CS 830: Randomness and Complexity 03/15/05

Lecture 17: Nisan's generator for Space Bounded Computation (Part 1)

instructor: Jin-Yi Cai Scribe: Rakesh Kumar and Yunpeng Li

1 Introduction

Today we will start looking at another main theme of the course, namely space bounded computation. In particular we will be interested in two main problems namely the deterministic space complexity of undirected connectivity(UCONN) and the deterministic simulation of randomized logspace(RL). The first problem has recently been solved by Reingold who showed that UCONN can be solved in deterministic logspace thereby showing that SL = L. We will review that result in due course. Another goal is the deterministic simulation of RL in $\log^{3/2} n$ space and polynomial time. This is still an open problem and we will see some results on the deterministic simulation of RL. In this lecture we will start looking at Nisan's generator for space bounded computation, which shows that RL can be simulated in $\log^2 n$ space and polynomial time.

One of the earliest results in this field is Savitch's theorem. Savitch showed how NL can be solved in DSPACE[$\log^2 n$]. This immediately leads to the following relations among complexity classes

$$\mathrm{SL} \subseteq \mathrm{RL} \subseteq \mathrm{NL} \subseteq \mathrm{DSPACE}[\log^2 n]$$

For a while this was the best result known to us. Then Nisan, Szemeredi and Wigderson showed that UCONN $\in$ DSPACE[$\log^{3/2} n$]. Later Armoni et. al showed that UCONN $\in$ DSPACE[$\log^{4/3} n$]. This remained the best result for the deterministic simulation of UCONN till Reingold showed that we can infact solve UCONN in deterministic logspace.

While on one hand researchers tried to analyze the problem of undirected connectivity, there were also effors to look at the deterministic simulation of RL. While Savitch's theorem does show that RL $\subseteq$ DSPACE[$\log^2 n$], it takes exponential time. Nisan first showed how RL can be deterministically simulated in $\log^2 n$ space and $poly(n)$ time. Thus he proved that RL $\subseteq$ DTISP[$2^{O(\log n)}, \log^2 n$]. Later on Saks and Zhou showed how RL and BPL can be simulated in $\log^{3/2} n$ space but the running time is exponential in the space required. Thus their proof yields that RL $\subseteq$ DTISP[$2^{\log^{3/2} n}, \log^{3/2} n$]. Cai, Chakaravarthy and van Melkebeek showed that there is a time space tradeoff between these two results. In particular they showed that for any $0 \leq \delta \leq 0.5$, BPL $\subseteq$ DTISP[$2^{O(\log^{3/2+\delta} n)}, \log^{3/2+\delta} n$]. We will look at this result in a later lecture.

Finally we may also discuss Reingold's new algorithm in the context of the JAG model. Reingold's algorithm can be seen to work in the JAG model. This is important since in the JAG model there is a $\log^2 n$ lower bound for the graph accessibility problem (GAP) on directed graphs. This indicates that any trivial modification of Reingold's algorithm is not likely to solve GAP on directed graphs.

Now let us start looking at Nisan's generator (which we have already seen), in the context of space bounded derandomization.

2 Recap Nisan's generator

In this section we will recall some of the main ideas about Nisan's generator that will be used in later sections. Recall that Nisan's generator was defined recursively as follows

$$\begin{aligned} G_0(x) &= x \\ \text{for } k \geq 0,\ G_{k+1}(x; h_1, \cdots, h_{k+1}) &= G_k(x; h_1, \cdots, h_k) \circ G_k(h_{k+1}(x); h_1, \cdots, h_k) \end{aligned}$$

We had also introduced a notion of a hash function h being ϵ-random.

DEFINITION 2.1 *Given a set $C \subseteq E \times F$, and a hash function $h : E \mapsto F$, h is said to be ϵ-random on C if*

$$| \, Pr_{x \in_U E}[x \circ h(x) \in C] - Pr_{x \in_U E, y \in_U F}[x \circ y \in C] \, | \leq \epsilon.$$

A key lemma in the proof of the pseudorandom behavior of Nisan's generator was the following lemma which was has been proved in an earlier lecture

LEMMA 2.2 *Nisan's lemma: Let $\mathcal{H}$ be a uniform family of hash functions $h : E \mapsto F$. Then $\forall C \subseteq E \times F$*

$$Pr_{h \in_U \mathcal{H}}[h \text{ is not } \epsilon\text{- random on } C] \leq \frac{\mu(C)}{\epsilon^2 |E|}$$

In the next section we will see how Nisan's generator can be used to simulate RL deterministically.

3 Simulating RL

Consider a randomized logspace machine M. We can assume that at every step the machine reads one random bit. Since the amount of space available to the machine is $S = O(\log n)$, hence the number of possible configurations of the machine is $O(2^S) = poly(n)$. Henceforth we will assume without loss of generality that the number of configurations is 2^S.
Now we can also consider the machine M as reading $t = O(\log n)$ bits at one time (instead of the more natural notion of reading a single bit). Specifically, we can consider our machine M to be a finite automaton Q having 2^S states over an alphabet of $\{0,1\}^t$. We will show that for a random choice of x and randomly chosen hash functions $h_1, \cdots, h_k$ (k will be chosen as $O(\log n)$), the behavior of the automaton on $G_k(x; h_1, \cdots, h_k)$ will be pretty close to its behavior on truly random bits.

Let $G_k(*; h_1, \cdots, h_k)$ be the distribution on $\{0,1\}^{t2^k}$ obtained from the output of Nisan's generator when we fix $h_1, \cdots, h_k$) and choose x uniformly from $\{0,1\}^t$. We will denote the uniform distribution on $\{0,1\}^{t2^k}$ by U_{t2^k}.
For a distribution D, we let $Q(D)$ denote the transition matrix of our automaton Q under the distribution D. Specifically, if p is a probability distribution on the states of the automaton Q, then $Q(D)p$ is the new probability distribution on the states of Q after it has read a block of l

random bits where D is a distribution on strings of length l.

