k-SAT with few clauses

Theorem (A)

For every natural number k, every k-SAT formula with less than 2^k clauses is satisfiable.

Proof: Consider a random truth assignment which sets variable x_i to 1 with probability $\frac{1}{2}$ and to 0 with probability $\frac{1}{2}$ for
$i = 1, 2, \ldots, n$.
Note that by this assignment, each of the 2^n possible truth assignments are equally likely (they all have probability 2^{-n}).

k-SAT with few clauses

For each clause C_i we let the random variable X_i take the value $X_i(t) = 1$ if t does **not** satisfy C_i and $X_i(t) = 0$ if t satisfies C_i.
Hence $E(X_i) = 2^{-k}$
Let $X = X_1(t) + X_2(t) + \ldots + X_m(t)$. So X counts the number of clauses that are not satisfied by t.
$E(X) = \sum_{i=1}^{m} E(X_i) = \sum_{i=1}^{m} 2^{-k} = 2^{-k} \sum_{i=1}^{m} 1 = m2^{-k} < 1$, since $m < 2^k$.
Hence, by Markov's inequality, $p(X \geq 1) \leq \frac{E(X)}{1} = E(X) < 1$ so $p(X = 0) > 0$. This shows that there is at least one of the 2^n truth assignments which satisfies all m clauses.

k-SAT with few clauses

The bound on the number of clauses in the theorem is best possible:
Suppose we have $n = k$ variables and all the 2^k clauses of size k over these variables (every clause contains each variable either with or without negation), then clearly this instance is not satisfiable, since no matter which truth assignment we take, some clause will have all literals evaluating to 0. But observe that removing just one we get a satisfiable instance by the theorem!

SAT with few clauses

Using the same argument as above we get the following bound for general SAT (clauses may have any size):

Theorem (B)

Let $\mathcal{F} = C_1 * C_2 * \ldots * C_m$ be an instance of SAT. If we have $\sum_{i=1}^{m} 2^{-|C_i|} < 1$, then $\mathcal{F}$ is satisfiable.

Corollary

For all $\epsilon > 0$ there exists a polynomial algorithm for solving any instance of SAT over n variables $x_1, x_2, \ldots, x_n$ in which all clauses have size at least ϵn.

Proof: Let $\epsilon > 0$ be given and let $\mathcal{F} = C_1 * C_2 * \ldots * C_m$ over the variables $x_1, x_2, \ldots, x_n$ satisfy that $|C_i| \geq \epsilon n$ for each $i \in \{1, 2, \ldots, m\}$.

SAT with few clauses

Suppose first that $m < 2^{\epsilon n}$. Then we have

$$\sum_{i=1}^{m} 2^{-|C_i|} \leq \sum_{i=1}^{m} 2^{-\epsilon n} = m 2^{-\epsilon n} < 1$$

Hence it follows from Theorem B that $\mathcal{F}$ is satisfiable and our algorithm stops with a "yes". Clearly this can be checked in time polynomial in $|\mathcal{F}|$ since we just need to check whether the number of clauses is less than $2^{\epsilon n}$. **Note that in this case we do not find a satisfying truth assignment!** We just answer correctly that there **exists** one.

SAT with few clauses

Now suppose that we found that there was at least $2^{\epsilon n}$ clauses. Then we simply check all the 2^n possible truth assignments to see whether one of these satisfies $\mathcal{F}$. If we find one that does, we stop and answer "yes" otherwise, after checking that none of them satisfy $\mathcal{F}$, we answer "no". The time required to do this is proportional to $2^n|\mathcal{F}|$, where $|\mathcal{F}|$ is the size of the formula $\mathcal{F}$ and hence of the input. Clearly $|\mathcal{F}| \geq 2^{\epsilon n}$ as all clauses have size at least 1 (in fact $|\mathcal{F}| \geq \epsilon n 2^{\epsilon n}$). From this we get that $2^n \leq |\mathcal{F}|^{\frac{1}{\epsilon}}$ so the running time of our algorithm is proportional to $2^n|\mathcal{F}| \leq |\mathcal{F}|^{1+\frac{1}{\epsilon}}$ which is a polynomial in $|\mathcal{F}|$ because ϵ is a constant (when we have chosen it).

The Lovasz Local Lemma

Let $A_1, ..., A_n$ be a set of "bad" events. We want to show that

$$Pr(\cap_{i=1}^{n} \bar{A}_i) > 0.$$

1. If $\sum_{i=1}^{n} Pr(A_i) < 1$ then $Pr(\cap_{i=1}^{n} \bar{A}_i) > 0.$
2. If all the A_i's are mutually independent and for all i $Pr(A_i) < 1$ then $Pr(\cap_{i=1}^{n} \bar{A}_i) = \Pi_{i=1}^{n} Pr(\bar{A}_i) > 0.$.
3. If each A_i depends only on a few other events: *The Lovasz Local Lemma*.

