Randomized Algorithms

南京大学

尹一通

Constraint Satisfaction Problem

- variables: $x_1, x_2, \ldots, x_n \in D$ (domain)
- constraints: $C_1, C_2, \ldots, C_m$
  - where $C_i(x_{i_1}, x_{i_2}, \ldots) \in \{\text{true}, \text{false}\}$
- CSP solution: an assignment of variables satisfying *all* constraints
- examples: SAT, graph colorability, ...
- *existence*: When does a solution exist?
- *search*: How to find a solution?

The Probabilistic Method

CSP $C_1, C_2, \ldots, C_m$ defined on $x_1, x_2, \ldots, x_n$

- sampling random values of $x_1, x_2, \ldots, x_n$
- Bad event A_i: constraint C_i is violated
- None of the bad events occurs with prob: $\Pr\left[\bigwedge_{i}^{m} \overline{A_i}\right]$
- The probabilistic method: being good is possible

$$\Pr\left[\bigwedge_{i=1}^{m} \overline{A_i}\right] > 0$$

Dependency Graph

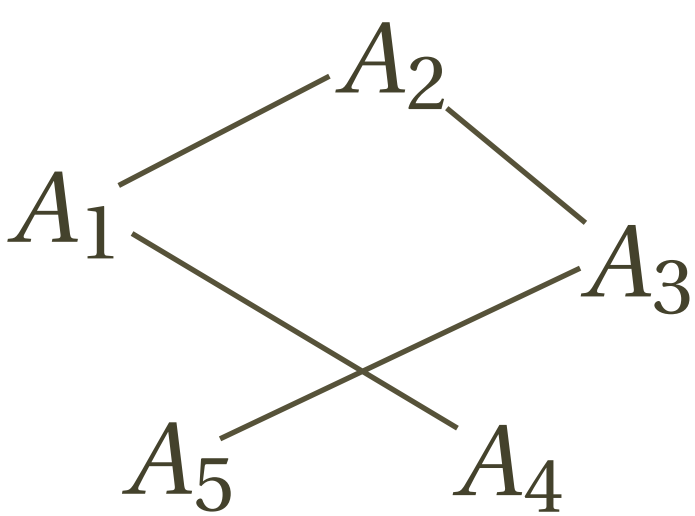
events: $A_1, A_2, \ldots, A_m$

dependency graph: $D(V,E)$

$$V = \{\ 1, 2, \ldots, m\ \}$$

$ij \in E$ ⇔ A_i and A_j are *dependent*

d : max degree of dependency graph



$A_1(X_1, X_4)$ $\quad A_4(X_4)$

$A_2(X_1, X_2)$ $\quad A_5(X_3)$

$A_3(X_2, X_3)$

$X_1, \ldots, X_4$ mutually independent

events: $A_1, A_2, \ldots, A_m$

each event is independent of all but at most d other events

Lovász Local Lemma (symmetric)

- $\forall i,\ \Pr[A_i] \le p$
- $\mathrm{e}p(d+1) \le 1$

$$\Longrightarrow \Pr\left[\bigwedge_{i=1}^{m} \overline{A_i}\right] > 0$$

Lovász Local Lemma (general)

$\exists \alpha_1, \ldots, \alpha_m \in [0,1)$

$\forall i, \Pr[A_i] \le \alpha_i \prod_{j \sim i} (1-\alpha_j)$

$$\Longrightarrow \Pr\left[\bigwedge_{i=1}^{m} \overline{A_i}\right] \ge \prod_{i=1}^{m} (1-\alpha_i)$$

k-SAT

- n Boolean variables: $x_1, x_2, \ldots, x_n \in \{\text{true}, \text{false}\}$
- conjunctive normal form:

  k-CNF $\quad \phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$

  "Is ϕ satisfiable?"

