightarrow \{0, 1\} \)
Read-once branching programs

Start

\( n + 1 \) layers

Width \( n \)

\( x = \begin{array}{cccccc}
1 & 0 & 0 & 0 & 1 & 1
\end{array} \)

Computes function \( f : \{0, 1\}^n \rightarrow \{0, 1\} \)
Fooling / Hitting ROBPs

\[ \Pr[x[f(x) = 1] - \Pr[z[f(Gen(z)) = 1]] \leq \varepsilon] \]

Suitable for derandomizing BPL

\[ \Pr[x[f(x) = 1] \geq \varepsilon] \Rightarrow \exists z, f(Gen(z)) = 1 \]

Suitable for derandomizing RL
Fooling / Hitting ROBPs

Pseudorandom generator: For every width-\(n\) ROBP,

\[
| \Pr_x[f(x) = 1] - \Pr_z[f(\text{Gen}(z)) = 1] | \leq \varepsilon
\]
Fooling / Hitting ROBPs

Pseudorandom generator: For every width-$n$ ROBP,

$$|\Pr_x[f(x) = 1] - \Pr_z[f(\text{Gen}(z)) = 1]| \leq \epsilon$$

Suitable for derandomizing BPL
Fooling / Hitting ROBPs

Pseudorandom generator: For every width-\(n\) ROBP,
\[
\left| \Pr_x[f(x) = 1] - \Pr_z[f(\text{Gen}(z)) = 1] \right| \leq \varepsilon
\]

Suitable for derandomizing BPL

Hitting set generator: For every width-\(n\) ROBP,
\[
\Pr_x[f(x) = 1] \geq \varepsilon \implies \exists z, f(\text{Gen}(z)) = 1
\]
Fooling / Hitting ROBP
s bits \rightarrow \text{Gen} \rightarrow n \text{ bits}

**Pseudorandom generator:** For every width-$n$ ROBP,
\[ |\Pr_x[f(x) = 1] - \Pr_z[f(\text{Gen}(z)) = 1]| \leq \varepsilon \]

Suitable for derandomizing BPL

**Hitting set generator:** For every width-$n$ ROBP,
\[ \Pr_x[f(x) = 1] \geq \varepsilon \implies \exists z, f(\text{Gen}(z)) = 1 \]

Suitable for derandomizing RL
Prior generators and main result

- Nonconstructive: PRG with seed length $O(\log n + \log(1/\varepsilon))$
Prior generators and main result

- Nonconstructive: PRG with seed length $O(\log n + \log(1/\varepsilon))$
- Babai, Nisan, Szegedy 1989: PRG with seed length $2^{O(\sqrt{\log n})} \cdot \log(1/\varepsilon)$
- Nisan 1990: PRG with seed length $O(\log 2^n + \log(1/\varepsilon) \log n)$
- Braverman, Cohen, Garg 2018: HSG with seed length $\tilde{O}(\log 2^n + \log(1/\varepsilon))$
- This work: HSG with seed length $O(\log 2^n + \log(1/\varepsilon))$
Prior generators and main result

- Nonconstructive: PRG with seed length $O(\log n + \log(1/\varepsilon))$

- Babai, Nisan, Szegedy 1989: PRG with seed length

  $$2^{O(\sqrt{\log n})} \cdot \log(1/\varepsilon)$$

- Nisan 1990: PRG with seed length

  $O(\log^2 n + \log(1/\varepsilon) \log n)$
Prior generators and main result

- Nonconstructive: PRG with seed length $O(\log n + \log(1/\varepsilon))$
- Babai, Nisan, Szegedy 1989: PRG with seed length $2^{O(\sqrt{\log n})} \cdot \log(1/\varepsilon)$
- Nisan 1990: PRG with seed length $O(\log^2 n + \log(1/\varepsilon) \log n)$
- Braverman, Cohen, Garg 2018: HSG with seed length $\tilde{O}(\log^2 n + \log(1/\varepsilon))$
Prior generators and main result

- Nonconstructive: PRG with seed length $O(\log n + \log(1/\varepsilon))$
- Babai, Nisan, Szegedy 1989: PRG with seed length
  $2^{O(\sqrt{\log n})} \cdot \log(1/\varepsilon)$
- Nisan 1990: PRG with seed length
  $O(\log^2 n + \log(1/\varepsilon) \log n)$
- Braverman, Cohen, Garg 2018: HSG with seed length
  $\tilde{O}(\log^2 n + \log(1/\varepsilon))$

- This work: HSG with seed length
  $O(\log^2 n + \log(1/\varepsilon))$
Comparison with [BCG '18]

- Our construction and analysis are simple
Comparison with [BCG '18]

- Our construction and analysis are simple
- Braverman, Cohen, Garg '18:

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Matrix bundles</td>
<td>29</td>
</tr>
<tr>
<td>5.2</td>
<td>Matrix bundles sequences</td>
<td>23</td>
</tr>
<tr>
<td>5.3</td>
<td>Gluing MBSs</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>Multiplication Rules for Matrix Bundle Sequences</td>
<td>26</td>
</tr>
<tr>
<td>6.1</td>
<td>The multiplication rules $\overrightarrow{\circ}$, $\overleftarrow{\circ}$ parameterized by a sampler</td>
<td>26</td>
</tr>
<tr>
<td>6.2</td>
<td>The multiplication rules $\overrightarrow{\circ}$, $\overleftarrow{\circ}$ parameterized by a sampler</td>
<td>29</td>
</tr>
<tr>
<td>6.3</td>
<td>The multiplication rules $\circlearrowleft$, $\circlearrowright$ parameterized by delta of samplers</td>
<td>34</td>
</tr>
<tr>
<td>7</td>
<td>Leveled Matrix Representations</td>
<td>39</td>
</tr>
<tr>
<td>8</td>
<td>The Family $\mathcal{F}(A, B)$</td>
<td>41</td>
</tr>
<tr>
<td>8.1</td>
<td>Basic properties of the MBSs in $\mathcal{F}(A, B)$</td>
<td>44</td>
</tr>
<tr>
<td>8.2</td>
<td>The slices of $\mathcal{F}(A, B)$</td>
<td>48</td>
</tr>
</tbody>
</table>
Comparison with [BCG ’18]

