

Lovász Local Lemma
and the
resampling algorithm

Lecture 1

Probability

Events, probabilities,
conditional probabilities,
union bound

Random variables,
expected value,
linearity of expectation
Markov's inequality

Algorithms

Approx. 3-SAT,

Big idea: we only want the average

$$\begin{array}{l} \text{Linearity of Expectation} \\ E[X+Y] = E[X] + E[Y] \quad (\text{stochastic variable}) \\ \text{"average sum = sum of averages"} \end{array}$$

let $Y_i = \begin{cases} 1 & \text{if } i\text{th clause satisfied} \\ 0 & \text{otherwise} \end{cases}$ ($i=1, \dots, m$)

$$E[Y_i] = 7/8 \quad (1 \text{ clause case})$$

$$E[\text{\# clauses satisfied}] = E\left[\sum_{i=1}^m Y_i\right] = \sum_{i=1}^m E[Y_i] = 7/8 m$$

let $Y_i = \begin{cases} 1 & \text{if } i\text{th clause satisfied} \\ 0 & \text{otherwise} \end{cases}$

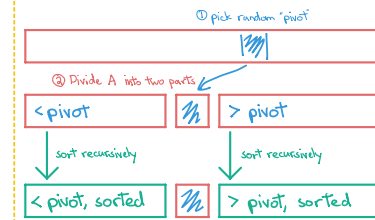
$$Z = \sum_{i=1}^m Y_i = \text{\# clauses satisfied}$$

$$E[Z] = \frac{1}{2} E[Z | x_1 = \text{true}] + \frac{1}{2} E[Z | x_1 = \text{false}]$$

$\Rightarrow$ either $E[Z | x_1 = \text{true}]$ or $E[Z | x_1 = \text{false}]$ is $\geq E[Z]$!

QuickSort,

Random divide and conquer



Expected running time of QuickSort

For each $i, j \in [n]$ w/ $i < j$,
 $X_{ij} = \begin{cases} 1 & \text{if } i\text{th smallest element is compared to } j\text{th smallest} \\ 0 & \text{otherwise} \end{cases}$

$\sum_{i,j} X_{ij}$ = total # comparisons made by the algorithm
 $\approx$ total running time

$E[\sum_{i,j} X_{ij}]$ = average running time

$$E[\sum_{i,j} X_{ij}] = \sum_{i,j} E[X_{ij}]$$

now analyze $E[X_{ij}]$ in isolation

$$E[X_{ij}] = \Pr[X_{ij} = 1] = \Pr[i\text{th element is compared to } j\text{th}]$$

i th and j th elem selected before any element in between happens w/ probability $\frac{2}{j-i+1}$

$$E[X_{ij}] = \frac{2}{j-i+1}$$

by linearity of expectation

$$E[\sum_{i,j} X_{ij}] = \sum_{i,j} E[X_{ij}] = \sum_{i,j} \frac{2}{j-i+1}$$

$$\leq 2n \left(\frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} \right)$$

$$\leq O(n \log n) \quad \text{with harmonic series}$$

Quickselect (HW)

Lecture 2

Probability

Hash Functions,

Universal Hash Functions

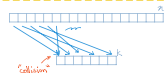
$h: [n] \rightarrow k$ s.t. for all pairs $i, j \in [n]$,
 $\Pr[h(i) = h(j)] \leq \frac{1}{k}$ (much weaker than ideal hash)

e.g. $h(x) = (ax + b) \bmod p \bmod k$

where • p prime, $> k$

• a in $\{1, \dots, p-1\}$ unif. random

• b in $\{0, \dots, p-1\}$ unif. random



Markov's Inequality

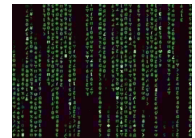
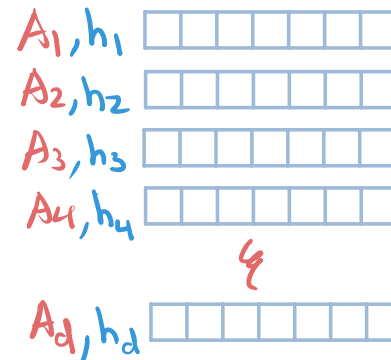
Markov's Inequality (special case)

Let $X \geq 0$ be nonnegative RV

$$\Pr[X \geq 2E[X]] \leq \frac{1}{2}$$

Algorithms

Frequency Estimation, Heavy Hitters (in streams)

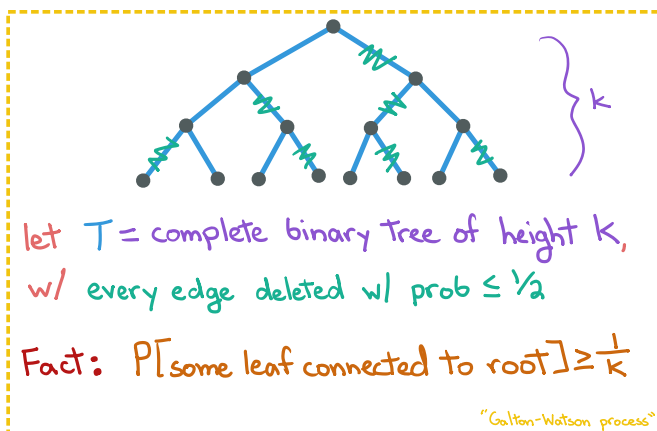


estimate f_i w/
 $\min_{j=1, \dots, d} A_j[h_j(i)]$

Lecture 4

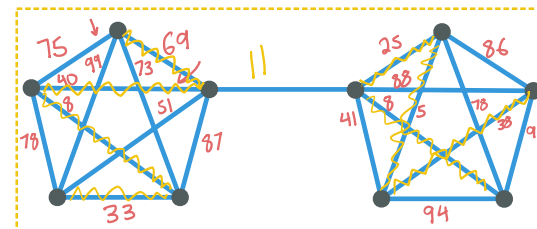
Probability

Galton-Watson processes



Algorithms

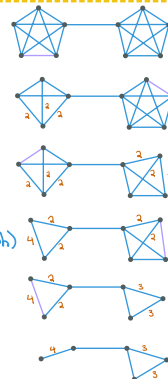
Random Contractions for min cut



Randomized contractions

Input: $G=(V,E)$, weights $c \in \mathbb{R}_{>0}^E$
(unweighted)

- ① while $|E| > 1$
 - Ⓐ sample $e \in E$ in proportion to c
 - Ⓑ "contract" e
- ② return set of edges (in original graph) contracted to remaining edge



Amplification by branching

Instead of restarting at the beginning, restart partway through



High-level algorithm

- ① randomly contract until prob. of ruining the min-cut is $\sim 1/2$
- ② Branch:
make two copies of current graph
recurse on both copies



Lecture 5

Probability

Multiplicative Chernoff Bound

Multiplicative Chernoff bound

let $X_1, \dots, X_n \in [0, 1]$ be independent

$$\mu = E[X_1 + \dots + X_n]$$

$$\epsilon \in (0, 1)$$

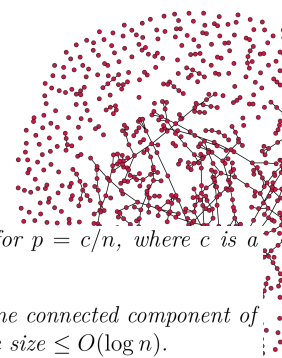
- $P[X_1 + \dots + X_n \geq (1 + \epsilon)\mu] \leq e^{-\epsilon^2 \mu / 3}$
- $P[X_1 + \dots + X_n \leq (1 - \epsilon)\mu] \leq e^{-\epsilon^2 \mu / 2}$

Algorithms (/ results)

Random graphs & the giant component

Erdős-Rényi graphs $G(n, p)$

- n vertices
- each edge sampled independently w/ probability p



Theorem 5.3. Consider a random graph $G \sim \mathcal{G}(n, p)$ for $p = c/n$, where c is a constant.

- If $c > 1$, then with high probability, there is exactly one connected component of G with $\Omega(n)$ vertices, and all other components have size $\leq O(\log n)$.
- For $c < 1$, then with high probability, all connected components of G has size $< O(\log n)$.

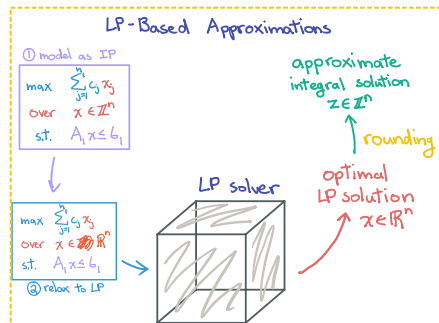
Lecture 6

Probability (& techniques)

Approximations via linear programming & randomized rounding

Linear programs (LPs)
Similar to IPs, except no integrality constraints

min/max $\sum_{j=1}^n c_j x_j$ linear objective
over $x \in \mathbb{R}^n$ linear integral variables
st. $A_1 x \leq b_1$ linear constraints
 $A_2 x = b_2$
 $A_3 x \geq b_3$



Continuous optimization vs Discrete optimization

randomization

Algorithms

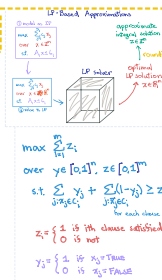
Max Sat Set Cover Covering Integer Programs

For each variable x_j , independently, randomly set $x_j = 1$ with probability p_j , and $x_j = 0$ otherwise.

$$E[\# \text{ sat. clauses}] \geq (1 - 1/e) \text{OPT}$$

k	oblivious	LP	average
1	1/2	$z_i \geq (3/4)z_i$	$\geq (3/4)z_i$
2	3/4	$(3/4)z_i \geq (3/4)z_i$	$\geq (3/4)z_i$
3	7/8	$(1 - 1/e)z_i \geq (3/4)z_i$	$\geq (3/4)z_i$
4	15/16	$(1 - 1/e)z_i \geq (3/4)z_i$	$\geq (3/4)z_i$
5	31/32	$(1 - 1/e)z_i \geq (3/4)z_i$	$\geq (3/4)z_i$
...	...	...	...
k	$1 - 1/2^k$	$(1 - 1/e)z_i \geq (3/4)z_i$	$\geq (3/4)z_i$

Can we get $\frac{3}{4}$ -APX?



Rounding LP solution
 $x \in [0, 1]^n \rightsquigarrow F \subseteq \{S_1, \dots, S_n\}$

For each set S_j , independently, take S_j with probability x_j .

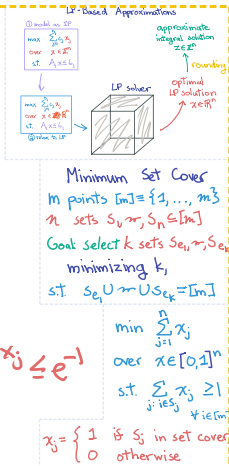
$$E[|F|] = \sum x_j \leq \text{OPT}$$

Fix a point i .

$$E[\# \text{ sets in } F \text{ covering } i] = \sum_{j \in S_i} x_j \geq z_i$$

$$P[F \text{ not cover } i] = \prod_{j \in S_i} (1 - x_j) \leq e^{-\sum_{j \in S_i} x_j} \leq e^{-z_i}$$

$$P[F \text{ cover } i] \geq 1 - e^{-1}$$



Lecture 7

Probability

Continuous Random Variables

Variance, Chebyshev's Variance Reduction

Median trick

Variance

$$\text{Var}(X) = E[(X - E[X])^2] = E[X^2] - (E[X])^2$$

Chebyshev's inequality

$$P[|X - E[X]| \geq d] \leq \frac{\text{Var}(X)}{d^2}$$

"median trick", "median of means"

- ① take mean to reduce variance, get constant prob. of error
- ② take median to amplify success prob. exponentially

Algorithms

Distinct Elements



Goal: estimate # of distinct elements in the stream

Suppose you had an ideal hash function

$$h: [m] \rightarrow [0, 1]$$

and the following facts:

let $Y_1, \dots, Y_k \in [0, 1]$ independent and uniformly random

$$X = \min \{Y_1, \dots, Y_k\}$$

$$\text{then } E[X] = \frac{1}{k+1}$$

$$\textcircled{1} X \leftarrow 1$$

② for each element e

$$\textcircled{a} X \leftarrow \min \{X, h(e)\}$$

$$/* E[X] = \frac{1}{k+1}, k = (\text{\# distinct elements}) */$$

$$\textcircled{3} \text{ return } \frac{1}{X} - 1$$

$$\text{recall } E[X] = \frac{1}{k+1}$$

$$\text{goal: show } |X - E[X]| \leq \frac{\epsilon}{k+1}$$

w/ good prob.

$$P[|X - E[X]| \geq \frac{\epsilon}{k+1}]$$

$$\leq \text{Var}(X) \cdot \left(\frac{k+1}{\epsilon}\right)^2 \leftarrow \text{Chebyshev's inequality}$$

where

$$\text{Var}(X) = \frac{k}{(k+1)^2(k+2)} \approx \frac{1}{k^2}$$

Q: how to reduce $\text{Var}(X)$?

we can get $(1 \pm \epsilon)$ -APX

w/ 10% error prob. in $O(\frac{1}{\epsilon^2})$ space

what about high probability?

