Randomness in Computing

Lecture 23

Last time
• Probabilistic method
  • Lovasz Local Lemma (LLL)
  • Algorithmic LLL

Today
• Probabilistic method
  • Algorithmic LLL
  • Applications of LLL
Algorithmic Lovasz Local Lemma for $k$SAT

If $d \leq 2^{k-3} = \frac{2^k}{8}$ for some $k$-CNF formula $\phi$, then $\phi$ is satisfiable.

Moreover, a satisfying assignment can be found in $O(m^2 \log m)$ time with probability at least $1 - 2^{-m}$. 

*Sofya Raskhodnikova; Randomness in Computing*
Moser-Tardos Algorithm for LLL

**Input:** a $k$CNF formula with clauses $C_1, \ldots, C_m$ on $n$ variables and with $d \leq 2^{k-3}$

1. Let $R$ be a random assignment where each variable is assigned 0 or 1 uniformly and independently.
2. While some clause $C$ is violated by $R$, run $\text{FIX}(C)$

**$\text{FIX}(C)$**

1. Pick new values for $k$ variables in $C$ uniformly and independently and update $R$.
2. While some clause $D$ that shares a variable with $C$ is violated by $R$, run $\text{FIX}(D)$

*D could be $C$ if we chose the same values as before*
Theorem (Correctness)

If Moser-Tardos terminates, it outputs a satisfying assignment.
Run time of Moser-Tardos

- **Assume:** $m \geq 2^k$ (o.w. trivial by other means)

**Theorem (Run time)**

If $d \leq 2^{k-3}$ then Moser-Tardos terminates after $O(m \log m)$ resampling steps with probability at least $1 - 2^{-m}$.

- **Proof idea:** Clever way to ``compress”” random bits if the algorithm runs for too long.

**Observation 2**

If a function $f : A \rightarrow B$ is injective (i.e., invertible on its range $f(A)$) then $|B| \geq |A|$.
• Suppose we stop Moser-Tardos after $T$ resampling steps.

Randomness used:

- $n$ bits for initial assignment
- $k$ bits for each resampling step

Total: $n + Tk$ bits

• Let $A$ be the set of all choices for $n + Tk$ bits

$$f_T((x_0, y_0)) = (x_T, z_T)$$
• Each call to FIX gets recorded as follows:
  If FIX(C) is called by the main algorithm
  1

  If FIX(D) is a recursive call made by FIX(C)
  1

• When a call to FIX returns,
  0 is written on the transcript
### Lemma 1

Function $f_T$ is invertible on all inputs $(x_0, y_0)$ for which Moser-Tardos does not terminate within $T$ steps when run with randomness $(x_0, y_0)$.

### Lemma 2

Length of transcript $z_T$ is at most $m(\lfloor \log_2 m \rfloor + 2) + T \cdot (k - 1)$.
Proof of Theorem

First, consider $T$ such that Moser-Tardos never terminates within $T$ resampling steps.

- There is a valid transcript $z_T$ for every choice of the random $n + Tk$ bits needed to run Moser-Tardos.
Now, consider $T$ such that Moser-Tardos fails to terminate w.p. $\geq \frac{1}{2^m}$ within $T$ resampling steps.

- Then $f_T$ is invertible on the set of size $\geq 2^{n+Tk-m}$
Proof of Lemma 1

**Lemma 1**

Function $f_T$ is invertible on all inputs $(x_0, y_0)$ for which Moser-Tardos does not terminate within $T$ steps when run with randomness $(x_0, y_0)$. 
### Algorithmic Lovasz Local Lemma for $k$SAT

<table>
<thead>
<tr>
<th>Algorithmic Lovasz Local Lemma for $k$SAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>If $d \leq 2^{k-3} = \frac{2^k}{8}$ for some $k$ CNF formula $\phi$, then $\phi$ is satisfiable.</td>
</tr>
<tr>
<td>Moreover, a satisfying assignment can be found in $O(m^2 \log m)$ time with probability at least $1 - 2^{-m}$.</td>
</tr>
</tbody>
</table>
Lovasz Local Lemma (LLL)

- Event $E$ is mutually independent from the events $E_1, ..., E_n$ if, for any subset $I \subseteq [n]$,
  \[
  \Pr[E \mid \bigcap_{j \in I} E_j] = \Pr[E].
  \]
- A dependency graph for events $B_1, ..., B_n$ is a graph with vertex set $[n]$ and edge set $E$, s.t. $\forall i \in [n]$, event $B_i$ is mutually independent of all events $\{B_j \mid (i, j) \notin E\}$.

**Lovasz Local Lemma**

Let $B_1, ..., B_n$ be events over a common sample space s.t.
1. max degree of the dependency graph of $B_1, ..., B_n$ is at most $d$
2. $\forall i \in [n], \Pr[B_i] \leq p$

If $ep(d + 1) \leq 1$ then $\Pr[\bigcap_{i \in [n]} \overline{B_i}] > 0$
Theorem

If \( e \left( \binom{k}{2} \binom{n}{k-2} + 1 \right) 2^{1 - \binom{k}{2}} \leq 1 \) then edges of \( K_n \) can be colored with 2 colors so that there is no monochromatic \( K_k \).

Proof:
Application 2: edge-disjoint paths

- \( n \) pairs of users need to communicate using edge-disjoint paths
- \( \forall i \in [n] \), pair \( i \) can choose a path from collection \( P_i \) of size \( m \).

**Theorem**

If \( \forall i \neq j \), each path in \( P_i \) shares edges with at most \( k \) paths in \( P_j \) and \( 2enk \leq m \) then there is a way to choose \( n \) edge-disjoint paths.

**Proof:**