R would have a maximal 4-tree of size r/d^3, the absence of such a 4-tree implies the absence of such a component.

We need to count the number of 4-trees of size $s = r/d^3$ in H. We can choose the root of the 4-tree in m ways. A tree with root v is uniquely defined by an Eulerian tour that starts and ends at v and traverses each edge of the tree twice, once in each direction. Since an edge of S represents a path of length 4 in H, at each vertex in the 4-tree the Eulerian path can continue in as many as d^4 different ways, and therefore the number of 4-trees of size $s = r/d^3$ in H is bounded by

$$m(d^4)^{2s} = md^{8r/d^3}.$$

The probability that the nodes of each such 4-tree survive in H' is at most

$$((d+1)2^{-k/2})^s = ((d+1)2^{-k/2})^{r/d^3}.$$

Hence the probability that H' has a connected component of size r is bounded by

$$md^{8r/d^3}((d+1)2^{-k/2})^{r/d^3} \leq m2^{(rk/d^3)(8\alpha+2\alpha-1/2)} = o(1)$$

for $r \geq c\log_2 m$ and for a suitably large constant c and a sufficiently small constant $\alpha > 0$. ∎

Thus, we have the following theorem.

Theorem 6.16: *Consider a k-SAT formula with m clauses, where k is an even constant and each variable appears in up to $2^{\alpha k}$ clauses for a sufficiently small constant $\alpha > 0$. Then there is an algorithm that finds a satisfying assignment for the formula in expected time that is polynomial in m.*

Proof: As we have described, if the first phase partitions the problem into subformulas involving only $O(k \log m)$ variables, then a solution can be found by solving each subformula exhaustively in time that is polynomial in m. The probability of the first phase partitioning the problem appropriately is $1 - o(1)$, so we need only run phase I a constant number of times on average before obtaining a good partition. The theorem follows. ∎

6.9. Lovasz Local Lemma: The General Case

For completeness we include the statement and proof of the general case of the Lovasz local lemma.

Theorem 6.17: *Let $E_1, \dots, E_n$ be a set of events in an arbitrary probability space, and let $G = (V, E)$ be the dependency graph for these events. Assume there exist $x_1, \dots, x_n \in [0,1]$ such that, for all $1 \leq i \leq n$,*

$$\Pr(E_i) \leq x_i \prod_{(i,j)\in E} (1 - x_j).$$

From Mitzenmacher and Upfal: Probability and Computing

Then

$$\Pr\left(\bigcap_{i=1}^{n} \bar{E}_i\right) \geq \prod_{i=1}^{n}(1 - x_i).$$

Proof: Let $S \subseteq \{1, \dots, n\}$. We prove by induction on $s = 0, \dots, n$ that, if $|S| \leq s$, then for all k we have

$$\Pr\left(E_k \mid \bigcap_{j \in S} \bar{E}_j\right) \leq x_k.$$

As in the case of the symmetric version of the local lemma, we must be careful that the conditional probability is well-defined. This follows using the same approach as in the symmetric case, so we focus on the rest of the induction.

The base case $s = 0$ follows from the assumption that

$$\Pr(E_k) \leq x_k \prod_{(k,j) \in E} (1 - x_j) \leq x_k.$$

For the inductive step, let $S_1 = \{j \in S \mid (k, j) \in E\}$ and $S_2 = S - S_1$. If $S_2 = S$ then E_k is mutually independent of the events $\bar{E}_i$, $i \in S$, and

$$\Pr\left(E_k \mid \bigcap_{j \in S} \bar{E}_j\right) = \Pr(E_k) \leq x_k.$$

We continue with the case $|S_2| < s$. We again use the notation

$$F_S = \bigcap_{j \in S} \bar{E}_j$$

and define F_{S_1} and F_{S_2} similarly, so that $F_S = F_{S_1} \cap F_{S_2}$.

Applying the definition of conditional probability yields

$$\Pr(E_k \mid F_S) = \frac{\Pr(E_k \cap F_S)}{\Pr(F_S)}. \tag{6.6}$$

By once again applying the definition of conditional probability, the numerator of (6.6) can be written as

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

and the denominator as

$$\Pr(F_S) = \Pr(F_{S_1} \mid F_{S_2}) \Pr(F_{S_2}).$$

Canceling the common factor then yields

$$\Pr(E_k \mid F_S) = \frac{\Pr(E_k \cap F_{S_1} \mid F_{S_2})}{\Pr(F_{S_1} \mid F_{S_2})}. \tag{6.7}$$

Since the probability of an intersection of events is bounded by the probability of each of the events and since E_k is independent of the events in S_2, we can bound the numerator of (6.7) by

$$\Pr(E_k \cap F_{S_1} \mid F_{S_2}) \leq \Pr(E_k \mid F_{S_2}) = \Pr(E_k) \leq x_k \prod_{(k,j) \in E} (1 - x_j).$$

To bound the denominator of (6.7), let $S_1 = \{j_1, \dots, j_r\}$. Applying the induction hypothesis, we have

$$\begin{aligned}
\Pr(F_{S_1} \mid F_{S_2}) &= \Pr\left(\bigcap_{j\in S_1} \bar{E}_j \mid \bigcap_{j\in S_2} \bar{E}_j\right) \\
&= \prod_{i=1}^{r}\left(1 - \Pr\left(E_{j_i} \mid \left(\bigcap_{t=1}^{i-1} \bar{E}_{j_t}\right) \cap \left(\bigcap_{j\in S_2} \bar{E}_j\right)\right)\right) \\
&\ge \prod_{i=1}^{r}(1 - x_{j_i}) \\
&\ge \prod_{(k,j)\in E}(1 - x_j).
\end{aligned}$$

Using the upper bound for the numerator and the lower bound for the denominator, we can prove the induction hypothesis:

$$\begin{aligned}
\Pr\left(E_k \mid \bigcap_{j\in S} \bar{E}_j\right) &= \Pr(E_k \mid F_S) \\
&= \frac{\Pr(E_k \cap F_{S_1} \mid F_{S_2})}{\Pr(F_{S_1} \mid F_{S_2})} \\
&\le \frac{x_k \prod_{(k,j)\in E}(1 - x_j)}{\prod_{(k,j)\in E}(1 - x_j)} \\
&= x_k.
\end{aligned}$$

The theorem now follows from:

$$\begin{aligned}
\Pr(\bar{E}_1, \dots, \bar{E}_n) &= \prod_{i=1}^{n} \Pr(\bar{E}_i \mid \bar{E}_1, \dots, \bar{E}_{i-1}) \\
&= \prod_{i=1}^{n}(1 - \Pr(E_i \mid \bar{E}_1, \dots, \bar{E}_{i-1})) \\
&\ge \prod_{i=1}^{n}(1 - x_i) > 0.
\end{aligned}$$

■

6.10. Exercises

Exercise 6.1: Consider an instance of SAT with m clauses, where every clause has exactly k literals.

(a) Give a Las Vegas algorithm that finds an assignment satisfying at least $m(1 - 2^{-k})$ clauses, and analyze its expected running time.

(b) Give a derandomization of the randomized algorithm using the method of conditional expectations.