- m clauses: $C_1, C_2, \ldots, C_m$
- each clause $C_i = \ell_{i_1} \vee \ell_{i_2} \vee \cdots \vee \ell_{i_k}$

  is a disjunction of k *distinct* literals

- each literal: $\ell_j \in \{x_r, \neg x_r\}$ for some r
- degree d : each clause shares variables

  with at most d other clauses

LLL for k-SAT

ϕ : k-CNF of max degree d

Theorem

$d \leq 2^{k-2}$ ⇒ $\exists$ satisfying assignment for ϕ

uniform random assignment $X_1, X_2, \ldots, X_n$

for clause C_i , bad event A_i : C_i is not satisfied

LLL: $e(d+1) \leq 2^k$ ⇒ $\Pr\left[\bigwedge_{i=1}^{n} \overline{A_i}\right] > 0$

Algorithmic *LLL*

ϕ : k-CNF of max degree d with m clauses on n variables

Theorem

$d \leq 2^{k-2}$ ⇒ $\exists$ satisfying assignment for ϕ

Theorem (Moser, 2009)

$d < 2^{k-3}$ ⇒ satisfying assignment can be found in $O(n + km \log m)$ *w.h.p.*

ϕ : k-CNF of max degree d with m clauses on n variables

Solve(ϕ)

pick a random assignment

$x_1, x_2, \ldots, x_n$;

while $\exists$ unsatisfied clause C

Fix(C);

Fix(C)

replace variables in C with random values;

while $\exists$ unsatisfied clause D overlapping with C

Fix(D);

ϕ : k-CNF of max degree d with m clauses on n variables

Solve(ϕ)
Pick a random assignment $x_1, x_2, \ldots, x_n$; while $\exists$ unsatisfied clause C **Fix(C)**;

Fix(C)
replace variables in C with random values; while $\exists$ unsatisfied clause D overlapping with C **Fix(D)**;

at top-level:

Observation: A clause C is satisfied and will keep satisfied once it has been fixed.

# of top-level calls to Fix(C) : $\leq m$ (# of clauses)

total # of calls to Fix(C) (including recursive calls): t

ϕ : k-CNF of max degree d with m clauses on n variables

Solve(ϕ)

Pick a random assignment

$x_1, x_2, \ldots, x_n;$

while ∃ unsatisfied clause C

Fix(C);

Fix(C)

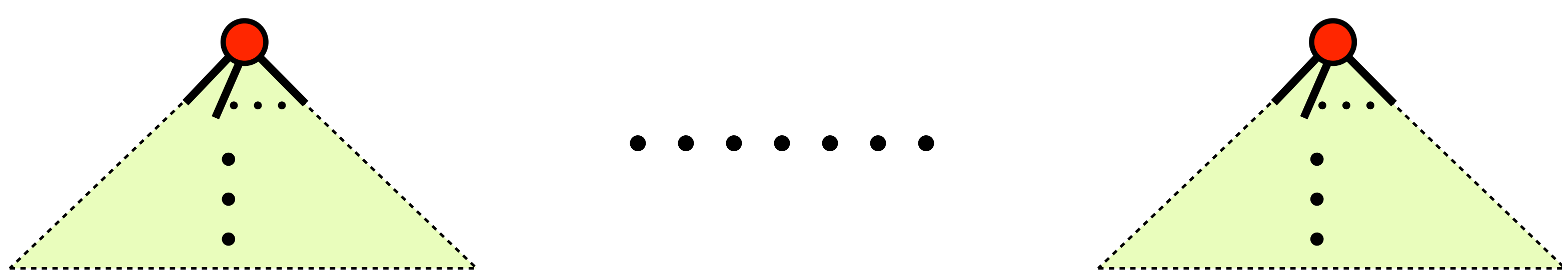
replace variables in C with random values;

while ∃ unsatisfied clause D overlapping with C

Fix(D);

