Constructive Proof of the Lovász Local Lemma

Ferienakademie im Sarntal — Course 1
Moderne Suchmethoden der Informatik: Trends und Potenzial

Katharina Angermeier

Naturwissenschaftliche Fakultät
FAU Erlangen-Nürnberg

26. September 2014

Outline

Notation and Basic Definitions

- The *conjunctive normal form* (CNF) is a special notation form for boolean formulas.
- Example:
  $\underbrace{(x_1 \vee x_2 \vee x_3)}_{\text{clause}} \wedge (\underbrace{x_1}_{\text{literal}} \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_4) \wedge (x_2 \vee \overline{x_3} \vee x_4)$
  This would be a 3-CNF formula with 4 clauses over the variables $\{x_1, x_2, x_3, x_4\}$.
  Variables in a clause do not repeat.
- In general: a k-CNF formula ($k \in \mathbb{N}$) is a CNF formula where every clause contains *exactly* k literals
- An *assignment* α over variable set V is a mapping $\alpha : V \rightarrow \{0, 1\}$ that extends to $\overline{V}$ via $\alpha(\overline{x}) := 1 - \alpha(x)$ for $x \in V$

Notation and Basic Definitions

$$(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_3 \vee \overline{x_4}) \wedge (\overline{x_1} \vee \overline{x_2} \vee x_4) \wedge (x_2 \vee \overline{x_3} \vee x_4)$$

- A formula is called *satisfiable* if there is a true-false assignment to the variables so that every clause has at least one literal that evaluates to true, in this case the assignment could be $(x_1, x_2, x_3, x_4) \mapsto (\mathit{true}, \mathit{true}, \mathit{false}, \mathit{true})$
- $\mathrm{vbl}(C)$ is the set of variables that occur in a clause C
- $\mathrm{vbl}(F) := \bigcup_{C \in F} \mathrm{vbl}(C)$ for F a CNF formula

Simple probabilistic argument

It takes at least 2^k clauses to construct an unsatisfiable k-CNF formula.

Justification: Suppose some k-CNF formula with fewer than 2^k clauses.
An assignment sampled uniformly at random violates each clause with probability 2^{-k}.
$\Rightarrow$ By linearity of expectation: The expected total number of violated clauses is smaller than 1.
$\Rightarrow$ Some of the assignments have to satisfy the whole formula.

Local constraints

The constraint on the formula size needs not only to be satisfied globally but even locally.

The *neighbourhood* $\Gamma(C) = \Gamma_F(C) := \{D \in F | vbl(D) \cap vbl(C) \neq \emptyset\}$ of a clause C is the set of clauses that share variables with C.

If we can change values in a clause C without causing too much damage in its neighbourhood, and if this property holds everywhere, then maybe we can find a globally satisfying assignment by just moving around violation issues.

If every clause in a k-CNF formula, $k \geq 1$, has a neighbourhood of size at most $2^k/e - 1$, then the whole formula admits a satisfying assignment.

Lovász Local Lemma, 1975

Other variant:

"In an unsatisfiable CNF formula clauses have to interleave - the larger the clauses, the more interleaving is required."

Useful Definitions

- The *conflict-neighbourhood* $\Gamma'(C) = \Gamma'_F(C) := \{D \in F | C \cap \overline{D} \neq \emptyset\}$ of a clause C is the set of clauses which share variables with C, at least one with opposite sign.
- *lopsided Local Lemma* shows the condition for neighbourhoods holds actually for conflict-neighbourhoods
- The *degree of* x is the number of occurrences of a variable x (with either sign) in a CNF formula,
  $deg(x) = deg_F(x) := |\{C \in F | x \in vbl(C)\}|$

Claim *If every variable in a k-CNF formula,* $k \geq 1$*, has degree at most* $2^k/(ek)$*, then the formula is satisfiable.*

Useful Definitions

For observing the quality of interleaving we define:

- A *linear CNF formula* is a CNF formula where any two clauses share at most one variable.
  Example: $(\overline{y_1} \vee \overline{y_2}) \wedge (y_1 \vee x) \wedge (y_2 \vee x) \wedge (z_1 \vee \overline{x}) \wedge (z_2 \vee \overline{x}) \wedge (\overline{z_1} \vee \overline{z_2})$
  This is a smallest unsatisfiable linear 2-CNF formula.

