Stanford University	**Spring 2016**
Math 233: Non-constructive methods in combinatorics	**Instructor: Jan Vondrák**
Lecture date: April 18, 2016	**Scribe: Huy Tuan Pham**

Lecture 8. Algorithmic Local Lemma

Here we discuss an algorithm for applications of the LLL found in the breakthrough work of [Moser and Tardos (2010)].

8.1 Setup

Given underlying independent random variables $X_1, ..., X_m$ with product measure μ. The "bad events" $\mathcal{E}_1, ..., \mathcal{E}_n$ are each determined by a certain subset of the random variables, which we denote $var(\mathcal{E}_i) \subset [n]$. The dependency graph G has vertices $[n]$ and edges $(i,j) \in E(G)$ whenever $var(\mathcal{E}_i) \cap var(\mathcal{E}_j) \neq \emptyset$. Note that this is a valid choice of a dependency graph, since each event $\mathcal{E}_i$ is independent of any conditioning on the variables outside of $var(\mathcal{E}_i)$.

Given that the conditions of the Lovász Local Lemma (or Shearer's Lemma) hold, we want to find a realization of the random variables $X_1, ..., X_m$ such that no events E_i happen.

8.2 The Moser-Tardos algorithm

1. Sample $X_1, X_2, ..., X_m$ from the distribution μ.
2. As long as any event $\mathcal{E}_i$ is satisfied by the current values of $X_1, \ldots, X_m$, choose any such i and resample $var(\mathcal{E}_i)$: replace $(X_a : a \in var(\mathcal{E}_i))$ by new independent samples.

It is clear that if the algorithm terminates, then we have found an assignment avoiding all events. The key is to analyze the expected number of resampling steps. By one resampling step, we mean the operation of resampling all the variables of an event.

Theorem 8.1 (Moser-Tardos) *The expected number of resampling steps before termination of the algorithm is at most* $\sum_{i=1}^n \frac{x_i}{1-x_i}$, *provided that* $\mathbb{P}[\mathcal{E}_i] \leq x_i \prod_{j \in \Gamma(i)}(1-x_j)$.

In applications, the x_i are usually small (certainly $x_i \leq \frac{1}{2}$), so this means the expected number of resampling operations is $O(n)$. Subsequently, [Kolipaka and Szegedy (2011)] extended the analysis to the setting of Shearer's lemma and proved the following (for the same algorithm).

Theorem 8.2 (Kolipaka-Szegedy) *The expected number of resampling steps before termination of the algorithm is at most* $\sum_{i=1}^n \frac{q_{\{i\}}}{q_\emptyset}$, *provided that Shearer's conditions are satisfied with coefficients* q_S.

We note that under LLL conditions, we have $\frac{q_{\{i\}}}{q_\emptyset} \leq \frac{x_i}{1-x_i}$, so this subsumes the theorem above. In the following, we present an exposition of the analysis close to [Kolipaka and Szegedy (2011)]

8.3 Analysis of the algorithm

We define the *execution log* of the algorithm as the sequence of events that get resampled: $(\mathcal{E}_{i_1}^{(1)}, \mathcal{E}_{i_2}^{(2)}, \ldots)$, where $\mathcal{E}_i^{(t)}$ denotes the fact that the event $\mathcal{E}_i$ was resampled at time t. We want to prove that

$$\mathbb{E}[\text{the number of times } \mathcal{E}_i \text{ gets resampled}] \leq \frac{x_i}{1-x_i}.$$

Stable set sequences. An important notion in the analysis will be that of stable set sequences. First, given the log, we define a directed graph R on vertices $\mathcal{E}_i^{(s)}$ as follows. For each pair of entries in the log, $E_i^{(s)}$ and $E_{i'}^{(s')}$, we add a directed edge $(E_i^{(s)}, E_{i'}^{(s')})$ to R if $s < s'$ and $(\mathcal{E}_i, \mathcal{E}_{i'}) \in E(G)$.

For a fixed entry $\mathcal{E}_i^{(t)}$ in the log, let us consider a subgraph $R_i^{(t)} \subset R$, induced by the vertices that have a directed path to $\mathcal{E}_i^{(t)}$. We call $\mathcal{E}_i^{(t)}$ the root of $R_i^{(t)}$. For each $\ell \geq 0$, we define a set of events:

$$I_\ell = \{j : \exists \mathcal{E}_j^{(s)} \in V(R_i^{(t)}), \text{ the longest path from } \mathcal{E}_j^{(s)} \text{ to the root } \mathcal{E}_i^{(t)} \text{ has length exactly } \ell\}.$$

Note that $I_0 = \{\mathcal{E}_i^{(t)}\}$. We have the following properties:

1. For every $\ell \geq 0$, I_ℓ is an independent set in G.

Proof: If $j, j' \in I_\ell$ and $(j, j') \in E(G)$, then there must be a directed edge $(\mathcal{E}_j^{(s)}, \mathcal{E}_{j'}^{(s')})$ in $R_i^{(t)}$ (we can assume wlog that $s < s'$). This means that $\mathcal{E}_j^{(s)}$ has a path to the root through $\mathcal{E}_{j'}^{(s')}$ that has length 1 more than the longest path from $\mathbb{E}_{j'}^{(s')}$ to the root. This contradicts the fact that the longest paths from both nodes to the root have length ℓ.

2. For every $\ell \geq 0$, $I_{\ell+1} \subseteq \Gamma^+(I_\ell)$.

Proof: For every $j \in I_{\ell+1}$, there is a longest directed path from $\mathcal{E}_j^{(s)}$ to the root of length $\ell+1$. So the next vertex on the path must have a longest path to the root of length ℓ. This vertex corresponds to an event $j' \in I_\ell$ and by construction of the directed graph, we have that $j \in \Gamma^+(\mathcal{E}_{j'})$.