① make q copies $Y_1, \dots, Y_q$ of

② return median of $Y_1, \dots, Y_q$

Algo correct $\Leftrightarrow \geq \frac{1}{2}$ of Y_i 's correct

let $Z = \# \text{correct } Y_i$'s $E[Z] = \frac{q}{2}$

$$P[Z \leq (1 - \frac{\epsilon}{4})E[Z]] \leq e^{-\left(\frac{\epsilon}{4}\right)^2 E[Z]/2} \leq \frac{1}{n^{O(n)}} \text{ for } q = O(\log n)$$

ideal hash $h: [m] \rightarrow [0, 1]$

let $Y_1, \dots, Y_k \in [0, 1]$

indep. and unifs. rand.

$X = \min \{Y_1, \dots, Y_k\}$

then $E[X] = \frac{1}{k+1}$

and $E[X^2] = \frac{2}{(k+1)(k+2)}$

① $X \leftarrow 1$

② for each element e

③ $X \leftarrow \min \{X, h(e)\}$

/* $E[X] = \frac{1}{k+1}, k = (\text{\# distinct})$ */

④ return $\frac{1}{X} - 1$

$\text{Var}(X) = E[(X - E[X])^2]$

$X_1, \dots, X_k$ indep. copies of X

$\bar{X} = \frac{1}{k}(X_1 + \dots + X_k)$

$\text{Var}(\bar{X}) = \frac{1}{k} \text{Var}(X)$

$P[X_1 + \dots + X_k \leq (1 - \epsilon)k] \leq e^{-\epsilon^2 k/4}$

Lecture 8

Probability

Gaussian random variables

Gaussian random variables

$$x \sim \mathcal{N}(\mu, \sigma^2)$$

mean variance

Def: the nicest, most pleasant, convenient random variable ever

Fact let $x \sim \mathcal{N}(\mu, \sigma^2)$, $d \in \mathbb{R}$.

$$\text{then } x+d \sim \mathcal{N}(\mu+d, \sigma^2)$$

$$dx \sim \mathcal{N}(d\mu, d^2\sigma^2)$$

Fact let $x_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$, $x_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$

$$\text{then } x_1 + x_2 \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$$

Fact let $x \sim \mathcal{N}(0, 1)$, $t < \frac{1}{2}$

$$\text{then } E[e^{tx^2}] = \frac{1}{\sqrt{1-2t}}$$

Algorithms

Dimensionality reduction

Bag-of-words

literally any sentence is fine

any ↑ is literally sentence English
five encodes text in high dimensions

Lemma 8.8. Let $A \sim \mathcal{N}^{k \times d}$ be a random matrix where each coordinate A_{ij} is an independently drawn sample from $\mathcal{N}$. Let $\epsilon > 0$ be sufficiently small. For any vector x ,

$$\mathbb{P}\left[(1-\epsilon)\|x\|^2 \leq \frac{1}{k}\|Ax\|^2 \leq (1+\epsilon)\|x\|^2\right] \geq 1 - 2e^{-k/8}$$

$$k \left\{ \left[\frac{1}{k} \|Ax\|^2 \right] \leftarrow \alpha_i \right\}$$

$$E[\|Ax\|^2] = \sum_{i=1}^k E[(Ax)_i^2] = k\|x\|^2$$

$$E[(\sum_{i=1}^k A_{ij} x_j)^2] = \|x\|^2$$

$$A_{ij} x_j \sim \mathcal{N}(0, x_j^2)$$

$$\sum_{j=1}^n A_{ij} x_j \sim \mathcal{N}(0, \|x\|^2)$$

$$(Ax) \sim \mathcal{N}^k(0, \|x\|^2)$$

$$\Pr[\|Ax\|^2 \geq k\|x\|^2 \cdot (1+\epsilon)] \leq e^{-\epsilon^2 k/8}$$

Dimensionality Reduction - high level

$O(\log n)$ dimensions suffice

where $n = \# \text{ points}$

$$\mathbb{R}^{\text{very large}} \xrightarrow{f(x)} \mathbb{R}^{O(\log n)}$$

One can map n points into $\mathbb{R}^{O(\log n/\epsilon^2)}$ while preserving all distances up to $(1 \pm \epsilon)$ -APX

Lecture 9

Probability

Locality-sensitive hashing

Fix $\theta \in [0, \pi]$. given q , either:

(a) output $p \in P$ w/ $\angle(p, q) \leq 4\theta$

(b) declare no $p \in P$ w/ $\angle(p, q) \leq \theta$

define $h: S^{d-1} \rightarrow \{-1, 1\}^k$ by

$$h_i(x) = \text{sign}(\langle g_i, x \rangle) \text{ w/ } g_i \sim \mathcal{N}^d$$

$$k = \pi(\log(n)/2\theta)$$



d-dimensional sphere

when querying y :

- if there is $x \in P$ w/ $\angle(x, y) \leq \theta$, then it collides w/ probability $\geq 1/\sqrt{n}$

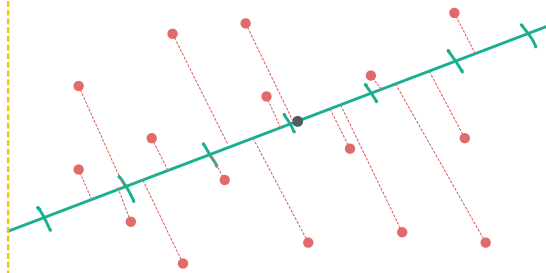
- expect to collide w/ at most one point $x \in P$ w/ $\angle(x, y) > 2\theta$

Lemma let $x, y \in S^{d-1}$
then $g \sim \mathcal{N}^d$

$$P[\text{sign}(\langle g, x \rangle) \neq \text{sign}(\langle g, y \rangle)] = \frac{\angle(x, y)}{\pi}$$

Random line embedding & Buckets

$$h(x) = \lfloor \langle g, x \rangle + d \rfloor \text{ w/ } g \sim \mathcal{N}^d, d \in [0, 1] \text{ unifs. rand.}$$

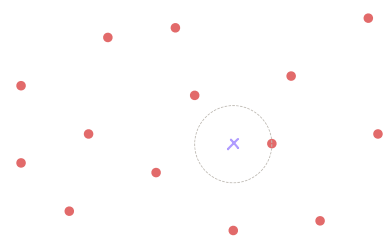


Algorithms

Approximate nearest neighbors

Nearest neighbor search

Input: n points $P \subset \mathbb{R}^d$



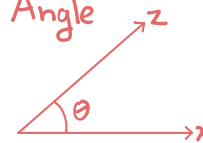
Query: given $q \in \mathbb{R}^d$, find closest point in P to q

Types of distances

- Euclidean distance

$$\|x - z\|_2 = \sqrt{\sum_i (x_i - z_i)^2}$$

- Angle



Lecture 10

Probability (and techniques)

Algorithms

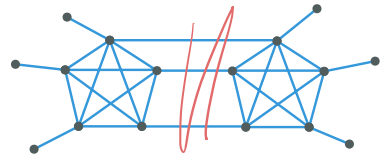
LP duality

L_1 -metric embeddings

Sparsest Cut

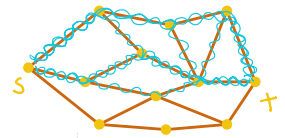
Special case: uniform sparsest cut
w/ uniform demands $b(u,v)=1 \forall u,v$

$$\text{"sparsity"} = \frac{c(\delta(s))}{|S||T|}$$



useful primitive for graph partitioning, divide&conquer

Max Flow
Min Cut



Path Packing LP

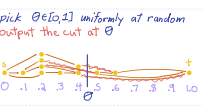
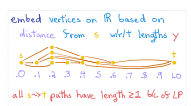
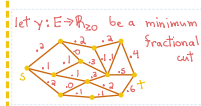
maximize $\sum_{p \in P} x_p$
over $x: P_{st} \rightarrow \mathbb{R}_{\geq 0}$
(all s-t paths)

s.t. $\sum_{p \in P} x_p \leq c_e$ for all edges e
"Ax ≤ c" (capacity of e)

Path Covering LP
(i.e. cut)

minimize $\sum_{e \in E} c_e y_e$
over $y: E \rightarrow \mathbb{R}_{\geq 0}$

s.t. $\sum_{e \in P} y_e \geq 1$ for all p ∈ P_{st}
"A'y ≥ b" (continuous analog of s-t cut)



$E[\text{capacity of cut}]$

$$\stackrel{\text{(linearity of expectation)}}{=} \sum_{e \in E} c_e \Pr[e \text{ cut}] \leq \sum_{e \in E} c_e y_e$$



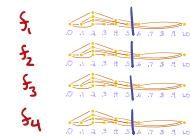
Q: how to round LP to sparse cut?

min $\sum_{e \in E} c_e d(u,v)$ over metrics $d: V \times V \rightarrow \mathbb{R}_{\geq 0}$
s.t. $\sum_{\{s,t\}} b(s,t) d(s,t) \geq 1$

Suppose $d(u,v)$ was of the form

$$d(u,v) = \sum_{i=1}^k |\xi_i(u) - \xi_i(v)| \text{ for } \xi_i: V \rightarrow [0,1]^k$$

(assume $\min \xi_i(a)=1, \max \xi_i(a)=1$)



" L_1 -metric"

Lecture 11

Probability (and techniques)

Algorithms

Other metric-based problems

(sometimes via correspondence between cuts $\leftrightarrow$ metrics)

Buy-at-bulk network design
k-median clustering
metric labeling
group Steiner tree
metrical task systems
vehicle routing
hierarchical placement
distributed k-server
covering Steiner tree
...



these problems are easy for tree metrics

Approximation by tree metrics

Input: undirected $G=(V,E)$ w/ edge weights $w: E \rightarrow \mathbb{R}_{\geq 0}$

(High-level)

