Today

Saks-Zhou Theorem

$BPL \subseteq L^{3/2}$.

CSS.413.1

Pseudorandomness

Lecture 16 (2021-10-21)

Instructor: Prahladh Harsha.

Last time: Nisan's PRG for small space

Theorem: There is a PRG $G: \{0,1\}^{\log^2 n} \to \{0,1\}^{poly(n)}$ that $1/n^2$-fools any $O(\log n)$-space machine and furthermore G can be computed in space $O(\log n)$.

Naive derandomization + (above) theorem

Corollary: BPL (randomized logspace) $\subseteq SPACE(\log^2 n)$ $(= L^2)$.

Today, an improvement

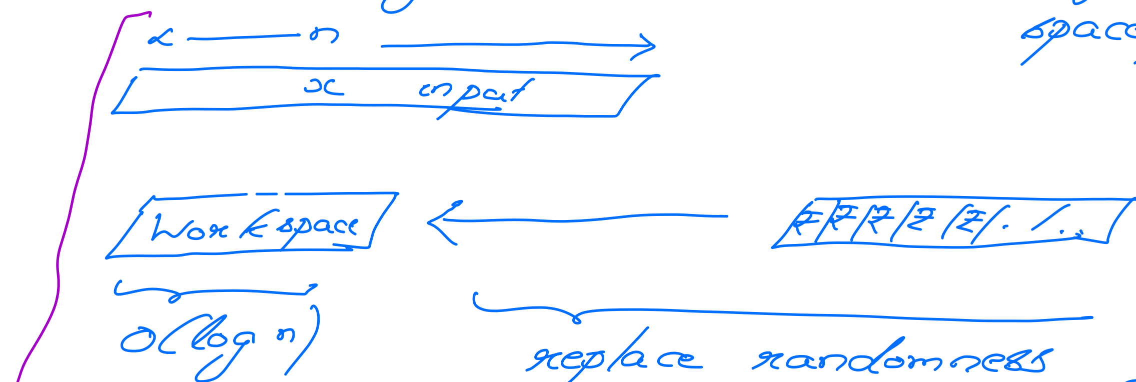
Theorem [Saks-Zhou] $BPL \subseteq L^{3/2}$

Open: $BPL \subseteq L$

Warmup:

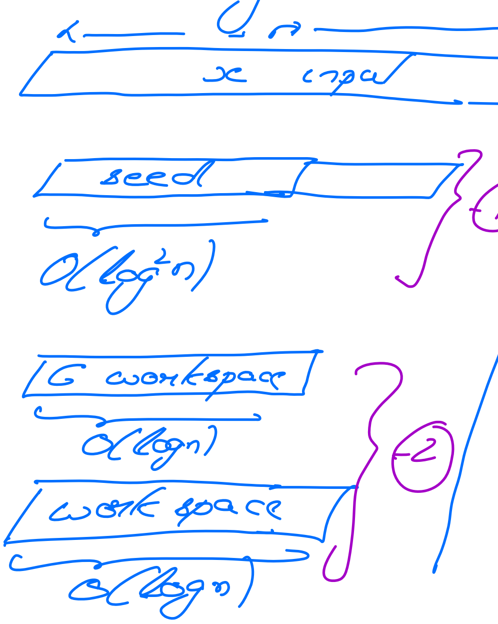
Proof of $BPL \subseteq L^2$ (from Nisan's PRG).

M - randomized TM (runs in $O(\log n)$ space)



replace randomness using Nisan's PRG.

Determinize it



Instead of reading the random bits from the random tape the deterministic simulation invokes the PRG G to obtain the random bits.

$$\text{Total space} = O(\log^2 n) + O(\log n) + O(\log n)$$
$$= O(\log^2 n)$$

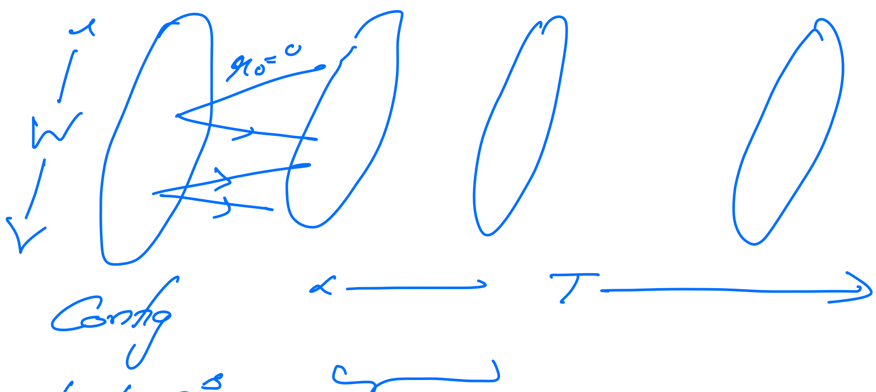
Space of Deterministic Simulator.

(1) Random Space (Space for seed)

(2) Processing Space. (both G & m/c M).