$\leq m$ recursion trees

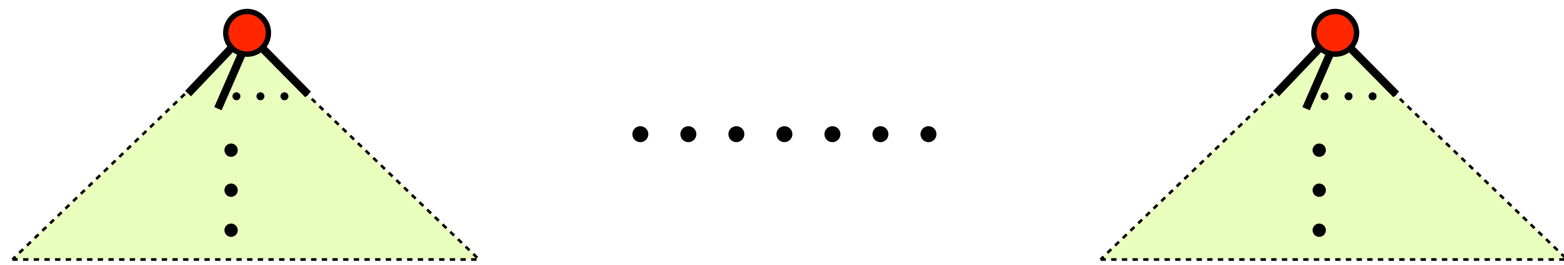
total # nodes: t



total # of random bits: $n+tk$ (assigned bits)

Observation: Fix(C) is called ⇒ assignment of C is uniquely determined

$\leq m$ recursion trees total # nodes: t



total # of random bits: $n+tk$ (assigned bits)

the sequence of random bits is *encoded to* :

final assignment: n bits

+

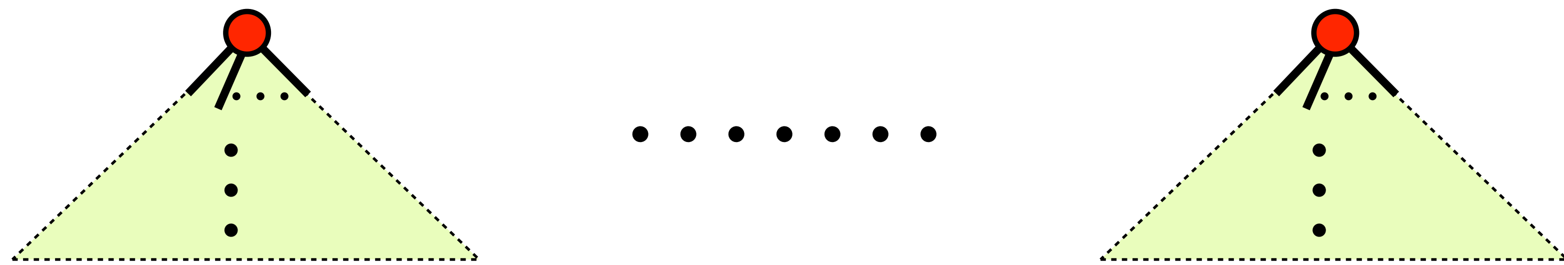
recursion trees: $\leq m\lceil \log_2 m \rceil + t(\log_2 d + 3)$ bits

for each recursion tree:

root: $\lceil \log_2 m \rceil$ bits

each internal node: $\leq \log_2 d + 3\,O(1)$ bits

$\leq m$ recursion trees total # nodes: t



total # of random bits: $n+tk$ (assigned bits)

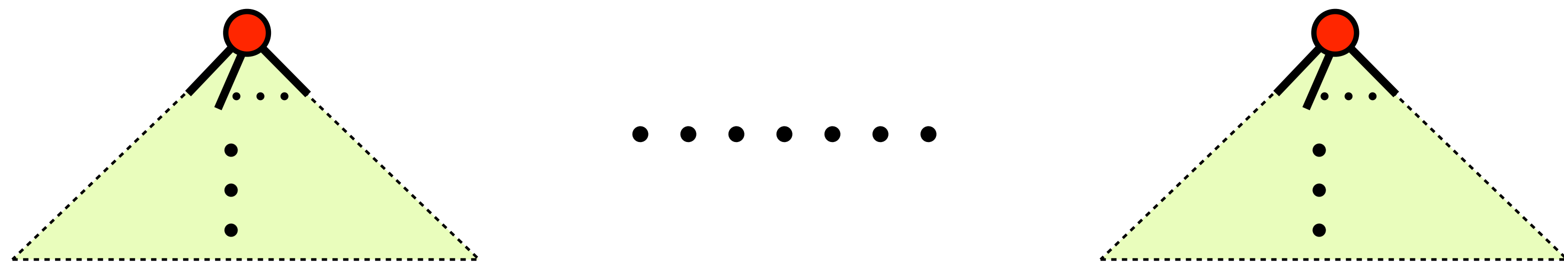
the sequence of random bits is *encoded to* :