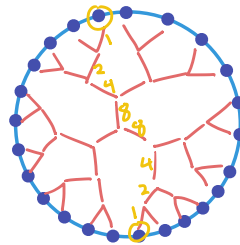
Goal: compute tree T (containing V as nodes)
so that shortest paths on T are close to T

$$d_T(u,v) \geq d_G(u,v)$$

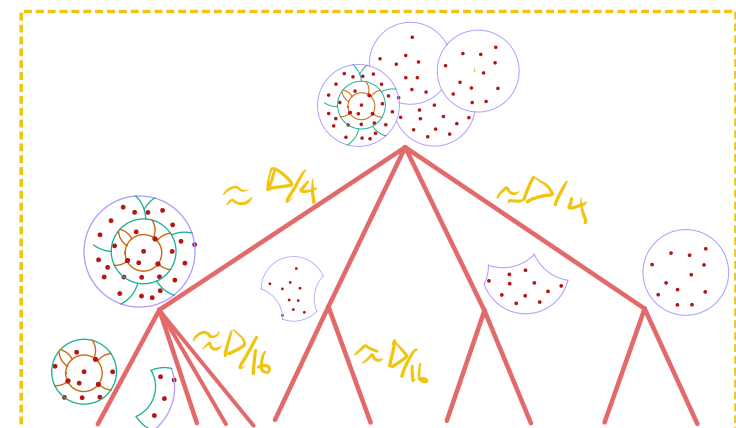
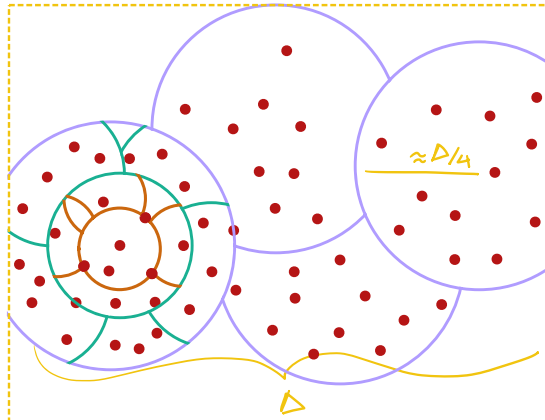
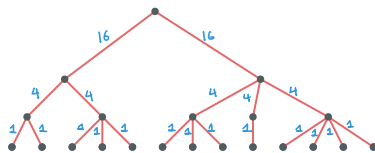
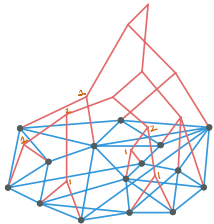
$$\text{For } e=f_u,v \quad \begin{matrix} \text{(distance in } T) \\ \text{(stretch } e) = \frac{d_T(u,v)}{d_G(u,v)} \\ \text{(distance in } G) \end{matrix}$$

ideally

$$(\text{stretch } e) = \text{polylog}(n) \quad \forall e$$



Theorem In randomized polynomial time,
randomized hierarchical dominating tree metric w/
 $E[\text{stretch } e] \leq O(\log n) \quad \forall e \in E$



Lecture 12

Probability (and techniques)

Additive Chernoff Inequality

Double-sampling

VC dimension

Additive Chernoff bound

let $X_1, \dots, X_n \in [0, 1]$ be independent
 $\mu = E[X_1 + \dots + X_n] / n$
 $\epsilon \in (0, 1)$

- $P[\frac{X_1 + \dots + X_n}{n} \geq \mu + \epsilon] \leq e^{-2\epsilon^2 n}$
- $P[\frac{X_1 + \dots + X_n}{n} \leq \mu - \epsilon] \leq e^{-2\epsilon^2 n}$

Algorithms

ϵ -samples and ϵ -nets

ϵ -sample $|M_Q(r) - \mu(r)| \leq \epsilon \quad \forall r \in R$

let Q sample k points w/ repetition
 $k = O\left(\frac{\ln(g(|P|))}{\epsilon^2}\right) \Rightarrow \epsilon$ -sample $k = O\left(\frac{\ln(g(|P|))}{\epsilon^2}\right) \Rightarrow \epsilon$ -net (w/ constant probability)

ϵ -sample theorem [VC71]
 $k \geq O\left(\frac{\ln(g(2k))}{\epsilon^2}\right) \Rightarrow \epsilon$ -sample (w/ constant probability)

e.g. $g(n) = O(n^d)$
 $k \geq O\left(\frac{d \ln(d/\epsilon)}{\epsilon^2}\right) \Rightarrow \epsilon$ -sample
 ↑ independent of $|P|$ (!)

$P[\frac{X_1 + \dots + X_n}{n} \geq \mu + \epsilon], P[\frac{X_1 + \dots + X_n}{n} \leq \mu - \epsilon] \leq e^{-2\epsilon^2 n}$ $g(n) = \max\{|RAQ| : \text{asf, lalsn}\}$

$\mu(r) = \frac{|P \cap r|}{|P|}$
 $M_Q(r) = \frac{|Q \cap r|}{|Q|}$
 $RAQ = \{r \cap q : r \in R, q \in Q\}$

ϵ -sample $|M_Q(r) - \mu(r)| \leq \epsilon \quad \forall r \in R$

let Q sample k points w/ repetition
 $k = O\left(\frac{\ln(g(|P|))}{\epsilon^2}\right) \Rightarrow \epsilon$ -sample $k = O\left(\frac{\ln(g(|P|))}{\epsilon^2}\right) \Rightarrow \epsilon$ -net (w/ constant probability)

ϵ -net theorem [HW87]
 $k \geq O\left(\frac{\ln(g(2k))}{\epsilon^2}\right) \Rightarrow \epsilon$ -net (w/ constant probability)

e.g. $g(n) = O(n^d)$
 $k \geq O\left(\frac{d \ln(d/\epsilon)}{\epsilon^2}\right) \Rightarrow \epsilon$ -net
 ↑ independent of $|P|$ (!)

$P[\frac{X_1 + \dots + X_n}{n} \geq \mu + \epsilon], P[\frac{X_1 + \dots + X_n}{n} \leq \mu - \epsilon] \leq e^{-2\epsilon^2 n}$ $g(n) = \max\{|RAQ| : \text{asf, lalsn}\}$

$\mu(r) = \frac{|P \cap r|}{|P|}$
 $M_Q(r) = \frac{|Q \cap r|}{|Q|}$
 $RAQ = \{r \cap q : r \in R, q \in Q\}$

Lecture 13

Probability
(and techniques)

Algorithms
(models and applications)

Consistency model

PAC learning model

Generalization error

Occam's Razor

Generalization by growth

Lecture 14

Probability
(and techniques)

Algorithms
(models and applications)

Entropy

(Optimal)
Error-Correcting
Codes

Lecture 14

Probability (and techniques)

Lovász Local Lemma

Algorithms (models and applications)

Moser's resampling
algorithm

3-SAT

"conjunctive normal form"

CNF formula

$$\varphi(x_1, \dots, x_n) = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_4 \vee \dots) \wedge \dots$$

m clauses, n variables

3 (distinct) variables per clause

The silliest APX algorithm

1. for $i = 1, \dots, n$

a. randomly set $x_i = \begin{cases} \text{TRUE} & \text{w/ prob .5} \\ \text{FALSE} & \text{w/ prob .5} \end{cases}$

We will:

① prove silly algo is $\frac{7}{8}$ -APX

② derandomize silly algo $\Rightarrow$ deterministic $\frac{7}{8}$ -APX

Even more surprising:

"probabilistically checkable proofs"

PCP Theorem

$(\frac{7}{8} + \epsilon)$ -APX is NP-Hard $\forall \epsilon > 0$

Approximating 3SAT: "Max 3-SAT"

$$\varphi(x_1, \dots, x_n) = (x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_4 \vee \dots) \wedge \dots$$

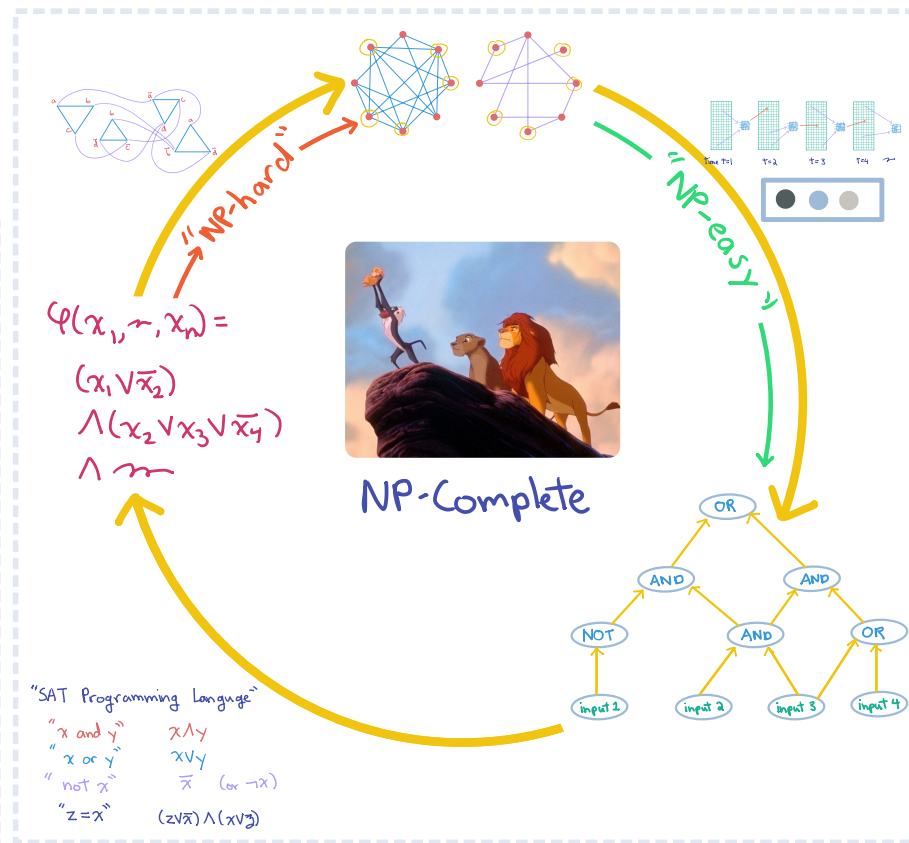
Goal: satisfy Max # clauses

Interesting even if φ is unsatisfiable

"d-Approximation algorithm" $d \in (0, 1)$

Algorithm satisfies $d \cdot \text{OPT}$ clauses

optimum (max) value



Suppose $m < 2^k$
(# clauses)

for random assignment $x_1, \dots, x_n \in \{0,1\}^n$:

$$\Pr[f(x_1, \dots, x_n) = \text{false}]$$

$$\leq \sum_{i=1}^m \Pr[\text{i-th clause unsatisfied}]$$

$$= \frac{m}{2^k} < 1$$

$\Rightarrow f$ must be satisfiable!

for $m > 2^k$, union bound not strong enough...

Resampling algorithm (for k-SAT)

1. Sample $x_1, \dots, x_n \in \{T, F\}$ (independently, uniformly at random)
2. while there is an unsatisfied clause C :
 - a. resample each x_i in C (independently, uniformly at random)
3. return $x_1, \dots, x_n$

Resampling algorithm (for k-SAT) (Moser, 2009)

1. Sample $x_1, \dots, x_n \in \{T, F\}$ (independently, uniformly at random)
2. while there is an unsatisfied clause C :
 - a. resample each x_i in C (independently, uniformly at random)
3. return $x_1, \dots, x_n$

Theorem Let $\mathcal{f}(x_1, \dots, x_n)$ be a k-SAT formula where each clause intersects at most d other clauses.

If $d \leq \frac{2^k}{e} - 1$, then

- $\mathcal{f}$ is satisfiable.
- resampling algo finds satisfying assignment while resampling each clause $\frac{1}{d}$ times in expectation

Resampling algorithm (for k-SAT) (Moser, 2009)

1. Sample $x_1, \dots, x_n \in \{T, F\}$ (independently, uniformly at random)
2. while there is an unsatisfied clause C :
 - a. resample each x_i in C (independently, uniformly at random)
3. return $x_1, \dots, x_n$

Theorem Let $\phi(x_1, \dots, x_n)$ be a k-SAT formula where each clause intersects at most d other clauses.

If $d \leq \frac{2^k}{e} - 1$, then

locally verifiable

- ϕ is satisfiable. $\leftarrow$ 1970's

- resampling algo finds satisfying assignment while resampling each clause $\frac{1}{d}$ times in expectation

2009

Lovász Local Lemma

(k-SAT is special case)

$A_1, \dots, A_n$ events

(e.g. A_i = i th clause unsatisfied)

" $A_i \sim A_j$ " $\Rightarrow$ A_i and A_j not mutually independent

(e.g. $A_i \sim A_j$ if i th and j th clause intersect)

Q: Is it possible for none of $A_1, \dots, A_n$ to occur?