▶ Our construction and analysis are simple
▶ Braverman, Cohen, Garg ’18:

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1</td>
<td>Matrix bundles</td>
<td>29</td>
</tr>
<tr>
<td>5.2</td>
<td>Matrix bundles sequences</td>
<td>23</td>
</tr>
<tr>
<td>5.3</td>
<td>Gluing MBSs</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>Multiplication Rules for Matrix Bundle Sequences</td>
<td>26</td>
</tr>
<tr>
<td>6.1</td>
<td>The multiplication rules parameterized by a sampler</td>
<td>26</td>
</tr>
<tr>
<td>6.2</td>
<td>The multiplication rules parameterized by a sampler</td>
<td>29</td>
</tr>
<tr>
<td>6.3</td>
<td>The multiplication rules parameterized by delta of samplers</td>
<td>34</td>
</tr>
<tr>
<td>7</td>
<td>Leveled Matrix Representations</td>
<td>39</td>
</tr>
<tr>
<td>8</td>
<td>The Family $\mathcal{F}(A, B)$</td>
<td>41</td>
</tr>
<tr>
<td>8.1</td>
<td>Basic properties of the MBSs in $\mathcal{F}(A, B)$</td>
<td>44</td>
</tr>
<tr>
<td>8.2</td>
<td>The slices of $\mathcal{F}(A, B)$</td>
<td>48</td>
</tr>
</tbody>
</table>

This work

Hitting Set Generator

Suitable for $\textbf{RL}$
Comparison with [BCG ’18]

► Our construction and analysis are simple
► Braverman, Cohen, Garg ’18:

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Matrix bundles</td>
<td>29</td>
</tr>
<tr>
<td>5.2</td>
<td>Matrix bundles sequences</td>
<td>23</td>
</tr>
<tr>
<td>5.3</td>
<td>Gluing MBSs</td>
<td>25</td>
</tr>
<tr>
<td>6</td>
<td>Multiplication Rules for Matrix Bundle Sequences</td>
<td>26</td>
</tr>
<tr>
<td>6.1</td>
<td>The multiplication rules $\frac{\om}{\oc}, \frac{\um}{\uc}$ parameterized by a sampler</td>
<td>26</td>
</tr>
<tr>
<td>6.2</td>
<td>The multiplication rules $\oc, \um$ parameterized by a sampler</td>
<td>29</td>
</tr>
<tr>
<td>6.3</td>
<td>The multiplication rules $\oc, \um$ parameterized by delta of samplers</td>
<td>34</td>
</tr>
<tr>
<td>7</td>
<td>Leveled Matrix Representations</td>
<td>39</td>
</tr>
<tr>
<td>8</td>
<td>The Family $F(A, B)$</td>
<td>41</td>
</tr>
<tr>
<td>8.1</td>
<td>Basic properties of the MBSs in $F(A, B)$</td>
<td>44</td>
</tr>
<tr>
<td>8.2</td>
<td>The slices of $F(A, B)$</td>
<td>48</td>
</tr>
</tbody>
</table>

Nisan ’90

- Pseudorandom Generator
  - Suitable for BPL

This work

- Hitting Set Generator
  - Suitable for RL
Comparison with [BCG ’18]

- Our construction and analysis are simple
- Braverman, Cohen, Garg ’18:

<table>
<thead>
<tr>
<th>9.1 Matrix bundles</th>
<th>29</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2 Matrix bundles sequences</td>
<td>23</td>
</tr>
<tr>
<td>5.3 Gluing MBSs</td>
<td>25</td>
</tr>
</tbody>
</table>

6 Multiplication Rules for Matrix Bundle Sequences
- 6.1 The multiplication rules $\overrightarrow{\cdot}, \overrightarrow{\circ}$ parameterized by a sampler | 26 |
- 6.2 The multiplication rules $\overrightarrow{\cdot}, \overrightarrow{\circ}$ parameterized by a sampler | 29 |
- 6.3 The multiplication rules $\overrightarrow{\cdot}, \overrightarrow{\circ}$ parameterized by delta of samplers | 34 |

7 Leveled Matrix Representations | 39 |

8 The Family $\mathcal{F}(A, B)$
- 8.1 Basic properties of the MBSs in $\mathcal{F}(A, B)$ | 44 |
- 8.2 The slices of $\mathcal{F}(A, B)$ | 48 |

---

<table>
<thead>
<tr>
<th>Nisan ’90</th>
<th>BCG ’18</th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pseudorandom Generator</td>
<td>“Pseudorandom Pseudodistribution”</td>
<td>Hitting Set Generator</td>
</tr>
<tr>
<td>Suitable for BPL</td>
<td>Suitable for BPL</td>
<td>Suitable for RL</td>
</tr>
</tbody>
</table>
Structural lemma for ROBPs

Let $f$ be a width-$n$, length-$n$ ROBP
Structural lemma for ROBPs

- Let $f$ be a width-$n$, length-$n$ ROBP
- Assume $\Pr[\text{accept}] = \varepsilon \ll 1/n^3$
Let $f$ be a width-$n$, length-$n$ ROBP

Assume $\Pr[\text{accept}] = \epsilon \ll 1/n^3$

Lemma: There is a vertex $u$ so that

$$\Pr[\text{reach } u] \geq \frac{1}{2n^3} \quad \text{and} \quad \Pr[\text{accept } | \text{ reach } u] \geq \epsilon n.$$
Proof of lemma \((\exists u, \Pr[u] \geq \frac{1}{2n^3} \land \Pr[\text{acc} \mid u] \geq \varepsilon n)\)

