13
Lovász Local Lemma

To apply the probabilistic method, a common problem is to bound the probability of bad events. Let $A_1, \dots, A_n$ be n bad events. If they are independent, we know that the probability of no bad events is $\prod \mathbf{Pr}\left(\overline{A_i}\right)$. Otherwise we have union bound. However, the union bound is too weak if bad events are not disjoint.

13.1 Local lemma and examples

Suppose we have a set of events $A_1, \dots, A_n$, each with probability p_i. If $\sum p_i < 1$, then by the union bound (or Markov's inequality), we know that $\mathbf{Pr}\left(\cap \overline{A}_i\right) > 0$ or even almost surely if $\sum p_i = o(1)$. If $\sum p_i = O(1)$ or even $\sum p_i \to \infty$, then we know nothing about $\mathbf{Pr}\left(\cap \overline{A}_i\right)$. Let X_i be the indicator of A_i. If $\mathbf{Var}\left[X\right] = o(\mathbb{E}\left[X\right]^2)$, then $\mathbf{Pr}\left(\cap \overline{A}_i\right) = \mathbf{Pr}(X = 0) = o(1)$. However, what do we need if we want to prove that $\mathbf{Pr}\left(\cap \overline{A}_i\right) > 0$?

In this section, we will introduce the celebrated *Lovász local lemma*. We start from the definition of dependency.

Definition 13.1 (Dependency graph). Suppose $A_1, \dots, A_n$ are n "bad events". For each A_i, let $N(i) \subseteq [n]$ be a set such that A_i is *independent* from all other events except those in $N(i)$, i.e., A_i is independent from $\{A_j \mid j \notin N(i) \cup \{i\}\}$.
Here we say an event A_0 is independent from $\{A_1, \dots, A_m\}$ if for any $B_i \in \{A_i, \overline{A}_i\}$, $\mathbf{Pr}(A_0 \mid B_1, B_2, \dots, B_m) = \mathbf{Pr}(A_0)$.
We usually represent the dependency relations by a (di)graph whose vertices are events, and $A_i \to A_j$ if and only if $j \in N(i)$.

Remark 13.2. Note that pairwise independence does not implies mutually independence. For the local lemma we need a stronger notion of independence. Consider $x_1, x_2, x_3 \in \{0, 1\}$ uniformly and A_i is the event that $\sum_{j \neq i} x_j = 0$. Then any two events are pairwise independent but are not independent if we consider the third event. Thus, the empty graph is not a valid dependency graph. But, any graph with at least two edges is a valid dependency graph.

Theorem 13.3 (Lovász Local Lemma, symmetric version)**.** *Let $A_1, \ldots, A_n$ be n events with probability $\mathbf{Pr}(A_i) \leq p$. Suppose that each A_i is independent from all other A_j except at most d of them. If $\mathrm{e}p(d+1) \leq 1$, then $\mathbf{Pr}(\cap \overline{A}_i) > 0$.*

Example 13.4. Consider the problem of k-SAT, a.k.a. k-CNF. A *formula* is in *conjunctive normal form* (CNF) if it is a *conjunction* of *clauses*, where each clause is a *disjunction* of *literals*, e.g., a formula of the form

$$\phi = (x_1 \vee \neg x_2 \vee \neg x_4) \wedge (x_2 \vee \neg x_3 \vee x_5) \wedge \cdots .$$

A formula is called k-SAT, or k-CNF if each clause consists of exactly k literals. Clearly, if each variable is chosen true or false independently and uniformly at random, the probability that a clause is not satisfied is 2^{-k}. However it is difficult to determine whether a formula is satisfiable since clauses are dependent.

Now we apply the Lovász local lemma. Construct a dependency graph on clauses, where two clauses are adjacent if they share a common variable. Thus we can show that if each variable appears in at most d clauses, where $d \leq 2^k/(\mathrm{e}k)$, then a k-CNF formula is satisfiable.

Example 13.5. Let's take another example of hypergraph coloring. Let $H = (V, E)$ be a hypergraph. A coloring c is proper if there doesn't exist a monochromatic edge. We can construct a dependency graph on edges such that for any two edges $e, f \in E$, $e \sim f$ if $e \cap f \neq \emptyset$. By the Lovász local lemma, if the hypergraph is k-uniform, the maximum vertex degree is at most Δ, and $\mathrm{e}k\Delta q^{1-k} \leq 1$, then H is q-colorable.