Claim *Any linear k-CNF formula with at most* $4^k/(4e^2k^3)$ *clauses is satisfiable.*

Algorithms

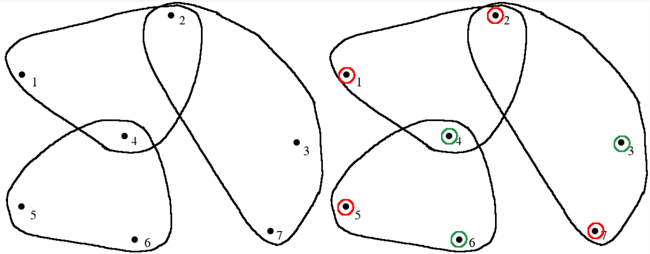
- Whenever the easily checkable conditions formulated above are satisfied, then the algorithmic problem of deciding satisfiability becomes trivial.
- The actual construction of a satisfying assignment is by no means obvious.

- Define $f(k)$, $k \in \mathbb{N}$, as the largest integer so that every k-CNF formula with no variable of degree exceeding $f(k)$ is satisfiable.
- $f(k) = \Theta(2^k/k)$
- For k-CNF formulas ($k \geq 3$) with max-degree at most $f(k) + 1$ the satisfiability problem becomes NP-complete.
- $l(k)$ is defined as the largest integer d such that every k-CNF formula F for which $|\Gamma_F(C)| \leq d$, for all $C \in F$, is satisfiable.
- $lc(k)$ is defined analogously, but with $|\Gamma'_F(C)| \leq d$.

Hypergraphs

- A *hypergraph* H is a pair (V, E) with V a finite set and $E \subseteq 2^V$.
- It is *k-uniform* if $|e| = k$ for all $e \in E$.
- H is called *2-colourable* if there is a colouring of the vertices in V by two colors red and green so that no hyperedge in E is monochromatic.

- Relation to satisfiablility of CNF formulas: $H = (V, E)$ is 2-colourable iff the CNF formula $E \cup \{\overline{e} | e \in E\}$, with V now considered as set of boolean variables, is satisfiable.



$$(x_1 \vee x_2 \vee x_4) \wedge (\overline{x_1} \vee \overline{x_2} \vee \overline{x_4}) \wedge (x_2 \vee x_3 \vee x_7) \wedge (\overline{x_2} \vee \overline{x_3} \vee \overline{x_7}) \wedge$$
$$(x_4 \vee x_5 \vee x_6) \wedge (\overline{x_4} \vee \overline{x_5} \vee \overline{x_6})$$

Local Lemma in Terms of SAT - Proof and Algorithm

Theorem 1 *Let* $k \in \mathbb{N}$ *and let* F *be a* k*-CNF formula. If* $|\Gamma(C)| \leq 2^k/e - 1$ *for all* $C \in F$*, then* F *is satisfiable.*

P. Erdős, L. Lovász: Problems and results on 3-chromatic hypergraphs and some related questions.

History of Theorem 1

1975 "existential" proof : short but non-constructive

1991 Beck proved the existence of a polynomial-time algorithm to find a satisfying assignment for all $C \in F$, F a k-CNF formula $\Gamma(C) \leq 2^{k/48}$.

1991 Alan simplified Beck's algorithm by randomness, and presented an algorithm that works for neighbourhoods of size up to $2^{k/8}$.

2000 Czumaj and Scheideler demonstrated that a variant of the method can be made to work for the case where clauses sizes vary.

2008 Srinivasan improved the time to essentially $2^{k/4}$.

2008 Moser published an polynomial-time algorithm for neighbourhood sizes up to $O(2^{k/2})$, later for 2^{k-5} neighbours.

2009 Moser and Tardos published a fully constructive proof.

$k = 1$	p
$\emptyset$	1
x_1	$\frac{1}{2}$
$x_1 \wedge x_2$	$\frac{1}{4}$
$x_1 \wedge x_2 \wedge x_3$	$\frac{1}{8}$
$x_1 \wedge x_2 \wedge x_3 \wedge x_4$	$\frac{1}{16}$

$k = 2$	p
$\emptyset$	1
$(x_1 \vee x_2)$	$\frac{3}{4}$
$(x_1 \vee x_2) \wedge (x_3 \vee x_4)$	$\frac{9}{16}$
$(x_1 \vee x_2) \wedge (x_3 \vee x_4) \wedge (x_5 \vee x_6)$	$\frac{27}{64}$

First Proof of Local Lemma - Existence

- F k-CNF formula, neighbourhood size at most $d := \frac{2^k}{e} - 1$
- If the probability of a random assignment α to satisfy F is positive, F is satisfiable.
- $F' \subset F$ subformula of F with one fewer clause, $C \in F \backslash F'$ one of the clauses removed
- α has probability $Pr(F')$ of satisfying F'
- We want to compute the drop in probability when adding back C. Claim: the factor is bounded by $(1 - e2^{-k})$, which means $Pr(F' \wedge C) \geq (1 - e2^{-k})Pr(F')$.
- If the factor is positive, the claim is proved.

