1. INTRODUCTION AND SETUP

This lecture, we will focus on a Moser-Tardos proof of Lovász Local Lemma. Recall the setup. Let $E_1, E_2, \ldots, E_n$ be random events. Let $P_1, P_2, \ldots, P_m$ be mutually independent random variables that event $E_i$ are determined by the values of $P_j$’s. For each $E_i$, let $vbl(E_i)$ be a set of $P_j$ that determine $E_i$.

Let $G$ be the dependency graph of events $\{E_i\}$. That is $G$ is a graph which $V = [n]$, and $(i, j)$ is an edge if $vbl(E_i) \cap vbl(E_j) \neq \emptyset$. We define $\Gamma_i$ by

$$\Gamma_i = \{ j \mid vbl(E_i) \cap vbl(E_j) \neq \emptyset \}$$

Also, we define $\Gamma_i^+ = \Gamma_i \cup \{i\}$. Finally, we assume that there are positive numbers $x_1, x_2, \ldots, x_n$ satisfying:

$$P(E_i) \leq x_i \prod_{j \in \Gamma_i} (1 - x_j)$$

We will define an algorithm that find values of $P_j$ that make all $E_i$ to be false.

2. ALGORITHM

1. Initialize random values for $P_1, \ldots, P_m$

2. If there are $E_i$ such that $E_i$ is true, pick such $E_i$ at random, and resample $P_j \in vbl(E_i)$, and recursively do (2). Otherwise, halt.

First, we prove a theorem:

**Theorem 1.** For $i \in \{1, 2, \ldots, n\}$, the expected number of time $E_i$ is resampled is $\leq \frac{x_i}{1 - x_i}$

The intuitive reason why the theorem above is true can be seen as follows: Consider a geometric distribution with probability of success $p$. The number of times we expect to run until we see the first unsuccess is equal to $\frac{1}{1 - p}$. However, the theorem above counts
only number of times resampled, so expected run time should be \( \frac{1}{1-p} - 1 = \frac{1}{1-p} \). Therefore, as in the usual version of the Lovasz Local Lemma, the values \( x_i \) can be viewed as acting like probabilities for the events \( E_i \) as if the events were independent.

### 3. Execution Logs and Witness Tree

Consider an execution of the algorithm. Define the log of the execution to be a sequence

\[ C = E_{i_1}E_{i_2} \ldots E_{i_t} \ldots \]

where we resampled \( E_{i_1} \) then \( E_{i_2} \) and so on.

A witness tree is a finite tree \( T \) such that each vertex \( u \) is labeled by some event \( E_{i_u} \) in such a way that, if \( v \) is a child of \( u \), then \( vbl(E_{i_u}) \cap vbl(E_{i_v}) \neq \emptyset \). The witness tree is proper if any two children of the same vertex have distinct labels.

Here, we will associate an execution of the algorithm to a witness tree. Given any execution log \( C = E_{i_1} \ldots E_{i_t} \ldots \), construct \( T^l_C \) inductively by

1. \( T^1_C(t) = \) a tree with one vertex, labeled by \( E_{i_t} \)
2. \( T^l_C(l-1) \) is defined by adding a child to a vertex \( u \) in \( T^l_C(l) \), such that \( u \) is a vertex with maximum depth (distance from the root) with the property that \( E_{i_u} \cap E_{i_{t-1}} \neq \emptyset \). If \( E_{i_{t-1}} \) is independent from all \( E_{i_v} \) in \( T \), then disregard \( E_{i_{t-1}} \), and let \( T^l_C(l-1) = T^l_C(l) \).

and set \( T^l_C = T_C(1) \). For given any log \( C \) of an execution, \( T^l_C \) has a following properties

**Claim 1.**

1. \( T^l_C \) is a proper witness tree
2. If \( u, v \) are siblings, then \( vbl(E_{i_u}) \cap vbl(E_{i_v}) = \emptyset \)
3. If \( u, v \) have the same depth, then \( vbl(E_{i_u}) \cap vbl(E_{i_v}) = \emptyset \)
4. If \( u \) has label \( E_{i_x} \), \( v \) has label \( E_{i_y} \), \( vbl(E_{i_x}) \cap vbl(E_{i_y}) \neq \emptyset \), and \( 1 \leq x < y \), then depth of \( u > \) depth of \( v \).