Definition

An event E is mutually independent of the events $E_1, \ldots, E_n$, if for any $T \subset [1, \ldots, n]$,

$$Pr(E \mid \cap_{j \in T} E_j) = Pr(E).$$

Definition

A dependency graph for a set of events $E_1, \ldots, E_n$ has n vertices $1, \ldots, n$. Events E_i is mutually independent of any set of events $\{E_j \mid j \in T\}$ iff there is no edge in the graph connecting i to any $j \in T$.

Theorem

Let $E_1, ..., E_n$ *be a set of events. Assume that*

1. *For all* i, $Pr(E_i) \leq p$;
2. *The degree of the dependency graph is bounded by* d.
3. $4dp \leq 1$

then

$$Pr(\cap_{i=1}^{n} \bar{E}_i) > 0.$$

Let $S \subset \{1, ..., n\}$. We prove by induction on $s = 0, ..., n-1$ that if $|S| \leq s$, for all k

$$Pr(E_k \mid \cap_{j \in S} \bar{E}_j) \leq 2p.$$

For $s = 0$, $S = \emptyset$ obvious.
W.l.o.g. renumber the events so that $S = \{1, ..., s\}$, and (k, j) is not and edge of the dependency graph for $j > d$.

$$
\begin{aligned}
Pr(E_k \mid \bigcap_{j=1}^{s} \bar{E}_j) &= \frac{Pr(E_k \cap \bigcap_{j=1}^{s} \bar{E}_j)}{Pr(\bigcap_{j=1}^{s} \bar{E}_j)} \\
&= \frac{Pr(E_k \cap \bigcap_{j=1}^{d} \bar{E}_j \mid \bigcap_{j=d+1}^{s} \bar{E}_j) Pr(\bigcap_{j=d+1}^{s} \bar{E}_j)}{Pr(\bigcap_{j=1}^{d} \bar{E}_j \mid \bigcap_{j=d+1}^{s} \bar{E}_j) Pr(\bigcap_{j=d+1}^{s} \bar{E}_j)} \\
&= \frac{Pr(E_k \cap \bigcap_{j=1}^{d} \bar{E}_j \mid \bigcap_{j=d+1}^{s} \bar{E}_j)}{Pr(\bigcap_{j=1}^{d} \bar{E}_j \mid \bigcap_{j=d+1}^{s} \bar{E}_j)}
\end{aligned}
$$

$$Pr(E_k \cap \bigcap_{j=1}^{d} \bar{E}_j \mid \bigcap_{j=d+1}^{s} \bar{E}_j) \leq Pr(E_k \mid \bigcap_{j=d+1}^{s} \bar{E}_j) = Pr(E_k) \leq p.$$

Using the induction hypothesis we prove:

$$\begin{aligned} Pr(\bigcap_{j=1}^{d} \bar{E}_j \mid \bigcap_{j=d+1}^{s} \bar{E}_j) &\geq 1 - \sum_{i=1}^{d} Pr(E_i \mid \bigcap_{j=d+1}^{s} \bar{E}_j) \\ &\geq 1 - \sum_{i=1}^{d} 2p \\ &\geq 1 - 2pd \geq 1/2. \end{aligned}$$

$$Pr(E_k \mid \bigcap_{j=1}^{s} \bar{E}_j) \leq \frac{p}{1/2} = 2p$$

Now we can complete the proof:

$$Pr\left(\bigcap_{j=1}^{n} \bar{E}_j\right) = \Pi_{i=1}^{n} Pr\left(\bar{E}_i \mid \bigcap_{j=1}^{i-1} \bar{E}_j\right)$$

$$= \Pi_{i=1}^{n}\left(1 - Pr\left(E_i \mid \bigcap_{j=1}^{i-1} \bar{E}_j\right)\right) \geq \Pi_{i=1}^{n}(1 - 2p) > 0.$$

Application: Edge-Disjoint Paths

Assume that n pairs of users need to communicate using edge-disjoint paths on a given network.
Each pair $i = 1, \ldots, n$ can choose a path from a collection F_i of m paths.

Theorem

If for each $i \neq j$, any path in F_i shares edges with no more than k paths in F_j, where $\frac{8nk}{m} \leq 1$, then there is a way to choose n edge-disjoint paths connecting the n pairs.

Proof

Consider the probability space defined by each pair choosing a path independently uniformly at random from its set of m paths.
$E_{i,j} =$ the paths chosen by pairs i and j share at least one edge.
A path in F_i shares edges with no more than k paths in F_j,

$$p = \Pr(E_{i,j}) \leq \frac{k}{m}.$$

Let d be the degree of the dependency graph.
Since event $E_{i,j}$ is independent of all events $E_{i',j'}$ when $i' \notin \{i, j\}$ and $j' \notin \{i, j\}$, we have $d < 2n$.

$$4dp < \frac{8nk}{m} \leq 1$$

$$\Pr(\cap_{i \neq j} \bar{E}_{i,j}) > 0.$$

Theorem

Consider a CNF formula with k literals per clause. Assume that each variable appears is no more than $T = \frac{2^k}{4k}$ clauses, then the formula has a satisfying assignment,

Proof.

