Topics in Theoretical Computer Science February 23, 2015

Lecture 2

Lecturer: Ola Svensson *Scribes: Pierre Sarton, Niklas Gehlen, Tobias Räderscheidt*

1 Last Lecture

In the previous lecture, we introduced the *probabilistic method*, which allowed us to prove the existence of a combinatorial object with specified properties. In general, the argument goes as follows: select an object at random from the set, and calculate the probability that it satisfies the required property; if this probability is strictly positive then such an object must exist. However this approach is not powerful enough when the events we are considering are not independent.

2 Lovász Local Lemma

Definition 1 *An event A is mutually independent from a set $\{B_i\}$ of events if for every subset β of events or their complements contained in $\{B_i\}$*

$$\Pr A \mid \beta = \Pr A$$

In a general setting, we have a set of n *bad* events $\{A_i\}$ that we are trying to avoid, such that $Pr\left[A_i\right] \leq p < 1$, for $i \in \{1, 2, \ldots, n\}$. If we assume that these events are independent, then their complements are independent as well and we can show that

$$\Pr \bigwedge_i \bar{A}_i \geq (1-p)^n > 0$$

However, if we remove the independence assumption, the union bound yields

$$\Pr \bigwedge_i \bar{A}_i \geq 1 - \sum_i \Pr \bar{A}_i$$

The *Lovász Local Lemma* improves upon the union bound in the case where the events are not *mutually independent*, but their dependencies are restricted. It was proved by Erdös and Lovász in 1975 [1].

Theorem 2 *(Lovász Local Lemma) Let $A_1, A_2, \ldots, A_n$ be a set of "bad" events with $\Pr A_i \leq p < 1$, and each A_i is dependent on at most d other A_j. If $p \cdot (d+1) \cdot e \leq 1$, then*

$$\Pr \bigwedge_{i=1}^{n} \bar{A}_i > 0$$

Before proving the theorem, we give an example showing that the bound is almost tight. Consider events $A_1, \ldots, A_{d+1}$, each happening with probability $\frac{1}{d+1}$. We assume that in each outcome exactly one bad event happens. Consider figure 1. In that case we have $p \cdot (d+1) = 1$ and $\Pr \bigwedge_{i=1}^{n} \bar{A}_i = 0$.

A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$	$\frac{1}{10}$

Figure 1: Example with $p = \Pr A_i = \frac{1}{10}$ and $d = 9$

This shows the tightness of the bound. We continue with the proof of the Lovász Local Lemma. To do so, we will use the following lemma:

Lemma 3 *For any $S \subset \{1, \ldots, n\}$ and $i \in \{1, \ldots, n\}$ the following holds:*

$$\Pr A_i \mid \bigwedge_{j \in S} \bar{A}_j \leq \frac{1}{d+1}$$

Proof We show the lemma by induction on the size m of S.

Base Case

For $m = 0$ we obtain

$$\Pr A_i \leq p \leq \frac{1}{(d+1) \cdot e} < \frac{1}{(d+1)}$$

where the second inequation follows by the condition of Lovász Local Lemma.

Inductive Case

Assume that the claim is true for all S with $|S| < m$. We prove the claim for an S with $|S| = m$. At first we partition S into $S = S_1 \cup S_2$ such that S_1 contains those events of S that A_i depends on. Recall that

$$\Pr A|B = \frac{\Pr A \wedge B}{\Pr B}$$

In the following we will apply this equation several times. Using this equation we get

$$\Pr A_i | \bigwedge_{j \in S} \bar{A}_j = \frac{\Pr A_i \wedge \bigwedge_{j \in S_1} \bar{A}_j | \bigwedge_{j \in S_2} \bar{A}_j}{\Pr \bigwedge_{j \in S_1} \bar{A}_j | \bigwedge_{j \in S_2} \bar{A}_j}$$

To bound this term further, we want to give an upper bound on the numerator and a lower bound on the denominator. The upper bound on the numerator is easily obtained as:

$$\Pr A_i \wedge \bigwedge_{j \in S_1} \bar{A}_j | \bigwedge_{j \in S_2} \bar{A}_j \leq \Pr A_i | \bigwedge_{j \in S_2} \bar{A}_j = \Pr A_i$$