Say \(u\) is a milestone if \(\Pr[\text{accept} \mid \text{reach } u] \in [\varepsilon n, 2\varepsilon n]\)
Proof of lemma \((\exists u, \Pr[u] \geq \frac{1}{2n^3} \land \Pr[\text{acc} \mid u] \geq \varepsilon n)\)

- Say \(u\) is a \text{milestone} if \(\Pr[\text{accept} \mid \text{reach } u] \in [\varepsilon n, 2\varepsilon n]\)
- Claim: Every accepting path passes through a milestone
Proof of lemma \((\exists u, \Pr[u] \geq \frac{1}{2n^3} \land \Pr[\text{acc} \mid u] \geq \epsilon n)\)

- Say \(u\) is a **milestone** if \(\Pr[\text{accept} \mid \text{reach } u] \in [\epsilon n, 2\epsilon n]\)

- **Claim:** Every accepting path passes through a milestone
  - **Proof:** Probability of acceptance at most doubles in each step

\[\Pr[\text{accept} \mid u] \leq \sum_{\text{milestone}} \Pr[\text{reach } u \text{ and accept}] \leq \sum_{\text{milestone}} \Pr[\text{reach } u] \cdot 2\epsilon n\]

\(\# \text{milestones} \leq n^2\), so for some milestone \(u\), \(\Pr[\text{reach } u] \geq \frac{1}{2n^3}\)
Proof of lemma \((\exists u, \Pr[u] \geq \frac{1}{2n^3} \land \Pr[\text{acc} \mid u] \geq \varepsilon n)\)

- Say \(u\) is a milestone if \(\Pr[\text{accept} \mid \text{reach } u] \in [\varepsilon n, 2\varepsilon n]\)

- Claim: Every accepting path passes through a milestone

- Proof: Probability of acceptance at most doubles in each step

<table>
<thead>
<tr>
<th>3% chance of accept</th>
<th>(\rightarrow)</th>
<th>1</th>
<th>6% chance of accept</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(\rightarrow)</td>
<td>0</td>
<td>0% chance of accept</td>
</tr>
</tbody>
</table>
Proof of lemma \((\exists u, \Pr[u] \geq \frac{1}{2n^3} \land \Pr[\text{acc} \mid u] \geq \varepsilon n)\)

- Say \(u\) is a **milestone** if \(\Pr[\text{accept} \mid \text{reach } u] \in [\varepsilon n, 2\varepsilon n]\)

- **Claim:** Every accepting path passes through a milestone

  - **Proof:** Probability of acceptance at most doubles in each step

  
  \[
  \begin{array}{c}
  3\% \text{ chance of accept} \rightarrow 1 \quad 6\% \text{ chance of accept} \\
  0 \quad 0\% \text{ chance of accept} \\
  \end{array}
  \]

- \(\varepsilon = \Pr[\text{accept}] \leq \sum_{u \text{ milestone}} \Pr[\text{reach } u \text{ and accept}]\)
Proof of lemma \((\exists u, \Pr[u] \geq \frac{1}{2n^3} \land \Pr[\text{acc} \mid u] \geq \varepsilon n)\)

- Say \(u\) is a **milestone** if \(\Pr[\text{accept} \mid \text{reach } u] \in [\varepsilon n, 2\varepsilon n]\)

- **Claim:** Every accepting path passes through a milestone

  - **Proof:** Probability of acceptance at most doubles in each step

  \[
  \varepsilon = \Pr[\text{accept}] \leq \sum_{u \text{ milestone}} \Pr[\text{reach } u \text{ and accept}] \\
  \leq \sum_{u \text{ milestone}} \Pr[\text{reach } u] \cdot 2\varepsilon n
  \]
Proof of lemma \((\exists u, \Pr[u] \geq \frac{1}{2n^3} \land \Pr[\text{acc} | u] \geq \varepsilon n)\)

- Say \(u\) is a **milestone** if \(\Pr[\text{accept} | \text{reach } u] \in [\varepsilon n, 2\varepsilon n]\)

- **Claim:** Every accepting path passes through a milestone

- **Proof:** Probability of acceptance at most doubles in each step

\[
\begin{align*}
\text{3\% chance of accept} & \quad \text{1} \quad \bullet \text{6\% chance of accept} \\
\text{0} & \quad \bullet \text{0\% chance of accept}
\end{align*}
\]

- \(\varepsilon = \Pr[\text{accept}] \leq \sum_{u \text{ milestone}} \Pr[\text{reach } u \text{ and accept}] \leq \sum_{u \text{ milestone}} \Pr[\text{reach } u] \cdot 2\varepsilon n\)

- \# milestones \(\leq n^2\), so for some milestone \(u\), \(\Pr[\text{reach } u] \geq \frac{1}{2n^3} \) \(\square\)
Iterating the structural lemma

\[ u_0 = \text{start} \]

\[ \Pr[\text{accept}] = \varepsilon \]
Iterating the structural lemma

\[ u_0 = \text{start} \]

\[ u_1 \]

\[ \text{Pr[accept]} = \varepsilon \quad n\varepsilon \]
Iterating the structural lemma

\[ u_0 = \text{start} \]

\[ \Pr[\text{accept}] = \varepsilon \]

\[ u_1 \]

\[ u_2 \]

\[ \text{acc} \]

\[ \Pr[\text{accept}] = \varepsilon \quad n\varepsilon \quad n^2\varepsilon \]
Iterating the structural lemma

\[ u_0 = \text{start} \]

\[ \Pr[\text{accept}] = \varepsilon \]

\[ u_1 \]

\[ u_2 \]

\[ u_3 \]

\[ n\varepsilon \]

\[ n^2\varepsilon \]

\[ n^3\varepsilon \]
Iterating the structural lemma

\[ u_0 = \text{start} \]

\[ \Pr[\text{accept}] = \varepsilon \]

\[ u_1 \]

\[ u_2 \]

\[ u_3 \]

\[ \text{acc} = u_t \]

\[ n\varepsilon \]

\[ n^2\varepsilon \]

\[ n^3\varepsilon \]

\[ n^t\varepsilon = 1 \]
Idea of our HSG

- Use Nisan’s generator for each individual hop $u_i \rightarrow u_{i+1}$
Idea of our HSG