Is $P[\bar{A}_1, \bar{A}_2, \dots, \bar{A}_n] > 0$?

YES if $A_1, \dots, A_n$ are independent

NO w/ no assumptions on dependence

Lovász Local Lemma

(k-SAT is special case)

$A_1, \dots, A_n$ events

(e.g. A_i = i th clause unsatisfied)

" $A_i \sim A_j$ " $\Rightarrow$ A_i and A_j not mutually independent

(e.g. $A_i \sim A_j$ is i th and j th clause intersect)

[Erdős and Lovász, 1975]

Theorem: Suppose $\exists \chi(A_1), \dots, \chi(A_n) \in (0,1)$ s.t.

$$P[A_i] \leq \chi(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - \chi(A_j)) \quad \forall A_i.$$

Then $P[\bar{A}_1, \dots, \bar{A}_n] > 0$

Theorem: Suppose $\exists x(A_1), \dots, x(A_n) \in (0,1)$ s.t.

$$P[A_i] \leq x(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - x(A_j)) \quad \forall A_i.$$

Then $P[\bar{A}_1, \dots, \bar{A}_n] > 0$

Corollary: Suppose each A_i is independent from all but d other events A_j , and $P[A_i] \leq p \quad \forall A_i$.

If $e_p(d+1) \leq 1$, then $P[\bar{A}_1, \dots, \bar{A}_n] > 0$.

Proof: set $x(A_i) = \frac{1}{d+1} \quad \forall A_i$

Theorem: Suppose $\exists x(A_1), \dots, x(A_n) \in (0,1)$ s.t.
 $P[A_i] \leq x(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - x(A_j)) \quad \forall A_i.$

Then $P[\bar{A}_1, \dots, \bar{A}_n] > 0$

Corollary: Suppose each A_i is independent from all but d other events A_j , and $P[A_i] \leq p \quad \forall A_i.$

If $ep(d+1) \leq 1$, then $P[\bar{A}_1, \dots, \bar{A}_n] > 0.$

Resampling algorithm (for k-SAT) (Moser, 2009)

1. Sample $x_1, \dots, x_n \in \{T, F\}$ (independently, uniformly at random)
2. while there is an unsatisfied clause C :
 - a. resample each x_i in C (independently, uniformly at random)
3. return $x_1, \dots, x_n$

Theorem Let $\phi(x_1, \dots, x_n)$ be a k-SAT formula where each clause intersects at most d other clauses.

If $d \leq \frac{2^k}{e} - 1$, then

• ϕ is satisfiable.

• resampling algo finds satisfying assignment while resampling each clause $\frac{1}{d}$ times in expectation



Lovász Local Lemma - variable setting (k-SAT is special case)

$A_1, \dots, A_n$ events

(e.g. A_i = i th clause unsatisfied)

" $A_i \sim A_j$ " $\Rightarrow A_i$ and A_j not mutually independent

(e.g. $A_i \sim A_j$ is i th and j th clause intersect)

Variables $Y = \{X_1, X_2, \dots\}$

(e.g., $x_1, \dots, x_n \in \{T, F\}$)

- Each X_i independent w/ its own distribution
- Each A_i determined by subset $S_i \subseteq Y$
- " $A_i \sim A_j$ " $\Rightarrow S_i \cap S_j \neq \emptyset$

(Moser-Tardos, 2010)

Lovász Local Lemma - variable setting
(k-SAT is special case)

$A_1, \dots, A_n$ events (e.g. A_i = i th clause unsatisfied)

" $A_i \sim A_j$ " $\Rightarrow A_i$ and A_j
~~not~~ mutually independent
(e.g. $A_i \sim A_j$ is i th and j th clause intersect)

Variables $Y = \{X_1, X_2, \dots\}$ (e.g. $x_i, \neg x_i \in \{T, F\}$)

- Each X_i independent w/ its own distribution
- Each A_i determined by subset $S_i \subseteq Y$
- " $A_i \sim A_j$ " $\Rightarrow S_i \cap S_j \neq \emptyset$

Resampling algorithm (for general LLL)

1. Sample all variables Y (independently, from their respective distributions)
2. while there is an event A_i
 - a. resample all of S_i (independently, from their respective distributions)

Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

Theorem: Suppose $\exists x(A_1), \dots, x(A_n) \in (0,1)$ s.t.
$$P[A_i] \leq x(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - x(A_j)) \quad \forall A_i.$$

Then $P[\bar{A}_1, \dots, \bar{A}_n] > 0$, and the

resampling algo finds a configuration of V inducing $\bar{A}_1, \dots, \bar{A}_n$ while resampling each A_i at most $\frac{x(A_i)}{1 - x(A_i)}$ times in expectation

Proof.

very subtle "coupling" argument

need to introduce:

1. "log"
2. "witness trees"

Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

Theorem: Suppose $\exists x(A_1), \dots, x(A_n) \in (0,1)$ s.t.

$$P[A_i] \leq x(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - x(A_j)) \quad \forall A_i.$$

Then $P[\bar{A}_1, \dots, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \dots, \bar{A}_n$ while resampling each A_i at most $\frac{x(A_i)}{1-x(A_i)}$ times in expectation

The "log"

Records resampled
events in chronological

Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

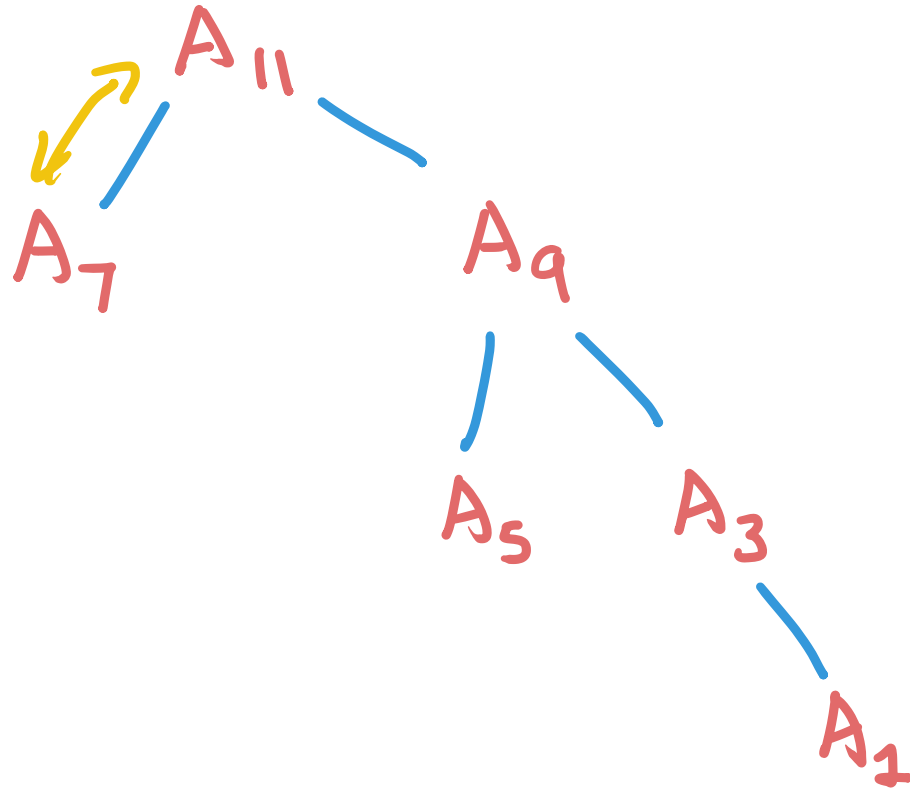
$A_5, A_7, A_1, A_{23}, A_{101}, \dots$

want to bound $E[|\log|]$

Witness trees

Def: rooted tree w/ nodes labeled by events

(labels can repeat)



Mapping prefixes of the log $\rightarrow$ witness trees

Suppose first l events are $A_{i_1}, A_{i_2}, \dots, A_{i_l}$

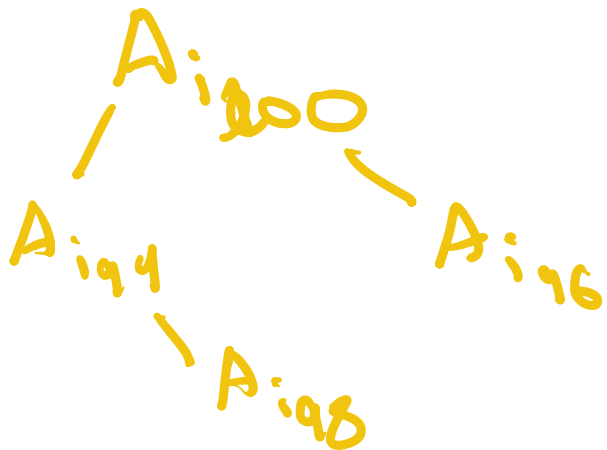
① Make root w/ label A_{i_l} .

② For $j=l-1, l-2, \dots, 1$

③ If $A_{i_j} \sim A_k$ for some label A_k in the tree

① pick such a node N of max depth

② make a new child of N w/ label A_{i_j}



Suppose first l events are $A_{i_1}, A_{i_2}, \dots, A_{i_l}$

① Make root w/ label A_{i_l} .

② For $j=l-1, l-2, \dots, 1$

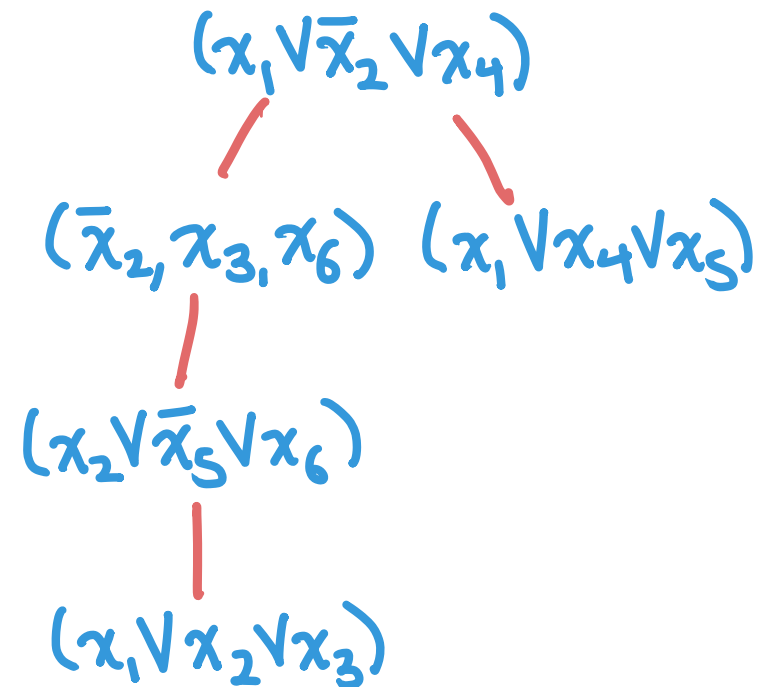
ⓐ If $A_{i_j} \sim A_k$ for some label A_k in the tree

① pick such a node N of max depth

② make a new child of N w/ label A_{i_j}

Log

$(x_1 \vee x_2 \vee x_3), (x_2 \vee \bar{x}_5 \vee x_6), (x_1, \bar{x}_8, x_9),$
 $(x_1 \vee x_4 \vee x_5), (\bar{x}_2, x_3, x_6), (x_1 \vee \bar{x}_2 \vee x_4)$



Suppose first l events are $A_{i_1}, A_{i_2}, \dots, A_{i_l}$

① Make root w/ label A_{i_l} .