Induction

- Suppose the latter claim has been proved for all subformulas F' up to a given size.
- trivial special case: If C is independent from F', the probability decreases by a factor of exactly $(1 - 2^{-k})$.
- Otherwise we remove all clauses of F' neighbouring C and get $F'' := F' \backslash \Gamma(C)$
- $\Rightarrow Pr(F'' \wedge \neg C) = 2^{-k} Pr(F'')$
- By adding back all clauses one by one to F'' to get F' we obtain
  $Pr(F') \geq (1 - e2^{-k})^d Pr(F'') \geq e^{-1} Pr(F'')$
  $Pr(F' \wedge \neg C) \leq Pr(F'' \wedge \neg C) = 2^{-k} Pr(F'')$

$$\Rightarrow \frac{Pr(F' \wedge \neg C)}{Pr(F')} \leq \frac{2^{-k}}{e^{-1}}$$

Second Proof of Local Lemma - Algorithm

- Algorithm: We repeatedly select any of the violated clauses and just select new uniformly random variables occurring in that clause until a satisfying assignment is obtained.
  Analysis:
- We record a log of corrections with the mapping $L : \mathbb{N}_0 \to F$
- Let $N : F \to \mathbb{N}_0 \cup \{\infty\}$ be random variables that count the number of times a given clause occurs in the log.
- We prove now that for each clause $C \in F$ the expected value $E[N(C)]$ is upper bounded by a constant.
- To continue we introduce witness trees. A witness tree is an unordered, rooted tree T along with a labelling $\sigma : V(T) \to F$ of its vertices $V(T)$ by clauses from F.
- We label the root vertex r $\sigma(r) := L(t)$.
- Now we traverse the log backwards and for each time step $s = t-1, t-2, \ldots, 0$, check if the clause $L(s)$ has any variables that it shares with any of the labels in the tree built so far.

- If $L(s)$ is independent from all clauses currently serving as labels, discard it.
  Otherwise select any deepest of the nodes the tree has in common with $L(s)$ and create a new child node of it, labelling that new child $L(s)$
- When arriving at $s = 0$ we have built a witness tree $T(t)$ that justifies correction step t.
- By traversing $T(t)$ in a breadth-first-search that starts at the root we obtain a sequence of clauses that is a subsegment of the execution log.
- The way we defined $T(t)$ assures two things:
  (a) The ordering in which the corrections have taken place is similar to the ordering in which we traverse the nodes.
  (b) When we traverse some node v representing correction step t, then all correction steps $t' < t$ that relate to step t do occur in the tree and have therefore been traversed before.

$\Rightarrow$ The number of times some variable x has occurred so far in labelling clauses corresponds to the number of times x has been reassigned new values before the corresponding correction step.

- What about when you have given a fixed witness tree T?
- We can reconstruct k of the random bits the algorithm has used.
- If the tree has n vertices, we can reconstruct nk bits in total.
- The probability that T can be constructed is exactly 2^{-nk}.
- For a fixed clause $C \in F$, number n we want the number of witness trees of order n which have C as the label of their root vertex.
- We embed each witness tree rooted at label C into an infinite tree that just enumerates neighbouring nodes.
- Consider an infinite tree with its root labelled C and such that each node v labelled $\sigma(v)$ has $|\Gamma(\sigma(v))|$ children labelled $\Gamma(\sigma(v))$.

- An infinite rooted $(\leq d)$-ary tree has at most $(ed)^n$ subtrees of size n.

  $\Rightarrow$ There are at most $(ed)^n$ witness trees of order n that have C as their root label.
- The expected number of witness trees of size n that can occur is bounded by $(ed2^{-k})^n$.
- Summing over all possible sizes $n \geq 1$ this becomes a geometric series that converges to a constant.

  $\Rightarrow$ There is at most a constant expected number of valid witness trees rooted at C.
- For each of the $N(C)$ times a clause C occurs in the execution log we can ask for a corresponding witness tree to justify that correction step.
- $N(C)$ is at most as large as the number of valid witness trees rooted at C, which is bounded by a constant in expectation. $\square$

A Stronger Variant - Conflicts

Theorem 3 *Let $k \in \mathbb{N}$ and let F be a k-CNF formula. If $|\Gamma'(C)| \leq 2^k/e - 1$ for all $C \in F$, then F is satisfiable.*

- Berman, Karpinski and Scott have demonstrated using the lopsided Local Lemma, that every 6-, 7-, 8- or 9-CNF formula in which every variable occurs at most 7, 13, 23 or 41 times, respectively, is satisfiable.