We continue with the lower bound on the denominator. W.l.o.g. let $S_1 = \{1, \ldots, r\}$. Therefore, we can rewrite the denominator as:

$$\begin{aligned}
\Pr \bigwedge_{j=1}^{r} \bar{A}_j | \bigwedge_{j \in S_2} \bar{A}_j =& \Pr \bar{A}_1 \mid \bar{A}_2 \wedge \bar{A}_3 \wedge \ldots \wedge \bar{A}_r \wedge \bigwedge_{j \in S_2} \bar{A}_j \\
&\cdot \Pr \bar{A}_2 \mid \bar{A}_3 \wedge \bar{A}_4 \wedge \ldots \wedge \bar{A}_r \wedge \bigwedge_{j \in S_2} \bar{A}_j \\
&\vdots \\
&\cdot \Pr \bar{A}_r \mid \bigwedge_{j \in S_2} \bar{A}_j \\
=& \left(1 - \Pr A_1 \mid \bar{A}_2 \wedge \bar{A}_3 \wedge \ldots \wedge \bar{A}_r \wedge \bigwedge_{j \in S_2} \bar{A}_j\right) \\
&\cdot \left(1 - \Pr A_2 \mid \bar{A}_3 \wedge \bar{A}_4 \wedge \ldots \wedge \bar{A}_r \wedge \bigwedge_{j \in S_2} \bar{A}_j\right) \\
&\vdots \\
&\cdot \left(1 - \Pr A_r \mid \bigwedge_{j \in S_2} \bar{A}_j\right) \\
\overset{I.B.}{\geq}& \left(1 - \frac{1}{d+1}\right) \cdot \left(1 - \frac{1}{d+1}\right) \cdot \ldots \cdot \left(1 - \frac{1}{d+1}\right) \\
=& \left(1 - \frac{1}{d+1}\right)^r \geq \left(1 - \frac{1}{d+1}\right)^d > \frac{1}{e}
\end{aligned}$$

We combine both bounds and obtain:

$$\begin{aligned}
\Pr A_i | \bigwedge_{j \in S} \bar{A}_j =& \frac{\Pr A_i \wedge \bigwedge_{j \in S_1} \bar{A}_j | \bigwedge_{j \in S_2} \bar{A}_j}{\Pr \bigwedge_{j \in S_1} \bar{A}_j | \bigwedge_{j \in S_2} \bar{A}_j} \\
\leq& \frac{\Pr A_i}{e^{-1}} \leq p \cdot e \leq \frac{1}{d+1}
\end{aligned}$$

This concludes the proof of the lemma. ■

We will now apply the lemma to prove the theorem.

Proof We need to show that desired event occurs with positive probability. This probability can be bounded by

$$\begin{aligned}
\Pr \bigwedge_{i=1}^{n} \bar{A}_i =& \Pr \bar{A}_1 \mid \bar{A}_2 \wedge \bar{A}_3 \wedge \ldots \wedge \bar{A}_n \\
&\cdot \Pr \bar{A}_2 \mid \bar{A}_3 \wedge \bar{A}_4 \wedge \ldots \wedge \bar{A}_n \\
&\vdots \\
&\cdot \Pr \bar{A}_n \\
\geq& \left(1 - \frac{1}{d+1}\right)^n > 0
\end{aligned}$$

where the inequality follows from lemma 3. ■

2.1 Application: k-SAT

We use the Lovász Local Lemma to prove that any k-CNF formula is satisfiable if a certain condition holds. First we define the k-SAT problem:

Input Boolean formula $\varphi = \bigwedge_{i=1}^{n} C_i$ in k-CNF.

with m boolean variables $x_1, \ldots, x_m$

and n clauses $C_1, \ldots, C_n$ being conjunctions of k different literals

Output Decide whether there is a satisfying interpretation of φ

Let us focus on a special case, the 3-SAT problem, as an example.