② For $j=l-1, l-2, \dots, 1$

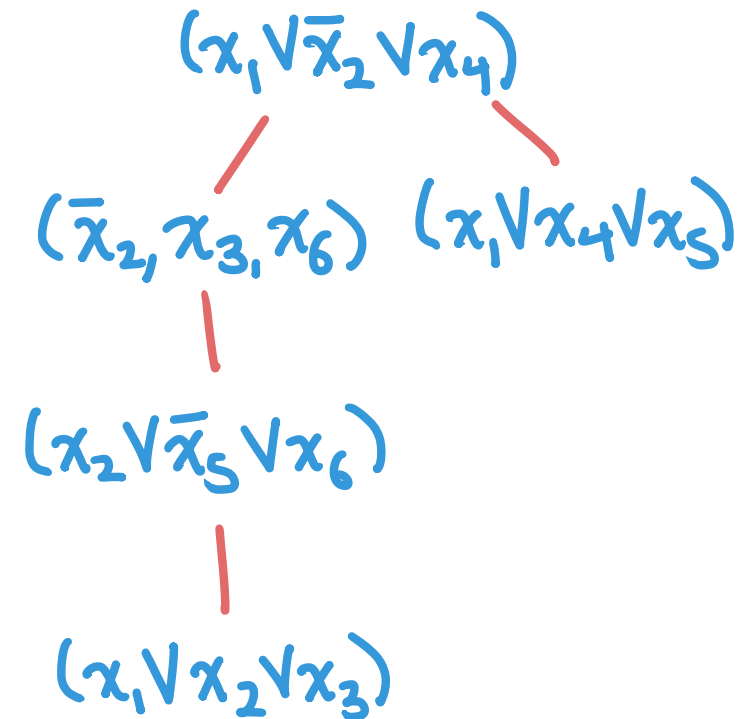
③ If $A_{i_j} \sim A_k$ for some label A_k in the tree

① pick such a node N of max depth

② make a new child of N w/ label A_{i_j}

Log

$(x_1, \vee x_2 \vee x_3), (x_2, \vee \bar{x}_5 \vee x_6), (x_7, \bar{x}_8, x_9),$
 $(x_1, \vee x_4 \vee x_5), (\bar{x}_2, x_3, x_6), (x_1, \vee \bar{x}_2 \vee x_4)$



Suppose first i events are $A_{i_1}, A_{i_2}, \dots, A_{i_i}$.

① Make root w/ label A_{i_i} .

② For $j=i-1, i-2, \dots, 1$

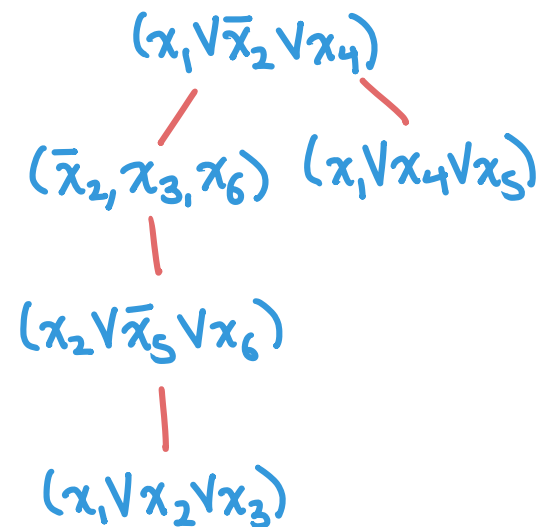
③ If $A_{i_j} \sim A_k$ for some label A_k in the tree

① pick such a node N of max depth

② make a new child of N w/ label A_{i_j}

Log

$(x_1 \vee x_2 \vee x_3), (x_2 \vee \bar{x}_5 \vee x_6), (x_7, \bar{x}_8, x_9),$
 $(x_1 \vee x_4 \vee x_5), (\bar{x}_2, x_3, x_6), (x_1 \vee \bar{x}_2 \vee x_4)$



Observation (for later)

Let T_ℓ = witness tree for $A_{i_1}, \dots, A_{i_\ell}$

Suppose $j < k$ and $A_{i_j} \sim A_{i_k}$

if A_{i_k} is a node in T_ℓ , then so is A_{i_j} ,

and $\text{depth}(A_{i_j}) > \text{depth}(A_{i_k})$



Suppose first i events are $A_{i_1}, A_{i_2}, \dots, A_{i_i}$.

① Make root w/ label A_{i_i} .

② For $j=i-1, i-2, \dots, 1$

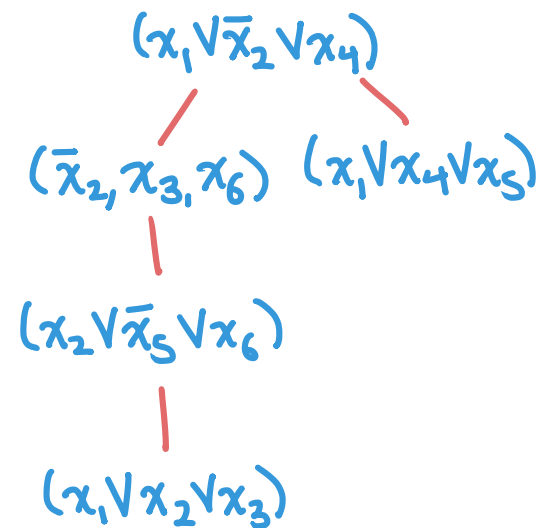
③ If $A_{i_j} \sim A_k$ for some label A_k in the tree

① pick such a node N of max depth

② make a new child of N w/ label A_{i_j}

Log

$(x_1 \vee x_2 \vee x_3), (x_2 \vee \bar{x}_5 \vee x_6), (x_7, \bar{x}_8, x_9),$
 $(x_1 \vee x_4 \vee x_5), (\bar{x}_2, x_3, x_6), (x_1 \vee \bar{x}_2 \vee x_4)$



Lemma let j, k index diff. entries in log
 then $T_j \neq T_k$

Proof. Suppose $A_{i_j} = A_{i_k}$.

log: $(A_{i_1}, A_{i_2}, \dots, \boxed{A_{i_j}}, \dots, A_{i_k})$



Fix A_i .

Goal: bound # times A_i resampled

let $W_i = \{\text{all witness trees w/ root } A_i\}$

for $j=1, 2, \dots$, let T_j be witness tree

for the first j entries of the log

(or $\emptyset$ if log has $< j$ entries)

$$E \left[\begin{array}{c} \# \text{ times } A_i \\ \text{resampled} \end{array} \right] = E \left[\begin{array}{c} \# \text{ times } A_i \\ \text{appears in log} \end{array} \right]$$
$$= \sum_{j=1}^{\infty} \sum_{T \in W_i} P_c [T_j = T]$$

Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

Theorem: Suppose $\exists \chi(A_i), \dots, \chi(A_n) \in (0,1)$ s.t.
 $P[A_i] \leq \chi(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - \chi(A_j)) \quad \forall A_i.$

Then $P[\bar{A}_1, \dots, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \dots, \bar{A}_n$ while resampling each A_i at most $\frac{\chi(A_i)}{1 - \chi(A_i)}$ times in expectation

Mapping prefixes of the log $\rightarrow$ witness trees

Suppose first i events are $A_{i_1}, A_{i_2}, \dots, A_{i_i}$.

① Make root w/ label A_{i_i} .

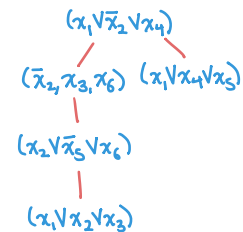
② For $j=i-1, i-2, \dots, 1$

ⓐ If $A_{i_j} \sim A_k$ for some label A_k in the tree

① pick such a node N of max depth

ⓐ make a new child of N w/ label A_{i_j}

Log
(x_1, x_2, x_3), ($x_2, \bar{x}_5, x_6$), ($x_1, \bar{x}_8, x_9$),
(x_1, x_4, x_5), ($\bar{x}_2, x_3, x_6$), ($x_1, \bar{x}_2, x_4$)



Lemma $j \neq k \Rightarrow T_j \neq T_k$

let $W_i = \{\text{all witness trees w/ root } A_i\}$

for $j=1,2,\dots$, let T_j be witness tree

for the first j entries of the log

(or $\emptyset$ if log has $< j$ entries)

$$E[\text{\# times } A_i \text{ resampled}] = E[\text{\# times } A_i \text{ appears in log}]$$

$$= \sum_{j=1}^{\infty} \sum_{T \in W_i} P_r[T_j = T]$$

$$= \sum_{T \in W_i} P_r[T_j = T \text{ for some } j]$$

Lemma let j, k index diss. entries in log
then $T_j \neq T_k$

Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

Theorem: Suppose $\exists \chi(A_i), \dots, \chi(A_n) \in (0,1)$ s.t.
 $P[A_i] \leq \chi(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - \chi(A_j)) \quad \forall A_i.$

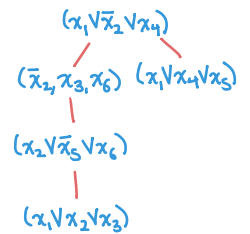
Then $P[\bar{A}_1, \dots, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \dots, \bar{A}_n$ while resampling each A_i at most $\frac{\chi(A_i)}{1 - \chi(A_i)}$ times in expectation

Mapping prefixes of the log $\rightarrow$ witness trees

Suppose first i events are $A_{i_1}, A_{i_2}, \dots, A_{i_i}$.

- ① Make root w/ label A_{i_1} .
- ② For $j=i-1, i-2, \dots, 1$
 - Ⓐ If $A_{i_j} \sim A_k$ for some label A_k in the tree
 - ① pick such a node N of max depth
 - ② make a new child of N w/ label A_{i_j}

Log
 $(x_1, x_2, x_3), (x_2, \bar{x}_5, x_6), (x_1, \bar{x}_8, x_9),$
 $(x_1, x_4, x_5), (\bar{x}_2, x_3, x_6), (x_1, \bar{x}_2, x_4)$



Lemma $j \neq k \Rightarrow T_j \neq T_k$

let $W_i = \{\text{all witness trees w/ root } A_i\}$

for $j=1,2,\dots$, let T_j be witness tree for the first j entries of the log (or $\emptyset$ if log has $< j$ entries)

$$\checkmark E \left[\begin{array}{c} \# \text{ times } A_i \\ \text{resampled} \end{array} \right] = \sum_{T \in W_i} \Pr[T_j = T \text{ for some } j]$$

$$\Pr[T_j = T \text{ for some } j]?$$

Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

Theorem: Suppose $\exists \chi(A_1), \dots, \chi(A_n) \in (0,1)$ s.t.
 $\Pr[A_i] \leq \chi(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - \chi(A_j)) \quad \forall A_i.$

Then $\Pr[\bar{A}_1, \dots, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \dots, \bar{A}_n$ while resampling each A_i at most $\frac{\chi(A_i)}{1 - \chi(A_i)}$ times in expectation

Mapping prefixes of the log $\rightarrow$ witness trees

Suppose first i events are $A_{i1}, A_{i2}, \dots, A_{ii}$.

① Make root w/ label A_{ii} .

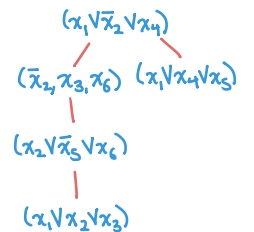
② For $j=i-1, i-2, \dots, 1$

ⓐ If $A_{ij} \sim A_k$ for some label A_k in the tree

① pick such a node N of max depth

ⓐ make a new child of N w/ label A_{ij}

Log
 $(x_1, x_2, x_3), (x_2, \bar{x}_5, x_6), (x_1, \bar{x}_8, x_9),$
 $(x_1, x_4, x_5), (\bar{x}_2, x_3, x_6), (x_1, \bar{x}_2, x_4)$



Lemma $j \neq k \Rightarrow T_j \neq T_k$

let $p(T) = \prod P[A_j]$
labels A_j in T
 (w/ multiplicity)

Lemma $\Pr[T_j = T \text{ for some } j] \leq p(T)$

Proof introduce "checking" procedure

check(T)

① For each event A_{ij} in T in decr. depth: (leaf first, root last)

① Sample variables in S_{ij} . If the sample does not induce A_{ij} , then return FAIL

② return SUCCESS

$p[\text{check}(T) = \text{SUCCESS}] = p(T)$

Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

Theorem: Suppose $\exists \chi(A_i), \dots, \chi(A_n) \in (0,1)$ s.t.
 $P[A_i] \leq \chi(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - \chi(A_j)) \quad \forall A_i.$

Then $P[\bar{A}_1, \dots, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \dots, \bar{A}_n$ while resampling each A_i at most $\frac{\chi(A_i)}{1 - \chi(A_i)}$ times in expectation

Mapping prefixes of the log $\rightarrow$ witness trees

Suppose first i events are $A_{i1}, A_{i2}, \dots, A_{ii}$.