Assume that the formula has m clauses.
For $i = 1, ..., m$, let E_i be the event "The random assignment does not satisfy clause i".

$$Pr(E_i) = \frac{1}{2^k}.$$

The event E_i is mutually independent of all the events related to clauses that do not share variables with clause i.
The degree of E_i in the dependency graph is bounded by kT.
Since

$$4dp \leq 4kT2^{-k} \leq 4k\frac{2^k}{4k}2^{-k} \leq 1$$

$$Pr(\bar{E_i}, ..., \bar{E_m}) > 0.$$

$\square$

Algorithm

Assume m clauses, ℓ variables, each clause has k literals, each variable appears in no more than $T = 2^{\alpha k}$ clauses.

First Part:

A clause is **Dangerous** at a given step if both

1. The clause is not satisfied;
2. At least $k/2$ of its variables were fixed.

For $i = 1$ to ℓ

If x_i is not in a dangerous clause assign it a random value in $\{0, 1\}$.

A **surviving clause** is a clause that is not satisfied at the end of phase one.

A surviving clause has no more than $k/2$ of its variables fixed.

A **deferred** variable is a variable that was not assigned a value in the first part.

Lemma

There is an assignment of values to the deferred variables such that all the surviving clauses are satisfied (thus the formula is satisfied).

Lemma

Let G' be the dependency graph on the surviving clauses. With high probability all connected components in G' have size $O(\log m)$.

Part Two:

Using exhaustive search assign values to the deferred variable to complete the truth assignment for the formula.

If a connected component has $O(\log m)$ clauses it has $O(k \log m)$ variables. Assuming $k = O(1)$ we can check all assignments in polynomial in m number of steps.

Lemma

There is an assignment of values to the deferred variables such that all the surviving clauses are satisfied (thus the formula is satisfied).

At the end of the first phase we have m' "surviving clauses' (all the rest are satisfied), each surviving clause has at least $k/2$ deferred variables.
Consider a random assignment of the deferred variables.
Let E_i be the event clause i (of the surviving clauses) is not satisfied.

$$p = Pr(E_i) \leq 2^{-k/2}.$$

The degree of the dependency graph is bounded by

$$d = kT < k2^{\alpha k}.$$

Since

$$4dp \le 4k2^{\alpha k}2^{-k/2} \le 1$$

there is a satisfying assignment of the deferred variables that (together with the assignment of the other variables) satisfies the formula.

Lemma

Let G' be the dependency graph on the surviving clauses. With high probability all connected components in G' have size $O(\log m)$.

Assume that there is a connected component R of size $r = |R|$. Since the degree of a vertex in R is bounded by d, there must be a set R' of $|R'| = r/d^3$ vertices in R which are at distance at least 4 from each other.

A clause "survives" the first part if it is at distance at most 1 from a dangerous clause. Thus, for each clause in R' there is a **distinct** dangerous clause, and these dangerous clauses are at distance 2 from each other.

The probability that a given clause is dangerous is at most $2^{-k/2}$.
The probability that a given clause C survives is at most $(d+1)2^{-k/2}$ (C must be unsatisfied after the first phase and either C is dangerous or at least one of its neighbours must be dangerous).
These events are independent for vertices in R'. Thus the probability of a particular connected component of r vertices is bounded by

$$\left((d+1)2^{-k/2}\right)^{r/d^3}$$

How many possible connected components of size r are in a graph of m nodes and maximum degree d?

Lemma

There are no more than md^{2r} possible connected components of size r in a graph of m vertices and maximum degree d.

Proof.

A connected component of size r has a spanning tree of $r - 1$ edges.
We can choose a "root" for the tree in m ways.
A tree can be defined by an Euler tour that starts and ends at the root and traverses each edge twice.
At each node the tour can continue in up to d ways. Thus, for a given root there are no more than d^{2r} different Euler tours. $\square$

Thus, the probability that at the end of the first phase there is a connected component of size $r = \Omega(\log m)$ is bounded by

$$md^{2r}\left((d+1)2^{-k/2}\right)^{r/d^3} = o(1)$$

for $d = k2^{\alpha k}$, $\alpha > 0$ sufficiently small.

Each deferred variable appears in only one component. A component of size $O(\log m)$ has only $O(\log m)$ variables. Thus, we can enumerate (try) all possibilities in time polynomial in m.

Theorem

Given a CNF formula of m clauses, each clause has $k = O(1)$ literals, each variables appears in up to $2^{\alpha k}$ clauses. For a sufficiently small $\alpha > 0$ there is an algorithm that finds a satisfying assignment to the formula in time polynomial in m.