We now use the Lovász local lemma to prove more existence problems.

Theorem 13.6 (Independent transversal)**.** *Let $G = (V, E)$ be a graph with maximal degree at most Δ. $V = V_1 \cup \ldots \cup V_r$ is a partition where $|V_i| \geq 2\mathrm{e}\Delta$ for any $1 \leq i \leq r$. Then, there exists an independent set which contains a vertex from each V_i.*

Proof. Let $k = \lceil 2\mathrm{e}\Delta \rceil$ and assume that $|V_i| = k$ for all $1 \leq i \leq r$. Pick $v_i \in V_i$ u.a.r. For any edge $e \in E$, let B_e be the event that both of its endpoints are chosen. Thus, $\mathbf{Pr}(B_e)\, leq \frac{1}{k^2}$. In the dependency graph, $B_e \sim B_f$ if there exists V_i that intersects both e and f. Therefore, the maximal degree of the dependency graph $d \leq 2k\Delta - 2$. Then, the Lovász Local Lemma applies. □

Remark 13.7. Some choices of bad events are better than others. If we define $A_{i,j} = \{v_i \sim v_j\}$ for any $1 \leq i < j \leq r$, then $\mathbf{Pr}(A_{i,j}) \leq \frac{\Delta}{k}$. In the dependency graph, $A_{i,j} \sim A_{k,l}$ if $\{i,j\} \cap \{k,l\} \neq \emptyset$. The maximal degree of the dependency graph is $d \leq 2k\Delta - 1$. However, this upper bound is still too large.

Theorem 13.8 (Alon & Linial, 1989). *For any directed graph G with minimal out-degree at least δ and maximal in-degree at most Δ contains a cycle of length divisible by k when*

$$k \leq \frac{\delta}{1 + \log(1 + \delta\Delta)}.$$

Proof. Assume that every vertex $v \in V$ has out-degree δ. (Otherwise, we delete some edges from v.) Assign $x_v \in \mathbb{Z}/k\mathbb{Z}$ to v uniformly randomly. Now, we look for cycles that the label increase by 1 at each step.

Let $A_v = \{\text{none out-neighbor of } v \text{ has label } x_v + 1\}$. Thus,

$$\mathbf{Pr}(A_v) = (1 - 1/k)^{\delta} \leq e^{-\delta/k}.$$

Let $N^{out}(v)$ be the set of out-neighbors of vertex v. Naively we may use the dependency graph where $A_u \sim A_v$ if and only if $\{u\} \cup N^{out}(u)$ intersects $\{v\} \cup N^{out}(v)$.

In fact we can construct a directed dependency graph and improve the bound. Note that $\mathbf{Pr}(A_v)$ is $(1 - 1/k)^{\delta}$ as long as $N^{out}(v)$ are free, even if v is assigned. So A_v is independent from all A_u's where $N^{out}(v)$ does not intersect $\{u\} \cup N^{out}(u)$. Therefore, the maximal degree of the dependency graph $d \leq \Delta\delta$. As

$$e^{1-\delta/k}(1 + \Delta\delta) \leq 1,$$

we are done by the Lovász Local Lemma. □

Remark 13.9. The dependency is not symmetric in this proof.

13.2 *Asymmetric version of local lemma*

In many cases, the probabilities of bad events are not necessary to use a same upper bound. Thus we introduce the following *asymmetric* *local lemma.*

Theorem 13.10 (Lovász local lemma, asymmetric version). *Let $A_1, \ldots, A_n$ be events and A_i is independent from $\{A_j \mid j \notin N(i) \cup \{i\}\}$. If there exists $x_1, \ldots, x_n \in [0,1)$ such that for any $1 \le i \le n$,*

$$\mathbf{Pr}(A_i) \le x_i \prod_{j \in N(i)} (1 - x_j),$$

then

$$\mathbf{Pr}\big(\cap \overline{A}_i\big) \ge \prod_{i=1}^{n} (1 - x_i).$$

To see the symmetric form, set $x_i = \frac{1}{d+1} < 1$ for all $1 \le i \le n$. Then,

$$x_i \prod_{j \in N(i)} (1 - x_j) \ge \frac{1}{d+1}\left(1 - \frac{1}{d+1}\right)^d > \frac{1}{\mathrm{e}(d+1)} \ge p.$$