① Make root w/ label A_{ii} .

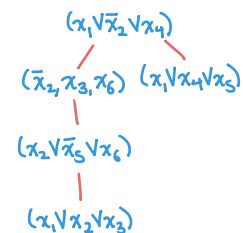
② For $j = i-1, i-2, \dots, 1$

① If $A_{ij} \sim A_k$ for some label A_k in the tree

① pick such a node N of max depth

② make a new child of N w/ label A_{ij}

Log
 $(x_1, x_2, x_3), (x_2, \bar{x}_5, x_6), (x_1, \bar{x}_8, x_9),$
 $(x_1, x_4, x_5), (\bar{x}_2, x_3, x_6), (x_1, \bar{x}_2, x_4)$



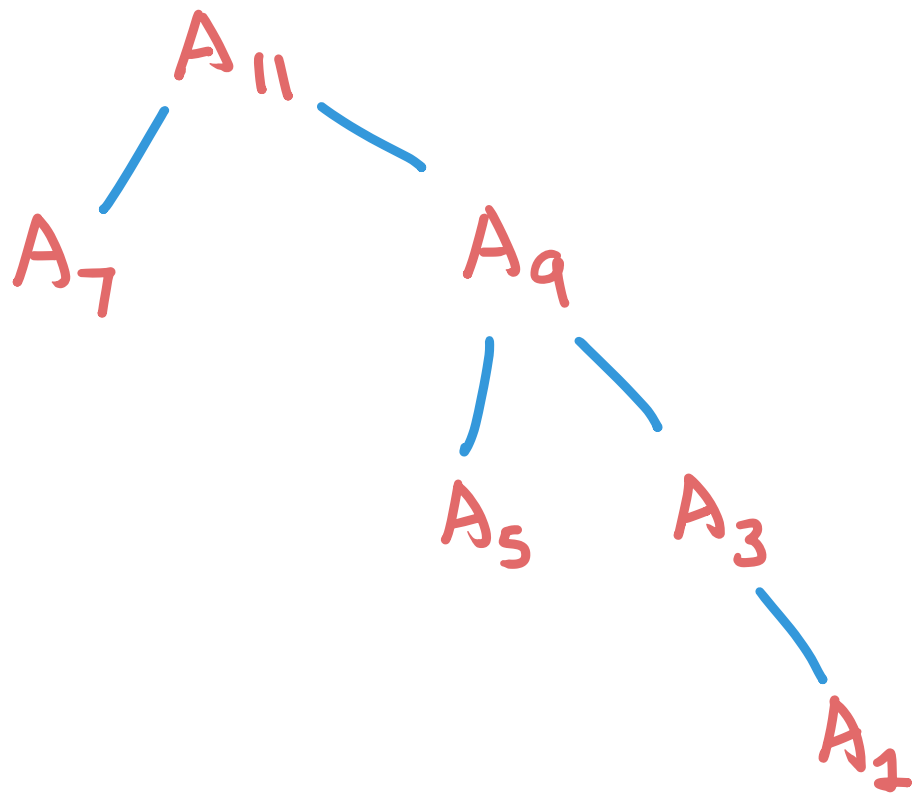
Lemma $j \neq k \Rightarrow T_j \neq T_k$

check(T)

① For each event A_{ij} in T in decr. depth: (leaf first, root last)

① Sample variables in S_{ij} . If the sample does not induce A_{ij} , then return FAIL

② return SUCCESS



Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

Theorem: Suppose $\exists \chi(A_i), \sim, \chi(A_n) \in (0,1)$ s.t.
 $P[A_i] \leq \chi(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - \chi(A_j)) \quad \forall A_i.$

Then $P[\bar{A}_1, \sim, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \sim, \bar{A}_n$ while resampling each A_i at most $\frac{\chi(A_i)}{1 - \chi(A_i)}$ times in expectation

Mapping prefixes of the log $\rightarrow$ witness trees

Suppose first i events are $A_{i1}, A_{i2}, \sim, A_{ii}$.

① Make root w/ label A_{ii} .

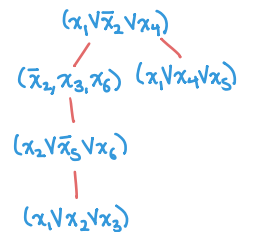
② For $j=i-1, i-2, \dots, 1$

① If $A_{ij} \sim A_k$ for some label A_k in the tree

① pick such a node N of max depth

② make a new child of N w/ label A_{ij}

Log
 $(x_1 \vee x_2 \vee x_3), (x_2 \vee \bar{x}_5 \vee x_6), (x_1, \bar{x}_8, x_9),$
 $(x_1 \vee x_4 \vee x_5), (\bar{x}_2, x_3, x_6), (x_1 \vee \bar{x}_2 \vee x_4)$



Lemma $j \neq k \Rightarrow T_j \neq T_k$

Lemma $\Pr[T_j = T \text{ for some } j] \leq p(T) = \prod P[A_j]$
labels A_j in T
(w/ multiplicity)

check(T)

- ① For each event A_{ij} in T in decr. depth:
(label) (leaf first, root last)
- ⓐ Sample variables in S_{ij} . If the sample does not induce A_{ij} , then return FAIL
- ② return SUCCESS

$$\Pr[\text{check}(T) \text{ succeeds}] = p(T) = \prod P[A_j]$$

labels A_j in T
(w/ multiplicity)

claim: $\Pr[T_j = T \text{ for some } j] \leq \Pr[\text{check}(T) \text{ succeeds}]$

Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

Theorem: Suppose $\exists \chi(A_i), \sim, \chi(A_n) \in (0,1)$ s.t.
 $P[A_i] \leq \chi(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - \chi(A_j)) \quad \forall A_i.$

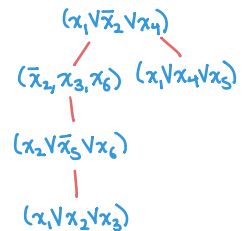
Then $P[\bar{A}_1, \sim, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \sim, \bar{A}_n$ while resampling each A_i at most $\frac{\chi(A_i)}{1 - \chi(A_i)}$ times in expectation

Mapping prefixes of the log $\rightarrow$ witness trees

Suppose first i events are $A_{i1}, A_{i2}, \sim, A_{ii}$.

- ① Make root w/ label A_{ii} .
- ② For $j = i-1, i-2, \dots, 1$
 - ⓐ If $A_{ij} \sim A_k$ for some label A_k in the tree
 - ① pick such a node N of max depth
 - ⓐ make a new child of N w/ label A_{ij}

Log
 $(x_1 \vee x_2 \vee x_3), (x_2 \vee \bar{x}_5 \vee x_6), (x_1, \bar{x}_8, x_9),$
 $(x_1 \vee x_4 \vee x_5), (\bar{x}_2, x_3, x_6), (x_1 \vee \bar{x}_2 \vee x_4)$



Lemma $j \neq k \Rightarrow T_j \neq T_k$

claim: $\Pr[T_j = T \text{ for some } j] \leq \Pr[\text{check}(T)]$

Imagine generating all (repeated) samples for all variables ahead of time.

V	1st sample	2nd sample	3rd sample	4th sample	5th	6th	7th	
X_1	$X_1^{(1)}$	$X_1^{(2)}$	$X_1^{(3)}$	$X_1^{(4)}$	$X_1^{(5)}$	$X_1^{(6)}$	$X_1^{(7)}$	~
X_2	$X_2^{(1)}$	$X_2^{(2)}$	$X_2^{(3)}$	$X_2^{(4)}$	$X_2^{(5)}$	$X_2^{(6)}$	$X_2^{(7)}$	~
X_3	$X_3^{(1)}$	$X_3^{(2)}$	$X_3^{(3)}$	$X_4^{(4)}$	$X_5^{(5)}$	$X_6^{(6)}$	$X_7^{(7)}$	~
$\{$								

we run resampling algo and check(T) using same samples

claim: if log induces T, then check(T) succeed

Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

Theorem: Suppose $\exists \chi(A_i), \sim, \chi(A_n) \in (0,1)$ s.t.
 $\Pr[A_i] \leq \chi(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - \chi(A_j)) \quad \forall A_i.$

Then $\Pr[\bar{A}_1, \sim, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \sim, \bar{A}_n$ while resampling each A_i at most $\frac{\chi(A_i)}{1 - \chi(A_i)}$ times in expectation

Mapping prefixes of the log $\rightarrow$ witness trees

Suppose first i events are $A_{i1}, A_{i2}, \dots, A_{ij}$.

① Make root w/ label A_{ij} .

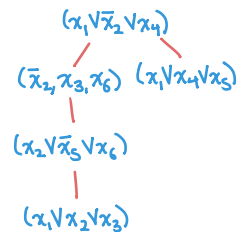
② For $j = i-1, i-2, \dots, 1$

ⓐ If $A_{ij} \sim A_k$ for some label A_k in the tree

① pick such a node N of max depth

ⓐ make a new child of N w/ label A_{ij}

Log
 $(x_1, x_2, x_3), (x_2, x_5, x_6), (x_1, x_8, x_9),$
 $(x_1, x_4, x_5), (\bar{x}_2, x_3, x_6), (x_1, x_2, x_4)$



check(T)

① For each event A_{ij} in T in decr. depth:
 (label) (leaf first, root last)

ⓐ Sample variables in S_{ij} . If the sample does not induce A_{ij} , then return FAIL

② return SUCCESS

we run resampling algo and check(T) using same samples

claim: if log induces T, then check(T) succeeds

V	1st sample	2nd sample	3rd sample	4th sample	5th	6th	7th	
X_1	$X_1^{(1)}$	$X_1^{(2)}$	$X_1^{(3)}$	$X_1^{(4)}$	$X_1^{(5)}$	$X_1^{(6)}$	$X_1^{(7)}$	~
X_2	$X_2^{(1)}$	$X_2^{(2)}$	$X_2^{(3)}$	$X_2^{(4)}$	$X_2^{(5)}$	$X_2^{(6)}$	$X_2^{(7)}$	~
X_3	$X_3^{(1)}$	$X_3^{(2)}$	$X_3^{(3)}$	$X_3^{(4)}$	$X_3^{(5)}$	$X_3^{(6)}$	$X_3^{(7)}$	~
$\vdots$								

each A_{ij} maps to sample of S_{ij} inducing A_{ij} (before resampling A_{ij})

say log = $A_{i1}, A_{i2}, A_{i3}, \dots$

say T has labels $A_{j1}, A_{j2}, A_{j3}, \dots$ in decr. depth

we need each A_{jk} to use sample of S_{jk} inducing A_{jk}

Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

Theorem: Suppose $\exists \chi(A_i), \dots, \chi(A_n) \in (0,1)$ s.t.
 $P[A_i] \leq \chi(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - \chi(A_j)) \quad \forall A_i.$

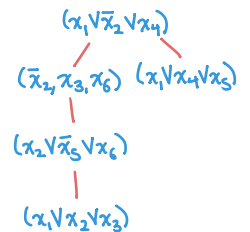
Then $P[\bar{A}_1, \dots, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \dots, \bar{A}_n$ while resampling each A_i at most $\frac{\chi(A_i)}{1 - \chi(A_i)}$ times in expectation

Mapping prefixes of the log $\rightarrow$ witness trees

Suppose first i events are $A_{i1}, A_{i2}, \dots, A_{ij}$.

