Lovász Local Lemma

Let $E_1, \ldots, E_n$ is set of "bad" events.

say want to show $\Pr\left(\bigcap_{i=1}^{n} \overline{E_i}\right) > 0$ "pos prob that nothing bad happens"

2 cases where easy

1. E_i are mutually indep
2. $\sum_{i=1}^{n} \Pr(E_i) < 1$ union bound suffices

LLL is clever combination

Defn: E mutually indep of $E_1, \ldots, E_n$ if for any subset $I \subseteq [1..n]$

$$\Pr\left(E \,\middle|\, \bigcap_{j \in I} E_j\right) = \Pr(E)$$

Defn: A dependency graph for $E_1, \ldots, E_n$ is $G = (V, E)$

$V = \{1, \ldots, n\}$ and E_i mutually indep of $\{E_j \mid (i,j) \notin E\}$

Thm Lóvasz Local Lemma

Let $E_1, \ldots, E_n$ be set of events s.t.

1. $\Pr(E_i) < p \quad \forall i$
2. the max degree in dependency graph is d
3. $4dp \le 1$

Then $\Pr\left(\bigcap_{i=1}^{n} \bar{E}_i\right) > 0$

Proof Show $\Pr\left(\bigcap_{i\in S} \bar{E}_i\right) > 0$ and $\Pr\left(E_k \,\middle|\, \bigcap_{i\in S} \bar{E}_i\right) \le 2p \quad \forall k$

by induction on $|S|$

Base case: $|S|=1 \quad \Pr(\bar{E}_i) = 1 - \Pr(E_i) = 1-p > 0$

Case 1: no edge $k - i$

$\Pr(E_k | \bar{E}_i) = \Pr(E_k) \le p$

Case 2: $\exists$ edge $(k-i)$

$\Pr(E_k | \bar{E}_i) = \dfrac{\Pr(E_k \cap \bar{E}_i)}{\Pr(\bar{E}_i)} \le \dfrac{p}{1-p} < 2p$

$(p < \tfrac{3}{4})$

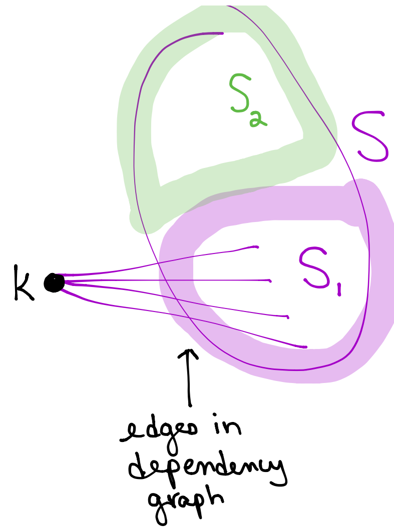
Induction Step: IH: $|T| < s \quad \Pr\left(\bigcap_{i\in T} \bar{E}_i\right) > 0 \quad \Pr\left(E_k \,\middle|\, \bigcap_{i\in T} \bar{E}_i\right) \le 2p$

wlog $S = \{1, \ldots, s\}$

$$\Pr\left(\bigcap_{i=1}^{s} \bar{E}_i\right) = \Pr(\bar{E}_1)\Pr(\bar{E}_2|\bar{E}_1)\Pr(\bar{E}_3|\bar{E}_1\cap\bar{E}_2)\cdots\Pr(\bar{E}_s|\bar{E}_1\cap\bar{E}_2\cap\ldots\cap\bar{E}_{s-1})$$

$$= \prod_{i=1}^{s}\left(1 - \Pr\left(E_i \,\middle|\, \bigcap_{j=1}^{i-1} \bar{E}_j\right)\right) \ge \prod_{i=1}^{s}(1-2p) > 0$$

Next, prove $\Pr\left(E_k \mid \bigcap_{i \in S} \overline{E_i}\right) \leq 2p$



If $S_2 = S$,

$$\Pr\left(E_k \mid \bigcap_{i \in S} \overline{E_i}\right) = \Pr(E_k) \leq p$$

So assume $S_2 < S$

$$F_S = \bigcap_{i \in S} \overline{E_i} \qquad F_{S_1} = \bigcap_{i \in S_1} \overline{E_i} \qquad F_{S_2} = \bigcap_{i \in S_2} \overline{E_i}$$

$$\Pr(E_k \mid F_S) = \frac{\Pr(E_k \cap F_S)}{\Pr(F_S)} = \frac{\Pr(E_k \cap F_{S_1} \mid F_{S_2})\cancel{\Pr(F_{S_2})}}{\Pr(F_{S_1} \mid F_{S_2})\cancel{\Pr(F_{S_2})}} = \frac{\Pr(E_k \cap F_{S_1} \mid F_{S_2})}{\Pr(F_{S_1} \mid F_{S_2})}$$

$$\Pr(E_k \cap F_{S_1} \mid F_{S_2}) \leq \Pr(E_k \mid F_{S_2}) = \Pr(E_k) \leq p$$

$$= 1 - \Pr\left(\bigcup_{i \in S_1} E_i \mid F_{S_2}\right)$$

$$\geq 1 - \sum_{i \in S_1} \Pr(E_i \mid F_{S_2}) \qquad \text{union bound}$$

$$\geq 1 - \sum_{i \in S_1} 2p \qquad \text{induction hypothesis, } |S_2| < |S|$$

$$\geq 1 - 2dp \geq \frac{1}{2}$$

$\Pr(E_k | E_s) \le \frac{p}{1/2} = 2p$

Application 1: k-SAT

Given k-SAT formula n vars, m clauses

Thm If no var appears in $> T = \frac{2^k}{4k}$ clauses, then formula has satisfying assignment

Proof Consider random assignment where each var is T with prob $\frac{1}{2}$ independently.