Proof. We claim that for any $i \notin S \subseteq [n]$, we have

$$\mathbf{Pr}\left(A_i \mid \bigcap_{j \in S} \overline{A}_j\right) \le x_i.$$

If it holds, then

$$\mathbf{Pr}\Big(\bigcap \overline{A}_i\Big) = \mathbf{Pr}\big(\overline{A}_i\big)\ \mathbf{Pr}\big(\overline{A}_2 \mid \overline{A}_1\big)\ \ldots \ge \prod_{i=1}^{n} (1 - x_i),$$

which completes the proof.

Now, let's prove our claim by induction on the size of S. Our claim is trivially true when $|S| = 0$.

We assume that for any set S' of which size is less than S, the claim always holds. Let's consider the set S. For $i \notin S$, let $S_1 = S \bigcap N(i)$ and $S_2 = S \setminus S_1$. Then we have

$$\mathbf{Pr}\left(A_i \mid \bigcap_{j \in S} \overline{A}_j\right) = \frac{\mathbf{Pr}\Big(A_i \bigcap (\bigcap_{j \in S_1} \overline{A}_j) \mid \bigcap_{j \in S_2} \overline{A}_j\Big)}{\mathbf{Pr}\Big(\bigcap_{j \in S_1} \overline{A}_j \mid \bigcap_{j \in S_2} \overline{A}_j\Big)} := \frac{\alpha}{\beta}.$$

Note that

$$\alpha \le \mathbf{Pr}\left(A_i \mid \bigcap_{j \in S_2} \overline{A}_j\right) = \mathbf{Pr}(A_i) \le x_i \cdot \prod_{j \in N(i)} (1 - x_j).$$

Also, let $S_1 = \{t_1, \dots, t_r\}$. We have

$$\begin{aligned}\beta &= \prod_{k=1}^{r} \mathbf{Pr}\left(\overline{A}_{t_k} \mid (\bigcap_{\ell=1}^{k-1} \overline{A}_{t_\ell}) \bigcap (\bigcap_{j \in S_2} \overline{A}_j) \right) \\ &\geq (1 - x_{t_1}) \dots (1 - x_{t_r}) \qquad \text{(by induction hypothesis)} \\ &\geq \prod_{j \in N(i)} (1 - x_j).\end{aligned}$$

Therefore, $\frac{\alpha}{\beta} \leq x_i$, which completes the proof. □

Remark 13.11. In 1985, Shearer proved that the constant e is best possible.

As an example, we now use asymmetric Lovász local lemma to bound Ramsey numbers.

Theorem 13.12 (Spencer, 1977). *If*

$$\mathrm{e}\left(\binom{k}{2}\binom{n}{k-2} + 1\right) 2^{1-\binom{k}{2}} < 1,$$

then $R(k,k) > n$.

Proof. Color K_n randomly. For any set of vertices S of size k, let E_S be the event that S induces a monochromatic K_k. Thus, $\mathbf{Pr}(E_S) = 2^{1-\binom{k}{2}}$.

For any k-vertex sets S, E_S is independent from all E_T where $|S \cap T| < 2$. Therefore, the maximal degree of the dependency graph is at most $\binom{k}{2}\binom{n}{k-2}$. Then, the Lovász Local Lemma applies. □

Remark 13.13. Optimizing the choice of n, it gives the best bound so far

$$R(k,k) > (\sqrt{2}/\mathrm{e} + o(1))\, k\, 2^{k/2}.$$

Recall that by the union bound we obtain $R(k,k) > (1/(\mathrm{e}\sqrt{2}) + o(1))\, k\, 2^{k/2}$, and by the alteration method we obtain $R(k,k) > (1/\mathrm{e} + o(1))\, k\, 2^{k/2}$. The Lovász Local Lemma does not improve much.

Let $K = \binom{n}{k}$ be the number of all events, then $d = |N(S)| \approx K^{1-O(1/k)}$. There are so many "dependencies", so the Lovász local lemma does not work well for diagonal Ramsey numbers. However, on the other hand, it performs well in asymmetric cases.