- Use Nisan’s generator for each individual hop $u_i \rightarrow u_{i+1}$
- Use a “hitter” to recycle the seed of Nisan’s generator from one hop to the next
Hitters (equivalent to dispersers)

- Assume query access to unknown $E \subseteq \{0, 1\}^m$ with density$(E) \geq \theta$
Hitters (equivalent to dispersers)

- Assume query access to unknown $E \subseteq \{0, 1\}^m$ with $\text{density}(E) \geq \theta$

- **Theorem (BGG ’93):** Algorithm that outputs some $z \in E$ with probability $1 - \delta$
Hitters (equivalent to dispersers)

- Assume query access to unknown $E \subseteq \{0, 1\}^m$ with density$(E) \geq \theta$

- **Theorem (BGG '93):** Algorithm that outputs some $z \in E$ with probability $1 - \delta$
  - # queries: $O(\theta^{-1} \cdot \log(1/\delta))$
Hitters (equivalent to dispersers)

- Assume query access to unknown $E \subseteq \{0, 1\}^m$ with density$(E) \geq \theta$
- **Theorem (BGG '93):** Algorithm that outputs some $z \in E$ with probability $1 - \delta$
  - # queries: $O(\theta^{-1} \cdot \log(1/\delta))$
  - # random bits: $O(m + \log(1/\delta))$
Hitters (equivalent to dispersers)

- Assume query access to unknown $E \subseteq \{0, 1\}^m$ with density $\text{density}(E) \geq \theta$

**Theorem** (BGG '93): Algorithm that outputs some $z \in E$ with probability $1 - \delta$

- # queries: $O(\theta^{-1} \cdot \log(1/\delta))$
- # random bits: $O(m + \log(1/\delta))$
Hitters (equivalent to dispersers)

- Assume query access to unknown $E \subseteq \{0, 1\}^m$ with density($E$) $\geq \theta$
- **Theorem** (BGG '93): Algorithm that outputs some $z \in E$ with probability $1 - \delta$
  - # queries: $O(\theta^{-1} \cdot \log(1/\delta))$
  - # random bits: $O(m + \log(1/\delta))$
Hitters (equivalent to dispersers)

- Assume query access to unknown $E \subseteq \{0, 1\}^m$ with density $\text{density}(E) \geq \theta$

- **Theorem (BGG '93):** Algorithm that outputs some $z \in E$ with probability $1 - \delta$
  - # queries: $O(\theta^{-1} \cdot \log(1/\delta))$
  - # random bits: $O(m + \log(1/\delta))$
For any $E$ with density($E$) $\geq \theta$, $\Pr_x[\exists y, \text{Hit}(x, y) \in E] \geq 1 - \delta$
Hitter as a function

For any $E$ with $\text{density}(E) \geq \theta$,

$$\Pr_{x} [\exists y, \text{Hit}(x, y) \in E] \geq 1 - \delta$$
Our HSG
Our HSG

\[ x \]

\[ y_1 \]

\[ y_2 \]

\[ y_3 \]

\[ y_t \]
Our HSG

\[ y_1 \]

\[ y_2 \]

\[ y_3 \]

\[ y_t \]
Our HSG
Our HSG
Our HSG

\[ x \rightarrow y_1 \rightarrow \text{Hit} \rightarrow \text{NisGen} \rightarrow \]
\[ y_2 \rightarrow \text{Hit} \rightarrow \text{NisGen} \rightarrow \]
\[ y_3 \rightarrow \text{Hit} \rightarrow \text{NisGen} \rightarrow \]
\[ y_t \rightarrow \text{Hit} \rightarrow \text{NisGen} \rightarrow \]

\[ n_1 \quad n_2 \quad n_3 \quad n_t \]
Our HSG
Our HSG

\[
x \rightarrow \text{Hit} \rightarrow \text{NisGen} \rightarrow \text{Output} = n
\]
Our HSG in symbols

For numbers $n_1, \ldots, n_t$ with $n_1 + \cdots + n_t = n$:

$$\text{Gen}(x, y_1, \ldots, y_t, n_1, \ldots, n_t) = \text{NisGen}(\text{Hit}(x, y_1))|_{n_1} \circ \cdots \circ \text{NisGen}(\text{Hit}(x, y_t))|_{n_t} \in \{0, 1\}^n$$
Our HSG in symbols

- For numbers $n_1, \ldots, n_t$ with $n_1 + \cdots + n_t = n$:

  \[
  \text{Gen}(x, y_1, \ldots, y_t, n_1, \ldots, n_t) = \text{NisGen}(|\text{Hit}(x, y_1)|_{n_1} \odot \cdots \odot \text{NisGen}(|\text{Hit}(x, y_t)|_{n_t}) \in \{0, 1\}^n
  \]

- Here $\odot = \text{concatenation}$, $|_r = \text{first } r \text{ bits}$
Our HSG in symbols

- For numbers $n_1, \ldots, n_t$ with $n_1 + \cdots + n_t = n$:

\[
\text{Gen}(x, y_1, \ldots, y_t, n_1, \ldots, n_t) = \\
\text{NisGen}(\text{Hit}(x, y_1))|_{n_1} \circ \cdots \circ \text{NisGen}(\text{Hit}(x, y_t))|_{n_t} \in \{0, 1\}^n
\]

- Here $\circ = \text{concatenation}$, $|_r = \text{first } r \text{ bits}$

- $|x| = O(\log^2 n)$, $|y_i| = O(\log n)$, $t = \frac{\log(1/\varepsilon)}{\log n}$
Our HSG in symbols