$$\leq n + m\lceil \log_2 m \rceil + t(\log_2 d + 3) \text{ bits}$$

Incompressibility Theorem (Kolmogorov)

N uniform random bits cannot be encoded to less than N - l bits with probability $1\text{-}O(2^{-l})$.

$\leq m$ recursion trees total # nodes: t



total # of random bits: $n+tk$ (assigned bits)

the sequence of random bits is *encoded to* :

$$\leq n + m\lceil \log_2 m \rceil + t(\log_2 d + 3) \text{ bits}$$

$\Rightarrow$ $t(k - 3 - \log_2 d) \leq m\lceil \log_2 m \rceil + \log n$ whp

when $d < 2^{k-3}$ $\Rightarrow$ $t \leq \dfrac{m\lceil \log_2 m \rceil + \log n}{k - 3 - \log_2 d}$

total running time: $n+tk = O(n + km \log m)$

Algorithmic LLL

ϕ : k-CNF of max degree d with m clauses on n variables

$$\phi = C_1 \wedge C_2 \wedge \cdots \wedge C_m$$

Theorem (Moser, 2009)

$d < 2^{k-3}$ ⇒ satisfying assignment can be found in O($n + km \log m$) *whp*

Solve(ϕ)

Pick a random assignment

$x_1, x_2, \ldots, x_n$;

while ∃ unsatisfied clause C

Fix(C);

Fix(C)

replace variables in C with random values;

while ∃ unsatisfied clause D overlapping with C

Fix(D);

events: $A_1, A_2, \ldots, A_m$

each event is independent of all but at most d other events

Lovász Local Lemma (symmetric)

- $\forall i, \ \Pr[A_i] \le p$
- $ep(d+1) \le 1$

$\Rightarrow$ $$\Pr\left[\bigwedge_{i=1}^{m} \overline{A_i}\right] > 0$$

Lovász Local Lemma (general)

$\exists \alpha_1, \ldots, \alpha_m \in [0,1)$

$\forall i, \Pr[A_i] \le \alpha_i \prod_{j \sim i} (1 - \alpha_j)$

$\Rightarrow$ $$\Pr\left[\bigwedge_{i=1}^{m} \overline{A_i}\right] \ge \prod_{i=1}^{m} (1 - \alpha_i)$$

mutually independent random variables: $X \in \mathcal{X}$

bad events: $A \in \mathcal{A}$ defined on variables in $\mathcal{X}$

vbl$(A) \subseteq \mathcal{X}$: set of variables on which A is defined

neighborhood: $\Gamma(A) = \{\, B \in \mathcal{A} \mid B \neq A$ and vbl$(A) \cap$ vbl$(B) \neq \emptyset \,\}$

inclusive neighborhood: $\Gamma^+(A) = \Gamma(A) \cup \{\, A \,\}$

"events that are dependent with A, excluding/including A itself"

Lovász Local Lemma (general)

$\exists \alpha : \mathcal{A} \to [0,1)$
$\forall A \in \mathcal{A}$:
$$\Pr[A] \leq \alpha(A) \prod_{B \in \Gamma(A)} (1 - \alpha(B))$$

⇒

$$\Pr\left[\bigwedge_{A \in \mathcal{A}} \overline{A}\right] \geq \prod_{A \in \mathcal{A}} (1 - \alpha(A))$$
$$> 0$$

mutually independent random variables: $X \in \mathcal{X}$

bad events: $A \in \mathcal{A}$ defined on variables in $\mathcal{X}$

vbl$(A) \subseteq \mathcal{X}$: set of variables on which A is defined

neighborhood: $\Gamma(A) = \{ B \in \mathcal{A} \mid B \neq A \text{ and } \mathrm{vbl}(A) \cap \mathrm{vbl}(B) \neq \emptyset \}$

inclusive neighborhood: $\Gamma^+(A) = \Gamma(A) \cup \{ A \}$

"events that are dependent with A, excluding/including A itself"