Let's first consider $R(k,3)$. Let p be a fixed parameter to be determined later. For each vertex, color it 0 with probability p, and 1 with probability $1-p$. Let S, T be two vertex sets where $|S| = 3$ and $|T| = k$. Define A_S as the event that S forms a monochromatic K_3

with color 0 and B_T as the event that T forms a monochromatic K_k with color 1. Clearly,

$$\mathbf{Pr}(A_S) = p^3,\ \mathbf{Pr}(B_T) = (1-p)^{\binom{k}{2}},$$

and two event are adjacent in the dependency graph if the intersection of their corresponding subsets has size at least 2.

For A_S, there exists at most $3(n-3)$ S' such that $A_S \sim A_{S'}$ and at most $\binom{n}{k}$ T' such that $A_S \sim B_{T'}$. For B_T, there exists at most $\binom{k}{2}(n-2) < \frac{k^2 n}{2}$ S' such that $B_T \sim A_{S'}$ and at most $\binom{n}{k}$ T' such that $B_T \sim B_{T'}$.

Apply the Lovász local lemma, if there exists p, x, y such that

$$\begin{cases} p^3 \le x(1-x)^{3n}(1-y)^{\binom{n}{k}} \\ (1-p)^{\binom{k}{2}} \le y(1-x)^{k^2 n/2}(1-y)^{\binom{n}{k}} \end{cases},$$

then $R(k,3) > n$. By setting $p = c_1 \cdot n^{-1/2}$, $k = c_2 \cdot n^{1/2} \log n$, $x = c_3 \cdot n^{-3/2}$ and $y = c_4 / \binom{n}{k}$, we have $R(k,3) > c_5 \cdot k^2 / \log^2 k$. The best known lower bound is $c_6 \cdot k^2 / \log k$.

Analogously, we can show $R(k,4) > k^{\frac{5}{2}+o(1)}$ by the asymmetric Lovász local lemma, which is better than any known result without the Lovász Local Lemma.

13.3 *Algorithmic local lemma*

In many problems, such as k-SAT or hypergraph colorings, the Lovász local lemma only tells us the existence of desired assignments. Can we find such a satisfying assignment in polynomial time?

Let's start from a computationally hard example. Let $q = 2^k$ and $f : [q] \to [q]$ be a bijection. Let $y \in [q]$ be a fixed element. We sample $x \in [q]$ uniformly at random. Define A_i as the bad event that $f(x)$ and y disagree at the i-th bit. All A_i's are mutually independent, so the local lemma applies. This means that there exists x such that $f(x) = y$. However, this conclusion is meaningless as we have already known that f is a bijection. Also, finding such an x may be extremely hard. For instance, consider the problem of *discrete logarithm* where $f(x) : \mathbb{F}_q \to \mathbb{F}_q = g^x$.

The example above shows that it's sometimes hard for us to find an assignment such that no "bad events" occur if we add no constraints to events. For simplicity, we only talk about random variable models, where each event only depends on some variables.

In 2010, Robin Moser and Gábor Tardos gave a *Las Vegas algorithm* to find a satisfying assignment in expected linear time, as long as the condition of the local lemma is satisfied. In this section, we will introduce a simple and elegant proof for a special case of algorithmic local lemma, due to Robin Moser in 2009.

They won Gödel Prize in 2020 based on this work.

Consider a k-SAT formula:

$$\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$$

of which each clause has exactly k literals. Robin Moser gave a fix-it algorithm to find a valid assignment as follows:

Algorithm 1: Moser's *fix-it* algorithm for *k-SAT* problem

Input: A k-SAT formula: $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_m$.

```
Function Solve(φ):
    randomly initialize v_1, ..., v_n
    while there exists unsatisfied clause C_i do
        Fix(C_i)

Function Fix(C_i):
    Resample the variables in C_i uniformly at random
    while there exists unsatisfied clause C_j overlapping with C_i do
        Fix(C_j)
```

Robin Moser proved that when each caluse does not intersect with too many other clauses, `Solve(`φ`)` can find a satisfying assignment in polynomial time with high probability. Precisely, the theorem is as follows:

This proof is actually based on Moser's talk in STOC 2009. Moser won the best paper award in STOC 2009 based on the constructive proof of local lemma, but the argument in his paper is quite a bit more complicated. Moser presented a version of the proof below in his conference talk, and his ideas were popularized by Fortnow and Tao. (Fortnow called Moser's talk "one of the best STOC talks ever".) Tao introduced the phase entropy compression argument to describe Moser's influential idea.