The proof is much easier to analyze in terms of matrix norms and so we take a little detour and introduce the concept of matrix norms.
A stochastic matrix is one in which every entry is positive ($M_{i,j} \geq 0$) and every column sum is 1 ($\forall i, \sum_{j=1}^{n} M_{j,i} = 1$). This is because the $(i,j)^{th}$ entry represents the probability of going from a state i to a state j. Thus any entry being a probability is non-negative and the sum of entries in a column is the sum of probabilities of going from a state to some state and this is 1.

Let $x \in \mathbb{R}^n$ and let $M \in \mathbb{R}^{n \times n}$.
The 1-norm of a vector is defined as the following.

$$||x||_1 = \sum_{i=1}^{n} |x_i|$$

Similarly the 1-norm of a matrix is defined as

$$||M||_1 = max_{x \neq 0} \frac{||Mx||}{||x||} = max_{||x||=1} ||Mx||$$

The following are left as exercises for the reader.

$$\begin{aligned} ||M+N|| &\leq ||M|| + ||N|| \\ ||MN|| &\leq ||M||.||N|| \\ ||M|| &\leq max_j \sum_{i=1}^{n} |M_{i,j}| \end{aligned}$$

We are now ready to prove the main lemma.

Lemma 3.1

$$Pr_{h_1,\cdots,h_k \in \mathcal{H}}[||Q(G_k(*;h_1,\cdots,h_k)) - Q(U_{t.2^k})|| > (2^k - 1)\epsilon] \leq \frac{2^{5S}.k}{\epsilon^2 2^t}$$

Proof. We prove this by induction on the value of k. The base case ($k = 0$) follows trivially from the fact that $G_0(*) = U_t$.
Assume that the statement is true for $k-1$ and we shall show that it is true for k.
For a fixed i,j and fixed hash functions $h_1, \cdots, h_{k-1}$ we define the set
$B_{i,j}^{h_1,\cdots,h_{k-1}} = \{x \mid G_{k-1}(x; h_1, \cdots, h_{k-1})$ takes Q from i to $j\}$. We will define two events
$E_1 : ||Q(G_{k-1}(*; h_1, \cdots, h_{k-1})) - Q(U_{t2^{k-1}})|| \leq (2^{k-1} - 1)\epsilon$.
E_2 : For all $i, j, l \in \{1, \cdots, 2^S\}$, h_k is $2^{-2S}\epsilon$ random for $(B_{i,l}^{h_1,\cdots,h_{k-1}}, B_{l,j}^{h_1,\cdots,h_{k-1}})$.
We will first prove the following

Claim 1

$$Pr[\neg E_1 \vee \neg E_2] \leq \frac{2^{5S}k}{\epsilon^2 2^t}$$

Proof. We will compute $Pr[\neg E_1]$ and $Pr[\neg E_2]$ separately and then use union bound to get the result.
Now $Pr[\neg E_1] \leq \frac{2^{5S}(k-1)}{\epsilon^2 2^t}$ by the induction hypothesis. Also

$$\begin{aligned}
Pr_{h_k}[\neg E_2] &\leq max_{h_1,\cdots,h_{k-1}} Pr_{h_k}[\exists i,j,l \mid h_k \text{ is not } 2^{-2S}\epsilon \text{ random for } (B_{i,l}^{h_1,\cdots,h_{k-1}}, B_{l,j}^{h_1,\cdots,h_{k-1}})] \\
&\leq max_{h_1,\cdots,h_{k-1}} \sum_i \sum_j \sum_l Pr_{h_k}[h_k \text{ is not } 2^{-2S}\epsilon \text{ random for } (B_{i,l}^{h_1,\cdots,h_{k-1}}, B_{l,j}^{h_1,\cdots,h_{k-1}})] \\
&\leq \sum_i \sum_j \sum_l \frac{2^{4S}}{\epsilon^2} . \frac{\mu(B_{i,l}^{h_1,\cdots,h_{k-1}})\mu(B_{l,j}^{h_1,\cdots,h_{k-1}})}{2^t} \text{ by lemma 2.2} \\
&\leq \sum_i \sum_l \frac{2^{4S}}{\epsilon^2} \frac{\mu(B_{i,l}^{h_1,\cdots,h_{k-1}})}{2^t} \sum_j \mu(B_{l,j}^{h_1,\cdots,h_{k-1}}) \\
&\leq \sum_i \sum_l \frac{2^{4S}}{\epsilon^2} \frac{\mu(B_{i,l}^{h_1,\cdots,h_{k-1}})}{2^t} \\
&\leq \sum_i \frac{2^{4S}}{2^t\epsilon^2} \sum_l \mu(B_{i,l}^{h_1,\cdots,h_{k-1}}) \\
&\leq \sum_i \frac{2^{4S}}{2^t\epsilon^2} \\
&\leq \frac{2^{5S}}{2^t\epsilon^2}
\end{aligned}$$

Thus we have

$$Pr[\neg E_1 \vee \neg E_2] \leq Pr[\neg E_1] + Pr[\neg E_2] \leq \frac{2^{5S}(k-1)}{\epsilon^2 2^t} + \frac{2^{5S}}{2^t\epsilon^2} = \frac{2^{5S}k}{2^t\epsilon^2}$$

This concludes the proof of Claim 1. In the next lecture we will show that if E_1 and E_2 hold then $||Q(G_k(*;h_1,\cdots,h_k)) - Q(U_{t.2^k})|| \leq (2^k-1)\epsilon$. This will prove the lemma.