An instance of 3-SAT consists of clauses, one can be represented this way :

$$(x_1 \vee x_2 \vee x_3)$$

We now state and prove two straightforward lemmas.

Lemma 4 *Every 3-SAT instance with six clauses is satisfiable.*

Proof We can use either Combinatorial or Probabilistic proof

Combinatorial

Assuming worst case there is at most 3 variables, which can create a total of 8 clauses. Let us take one of the two clauses that are not in the six clauses of our instance, for example $(x_1 \vee x_2 \vee x_3)$. We can choose an assignment of x_1, x_2, x_3 such that this clause is false, thus making all other clauses true.

Probabilistic

Consider a uniformly random truth assignment. For each clause C in the instance,
Pr[C is false]= 2^{-3},
from the Union Bound we get : $\Pr[\exists \text{ clause that is false}] \leq 6 * 2^{-3}$
$\Rightarrow \Pr[\text{Satisfiable}] \geq 1 - 6 * 2^{-3} = \frac{1}{4} > 0$. ■

Lemma 5 *Any 3-SAT instance where every variable appears exactly once is satisfiable.*

Proof Consider a uniformly random truth assignment.
Let A_1, ..., A_m be the events that the i-th clause in the instance is false. $\Pr[A_i \text{ is false}] = \frac{1}{8}$
Observe that A_is are mutually independent.

$$\Pr \bigwedge_{i=1}^{n} \bar{A}_i = \left(\frac{7}{8}\right)^m > 0$$

■

We now formalize the condition to guarantee the satisfiability of an k-CNF formula.

Lemma 6 *Any instance φ of k-SAT in which no variable appears in more than $\frac{2^{k-2}}{k}$ clauses is satisfiable.*

Proof We take an assignment $X : \{x_1, \ldots, x_m\} \to \{true, false\}$ uniformly at random. Let A_i be the "bad" event that clause C_i is unsatisfied by the random assignment X.
Let us determine the probability $\Pr A_i$. Since each clause consists of k different literals and each literal evaluates to false with probability $\frac{1}{2}$ we get:

$$\Pr A_i = \frac{1}{2^k} =: p$$

Now let us bound the maximum dependency of A_i. Let $j \neq i$. It is obvious that A_i is dependent on A_j if and only if C_i and C_j share a variable. Each variable may occur in at most $\frac{2^{k-2}}{k}$ clauses. Since C_i consists of k literals, it can be dependent on at most $k\frac{2^{k-2}}{k} = 2^{k-2}$ other clauses.
Thus we can apply the Lovász Local Lemma with $p = \frac{1}{2^{k-2}}$ and $d = 2^{k-2}$ and get $p \cdot (d+1) \cdot e \leq p \cdot d \cdot 4 = 1$ for sufficiently large d. Applying the Lovász Local Lemma we get that the probability of getting a satisfying assignment $\Pr[\bigwedge_{i=1}^{n} \bar{A}_i] > 0$. Using the probabilistic method, we can conclude that there is in fact a satisfying assignment for φ. ■

2.2 Exercise 1: Application: 2-COLORING in a hyper graph

We use Lovász Local Lemma to prove that certain hyper graphs always have a 2-Coloring. A hyper graph is a tuple $H = (V, E)$ with vertices V and hyper edges E. That is, each edge is a subset of the set of vertices V. A 2-Coloring is an assignment $c : V \to \{red, blue\}$ of colors to vertices such that there is no monochromatic hyper edge. A monochromatic edge is an edge connecting only nodes of the same color.

Exercise 1 Let $H = (V, E)$ be a hyper graph in which every edge has at least k elements and intersects at most d other edges. For which k, d has H a 2-Coloring?

Solution We take a coloring uniformly at random. Let A_e be the bad event that edge e became monochromatic. This is the case when all vertices of e are either blue or all red. Now consider the probability of A_e:

$$\Pr A_e = \left(\frac{1}{2}\right)^k + \left(\frac{1}{2}\right)^k = 2^{1-k} =: p$$

We need to choose d such that the condition of the Lovász Local Lemma is satisfied. That is, $p \cdot (d+1) \cdot e \leq 1$. This can be achieved by choosing $d \leq \frac{2^{1-k}}{e} - 1$.