Theorem 13.14 (Robin Moser). *Let d be the maximum degree of clauses, i.e., each cluase intersects with at most d clauses (including itself). Then,* `Solve(`φ`)` *finds a satisfying assignment in polynomial time with high probability as long as $d \le 2^{k-3}$.*

Proof. Consider the recursion tree.

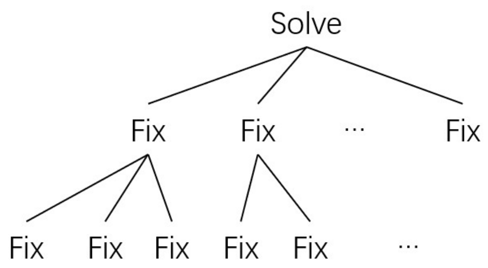

Figure 13.1: The graph shows an example of the recursion tree.

Suppose there are T times of `Fix` calls before terminating. Clearly, `Solve(`φ`)` used $n + kT$ random bits in total. We now argue that if T is sufficiently large, the number of 01 bits used by the recursion tree is smaller than kT.

A key observation is that for each sub-tree rooted at some `Fix(`C`)`, all satisfied clauses before `Fix(`C`)` cannot become unsatisfied after all `Fix` calls in the sub-tree have been executed. Thus we can see that all clauses in the first level are distinct. Denote by m $0/1$ bits whether each clause is fixed at the first level.

For any other node in the recursion tree, it is clear that each `Fix` call has at most d children. So we only need to record it by its index in the children of its parent node. Denote by $\lceil \log_2 d \rceil$ bits.

To record the structure of the recursion tree, consider its DFS sequence. For each node, use "1" to denote that the node is pushed in stack, and use "0" to record the outing. Overall, the number of bits we need is at most $m + (\lceil \log_2 d \rceil + 2)T \leq m + (k-1)T$. Finally, we use another n bits to record the final assignment. Note that every random bit used in Moser's algorithm is determined uniquely by the final assignment and the recursion tree, since for each `Fix(`C`)` call, the assignment of variables in C is determined before `Fix(`C`)`. Therefore, we use $m + (k-1)T + n$ bits to uniquely encode a sequence of $n + kT$ random bits.

Now, we need the following *incompressibility theorem*.

Claim 13.15 (*Incompressibility theorem*). N uniform random bits cannot be encoded to no more than $N - \ell$ bits with probability at least $1 - O(2^{-\ell})$.

In Moser's proof, $n + kT$ random bits are encoded to $n + m + (k - 1)T$ bits. However, T is not a fixed integer, which implies that we cannot use the incompressibility theorem directly. We need to find some other methods.

Let's fix $t = m + \log_2 n$, and we provide only $n + kt$ random bits in total. The algorithm will be forced to terminate if all random bits have been used up. If the algorithm succeeds after T steps of `Fix` calls, $n + kt$ random bits are decoded into $n + m + (k-1)T + k(t - T)$ bits. Otherwise when the algorithm fails to find a satisfying assignment, then $n + kt$ random bits are encoded into at most $n + m + (k-1)t$ bits. According the incompressibility theorem, the probability that the algorithm fails is at most

$$2^{(n+m+(k-1)t)-(n+kt)} = 2^{-\log_2 n} = 1/n,$$

which completes the proof. □

The general algorithm was given by Robin Moser and Gábor Tardos.

Algorithm 2: Moser–Tardos algorithm

Input: n bad events $A_1, \ldots, A_n$ in the variable model.