- For numbers $n_1, \ldots, n_t$ with $n_1 + \cdots + n_t = n$:

\[
\text{Gen}(x, y_1, \ldots, y_t, n_1, \ldots, n_t) = \text{NisGen}(\text{Hit}(x, y_1))|_{n_1} \circ \cdots \circ \text{NisGen}(\text{Hit}(x, y_t))|_{n_t} \in \{0, 1\}^n
\]

- Here $\circ = \text{concatenation}$, $|_r = \text{first } r \text{ bits}$

- $|x| = O(\log^2 n)$, $|y_i| = O(\log n)$, $t = \frac{\log(1/\varepsilon)}{\log n}$

- So seed length $= O(\log^2 n + \log(1/\varepsilon))$
Proof of correctness of our HSG

\[ u_0 = \text{start} \]

\[ \Pr[\text{accept}] = \varepsilon \]
Proof of correctness of our HSG

$u_0 = \text{start}$

$Pr[\text{accept}] = \varepsilon$

$n\varepsilon$
Proof of correctness of our HSG

\[ \Pr[\text{accept}] = \varepsilon \]

\[ u_0 = \text{start} \]

\[ u_1 \]

\[ u_2 \]

\[ n_1 \quad n_2 \]

\[ \text{acc} \]
Proof of correctness of our HSG

$u_0 = \text{start}$

$\Pr[\text{accept}] = \varepsilon$

$n_1 \varepsilon \quad n_2 \varepsilon \quad n_3 \varepsilon$

$u_1 \quad u_2 \quad u_3 \quad \text{acc}$
Proof of correctness of our HSG

\[ u_0 = \text{start} \quad \text{Pr}[\text{accept}] = \varepsilon \quad n_1 \epsilon \quad n_2 \epsilon \quad u_2 \quad n_3 \epsilon \quad u_3 \quad \text{acc} = u_t \quad n_t \epsilon = 1 \]
Define $E_i \subseteq \{0, 1\}^m$ by

$$E_i = \{z \mid \text{start at } u_{i-1}, \text{ read } \text{NisGen}(z) \Rightarrow \text{ reach } u_i\}$$
Proof of correctness of our HSG (continued)

- Define $E_i \subseteq \{0, 1\}^m$ by

$$E_i = \{z \mid \text{start at } u_{i-1}, \text{ read } \text{NisGen}(z) \implies \text{reach } u_i\}$$

- $\Pr[\text{reach } u_i \mid \text{reach } u_{i-1}] \geq \frac{1}{2n^3} \implies \text{density}(E_i) > \frac{1}{4n^3}$
Proof of correctness of our HSG (continued)

- Define $E_i \subseteq \{0, 1\}^m$ by
  $$E_i = \{z \mid \text{start at } u_{i-1}, \text{ read } \text{NisGen}(z) \implies \text{reach } u_i\}$$

- $\Pr[\text{reach } u_i \mid \text{reach } u_{i-1}] \geq \frac{1}{2n^3} \implies \text{density}(E_i) > \frac{1}{4n^3}$

- Hitter property: $\Pr_x[\exists y, \text{Hit}(x, y) \in E_i] > 1 - \frac{1}{t}$
Proof of correctness of our HSG (continued)

- Define $E_i \subseteq \{0, 1\}^m$ by
  
  $$E_i = \{ z \mid \text{start at } u_{i-1}, \text{ read NisGen}(z) \Rightarrow \text{ reach } u_i \}$$

- $\Pr[\text{reach } u_i \mid \text{reach } u_{i-1}] \geq \frac{1}{2n^3} \implies \text{density}(E_i) > \frac{1}{4n^3}$

- Hitter property: $\Pr_x[\exists y, \text{Hit}(x, y) \in E_i] > 1 - \frac{1}{t}$

- Union bound: There is one $x$ so that for all $i$,

  $$\exists y_i, \text{Hit}(x, y_i) \in E_i.$$
Proof of correctness of our HSG (continued)

- Define $E_i \subseteq \{0, 1\}^m$ by
  \[ E_i = \{ z \mid \text{start at } u_{i-1}, \text{ read NisGen}(z) \implies \text{reach } u_i \} \]

- \[ \Pr[\text{reach } u_i \mid \text{reach } u_{i-1}] \geq \frac{1}{2n^3} \implies \text{density}(E_i) > \frac{1}{4n^3} \]

- Hitter property: \[ \Pr_x[\exists y, \text{Hit}(x, y) \in E_i] > 1 - \frac{1}{t} \]

- Union bound: There is one $x$ so that for all $i$, \[ \exists y_i, \text{Hit}(x, y_i) \in E_i. \]

- \[ f(\text{Gen}(x, y_1, \ldots, y_t, n_1, \ldots, n_t)) = 1 \]
Suppose language $L$ can be decided by a randomized log-space algorithm $A$ that always halts with

\[ x \in L \implies \Pr[A(x) \text{ accepts}] \geq \varepsilon = \varepsilon(n) \]
\[ x \notin L \implies \Pr[A(x) \text{ accepts}] = 0. \]
Application: Derandomizing small-success $\textbf{RL}$

- Suppose language $L$ can be decided by a randomized log-space algorithm $A$ that always halts with

  $$x \in L \implies \Pr[A(x) \text{ accepts}] \geq \varepsilon = \varepsilon(n)$$
  $$x \notin L \implies \Pr[A(x) \text{ accepts}] = 0.$$  

- $\varepsilon = \frac{1}{2} \implies L \in \textbf{RL}$. 