Lovász Local Lemma (general)

$\exists \alpha : \mathcal{A} \to [0,1)$
$\forall A \in \mathcal{A}:$

$$\Pr[A] \leq \alpha(A) \prod_{B \in \Gamma(A)} (1 - \alpha(B))$$

⇒

$\exists$ values of variables in $\mathcal{X}$ violating all events $A \in \mathcal{A}$ simultaneously.

Algorithmic LLL

bad events $A \in \mathcal{A}$ defined on

mutually independent random variables $X \in \mathcal{X}$

vbl(A): set of variables on which A is defined

neighborhood $\Gamma(A)$ and *inclusive* neighborhood $\Gamma^+(A)$

Assumption:

I. We can efficiently sample an independent evaluation of every random variable $X \in \mathcal{X}$.

II. We can efficiently check the violation of every event $A \in \mathcal{A}$.

RandomSolver:

sample all $X \in \mathcal{X}$;

while ∃ a non-violated bad event $A \in \mathcal{A}$:

  resample all $X \in$ vbl(A);

bad events $A \in \mathcal{A}$ defined on

mutually independent random variables $X \in \mathcal{X}$

vbl(A): set of variables on which A is defined

neighborhood $\Gamma(A)$ and *inclusive* neighborhood $\Gamma^+(A)$

RandomSolver:
sample all $X \in \mathcal{X}$;
while ∃ a non-violated bad event $A \in \mathcal{A}$:
    resample all $X \in$ vbl(A);

Moser-Tardos 2010:

$\exists \alpha : \mathcal{A} \to [0,1)$
$\forall A \in \mathcal{A}$:

$$\Pr[A] \le \alpha(A) \prod_{B \in \Gamma(A)} (1 - \alpha(B))$$

⇒ *RandomSolver* finds values of all $X \in \mathcal{X}$ violating all $A \in \mathcal{A}$ within expected $\sum_{A \in \mathcal{A}} \frac{\alpha(A)}{1 - \alpha(A)}$ resamples.

bad events $A \in \mathcal{A}$ defined on

mutually independent random variables $X \in \mathcal{X}$

vbl(A): set of variables on which A is defined

neighborhood $\Gamma(A)$ and *inclusive* neighborhood $\Gamma^+(A)$

RandomSolver:

sample all $X \in \mathcal{X}$;

while $\exists$ a non-violated bad event $A \in \mathcal{A}$:

  resample all $X \in$ vbl(A);

Moser-Tardos 2010:

- $\forall A \in \mathcal{A}$, $\Pr[A] \le p$
- $ep(d+1) \le 1$
  where $d = \max_A |\Gamma(A)|$

RandomSolver finds values of all $X \in \mathcal{X}$ violating all $A \in \mathcal{A}$ within expected $|\mathcal{A}|/d$ resamples.

k-SAT

ϕ : k-CNF of max degree d with m clauses on n variables

RandomSolver:
pick a random assignment $x_1, x_2, \ldots, x_n$;
while $\exists$ an unsatisfied clause C:
    replace variables in C with random values;

$d \leq 2^{k-2}$
($e(d+1) \leq 2^k$)

$\Rightarrow$

RandomSolver returns a satisfying assignment within expected $O(n + km/d)$ time

bad events $A \in \mathcal{A}$ defined on

mutually independent random variables $X \in \mathcal{X}$

vbl(A): set of variables on which A is defined

neighborhood $\Gamma(A)$ and *inclusive* neighborhood $\Gamma^{+}(A)$

RandomSolver:

sample all $X \in \mathcal{X}$;

while ∃ a non-violated bad event $A \in \mathcal{A}$:

resample all $X \in$ vbl(A);

Moser-Tardos 2010:

$\exists \alpha : \mathcal{A} \to [0,1)$

$\forall A \in \mathcal{A}$:

$$\Pr[A] \le \alpha(A) \prod_{B \in \Gamma(A)} (1 - \alpha(B))$$

RandomSolver finds values of all $X \in \mathcal{X}$ violating all $A \in \mathcal{A}$ within expected $\sum_{A \in \mathcal{A}} \frac{\alpha(A)}{1-\alpha(A)}$ resamples.