Function `Solve`:
- Initialize each variable a random value independently.
- **while** *some bad event A_i occurs* **do**
  - re-sample all variables that A_i depends on

Theorem 13.16 (Robin Moser & Gábor Tardos, 2010)**.** *If the condition of Lovász Local Lemma holds, then Moser-Tardos algorithm returns an assignment that no bad event occurs in expected linear time. In particular, the expected rounds of re-sampling is no more than*

$$E = \sum_{i=1}^{n} \frac{x_i}{1 - x_i}.$$

Proof. Let the execution log L be the sequence of A_i's that are picked in the `while` loop. $|L|$ may be infinite, but we claim that $\mathbb{E}[|L|] \le E$.

Construct witness trees as follows for each time $t \le |L|$. Let $L = (A_{\ell_1}, A_{\ell_2}, \ldots, A_{\ell_t}, \ldots)$. Read prefix right to left: $A_{\ell_t}, \ldots, A_{\ell_1}$.

- Let the root of the witness tree $T(t)$ be a vertex labelled with A_{ℓ_t} on it.
- For $s = t - 1, \ldots, 1$:
  - If none of the events corresponding to vertices in T shares variables with A_{ℓ_s}, continue.
  - Otherwise, find a deepest node v such that the event $A_{[v]}$ shares common variables with A_{ℓ_s}, namely, $vbl(A_{[v]}) \cap vbl(A_{\ell_s}) \neq \emptyset$, and then add a node labelled with A_{ℓ_s} as v's child.

The margin picture demonstrates a valid witness tree as an example.

Now, consider properties of the witness trees with node labels. For convenience, denote by $[v]$ the label (the index of event) on node v.

- $T(t_1) \neq T(t_2)$ for different times $t_1 \neq t_2$.

  If $A_{\ell_{t_1}} \neq A_{\ell_{t_2}}$, then the roots of $T(t_1)$ and $T(t_2)$ have different labels. If $A_{\ell_{t_1}} = A_{\ell_{t_2}} = A_r$, then A_r appears different times in the node labels of $T(t_1)$ and $T(t_2)$, which implies that $T(t_1) \neq T(t_2)$.

- For any $T = T(t)$ and $u, v \in T$ of the same depth, $vbl(A_{[u]}) \cap vbl(A_{[v]}) = \emptyset$.

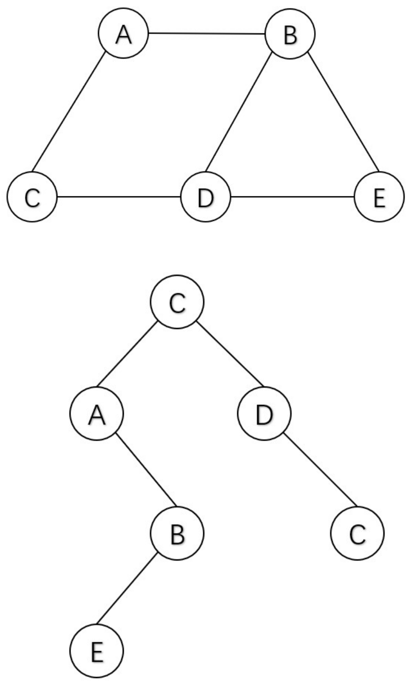

Figure 13.2: The first picture is the dependency graph of events, while the second one is a valid witness tree when $L = (C, E, B, D, A, B, B, E, C)$.

The first property implies that

$$\mathbb{E}[|L|] = \sum_T \mathbb{E}[X_T] = \sum_T \mathbf{Pr}(T \text{ is a witness tree}).$$

We claim that

$$\mathbf{Pr}(T \text{ appears as a witness tree for some time } t) \le \prod_{v \in T} \mathbf{Pr}\Big(A_{[v]}\Big) .$$

In order to illustrate the above inequality more clearly, we give two simple examples. Consider T is a tree with one single vertex labeled with A as its root. If T is a valid witness tree for some time t, then event A is picked at time t, and no events picked before time t share a common variable with A, so A occurs at beginning, which implies that

$$\mathbf{Pr}(T \text{ appears as a witness tree for some time } t) \le \mathbf{Pr}(A).$$

If T is a tree with two vertices where A is its root while B is a child of A, then clearly B happens at the beginning. After re-sampling $vbl(B)$, event A occurs. Therefore, the probability that T is a valid witness tree is no larger than $\mathbf{Pr}(B)$ $\mathbf{Pr}(A)$.