Saks, Zhou '95: \( \textbf{RL} \subseteq \text{DSPACE}(\log 3/2n) \)

- In general, Saks and Zhou showed $L \in \text{DSPACE}(\log 3/2n + \sqrt{\log n \log(1/\varepsilon)})$ 

- Theorem: $L \in \text{DSPACE}(\log 3/2n + \log n \log \log(1/\varepsilon))$
Suppose language \( L \) can be decided by a randomized log-space algorithm \( A \) that always halts with

\[
x \in L \implies \Pr[A(x) \text{ accepts}] \geq \varepsilon = \varepsilon(n)
\]

\[
x \not\in L \implies \Pr[A(x) \text{ accepts}] = 0.
\]

\[
\varepsilon = \frac{1}{2} \implies L \in \text{RL}. \quad \text{Saks, Zhou '95: RL} \subseteq \text{DSPACE}(\log^{3/2} n)
\]
Application: Derandomizing small-success \textbf{RL}

- Suppose language \( L \) can be decided by a randomized log-space algorithm \( A \) that always halts with

\[
  x \in L \implies \Pr[\text{\( A(x) \) accepts}] \geq \varepsilon = \varepsilon(n)
\]

\[
  x \not\in L \implies \Pr[\text{\( A(x) \) accepts}] = 0.
\]

- \( \varepsilon = \frac{1}{2} \implies L \in \textbf{RL}. \) Saks, Zhou ’95: \textbf{RL} \subseteq \textbf{DSPACE}(\log^{3/2} n)

- In general, Saks and Zhou showed

\[
  L \in \textbf{DSPACE}(\log^{3/2} n + \sqrt{\log n \log(1/\varepsilon)})
\]
Application: Derandomizing small-success $\textbf{RL}$

- Suppose language $L$ can be decided by a randomized log-space algorithm $A$ that always halts with

  \[
  x \in L \implies \Pr[A(x) \text{ accepts}] \geq \varepsilon = \varepsilon(n)
  \]
  \[
  x \notin L \implies \Pr[A(x) \text{ accepts}] = 0.
  \]

- $\varepsilon = \frac{1}{2} \implies L \in \textbf{RL}$. Saks, Zhou ’95: $\textbf{RL} \subseteq \textbf{DSPACE}(\log^{3/2} n)$

- In general, Saks and Zhou showed

  \[
  L \in \textbf{DSPACE}(\log^{3/2} n + \sqrt{\log n \log(1/\varepsilon)})
  \]

- Theorem:

  \[
  L \in \textbf{DSPACE}(\log^{3/2} n + \log n \log \log(1/\varepsilon))
  \]
Derandomization algorithm for small-success RL

\begin{align*}
\text{start} & \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \bullet \b
Derandomization algorithm for small-success RL

Saks, Zhou '95: Can distinguish in $O(\log^{3/2} n)$ space between $\Pr[reach_v | reach_u] = 0$ vs. $\Pr[reach_v | reach_u] \geq \frac{1}{2^n}$

In second case, add red edge $(u, v)$
Derandomization algorithm for small-success RL

Saks, Zhou '95: Can distinguish in $O(\log^{3/2} n)$ space between

$$\Pr[\text{reach } v \mid \text{reach } u] = 0 \quad \text{vs.} \quad \Pr[\text{reach } v \mid \text{reach } u] \geq \frac{1}{2n^3}$$
Saks, Zhou ’95: Can distinguish in $O(\log^{3/2} n)$ space between

\[ \Pr[\text{reach } v \mid \text{reach } u] = 0 \text{ vs. } \Pr[\text{reach } v \mid \text{reach } u] \geq \frac{1}{2n^3} \]

In second case, add red edge $(u, v)$
Saks, Zhou ’95: Can distinguish in $O(\log^{3/2} n)$ space between

$$\Pr[\text{reach } v \mid \text{reach } u] = 0 \quad \text{vs.} \quad \Pr[\text{reach } v \mid \text{reach } u] \geq \frac{1}{2n^3}$$

In second case, add red edge $(u, v)$
Saks, Zhou '95: Can distinguish in $O(\log^{3/2} n)$ space between

$$\Pr[\text{reach } v \mid \text{reach } u] = 0 \quad \text{vs.} \quad \Pr[\text{reach } v \mid \text{reach } u] \geq \frac{1}{2n^3}$$

In second case, add red edge $(u, v)$
Derandomization algorithm for small-success \textbf{RL} (2)

- Use Savitch’s algorithm to check for path of length $t = \frac{\log(1/\varepsilon)}{\log n}$ from start to acc using red edges

- If $x \in L$, such a path exists by structural lemma

- If $x \not\in L$, no path exists
Derandomization algorithm for small-success $\mathbf{RL}$ (2)

- Use Savitch’s algorithm to check for path of length $t = \frac{\log(1/\varepsilon)}{\log n}$ from start to acc using red edges.

- If $x \in L$, such a path exists by structural lemma.
Derandomization algorithm for small-success RL (2)

- Use Savitch’s algorithm to check for path of length \( t = \frac{\log(1/\varepsilon)}{\log n} \) from start to acc using red edges.
- If \( x \in L \), such a path exists by structural lemma.
- If \( x \notin L \), no path exists.
Restricted case: Derandomizing low-randomness $\text{RL}$

- How many random bits can be derandomized in $O(\log n)$ space?

- Ajtai, Komlós, Szemerédi '87: HSG with seed length $O(\log n)$ for $r \leq O\left(\log^2 n \log \log n\right)$, $\epsilon = 1^{\text{poly}(n)}$.

- Nisan, Zuckerman '93: PRG with seed length $O(\log n)$ for $r \leq \text{polylog } n$, $\epsilon = 2^{-\log 0.99 n}$.

- Theorem: HSG with seed length $O(\log(\frac{n}{\epsilon}))$ for $r \leq \text{polylog } n$. 