This motivates the following definition.

Definition 8.3 *A stable set sequence for G is a finite sequence of sets $\mathcal{I} = (I_0, I_1, ..., I_r)$ such that for every $0 \leq \ell \leq r$, I_ℓ is an independent set in G and for every $0 \leq \ell < r$, $I_{\ell+1} \subseteq \Gamma^+(I_\ell)$.*

By the discussion above, every sequence $\mathcal{I} = (I_0, I_1, ..., I_r)$ produced from a log of execution of the algorithm is a stable set sequence (note that it must be finite, since for a fixed root $\mathcal{E}_i^{(t)}$ the induced subgraph $R_i^{(t)}$ is finite).

Definition 8.4 *A stable set sequence $\mathcal{I}$ is said to be a witness of a resampling $\mathcal{E}_i^{(t)}$ if it is produced from the log by the above process, starting from root $\mathcal{E}_i^{(t)}$. We say that $\mathcal{I}$ occurs in the execution log if there is t such that $\mathcal{I}$ is a witness of the resampling $\mathcal{E}_i^{(t)}$.*

The crucial lemma in the Moser-Tardos analysis is the following.[1]

[1] In the Moser-Tardos paper, the role of the stable set sequences was played by "witness trees" but the concept is essentially the same; the stable sets I_ℓ can be obtained as layers in a witness tree.

Lemma 8.5 *For every stable set sequence $\mathcal{I} = \{I_0, I_1, ..., I_r\}$,*

$$\mathbb{P}[\mathcal{I} \text{ occurs in the log}] \leq \prod_{\ell=0}^{r} \prod_{i \in I_\ell} p_i$$

where $p_i = \mathbb{P}[\mathcal{E}_i]$.

Proof: We first modify the algorithm as follows (which does not change its behavior). We prepare an infinite table of samples to be used: For each X_a, the $a-$th column of the table contains an infinite sequence $X_a^1, X_a^2, X_a^3, \ldots$, each sampled independently according to the distribution of X_a. The algorithm maintains a pointer $\pi(a)$ for each $a \in [m]$. We start with $\pi(a) = 1$ for each $a \in [m]$. The "current values" of X_a are given by $X_a^{\pi(a)}$. Whenever the algorithm "resamples" X_a, we increment $\pi(a)$ by 1, which means moving on to the next sample. Clearly, this is equivalent to the original description of the algorithm.

We claim that if a certain stable set sequence $\mathcal{I}$ occurs in the execution log, then for each of its events we can determine a particular set of samples in the table that must satisfy the event. Given $\mathcal{I} = (I_0, I_1, \ldots)$, we obtain the locations of these samples as follows: For every $a \in var(\mathcal{E}_j)$ where $j \in I_\ell$, let $n_{a,\ell}$ denote the number of indices $\ell' \leq \ell$ such that $j' \in I_{\ell'}$ and $a \in var(\mathcal{E}_j)$. (Note that for each ℓ', at most one event in $I_{\ell'}$ can depend on X_a, since $I_{\ell'}$ is an independent set.) Then, we claim that $(X_a^{n(a,\ell)} : a \in var(\mathcal{E}_j))$ are exactly the samples of X_a that were checked by the algorithm to determine that $\mathcal{E}_j$ occurs, before the resampling that makes $\mathcal{E}_j$ a member of I_ℓ.

This is because the only times when the pointer $\pi(a)$ is incremented is when we resample an event depending on X_a. If $\mathcal{E}_j \in I_\ell$ and this is due to a resampling at time s, then any event resampled before time s that also depends on X_a will be part of the directed graph $R_j^{(s)}$ and hence also part of the stable set sequence. These are the only times when the pointer $\pi(a)$ is incremented prior to the resampling $\mathcal{E}_j^{(s)}$ and hence the value of $\pi(a)$ just before this resampling is exactly $n(a, \ell)$.

Now we know that in order for $\mathcal{I} = (I_0, I_1, \ldots, I_r)$ to occur, it must be the case that for each $0 \leq \ell \leq r$ and for each event $\mathcal{E}_j \in I_\ell$, the samples $(X_a^{n(a,\ell)} : a \in var(\mathcal{E}_j))$ satisfy the event $\mathcal{E}_j$. (Otherwise the event is not satisfied when the algorithm resamples it — but that is never the case.) This happens with probability $\mathbb{P}[\mathcal{E}_j]$. Also, the samples $X_a^{n(a,\ell)}$ for different values of k are distinct; this is directly from the definition of $n(a, \ell)$. By the independence of the samples in the table, the probability that for each $\mathcal{E}_j \in I_\ell$, $0 \leq \ell \leq r$, the samples $(X_a^{n(a,\ell)} : a \in var(\mathcal{E}_j))$ satisfy $\mathcal{E}_j$, is $\prod_{s=0}^{r} \prod_{j \in I_s} p_j$. □

We note that this is an upper bound on $\mathbb{P}[\mathcal{I}$ occurs in the log] rather than its exact value, because the presence of appropriate samples in the table does not guarantee that $\mathcal{I}$ will occur: The sequence of resamplings could unfold in different ways and $\mathcal{I}$ might not be a valid witness sequence. The presence of appropriate samples in the table is only a necessary condition for $\mathcal{I}$ to occur.

We will finish the analysis next time.

References

[Moser and Tardos (2010)] Robin Moser and Gabor Tardos. A constructive proof of the general Lovász Local Lemma. *Journal of the ACM* 57(2), 2010.

[Kolipaka and Szegedy (2011)] Kashyap Kolipaka and Mario Szegedy. Moser and Tardos meet Lovász. In *Proceedings of ACM-SIAM STOC*, 2011.