Now, we start to prove our claim strictly. Let a simulation of T to be the following process: visit all nodes of T in the reverse BFS order. At each node v, resample all the variables that $A_{[v]}$ depends on, and then check if the event occurs. The simulation succeeds if all bad events encountered do occur. Assume for each variable, we have a list of (infinite many) $0/1$ values, of which each is independently sampled and then fixed. When simulating the Moser–Tardos algorithm and simulating a witness tree, we look up the (same) value table of each variable instead of sampling.

Here we actually construct a coupling between two processes.

For each $v \in T$ and any $u \in T$ with $vbl(A_{[u]}) \cap vbl(A_{[v]}) \neq \emptyset$, u is deeper than v if and only if $A_{[u]}$ appears before $A_{[v]}$ in the execution log. For any $z \in vbl(A_{[v]})$, let $n_{z,v}$ be the number of u's before v such that $z \in vbl(A_{[u]})$. In the simulation of the Moser–Tardos algorithm, there is an initialization of all variables, so the first value of each variable in the list is looked up at the begining. Then, during execution, for each node v of the witness tree, we know that $A_{[v]}$ occurs where each variable z of $A_{[v]}$ is assigned with the $(n_{z,v}+1)$-th value. For the simulation of a witness tree T, we also look up the $(n_{z,v}+1)$-th value of variable z at the time checking $A_{[v]}$. Since we look up the same value table, if T appears as a witness tree at some time t, the simulation of T must succeed. Thus, we have

$$\mathbf{Pr}(T \text{ is a witness tree}) \le \mathbf{Pr}(\text{simulation of } T \text{ succeeds}) = \prod_{v \in T} \mathbf{Pr}\Big(A_{[v]}\Big) ,$$

which proves our claim.

Let W be the set of all possible witness trees.

$$\mathbb{E}[|L|] = \sum_{T \in W} \mathbf{Pr}(T = T(t) \text{ for some } t) \le \sum_{T \in W} \prod_{v \in T} \mathbf{Pr}\Big(A_{[v]}\Big)$$

If $T \in W$, then T has the following properties:

- T is finite;
- For any $u \to v$ in T, variables in $A_{[u]}$ and $A_{[v]}$ are overlapping;
- For any $u, v \in T$ have the same depth, $A_{[u]}$ and $A_{[v]}$ are disjoint.

Let W' be the set of trees that only satisfy the second property, and W'_k be the set of trees in W' and rooted at event A_k. We generate trees in W'_k by a random process (Galton-Watson process):

- Let the label of the root be k.
- For any vertex v, we find all its "potential" children $N^+(v) = N([v]) \cup \{[v]\}$ whose variables overlap with $vbl(A_{[v]})$.
- For each "potential" child A_i, add a vertex labelled with i as the child of vertex v in the tree with probability x_i (x_i is the value corresponding to event A_i in the statement of the local lemma) and call it an alive children of v. Denote by $D(v)$ the set of alive children of v.

Let P_T be the probability that Galton-Watson process generates T. Thus, we have

$$\begin{aligned}
P_T &= \frac{1}{x_k} \prod_{v \in T} x_{[v]} \prod_{v \in T} \prod_{i \in N^+(v) \setminus D(v)} (1 - x_i) \\
&= \frac{1 - x_k}{x_k} \prod_{v \in T} \frac{x_{[v]}}{1 - x_{[v]}} \prod_{i \in N^+(v)} (1 - x_i) \\
&= \frac{1 - x_k}{x_k} \prod_{v \in T} x_{[v]} \prod_{i \in N(v)} (1 - x_i) \\
&\ge \frac{1 - x_k}{x_k} \prod_{v \in T} \mathbf{Pr}\Big(A_{[v]}\Big) .
\end{aligned}$$

Clearly, $\sum_{T \in W'_k} P_T \le 1$. Therefore,

$$\sum_{T \in W_k} \prod_{v \in T} \mathbf{Pr}\Big(A_{[v]}\Big) \le \sum_{T \in W'_k} P_T \cdot \frac{x_k}{1 - x_k} \le \frac{x_k}{1 - x_k},$$

which implies that

$$\mathbb{E}[|L|] \le \sum_{k=1}^{n} \frac{x_k}{1 - x_k} .$$

This completes the whole proof. □