- ① Make root w/ label A_{ij} .
- ② For $j=i-1, i-2, \dots, 1$
 - a. If $A_{ij} \sim A_k$ for some label A_k in the tree
 - ① pick such a node N of max depth
 - ② make a new child of N w/ label A_{ij}

Log
 $(x_1 \vee x_2 \vee x_3), (x_2 \vee \bar{x}_5 \vee x_6), (x_1, \bar{x}_8, x_9),$
 $(x_1 \vee x_4 \vee x_5), (\bar{x}_2, x_3, x_6), (x_1 \vee \bar{x}_2 \vee x_4)$



check(T)

- ① For each event A_{ij} in T in decr. depth: (label) (leaf first, root last)
 - a. Sample variables in S_{ij} . If the sample does not induce A_{ij} , then return FAIL
 - ② return SUCCESS

we run resampling algo and check(T) using same samples

claim: if log induces T, then check(T) succeeds

V	1st sample	2nd sample	3rd sample	4th sample	5th	6th	7th	
X_1	$X_1^{(1)}$	$X_1^{(2)}$	$X_1^{(3)}$	$X_1^{(4)}$	$X_1^{(5)}$	$X_1^{(6)}$	$X_1^{(7)}$	~
X_2	$X_2^{(1)}$	$X_2^{(2)}$	$X_2^{(3)}$	$X_2^{(4)}$	$X_2^{(5)}$	$X_2^{(6)}$	$X_2^{(7)}$	~
X_3	$X_3^{(1)}$	$X_3^{(2)}$	$X_3^{(3)}$	$X_3^{(4)}$	$X_3^{(5)}$	$X_3^{(6)}$	$X_3^{(7)}$	~
$\vdots$								

Fix node N in T w/ label A_i ,
corresponding to entry E

for $X_j \in S_i$, E uses sample $X_j^{(h)}$ where
 $h = \#$ entries before (and including) E
w/ event using X_j

Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

Theorem: Suppose $\exists \chi(A_i), \sim, \chi(A_n) \in (0,1)$ s.t.

$$P[A_i] \leq \chi(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - \chi(A_j)) \quad \forall A_i.$$

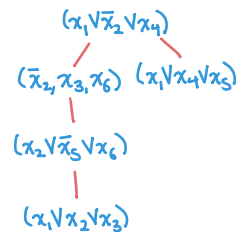
Then $P[\bar{A}_1, \sim, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \sim, \bar{A}_n$ while resampling each A_i at most $\frac{\chi(A_i)}{1 - \chi(A_i)}$ times in expectation

Mapping prefixes of the log $\rightarrow$ witness trees

Suppose first i events are $A_{i1}, A_{i2}, \sim, A_{ii}$.

- ① Make root w/ label A_{ii} .
- ② For $j = i-1, i-2, \dots, 1$
 - a. If $A_{ij} \sim A_k$ for some label A_k in the tree
 - ① pick such a node N of max depth
 - ② make a new child of N w/ label A_{ij}

Log
 $(x_1, x_2, x_3), (x_2, \bar{x}_5, x_6), (x_1, \bar{x}_8, x_9),$
 $(x_1, x_4, x_5), (\bar{x}_2, x_3, x_6), (x_1, \bar{x}_2, x_4)$



check(T)

- ① For each event A_{ij} in T in decr. depth: (label) (leaf first, root last)
 - a. Sample variables in S_{ij} . If the sample does not induce A_{ij} , then return FAIL
 - ② return SUCCESS

Recall:

Suppose first i events are $A_{i_1}, A_{i_2}, \dots, A_{i_i}$.

① Make root w/ label A_{i_i} .

② For $j=i-1, i-2, \dots, 1$

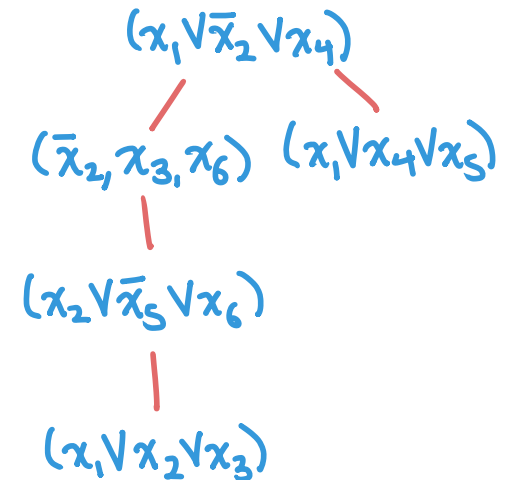
③ If $A_{i_j} \sim A_k$ for some label A_k in the tree

① pick such a node N of max depth

② make a new child of N w/ label A_{i_j}

Log

$(x_1, \bar{x}_2, x_3), (x_2, \bar{x}_5, x_6), (x_7, \bar{x}_8, x_9),$
 $(x_1, x_4, x_5), (\bar{x}_2, x_3, x_6), (x_1, \bar{x}_2, x_4)$

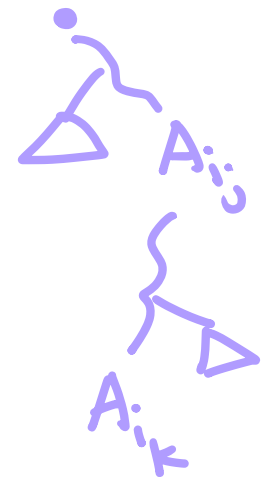


Observation (for later)

Let T_e = witness tree for $A_{i_1}, \dots, A_{i_e}$

Suppose $j < k$ and $A_{i_j} \sim A_{i_k}$

if A_{i_k} is a node in T_e , then so is A_{i_j} ,
 and A_{i_j} has greater depth than A_{i_k}



we run resampling algo and check(T) using same samples

claim: if log induces T, then check(T) succeeds

V	1st sample	2nd sample	3rd sample	4th sample	5th	6th	7th	
X_1	$X_1^{(1)}$	$X_1^{(2)}$	$X_1^{(3)}$	$X_1^{(4)}$	$X_1^{(5)}$	$X_1^{(6)}$	$X_1^{(7)}$	~
X_2	$X_2^{(1)}$	$X_2^{(2)}$	$X_2^{(3)}$	$X_2^{(4)}$	$X_2^{(5)}$	$X_2^{(6)}$	$X_2^{(7)}$	~
X_3	$X_3^{(1)}$	$X_3^{(2)}$	$X_3^{(3)}$	$X_3^{(4)}$	$X_3^{(5)}$	$X_3^{(6)}$	$X_3^{(7)}$	~
$\vdots$								

Fix node N in T w/ label A_j , corresponding to entry E

for $X_j \in S_i$, E uses sample $X_j^{(h)}$ where $h = \# \text{ entries before (and including) E w/ event using } X_j$

Observation $\Rightarrow$ so does N!

$\Rightarrow$ check(T) passes N's iteration

Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

Theorem: Suppose $\exists \chi(A_i), \sim, \chi(A_n) \in (0,1)$ s.t.
 $P[A_i] \leq \chi(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - \chi(A_j)) \quad \forall A_i.$

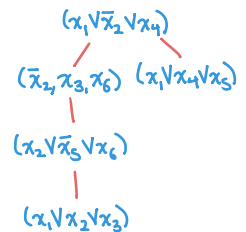
Then $P[\bar{A}_1, \sim, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \sim, \bar{A}_n$ while resampling each A_i at most $\frac{\chi(A_i)}{1 - \chi(A_i)}$ times in expectation

Mapping prefixes of the log $\rightarrow$ witness trees

Suppose first i events are $A_{i1}, A_{i2}, \sim, A_{ii}$.

- ① Make root w/ label A_{ii} .
- ② For $j = i-1, i-2, \dots, 1$
 - a. If $A_{ij} \sim A_k$ for some label A_k in the tree
 - ① pick such a node N of max depth
 - ② make a new child of N w/ label A_{ij}

Log
 $(x_1, x_2, x_3), (x_2, \bar{x}_5, x_6), (x_1, \bar{x}_8, x_9),$
 $(x_1, x_4, x_5), (\bar{x}_2, x_3, x_6), (x_1, \bar{x}_2, x_4)$



check(T)

- ① For each event A_{ij} in T in decr. depth: (leaf first, root last)
 - a. Sample variables in S_{ij} . If the sample does not induce A_{ij} , then return FAIL
 - ② return SUCCESS

claim: $\Pr[T_j = T \text{ for some } j] \leq \Pr[\text{check}(T)]$

Imagine generating all (repeated) samples for all variables ahead of time.

V	1st sample	2nd sample	3rd sample	4th sample	5th	6th	7th	
X_1	$X_1^{(1)}$	$X_1^{(2)}$	$X_1^{(3)}$	$X_1^{(4)}$	$X_1^{(5)}$	$X_1^{(6)}$	$X_1^{(7)}$	~
X_2	$X_2^{(1)}$	$X_2^{(2)}$	$X_2^{(3)}$	$X_2^{(4)}$	$X_2^{(5)}$	$X_2^{(6)}$	$X_2^{(7)}$	~
X_3	$X_3^{(1)}$	$X_3^{(2)}$	$X_3^{(3)}$	$X_4^{(4)}$	$X_5^{(5)}$	$X_6^{(6)}$	$X_7^{(7)}$	~
$\{$								

we run resampling algo and check(T) using same samples

✓ claim: if log induces T, then check(T) succeeds
(hence overall claim)

Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

Theorem: Suppose $\exists \chi(A_i), \sim, \chi(A_n) \in (0,1)$ s.t.
 $\Pr[A_i] \leq \chi(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - \chi(A_j)) \quad \forall A_i.$

Then $\Pr[\bar{A}_1, \sim, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \sim, \bar{A}_n$ while resampling each A_i at most $\frac{\chi(A_i)}{1 - \chi(A_i)}$ times in expectation

Mapping prefixes of the log $\rightarrow$ witness trees

Suppose first i events are $A_{i1}, A_{i2}, \dots, A_{ij}$.

① Make root w/ label A_{ij} .

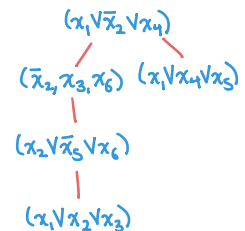
② For $j = i-1, i-2, \dots, 1$

ⓐ If $A_{ij} \sim A_k$ for some label A_k in the tree

① pick such a node N of max depth

ⓐ make a new child of N w/ label A_{ij}

Log
 $(x_1 \vee x_2 \vee x_3), (x_2 \vee \bar{x}_5 \vee x_6), (x_1, \bar{x}_8, x_9),$
 $(x_1 \vee x_4 \vee x_5), (\bar{x}_2, x_3, x_6), (x_1 \vee \bar{x}_2 \vee x_4)$



check(T)

① For each event A_{ij} in T in decr. depth:
 (label) (leaf first, root last)

ⓐ Sample variables in S_{ij} . If the sample does not induce A_{ij} , then return FAIL

② return SUCCESS

Lemma $\Pr[T_j = T \text{ for some } j] \leq p(T) = \prod P[A_j]$
labels A_j in T
(w/ multiplicity)

check(T)

- ① For each event A_{ij} in T in decr. depth:
(label) (leaf first, root last)
 - Ⓐ Sample variables in S_{ij} . If the sample does not induce A_{ij} , then return FAIL
- ② return SUCCESS

$$\Pr[\text{check}(T) \text{ succeeds}] = p(T) = \prod P[A_j]$$

labels A_j in T
(w/ multiplicity)

✓ claim: $\Pr[T_j = T \text{ for some } j] \leq \Pr[\text{check}(T) \text{ succeeds}]$

Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

Theorem: Suppose $\exists \chi(A_i), \sim, \chi(A_n) \in (0,1)$ s.t.
 $P[A_i] \leq \chi(A_i) \prod_{A_j \sim A_i} (1 - \chi(A_j)) \quad \forall A_i$
 $A_j \sim A_i$
 $A_j \neq A_i$

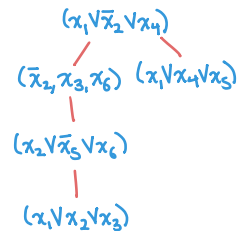
Then $P[\bar{A}_1, \sim, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \sim, \bar{A}_n$ while resampling each A_i at most $\frac{\chi(A_i)}{1 - \chi(A_i)}$ times in expectation

Mapping prefixes of the log $\rightarrow$ witness trees

Suppose first i events are $A_{i1}, A_{i2}, \sim, A_{ii}$.