RandomSolver:

sample all $X \in \mathcal{X}$;

while $\exists$ a non-violated $A \in \mathcal{A}$:

resample all $X \in \mathsf{vbl}(A)$;

execution log Λ:

$$\Lambda_1, \Lambda_2, \Lambda_3, \ldots \in \mathcal{A}$$

random sequence of resampled events

$$N_A = |\{\, i \mid \Lambda_i = A \,\}|$$

total # of times of A is resampled

Moser-Tardos 2010:

$\exists \alpha : \mathcal{A} \to [0,1)$

$\forall A \in \mathcal{A}$:

$$\Pr[A] \le \alpha(A) \prod_{B \in \Gamma(A)} (1 - \alpha(B))$$

$\Rightarrow$

$\forall A \in \mathcal{A}$:

$$\mathbb{E}[N_A] \le \frac{\alpha(A)}{1 - \alpha(A)}$$

RandomSolver:

sample all $X \in \mathcal{X}$;

while $\exists$ a non-violated $A \in \mathcal{A}$:

resample all $X \in \text{vbl}(A)$;

execution log Λ:

$\Lambda_1, \Lambda_2, \Lambda_3, \ldots \in \mathcal{A}$

random sequence of resampled events

witness tree: A witness tree τ is a labeled tree in which every vertex v is labeled by an event $A_v \in \mathcal{A}$, such that *siblings* have distinct labels.

$T(\Lambda, t)$ is a witness tree constructed from exe-log Λ:

- initially, T is a single root with label Λ_t
- for $i = t-1, t-2, \ldots, 1$
  - if $\exists$ a vertex v in T with label $A_v \in \Gamma^+(\Lambda_i)$
    - add a new child u to the deepest such v and label it with Λ_i
- $T(\Lambda, t)$ is the resulting T

$T(\Lambda, s) \neq T(\Lambda, t)$ for $s \neq t$

$\mathcal{T}_A$: set of all witness trees with root-label A

$$\Rightarrow \quad \mathbf{E}[N_A] = \sum_{\tau \in \mathcal{T}_A} \Pr[\exists t, T(\Lambda, t) = \tau]$$

RandomSolver:

sample all $X \in \mathcal{X}$;

while $\exists$ a non-violated $A \in \mathcal{A}$:

resample all $X \in$ vbl(A);

execution log Λ:

$$\Lambda_1, \Lambda_2, \Lambda_3, \ldots \in \mathcal{A}$$

random sequence of resampled events

LLL hypothesis: $\exists \alpha : \mathcal{A} \to [0,1)$

$$\forall A \in \mathcal{A} : \quad \Pr[A] \le \alpha(A) \prod_{B \in \Gamma(A)} (1 - \alpha(B))$$

total # of times of A is resampled

$$N_A = |\{\, i \mid \Lambda_i = A \,\}|$$

$$\mathbf{E}[N_A] = \sum_{\tau \in \mathcal{T}_A} \Pr[\exists t, T(\Lambda, t) = \tau]$$

(lemma 1) $$\le \sum_{\tau \in \mathcal{T}_A} \prod_{v \in \tau} \Pr[A_v]$$

(hypothesis of LLL) $$\le \sum_{\tau \in \mathcal{T}_A} \prod_{v \in \tau} \left[\alpha(A_v) \prod_{B \in \Gamma(A_v)} (1 - \alpha(B)) \right]$$