Here, we define another way to sampling each \( P_i \).

**Definition 1.** Fix a proper witness tree \( T \). A tree sampling of \( T \) is a sequence of sampling variables in \( E_{i_u} \) according to the depth of \( u \), from largest to smallest.

The next lemma states an equivalence between oszer-Tardós resampling and the evaluation of events in a tree sampling. Both algorithms run by repeatedly choosing an event \( E \) and choosing new random values for the variables \( vbl(E) \) which determine \( E \). For this, we assume that the algorithms obtain their randomness as follows: the algorithm is initialized with infinite sequences of (random) values for each random variable \( P_i \). That is, each
A random variable is given an infinite sequence of random values, and when the algorithm needs a new value for $P_i$, it uses the next value in the sequence of values for $P_i$.

The purpose of introducing tree sampling is that the randomness in either algorithm, the Moser-Tardós algorithm or tree sampling, provide an equivalent order of variable resampling for the random variables of each event. In the Moser-Tardós, we resampled first $E_{i_1}$ then $E_{i_2}$ and so on, while the order of $t$ treesampling is determined by depth of $u$. At first glance, it might not be obvious that the sequence of sampling is the same in Moser-Tardós and the tree sampling. For instance, there are no reasons why $E_{i_1}$ has the largest depth in $T$. However, two resampling sequences are “equivalent” in their effects on each event, since even if $E_{i_1}$ is not resampled first in the tree sampling, $E_{i_1}$ is independent from any $E_{i_u}$, where $u$ has a larger depth than the vertex of $E_{i_1}$. This is true by the property of a witness tree. Since they are independent, it does not really matter whether we resample $E_{i_1}$ first or not.

Tree sampling yields another way to look at the problem, as shown in Lemma 1.

**Lemma 1.** For $T$ a proper witness tree, $C$ the (random) log produced by the Moser-Tardós algorithm, the probability that $\exists t$ such that $T_C^t = T$ is $\leq \prod_{u \in T} Pr[E_{i_u}]$.

**Proof.** Assume that $T_C^t = T$. We claim that it is equivalent whether we resample variables using Moser-Tardós or tree sampling. Namely, each event $E_i$ which is chosen for the resampling in the Moser-Tardós algorithm, is evaluated using the same random values as is used for evaluating the event in the tree sampling algorithm. More precisely, the $j$-th time the event $E_i$ is chosen for resampling by the Moser-Tardós algorithm, the event $E_i$'s value has been computed using the same random values as are used by the tree sampling algorithm when it evaluates $E_i$ the $j$-th time.

This claim is proved by noting that the depth in the tree of event $E$ respects the orders in which random variables are used.

In the Moser-Tardós’s algorithm, each $E_{i_u}$ chosen for resampling has value, therefore, each time we resample variables in $E_{i_u}$ the tree sampling, it must evaluate as true, which happens with probability $Pr[E_{i_u}]$. Thus, the probability that there exists $t$ such that $T_C^t = T$ is at most the probability that all $E_{i_u}$ evaluated true, which is equal to $Pr[E_{i_u}]$.

**Definition 2.** $T_{E_i} = \{ T \mid T$ is a proper witness tree with a root labeled by $E_i \}$

**Lemma 2.** For any $E_i$

$$\sum_{T \in T_{E_i}} \prod_{u \in T} P[E_{i_u}] \leq \frac{x_i}{1 - x_i}$$

Then, we can conclude that the number of times $E_i$ resampled $\leq$ the expected number of $T \in T_{E_i}$ such that there exist $t$, $T_C^t = T$. Theorem 1 is an immediate consequence of Lemma 1 and 2.
To prove the Lemma 2, it suffices to prove that

$$\sum_{T \in T_{E_i}} \prod_{u \in T} \left( x_{i_u} \prod_{j \in \Gamma_{i_u}} (1 - x_j) \right) \leq \frac{x_i}{1 - x_i}$$

To simplify notation, we let $x'_i = x_i \prod_{j \in \Gamma_i} (1 - x_j)$. Thus, we want to show that

$$\sum_{T \in T_{E_i}} \prod_{u \in T} x'_{i_u} \leq \frac{x_i}{1 - x_i}$$

This will be shown in the next lecture.