Restricted case: Derandomizing low-randomness \(\textbf{RL}\)

- How many random bits can be derandomized in \(O(\log n)\) space?
- \((\log n)\)-space algorithm that uses \(r\) random bits \(\implies\) ROBP with width \(n\) and length \(r\)
Restricted case: Derandomizing low-randomness $\mathbf{RL}$

- How many random bits can be derandomized in $O(\log n)$ space?
- $(\log n)$-space algorithm that uses $r$ random bits $\implies$ ROBP with width $n$ and length $r$
- Ajtai, Komlós, Szemerédi '87: HSG with seed length $O(\log n)$ for $r \leq O\left(\frac{\log^2 n}{\log \log n}\right)$, $\varepsilon = \frac{1}{\text{poly}(n)}$
Restricted case: Derandomizing low-randomness RL

- How many random bits can be derandomized in $O(\log n)$ space?
- $(\log n)$-space algorithm that uses $r$ random bits $\implies$ ROBP with width $n$ and length $r$
- Ajtai, Komlós, Szemerédi '87: HSG with seed length $O(\log n)$ for $r \leq O\left(\frac{\log^2 n}{\log \log n}\right)$, $\varepsilon = \frac{1}{\text{poly}(n)}$
- Nisan, Zuckerman '93: PRG with seed length $O(\log n)$ for $r \leq \text{polylog } n$, $\varepsilon = \frac{1}{2^{\log^{0.99} n}}$
Restricted case: Derandomizing low-randomness \textbf{RL}

- How many random bits can be derandomized in $O(\log n)$ space?
- $(\log n)$-space algorithm that uses $r$ random bits $\implies$ ROBP with width $n$ and length $r$
- Ajtai, Komlós, Szemerédi '87: HSG with seed length $O(\log n)$ for $r \leq O\left(\frac{\log^2 n}{\log \log n}\right)$, $\varepsilon = \frac{1}{\text{poly}(n)}$
- Nisan, Zuckerman '93: PRG with seed length $O(\log n)$ for $r \leq \text{polylog } n$, $\varepsilon = \frac{1}{2^{\log^{0.99} n}}$
- \textbf{Theorem}: HSG with seed length $O(\log(n/\varepsilon))$ for $r \leq \text{polylog } n$
Optimal HSG for $r \leq \text{polylog } n$

- The generator: Same as main construction but with the Nisan-Zuckerman PRG in place of Nisan’s PRG.
Optimal HSG for $r \leq \text{polylog } n$

- The generator: Same as main construction but with the Nisan-Zuckerman PRG in place of Nisan’s PRG

- Analysis difficulty: Vertex $u$ from structural lemma merely satisfies

  $$\Pr[\text{reach } u] \geq \frac{1}{2nr^2}.$$
Optimal HSG for $r \leq \text{polylog } n$

- The generator: Same as main construction but with the **Nisan-Zuckerman PRG** in place of Nisan’s PRG
- Analysis difficulty: Vertex $u$ from structural lemma merely satisfies

$$\Pr[\text{reach } u] \geq \frac{1}{2nr^2}.$$

- Nisan-Zuckerman PRG has **too much error**
Optimal HSG for $r \leq \text{polylog } n$

- The generator: Same as main construction but with the Nisan-Zuckerman PRG in place of Nisan’s PRG

- Analysis difficulty: Vertex $u$ from structural lemma merely satisfies

\[ \Pr[\text{reach } u] \geq \frac{1}{2nr^2}. \]

- Nizan-Zuckerman PRG has too much error

- Solution: Better structural lemma!
Better structural lemma

Let $f$ be a length-$r$ ROBP of any width
Better structural lemma

- Let \( f \) be a length-\( r \) ROBP of any width

- Assume \( \Pr[\text{accept}] = \varepsilon \ll 1/r^2 \)
Better structural lemma

- Let \( f \) be a length-\( r \) ROBP of any width
- Assume \( \Pr[\text{accept}] = \varepsilon \ll 1/r^2 \)
- **Lemma**: There is a subset \( U \) of some layer so that
  
  \[
  \Pr[\text{reach } U] \geq \frac{1}{2r^2} \quad \text{and} \quad \forall u \in U, \ \Pr[\text{accept} \mid \text{reach } u] \geq \varepsilon r.
  \]
Better structural lemma

- Let $f$ be a length-$r$ ROBP of any width
- Assume $\Pr[\text{accept}] = \varepsilon \ll \frac{1}{r^2}$

**Lemma**: There is a subset $U$ of some layer so that

\[
\Pr[\text{reach } U] \geq \frac{1}{2r^2} \quad \text{and} \quad \forall u \in U, \; \Pr[\text{accept } | \text{ reach } u] \geq \varepsilon r.
\]

**Proof**: Similar to the proof of the original structural lemma
Let $f$ be a length-$r$ ROBP of any width

Assume $\Pr[\text{accept}] = \varepsilon \ll 1/r^2$

**Lemma**: There is a subset $U$ of some layer so that

$$\Pr[\text{reach } U] \geq \frac{1}{2r^2} \quad \text{and} \quad \forall u \in U, \ Pr[\text{accept } \mid \text{reach } u] \geq \varepsilon r.$$ 

**Proof**: Similar to the proof of the original structural lemma

(Error of NZ generator) $\ll \frac{1}{2r^2} = \frac{1}{\text{polylog } n}$
Application: Randomness vs. nondeterminism

\[ \text{RL} \subseteq \text{NL} \]
Application: Randomness vs. nondeterminism

-RL ⊆ NL

- Theorem: For any $r = r(n)$

\[(RL \text{ with } r \text{ coins}) ⊆\]
Application: Randomness vs. nondeterminism

- **RL \( \subseteq \) NL

- **Theorem:** For any \( r = r(n) \) and any constant \( c \),

\[(\text{RL with } r \text{ coins}) \subseteq \left(\text{NL with } \frac{r}{\log^c n} \text{ nondeterministic bits}\right)\]
Application: Randomness vs. nondeterminism

- **RL ⊆ NL**

- **Theorem:** For any \( r = r(n) \) and any constant \( c \),

\[
(\text{RL with } r \text{ coins}) \subseteq \left( \text{NL with } \frac{r}{\log^c n} \text{ nondeterministic bits} \right)
\]
Simulating $r$ coins with $r/\log^{c}n$ nondeterministic bits
Simulating $r$ coins with $r/\log^c n$ nondeterministic bits