E_i event that clause i not satisfied

- $p = \Pr(E_i) = 2^{-k}$
- E_i mutually indep of clauses that don't share vars

$\Rightarrow d \le kT = 2^{k-2} \quad \Rightarrow 4dp \le 1 \quad \underset{LLL}{\Rightarrow} \Pr\left(\bigcap_{i=1}^{m} \overline{E_i}\right) > 0$ satisfying assignment

Efficient Algs for finding outcome guaranteed by LLL

..... [Moser] [Moser, Tardos]

$\boxed{\text{Thm}}$ S set of m length k clauses, s.t. support of each clause intersects $\leq 2^{k-c}$ clauses [c is a sufficiently large constant]

Then the clauses in S simultaneously satisfiable

Proof: $\boxed{\text{constructive version}}$

Fix ordering $C_1, \ldots, C_m$ of clauses

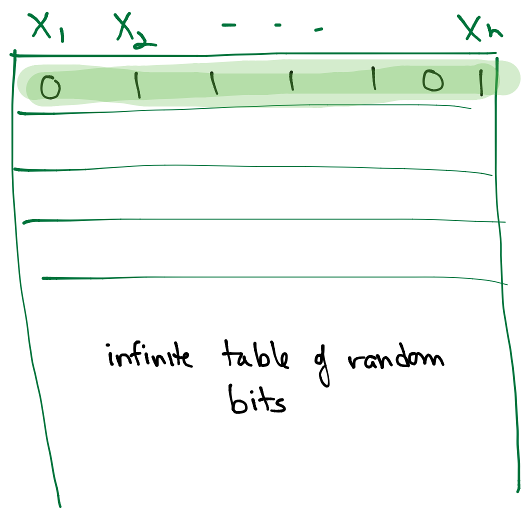
$\boxed{\text{Solve SAT}}$

Choose random assignment of vars $x_1, \ldots, x_n$

while $\exists$ clause not satisfied, pick arbitrary clause that is not satisfied, and assign new random values to the vars init

Claim: This terminates in poly time w.h.p.

whenever random bit needed, say for x_i, go to x_i column & take next "unused" bit.



Consider sequence of clauses alg resamples

$C_1, C_2, \ldots, C_t$

Describe part of execution (source of random bits) relevant to C_t using tree whose nodes are clauses

built inductively $i = t \ldots 1$

T_t^t single node: C_t

T_t^i derived from T_t^{i+1}

if $C_i \cap C_j = \phi \ \forall C_j \in T_t^{i+1}$, then $T_t^i = T_t^{i+1}$

o.w. C_i appended to deepest clause C_j that shares vars with C_i

Claim: T_t' uniquely determines which locations in R the random values for vars of each clause are taken from $\forall$ clauses in tree

Pf: Postorder traversal

Observation: If 2 clauses C_i & C_j $i<j$ in tree are at same depth in tree, then they are disjoint

consider which came first

Tree is **feasible** if all clauses in tree are not satisfied by the corresponding values in R

$$\boxed{\Pr(\text{tree } T_t' \text{ with } q \text{ clauses is feasible}) = p^q}$$

$$\boxed{p = 2^{-k}}$$

(tree being feasible doesn't mean actually generated)

If alg runs for qm steps, $\Rightarrow$ R contains feasible tree of size $\geq q$

qm clauses $\Rightarrow$ some clause repeated $\geq q$ times

all occurences in same tree

$\Rightarrow \Pr(\text{alg runs for} \geq qm \text{ steps}) \leq \Pr(R \text{ contains feasible tree of size} \geq q)$

Claim: # legally labelled trees of size $q \leq m \binom{dq}{q-1}$

- nodes labelled w/ clauses
- adjacent nodes overlap
- nodes at same level don't intersect

Proof of claim: Fix ordering of clauses

- m choices for root
- for each clause chosen, append vector of length d of 0's

  $d =$ max # $\cap$ clauses (2^{k-c})
- scan coordinates one-by-one, setting entries corresponding to "children" clauses (and append d coordinates for each)

any legal tree can be represented this way

& any vector corresponds to at most one legal tree

$\Rightarrow$ # legal trees with q nodes $\leq m\binom{dq}{q-1} \leq m(de)^q$

Stirling's Formula

$\Rightarrow \Pr_R(\exists$ feasible legal tree of size Q or more$)$

$$\leq m \sum_{q=Q}^{\infty} (de)^q p^q \leq m \frac{(dpe)^Q}{1-dpe}$$

$d = 2^{k-c}$ $\quad p = 2^{-k}$

$= o(1)$ for $Q = \varepsilon \log m$

$\Rightarrow \Pr(\#$ clause resamplings $\geq \varepsilon m \log m) = o(1)$

Can be viewed as "entropy compression" argument.