Space Bdd Computation = Matrix Exponentiation

Space s m/c can be expressed as a branching program



Config $|W| \approx 2^s$

$$M \in \mathbb{R}^{W \times W}$$

$M[i,j] = \Pr[$Machine moves from state j given that it is state $i]$

Typically $M[i,j] \in \{0, \frac{1}{2}, 1\}$

M - row stochastic matrix.

$M^T[\text{start}, \text{acc}]$ = determines if o/p is acc.

BPL "reduces" to space-efficient computation of matrix exponentiation

W.l.o.g. $T = 2^r$ (power of 2)

(by making acc & rej states as sink states).

Problem.

Input: $M \in \mathbb{R}^{d \times d}$ (stochastic matrix)

2^r (exponentiation parameter)

$i, j \in [d]$

Output: $M^{2^r}[i,j]$

__Soln:__ Repeated Squaring

$M \longmapsto M^2$

$$\left.M^2[i,j] = \sum_{k=1}^{d} M[i,k]\, M[k,j]\right\} O(\log d) \text{ space}$$

$$M^{2^r} = \underbrace{(((M^2)^2)^2 \cdots)^2}_{r \text{ times}}$$

Space – $O(r \log d)$.

Observation: Suffices to compute M^{2^r} approximately

Notion of approximation

$$\|A\|_\infty \triangleq \max_c \sum_d |A(i,j)|$$

Properties: (1) Subadditivity $\|A+B\|_\infty \le \|A\|_\infty + \|B\|_\infty$

(2) Submultiplicativity : $\|AB\|_\infty \le \|A\|_\infty \cdot \|B\|_\infty$

(3) A, B, A', B' – stochastic matrices

$$\|AB - A'B'\|_\infty \le \|A - A'\|_\infty + \|B - B'\|_\infty$$

Problem:

Approximate Matrix Repeated Squaring (AMRS)

Input: M – $d \times d$ substochastic matrix

2^r – exponentiation parameter

2^a – accuracy parameter

Output: M' – $d \times d$. substochastic matrix
$$\| M' - M^{2^r} \|_\infty \le \frac{1}{2^a}$$

substochastic matrix
- $M(i,j) \ge 0$
- $\forall i, \ \sum_j M(i,j) \le 1$

<u>Obs 1:</u> Repeated Squaring solves AMRS exactly.

<u>Parameters:</u> d – dim matrix
r – exponentiation parameter
a – accuracy parameter

$$s = \max\{r, a, \log d\}$$

(logn scenario, $s \approx r \approx a$
$\approx \log d$
$\approx O(\log n)$)

<u>Obs 2</u> Nisan's PRG yields a soln to AMRS, which we

will refer to Pseudorandom Repeated Squaring (PRS)

Recall Nisan's PRG.

$$G : \{0,1\}^{s+O(ks)} \longrightarrow \{0,1\}^{s \cdot 2^k}$$

$$(x, h_1 \ldots h_k) \longmapsto \underbrace{\qquad\qquad}_{s \cdot 2^k\text{-bits long.}}$$

G - constructible in O(s)-space.

$$\left| \mathbb{E}_{x \leftarrow \{0,1\}^{s+O(ks)}}[T(G(x))] - \mathbb{E}_{z \leftarrow \{0,1\}^{s2^k}}[T(z)] \right| \leq \varepsilon$$

for all O(s)-space m/c T.

Lemma: (Abstraction of Nisan's PRG).
There exists an algorithm PRS.

Input: M - d×d. sub stochastic

2^r - exp. param

2^a - accuracy parameter

$\bar{h} = (h_1 \ldots h_k)$ - seed. $h' \in \{0,1\}^{O(r \log \frac{1}{\varepsilon})}$.

Output: M' s.t $\|M' - M^{2^r}\| \le \frac{1}{2^a}$

w/ prob $1-\varepsilon$ over the choice of $\bar{e}$

	RS	PRS (Nisan)	SZ
Random Space	0	$O(rs)$	$O(\sqrt{r}\, s)$
Processing Space	$O(rs)$	$O(s)$	$O(\sqrt{r}\, s)$

Idea: Use PRS recursively

$$r = r_1 r_2$$

$$M^{2^r} = \underbrace{\left(\left(\left(M^{2^{r_1}}\right)^{2^{r_1}}\right)^{2^{r_1}} \cdots \right)^{2^{r_1}}}_{r_2 \text{ times.}}$$

More precisely

$$\Lambda^{(r)}(M) = M^{2^r}$$

$$\Lambda^{(r)}(M) = M^{2^r} = \left(\Lambda^{(r_1)}\right)^{r_2}(M)$$

Algorithmically

$$
\left.\begin{array}{l}
N_0 \leftarrow M \\
\text{For } i \leftarrow 1 \text{ to } r_2 \\
\quad \underbrace{N_i \leftarrow N_{i-1}^{2^{r_1}}}
\end{array}\right\}
$$

Idea: use PRS to perform this step.