(lemma 2) $$\le \frac{\alpha(A)}{1 - \alpha(A)}$$

RandomSolver:

sample all $X \in \mathcal{X}$;

while $\exists$ a non-violated $A \in \mathcal{A}$:

resample all $X \in \text{vbl}(A)$;

execution log Λ:

$$\Lambda_1, \Lambda_2, \Lambda_3, \ldots \in \mathcal{A}$$

random sequence of resampled events

$T(\Lambda, t)$ is a witness tree constructed from exe-log Λ:

- initially, T is a single root with label Λ_t
- for $i = t\text{-}1, t\text{-}2, \ldots, 1$
  - if $\exists$ a vertex v in T with label $A_v \in \Gamma^+(\Lambda_i)$
    - add a new child u to the deepest such v and label it with Λ_i
- $T(\Lambda, t)$ is the resulting T

Lemma 1 For any particular witness tree τ:

$$\Pr[\exists t, T(\Lambda, t) = \tau] \leq \prod_{v \in \tau} \Pr[A_v]$$

grow a random witness tree $T_A \in \mathcal{T}_A$:

- initially, T_A is a single root with label A
- for $i = 1, 2, \ldots$
  - for every vertex v at depth i (root has depth 1) in T_A
  - for every $B \in \Gamma^+(A_v)$:
    - add a new child u to v independently with probability $\alpha(B)$;
    - and label it with B;
- stop if no new child added for an entire level

Lemma 2 For any particular witness tree $\tau \in \mathcal{T}_A$:

$$\Pr[T_A = \tau] = \frac{1 - \alpha(A)}{\alpha(A)} \prod_{v \in \tau} \left[\alpha(A_v) \prod_{B \in \Gamma(A_v)} (1 - \alpha(B)) \right]$$

RandomSolver:

sample all $X \in \mathcal{X}$;

while $\exists$ a non-violated $A \in \mathcal{A}$:

resample all $X \in \text{vbl}(A)$;

execution log Λ:

$$\Lambda_1, \Lambda_2, \Lambda_3, \ldots \in \mathcal{A}$$

random sequence of resampled events

LLL hypothesis: $\exists \alpha : \mathcal{A} \to [0,1)$

$\forall A \in \mathcal{A} : \quad \Pr[A] \le \alpha(A) \prod_{B \in \Gamma(A)} (1 - \alpha(B))$

total # of times of A is resampled

$$N_A = |\{ i \mid \Lambda_i = A \}|$$

$$\mathbf{E}[N_A] = \sum_{\tau \in \mathcal{T}_A} \Pr[\exists t, T(\Lambda, t) = \tau]$$

(lemma 1) $$\le \sum_{\tau \in \mathcal{T}_A} \prod_{v \in \tau} \Pr[A_v]$$

(hypothesis of LLL) $$\le \sum_{\tau \in \mathcal{T}_A} \prod_{v \in \tau} \left[\alpha(A_v) \prod_{B \in \Gamma(A_v)} (1 - \alpha(B)) \right]$$

(lemma 2) $$\le \frac{\alpha(A)}{1 - \alpha(A)} \sum_{\tau \in \mathcal{T}_A} \Pr[T_A = \tau] \quad \le \frac{\alpha(A)}{1 - \alpha(A)}$$

bad events $A \in \mathcal{A}$ defined on

mutually independent random variables $X \in \mathcal{X}$

vbl(A): set of variables on which A is defined

neighborhood $\Gamma(A)$ and *inclusive* neighborhood $\Gamma^+(A)$

RandomSolver:
sample all $X \in \mathcal{X}$;
while ∃ a non-violated bad event $A \in \mathcal{A}$:
    resample all $X \in$ vbl(A);

Moser-Tardos 2010:

$\exists \alpha : \mathcal{A} \to [0,1)$
$\forall A \in \mathcal{A}:$
$$\Pr[A] \le \alpha(A) \prod_{B \in \Gamma(A)} (1 - \alpha(B))$$

⇒ *RandomSolver* finds values of all $X \in \mathcal{X}$ violating all $A \in \mathcal{A}$ within expected $\sum_{A \in \mathcal{A}} \frac{\alpha(A)}{1 - \alpha(A)}$ resamples.