- ① Make root w/ label A_{ii} .
- ② For $j = i-1, i-2, \dots, 1$
 - Ⓐ If $A_{ij} \sim A_k$ for some label A_k in the tree
 - ① pick such a node N of max depth
 - Ⓐ make a new child of N w/ label A_{ij}

Log
 $(x_1, x_2, x_3), (x_2, \bar{x}_5, x_6), (x_1, \bar{x}_8, x_9),$
 $(x_1, x_4, x_5), (\bar{x}_2, x_3, x_6), (x_1, \bar{x}_2, x_4)$



let $W_i = \{\text{all witness trees w/ root } A_i\}$

for $j=1,2,\dots$, let T_j be witness tree for the first j entries of the log (or $\emptyset$ if log has $< j$ entries)

$$\checkmark E[\text{\# times } A_i \text{ resampled}] = \sum_{T \in W_i} P_r[T_j = T \text{ for some } j]$$

$$\checkmark P_r[T_j = T \text{ for some } j] \leq p(T) = \prod_{\text{labels } A_j \text{ in } T} P[A_j]$$

(w/ multiplicity)

claim: $\sum_{T \in W_i} p(T) \leq \frac{x(A)}{1-x(A)}$

(which completes the proof!)

Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

Theorem: Suppose $\exists x(A_1), \dots, x(A_n) \in (0,1)$ s.t.
 $P[A_i] \leq x(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1-x(A_j)) \quad \forall A_i$

Then $P[\bar{A}_1, \dots, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \dots, \bar{A}_n$ while resampling each A_i at most $\frac{x(A_i)}{1-x(A_i)}$ times in expectation

Mapping prefixes of the log $\rightarrow$ witness trees

Suppose first i events are $A_{i_1}, A_{i_2}, \dots, A_{i_i}$.

① Make root w/ label A_{i_1} .

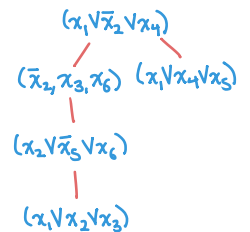
② For $j=i-1, i-2, \dots, 1$

ⓐ If $A_{i_j} \sim A_k$ for some label A_k in the tree

① pick such a node N of max depth

ⓐ make a new child of N w/ label A_{i_j}

Log
 $(x_1 \vee x_2 \vee x_3), (x_2 \vee \bar{x}_5 \vee x_6), (x_1, \bar{x}_8, x_9),$
 $(x_1 \vee x_4 \vee x_5), (\bar{x}_2, x_3, x_6), (x_1 \vee \bar{x}_2 \vee x_4)$



Lemma: $\sum_{T \in \mathcal{W}_i} p(T) \leq \frac{x(A)}{1-x(A)}$ ($p(T) = \prod_{\substack{\text{labels } A_j \text{ in } T \\ (\text{w/ multiplicity})}} P[A_j]$)

Theorem: Suppose $\exists x(A_1), \dots, x(A_n) \in (0,1)$ s.t.
 $P[A_i] \leq x(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1-x(A_j)) \quad \forall A_i.$

Then $P[\bar{A}_1, \dots, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \dots, \bar{A}_n$ while resampling each A_i at most $\frac{x(A_i)}{1-x(A_i)}$ times in expectation

Proof: Galton-Watson trees!

Round 1: Create root w/ label A_i .

Round $n+1$: $\forall$ nodes N created in prev. round, w/ label A_j , and each $A_k \sim A_j$, create child of N w/ label A_k w/ prob. $x(A_k)$

Fix $T \in \mathcal{W}_i$. let $q_T = P[\uparrow \text{ produces } T]$

Lemma: $\sum_{T \in \mathcal{W}_i} p(T) \leq \frac{x(A_i)}{1-x(A_i)}$

$$(p(T) = \prod_{\substack{\text{labels } A_j \text{ in } T \\ (\text{w/ multiplicity})}} P[A_j])$$

Theorem: Suppose $\exists x(A_i), \dots, x(A_n) \in (0,1)$ s.t.
 $P[A_i] \leq x(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1-x(A_j)) \quad \forall A_i$

Then $P[\bar{A}_1, \dots, \bar{A}_n] > 0$, and the resampling algo finds a configuration of $\mathcal{V}$ inducing $\bar{A}_1, \dots, \bar{A}_n$ while resampling each A_i at most $\frac{x(A_i)}{1-x(A_i)}$ times in expectation

Fix $T \in \mathcal{W}_i$. let $q_T = P[\text{produces } T]$

for node N in T , let
 $A_N = \text{event labeling } N$

$\mathcal{B}_N = \{A_k: A_k \sim A_N, \text{ but no child of } A_N \text{ labeled } A_k\}$

Round 1: Create root w/ label A_i .

Round nt+1: $\forall$ nodes N created in prev. round, w/ label A_j , and each $A_k \sim A_j$, create child of N w/ label A_k w/ prob. $x(A_k)$

$$q_T = \frac{1}{x(A_i)} \underbrace{\prod_{N \in T} x(A_N)}_{\text{nodes/events sampled}} \underbrace{\prod_{A_k \in \mathcal{B}_N} (1-x(A_k))}_{\text{nodes not sampled}}$$

$$= \frac{1-x(A_i)}{x(A_i)} \prod_{N \in T} \left(\frac{x(A_N)}{1-x(A_N)} \prod_{A_k \sim A_N} (1-x(A_k)) \right)$$

$$\frac{1 - \chi(A_i)}{\chi(A_i)} \prod_{N \in T} \left(\frac{\chi(A_N)}{\cancel{1 - \chi(A_N)}} \prod_{\substack{A_k \sim A_N \\ A_k \neq A_N}} (1 - \chi(A_k)) \right)$$

$$= \frac{1 - \chi(A_i)}{\chi(A_i)} \prod_{N \in T} \frac{\chi(A_N)}{\cancel{1 - \chi(A_N)}} \prod_{\substack{A_k \sim A_N \\ A_k \neq A_N}} (1 - \chi(A_k))$$

For $T \in \mathcal{W}_i$:

$$q_T = \frac{1 - \chi(A_i)}{\chi(A_i)} \prod_{N \in T} \chi(A_N) \prod_{\substack{A_j \sim A_N \\ A_j \neq A_N}} (1 - \chi(A_j))$$

$$P(T) = \prod_{\substack{\text{labels } A_j \text{ in } T \\ (\text{w/ multiplicity})}} P[A_j]$$

$$\leq \prod_{N \in T} \chi(A_N) \prod_{\substack{A_j \sim A_N \\ A_j \neq A_N}} (1 - \chi(A_j))$$

$$= \frac{\chi(A_i)}{1 - \chi(A_i)} q_T$$

Theorem: Suppose $\exists \chi(A_1), \dots, \chi(A_n) \in (0, 1)$ s.t.
 $P[A_i] \leq \chi(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - \chi(A_j)) \quad \forall A_i.$

Then $P[\bar{A}_1, \dots, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \dots, \bar{A}_n$ while resampling each A_i at most $\frac{\chi(A_i)}{1 - \chi(A_i)}$ times in expectation

Round 1: Create root w/ label A_i .

Round nt+1: $\forall$ nodes N created in prev. round, w/ label A_j , and each $A_k \sim A_j$, create child of N w/ label A_k w/ prob. $\chi(A_k)$

$$\checkmark \quad p(T) \leq \frac{\chi(A_i)}{1-\chi(A_i)} q_T$$

$$\sum_{T \in W_i} p(T) \leq \sum_{T \in W_i} \frac{\chi(A_i)}{1-\chi(A_i)} q_T$$

$$\leq \frac{\chi(A_i)}{1-\chi(A_i)}$$

$$(b/c \sum_{T \in W_i} q_T \leq 1 !)$$

Theorem: Suppose $\exists \chi(A_1), \dots, \chi(A_n) \in (0,1)$ s.t.
 $P[A_i] \leq \chi(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1-\chi(A_j)) \quad \forall A_i.$

Then $P[\bar{A}_1, \dots, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \dots, \bar{A}_n$ while resampling each A_i at most $\frac{\chi(A_i)}{1-\chi(A_i)}$ times in expectation

Round 1: Create root w/ label A_i .

Round $n+1$: $\forall$ nodes N created in prev. round, w/ label A_j , and each $A_k \sim A_j$, create child of N w/ label A_k w/ prob. $\chi(A_k)$

let $W_i = \{\text{all witness trees w/ root } A_i\}$

for $j=1,2,\dots$, let T_j be witness tree for the first j entries of the log (or $\emptyset$ if log has $< j$ entries)

$$\checkmark E[\text{\# times } A_i \text{ resampled}] = \sum_{T \in W_i} P_r[T_j = T \text{ for some } j]$$

$$\checkmark P_r[T_j = T \text{ for some } j] \leq p(T) = \prod_{\substack{\text{labels } A_j \text{ in } T \\ (\text{w/ multiplicity})}} P[A_j]$$

$$\checkmark \text{claim: } \sum_{T \in W_i} p(T) \leq \frac{x(A)}{1-x(A)}$$

(which completes the proof!)

Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

Theorem: Suppose $\exists x(A_1), \dots, x(A_n) \in (0,1)$ s.t.
 $P[A_i] \leq x(A_i) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1-x(A_j)) \quad \forall A_i$

Then $P[\bar{A}_1, \dots, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \dots, \bar{A}_n$ while resampling each A_i at most $\frac{x(A_i)}{1-x(A_i)}$ times in expectation

Mapping prefixes of the log $\rightarrow$ witness trees

Suppose first i events are $A_{i_1}, A_{i_2}, \dots, A_{i_i}$.

① Make root w/ label A_{i_1} .

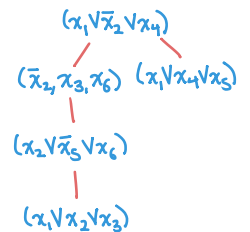
② For $j=i-1, i-2, \dots, 1$

ⓐ If $A_{i_j} \sim A_k$ for some label A_k in the tree

① pick such a node N of max depth

ⓐ make a new child of N w/ label A_{i_j}

Log
 $(x_1 \vee x_2 \vee x_3), (x_2 \vee \bar{x}_5 \vee x_6), (x_1, \bar{x}_8, x_9),$
 $(x_1 \vee x_4 \vee x_5), (\bar{x}_2, x_3, x_6), (x_1 \vee \bar{x}_2 \vee x_4)$



Resampling algorithm (for general LL) (Moser-Tardos, 2010)

1. Sample all variables V (independently, from their respective distributions)
2. while there is an event A_i :
 - a. resample all of S_i (independently, from their respective distributions)

Theorem: Suppose $\exists x(A_1), \dots, x(A_n) \in (0,1)$ s.t.

$$P[A_1] \leq x(A_1) \prod_{\substack{A_j \sim A_i \\ A_j \neq A_i}} (1 - x(A_j)) \quad \forall A_i.$$

Annotations:
- $\frac{1}{2^k}$ (with a crossed-out $\frac{1}{2}$)
- $\perp$ (under the product)
- A yellow bracket under the product with labels $A_j \sim A_i$ and $A_j \neq A_i$

Then $P[\bar{A}_1, \dots, \bar{A}_n] > 0$, and the resampling algo finds a configuration of V inducing $\bar{A}_1, \dots, \bar{A}_n$ while resampling each A_i at most $\frac{x(A_i)}{1-x(A_i)}$ times in expectation