2.3 Exercise 2: Subtrees

The proof of Lovász Local Lemma given beforehand was non-constructive. Later on we will give a constructive proof. To do so, the following statement will be useful.

Exercise 2 Consider a graph G of degree at most $d + 2$. Give an upper bound on the number of subtrees of G consisting of s nodes.

Solution For a fixed root we order the vertices in an arbitrarily (but fixed) order. Note, that this also orders the neighbors of a vertex uniquely. We will encode each tree with a bitstring of length $(d + 1) \cdot s$ containing $s - 1$ ones. We count the number of trees rooted at a vertex v.
Assume we had such an encoding. Then, the number of trees can be at most the number of bitstrings of this kind. That is:

$$\binom{(d+1) \cdot s}{s-1} \leq \left(\frac{(d+1) \cdot s \cdot e}{s-1}\right)^s \leq ((d+2) \cdot e)^s$$

It remains to be shown how a tree T can be encoded in such a fashion. The general idea is to do a pre-order traversal of T starting at the root v. When we discover a new vertex u, we output a string of length $(d+1)$. This string indicates whether for every neighbor w of u, w is contained in the tree. In particular, the string has a "1" at the ith position if and only if, the ith neighbor is a child of u in T. This is visualized in figure 2:

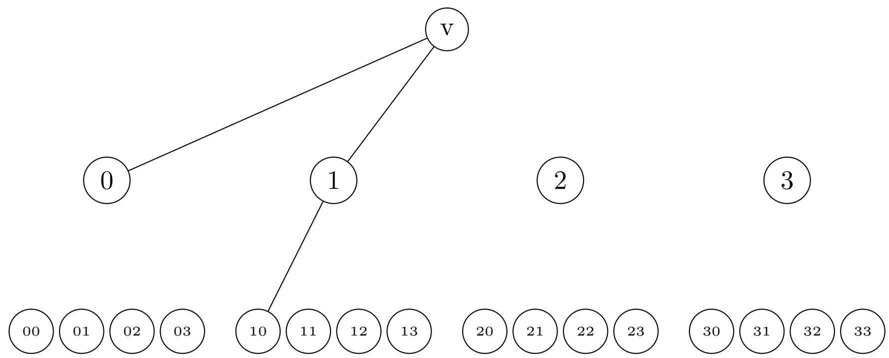

Figure 2: Tree of size 4 with coding 1100 0000 1000 0000

Because we have n possible roots, the total number of possible trees can be bounded by $n \cdot ((d+2) \cdot e)^s$.

2.4 A Constructive Proof of Lovász Local Lemma

The previous proof in this lecture was non-constructive. It did not show how to obtain an algorithm that computes the desired object. In this subsection we present the constructive version of this proof in [2] and [3] by giving a randomized algorithm for k-SAT. For the general case a similar algorithm can be constructed. We use the notation as before.

Algorithm 1 Randomized k-SAT

```
Pick random truth assignment X : {x_1, ..., x_m} → {true, false}
while there is an unsatisfied clause C do
    Reflip all variables in C
end while
Return the satisfying assignment
```

Obviously this algorithm returns a satisfying assignment on termination. It remains to be shown that it in fact terminates in acceptable runtime.

Theorem 7 *Let $p \cdot (d+1) \cdot e \leq 1 - \varepsilon$ for a constant $\varepsilon > 0$. Then the expected amount of reflips in the algorithm will be in $\Theta(n)$.*

Proof Let $C_1, C_2, \ldots$ be the (possibly infinite) sequence of clauses that are reflipped by the algorithm in that order. With each C_t we associate the tree T_t with a root node labeled as C_t. We add clause $C_1, \ldots C_{t-1}$ in reversed order as follows: For $i = t-1, \ldots, 1$ we ignore C_i if it does not share any variable with any clause in T_t. Otherwise, we add C_i as a child of the deepest clause sharing at least one variable with C_i.