How?

$$
\left[\begin{array}{l}
\text{Pick } \ell_1 \ldots \ell_{r_2} \leftarrow \{0,1\}^{r_1 s} \\
M_0 \leftarrow M \\
\text{For } i \leftarrow 1 \text{ to } r_2 \\
\quad M_i \leftarrow PRS(M_{i-1}, r_1, a; \ell_i)
\end{array}\right]
$$

Random space $\quad r_2 \cdot O(r_1 s) = O(rs)$

Processing space $\quad r_2 \, O(s) = O(r_2 s)$

Qn: Can we use the same seed $\bar{\ell}$ for all r_2 runs of PRS?

ie, Does the following alg work

$$
\left[\begin{array}{l}
\text{Pick } \bar{\ell} \leftarrow \{0,1\}^{O(r_1 s)}
\end{array}\right.
$$

$M_0 \leftarrow M$

For $i \leftarrow 1$ to n_2

$\quad M_i \leftarrow PRS(M_{i-1}, \eta, a_i; \ell)$

Unfortunately, we don't know how to prove this works as the M_i's are not independent of the random seed ℓ.

$\bar{\ell}$ - works w/ N_i's but not M_i's.

Idea 2: Truncate M_i to t bits of accuracy.

More precisely,

$z \in \mathbb{R}$, $\quad \lfloor z \rfloor_t = 2^{-t} \lfloor 2^t z \rfloor$

Truncation.

$\lfloor M \rfloor_t$ — entry-by-entry truncate.

To ensure the truncation of M_i & N_i are identical

- perturb every entry

Perturbation.

$\delta \in (0,1) \qquad z \in \mathbb{R}$

$$\Sigma_\delta z = \max\{z-\delta, 0\}$$

$\Sigma_\delta(M)$ - entry-by-entry perturbation.

AMRS - Algorithm

Input: M - $d \times d$ substochastic matrix

2^r - exp $(r = r_1 r_2)$.

2^a - accuracy

t - truncation parameter

k - perturbation parameter

$k \geq t$

$k = t + D \quad (D \geq 0)$.

Algorithm:

Random: $\bar{\ell} \leftarrow \{0,1\}^{r_1 s}$

$q_1, q_2 \ldots \quad q_{r_2} \in \{0,1\}^D$

$(q_i \in [0, 2^D - 1]$

1. $M_0 \leftarrow M$

2. For $i \leftarrow 1$ to r_2.

$$M_i' \leftarrow PRS(M_{i-1}, r_1, a; \bar{\ell})$$

$$\tilde{M}_i \leftarrow \sum_{\delta_i}(M_i')$$

where $\delta_i = \frac{q_i}{2^k}$

$$M_i \leftarrow \lfloor \tilde{M}_i \rfloor_t.$$

3. Output M_{r_2}

Meta Algorithm:

$N_0 \leftarrow M$

For $i \leftarrow 1$ to r_2

$$N_i' \leftarrow N_{i-1}^{2^{r_1}}$$

$$\tilde{N}_i \leftarrow \sum_{\delta_i} N_i \quad \text{where} \quad \delta_i = \frac{q_i}{2^k}$$

$$N_i \leftarrow \lfloor \tilde{N}_i \rfloor_t$$

Missing theorem stmts to complete SZ analysis.

Main Theorem [Saks Zhou]

The output M_{r_2} of AMRS-algorithm approximates M^{2^r} w/ accuracy $\dfrac{2^{D+2r+\log d}}{2^k}$ and error probability at most $r_2(\varepsilon + 2d^2/2^D)$

This thm is proved via the following intermediate claims.

Claim 1: For any choice of $q_1 \dots q_{r_2} \in \{0,1\}^D$ the sequence $N_0, N_1, \dots N_{r_2}$ satisfies

$$\| N_{r_2} - M^{2^r} \|_\infty \le \frac{2^{D+2r+\log d}}{2^k}$$

Definition:

① We say that a random seed $\bar{\ell}$ is (M,ε)-good if:

$$\| PRS(M, r_1, \log\tfrac{1}{\varepsilon}; \ell) - M^{2^{r_1}} \|_\infty \le \varepsilon.$$

(2) A real number r is (b,t)-dangerous for positive integers $b > t$ if

$$r = 2^{-t} I + \rho \text{ for some integer } I$$
$$\& \ \rho \in [-2^{-b}, 2^{-b})$$

Claim 2: If the random seeds $\bar{l}$ & $q_1 \ldots q_r$ satisfy the following two properties:

(a) $\forall i \in [r_2]$, none of the entries in any of the matrices $\tilde{N}_i$ are (k,t)-dangerous

(b) $\bar{l}$ is (N_i, ε)-good for $\forall i \in [r_2]$

the $M_{r_2} = N_{r_2}$ (infact $M_i = N_i$, $\forall i \in [r_2]$)

Claim 3: For any $q_1 \ldots q_{r_2} \in \{0,1\}^D$

$$\Pr_{\bar{l}}\left[\exists i \in [r_2],\ \bar{l} \text{ is not } (N_i, \varepsilon)\text{-good}\right] \le r_2 \varepsilon$$

Claim 4: $\Pr_{q_1 \ldots q_{r_2}}\left[\exists i \in [r_2],\ \tilde{N}_i \text{ is } (k,t)\text{-dangerous}\right] \le 2 r_2 d^2 / 2^D$.