Pseudorandom
Simulating $r$ coins with $r/\log^c n$ nondeterministic bits

\[ \varepsilon < \frac{1}{2r} \]
Proof that this works: Suppose $\Pr[\text{accept}] = \alpha$
Proof that this works: Suppose $\Pr[\text{accept}] = \alpha$

Let $L$ be the layer reached after $\log^{c+1} n$ steps
Simulating $r$ coins with $r/\log^c n$ nondeterministic bits (2)

- Proof that this works: Suppose $\Pr[\text{accept}] = \alpha$
- Let $L$ be the layer reached after $\log^{c+1} n$ steps
- Define $U = \{u \in L : \Pr[\text{accept} \mid \text{reach } u] \geq \alpha - \varepsilon\}$
Proof that this works: Suppose $\Pr[\text{accept}] = \alpha$

Let $L$ be the layer reached after $\log^{c+1} n$ steps

Define $U = \{ u \in L : \Pr[\text{accept} | \text{reach } u] \geq \alpha - \varepsilon \}$

Then $\alpha = \Pr[\text{accept}]$

$$= \sum_{u \in U} \Pr[u] \cdot \Pr[\text{acc} | u] + \sum_{u \in L \setminus U} \Pr[u] \cdot \Pr[\text{acc} | u]$$
Simulating $r$ coins with $r / \log^c n$ nondeterministic bits (2)

- Proof that this works: Suppose $\Pr[\text{accept}] = \alpha$
- Let $L$ be the layer reached after $\log^{c+1} n$ steps
- Define $U = \{u \in L : \Pr[\text{accept} | \text{reach } u] \geq \alpha - \varepsilon\}$
- Then $\alpha = \Pr[\text{accept}]
  \begin{align*}
  &= \sum_{u \in U} \Pr[u] \cdot \Pr[\text{acc} | u] + \sum_{u \in L \setminus U} \Pr[u] \cdot \Pr[\text{acc} | u] \\
  &\leq \Pr[U] + (\alpha - \varepsilon)
  \end{align*}
Simulating $r$ coins with $r / \log^c n$ nondeterministic bits (2)

- Proof that this works: Suppose $\Pr[\text{accept}] = \alpha$
- Let $L$ be the layer reached after $\log^{c+1} n$ steps
- Define $U = \{ u \in L : \Pr[\text{accept} \mid \text{reach } u] \geq \alpha - \varepsilon \}$
- Then $\alpha = \Pr[\text{accept}]
  \begin{align*}
  &= \sum_{u \in U} \Pr[u] \cdot \Pr[\text{acc} \mid u] + \sum_{u \in L \setminus U} \Pr[u] \cdot \Pr[\text{acc} \mid u] \\
  &\leq \Pr[U] + (\alpha - \varepsilon) \\
  \Pr[U] &\geq \varepsilon.
  \end{align*}
Simulating $r$ coins with $r/\log^c n$ nondeterministic bits (2)

- Proof that this works: Suppose $\Pr[\text{accept}] = \alpha$
- Let $L$ be the layer reached after $\log^{c+1} n$ steps
- Define $U = \{u \in L : \Pr[\text{accept} | \text{reach } u] \geq \alpha - \varepsilon\}$
- Then $\alpha = \Pr[\text{accept}]
  = \sum_{u \in U} \Pr[u] \cdot \Pr[\text{acc} | u] + \sum_{u \in L \setminus U} \Pr[u] \cdot \Pr[\text{acc} | u]
  \leq \Pr[U] + (\alpha - \varepsilon)
  \Pr[U] \geq \varepsilon.$
- So some seed $x$ leads to $U$. Induct
General theorem: Reduction to $1/\text{poly}$ error case

- Assume efficient PRG for ROBPs with seed length $m$ and error $\frac{1}{r^2}$
General theorem: Reduction to $1/\text{poly}$ error case

- Assume efficient PRG for ROBPs with seed length $m$ and error $\frac{1}{r^2}$

- **Theorem**: For every $\varepsilon > 0$, there’s an efficient HSG for ROBPs with seed length

  $$O(m + \log(nr/\varepsilon))$$
The case polylog $n \ll r \ll n$

- **Theorem**: HSG for width-$n$, length-$r$ ROBPs with seed length
  \[ O \left( \frac{\log(nr) \log r}{\max\{1, \log \log n - \log \log r\}} + \log(1/\varepsilon) \right) \]

- **Proof**: Plug in PRG of [Armoni '98]
Open questions

- **Conjecture**: For any $r = r(n)$, for any constant $c$,

$$\text{(BPL with } r \text{ coins)} = \left( \text{BPL with } \frac{r}{\log^c n} \text{ coins} \right)$$
Open questions

- **Conjecture**: For any \( r = r(n) \), for any constant \( c \),

\[
(BPL \text{ with } r \text{ coins}) = (BPL \text{ with } \frac{r}{\log^c n} \text{ coins})
\]

- True for \( r \leq 2^{\log^{0.99} n} \) by Nisan-Zuckerman
Open questions

- **Conjecture**: For any $r = r(n)$, for any constant $c$,

\[(\text{BPL with } r \text{ coins}) = \left( \text{BPL with } \frac{r}{\log^c n} \text{ coins} \right)\]

- True for $r \leq 2^{\log^{0.99} n}$ by Nisan-Zuckerman

- ACR '96: Explicit HSG for circuits $\implies P = BPP$. Similar theorem for BPL?
Open questions

▶ **Conjecture**: For any $r = r(n)$, for any constant $c$,

$$(\text{BPL with } r \text{ coins}) = \left( \text{BPL with } \frac{r}{\log^c n} \text{ coins} \right)$$

▶ True for $r \leq 2^{\log^{0.99} n}$ by Nisan-Zuckerman

▶ ACR '96: Explicit HSG for circuits $\implies$ $\text{P} = \text{BPP}$. Similar theorem for $\text{BPL}$?

▶ Thanks! Questions?