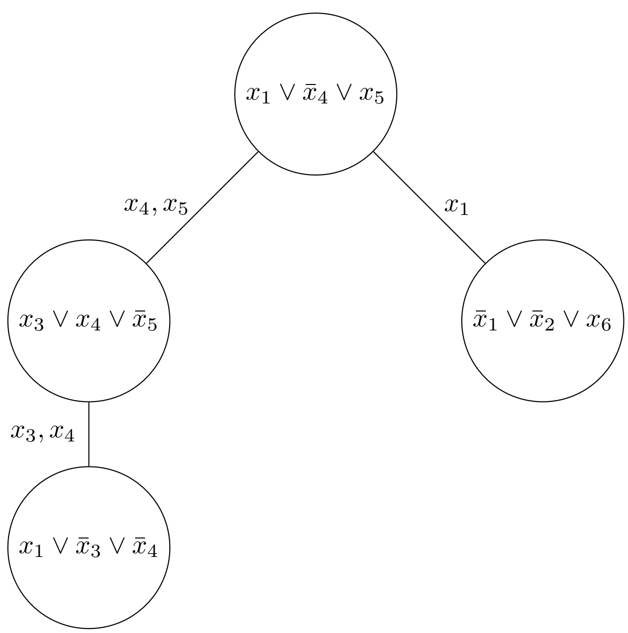

Figure 3: Example with reflipped clauses $(x_1 \vee \bar{x}_3 \vee \bar{x}_4), (\bar{x}_1 \vee \bar{x}_2 \vee x_6), (x_3 \vee x_4 \vee \bar{x}_5)$ and $(x_1 \vee \bar{x}_4 \vee x_5)$

We use the constructed tree T_t to prove the theorem by bounding the expected number of reflips. To do so, we have to measure the probability of a certain tree appearing.

Claim Consider any tree T_t of s nodes. The probability of this tree appearing is at most 2^{-sk}.
Proof Consider a leaf of maximal depth in T_t. The corresponding clause does not share any variables with a clause which has been reflipped earlier. Thus, this clause was unsatisfied by the original assignment. This happens with probability 2^{-k}. Since every variable in this clause is reflipped, the assignment of variables of higher clauses is independent. Hence, each node independently appears with probability 2^{-k}. Because the tree consists of s nodes, its probability of appearing is 2^{-sk}. ■

We use the claim to calculate the estimated amount of reflips for finding a satisfying assignment. Note that, by definition, two trees T_i and T_j with $i < j$ can't be identical, hence it suffices to count the number of distinct trees.

$$\begin{aligned}
\mathbb{E}[\#reflips] &= \mathbb{E}[\#\text{ trees appearing}] \\
&= \sum_{s=1}^{\infty} \mathbb{E}[\#\text{ trees of size } s] \\
&\leq n \cdot \sum_{s=1}^{\infty} \overbrace{((d+1)\cdot e)^s}^{\text{bound on the amount of trees}} \cdot \underbrace{2^{-sk}}_{\text{probility of a tree appearing}} \\
&= n \cdot \sum_{s=1}^{\infty} ((d+1)\cdot e \cdot 2^{-k})^s \\
&= n \cdot \sum_{s=1}^{\infty} ((d+1)\cdot e \cdot p)^s \\
&\leq n \cdot \sum_{s=1}^{\infty} (1-\varepsilon)^s \\
&\leq \frac{n}{1-(1-\varepsilon)} = \frac{n}{\varepsilon} = \Theta(n)
\end{aligned}$$

This finishes the proof of the theorem. ■

References

[1] Paul Erdos and László Lovász. Problems and results on 3-chromatic hypergraphs and some related questions. *Infinite and finite sets*, 10:609–627, 1975.

[2] Robin A Moser. A constructive proof of the lovász local lemma. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 343–350. ACM, 2009.

[3] Robin A Moser and Gábor Tardos. A constructive proof of the general lovász local lemma. *Journal of the ACM (JACM)*, 57(2):11, 2010.