Bounded Variable Degree

- A k-CNF formula in which no variable occurs in more than d clauses is called a (k, d)-CNF formula.
- $f(k)$ is now defined as the unique integer so that all $(k, f(k))$-CNF formulas are satisfiable.
- $0 \leq f(k) \leq 2^k$
- Tovey was the first to consider $f(k)$ in 1984
- He showed $f(k) \geq k$ and conjectured that all $(k, 2^{k-1} - 1)$-CNF formulas are satisfiable.
- $k(d-1) \leq 2^k/e - 1$ implies that every (k, d)-CNF formula is satisfiable
- Kratochvíl, Savický and Tuza established 1993 that and the bounds of $f(k) \geq \lfloor 2^k/(ek) \rfloor$ and $f(k) \leq 2^{k-1} - 2^{k-4} - 1$
- Savický and Sgall showed $f(k) = O(k^{-0.26} 2^k)$ (2000), Hoory and Szeider improved it to $f(k) = O((2^k logk)/k)$ (2006). Recently Gebauer settled $f(k) = \Theta(2^k/k)$

Theorem 4 *For k a large enough integer,*

$$\lfloor 2^k/ek \rfloor \leq f(k) < 2^{k+1}/k.$$

If k is a sufficiently large power of 2 we have $f(k) < 2^k/k$.

- Proof of the upper bound with a Combinatorial game:
- Maker wants to completely occupy a hyperedge and Breaker tries to prevent this.
- The problem is to find the minimum $d = d(k)$ such that there is a k-uniform hypergraph of maximum vertex degree d where Maker has a winning strategy.
- If the Maker uses a pairing strategy, this game is equivalent to unsatisfiability.

- A hypergraph H, pairing P can be interpreted as a CNF formula F where the hyperedges of H are clauses and two vertices of a pair of P are complementary literals.
- Maker wins the game on H using the pairing strategy according to P if and only if F is unsatisfiable.

If there is a k-uniform hypergraph of maximum vertex degree d with a winning pairing strategy for Maker, then there is an unsatisfiable $(k, 2d) - CNF$ *formula.*

Small Values

Lemma *Let F be a minimal unsatisfiable CNF formula. Consider x and $C \in F$ with $x \in vbl(C)$. Then there is a clause D with the property that x is the* unique *variable that appears in C and D with opposite signs.*

Proof. F is minimal $\Rightarrow F \backslash \{C\}$ has satisfying assignment α.
α cannot satisfy C, because F is assumpted to be unsatisfiable.
We switch the value of x to satisfy C.
Now some other clause $D \in F$ is violated.
$\Rightarrow D$ serves the purpose

Lemma 3 *(1) $f(k) \geq k$ for $k \geq 1$ and (2) $l(k) \geq lc(k) \geq k$ for $k \geq 2$*

- $k \geq 1$, F a k-CNF formula over a variable set V, no variable occurring in more than k clauses.
- Consider the incidence graph between clauses and variables.
- Hall's condition for a matching covering all clause-vertices holds.
- An assignment is now defined by letting every variable x that is matched to a clause C map to the value so that it satisfies C.
- The matching property prevents conflicts and no matter how we complete the assignment for unmatched variables it will satisfy all clauses. $\Rightarrow$ (1)

(2) $l(k) \geq lc(k) \geq k$ for $k \geq 2$

- Let $k \geq 2$. We will actually prove $lc(k) \geq \lceil (f(k)+1)/2 \rceil + k - 2$. This yields $\lceil (3k-3)/2 \geq k \rceil$.
- This means we have to show that every unsatisfiable k-CNF formula F contains a clause C with $|\Gamma'(C)| \geq \lceil (f(k)+1)/2 \rceil + k - 1$.
- Minimal unsatisfiable k-CNF formula $G \subseteq F$. G has variable x with $deg_G(x) \geq f(k) + 1$, w. l. o. g. we assume $\overline{x}$ occurs at least $\lceil (f(k)+1)/2 \rceil$ times.
- We choose $C \in G$ with literal x. $\Gamma'_G(C)$ contains all clauses with $\overline{x}$.
- $\forall z \in C \backslash \{x\} \exists D_z \in G$: z is the unique variable that appears in C and D_z with opposite signs.
- $\Rightarrow |\Gamma'_G(C)| \geq \lceil (f(k)+1)/2 \rceil + k - 1$
- With $\Gamma'_F(C) \supseteq \Gamma'_G(C)$ this concludes the argument.

- $f(k) = k$ is known for $k \leq 4$, the best known bounds for $k = 5$ are $5 \leq f(5) \leq 7$.
- $k = 6$ is the first value for which the bound in Lemma 3(1) is known not to be tight: $7 \leq f(6) \leq 11$.

Linear Formulas

- A CNF formula F is linear if
  $|vbl(C) \cap vbl(D)| \leq 1, \quad C, D \in F, C \neq D$
- A hypergraph $H = (V, E)$ is linear if $|e \cap f| \leq 1$ for any two distinct edges $e, f \in E$.
- Given a k-uniform non-2-colorable hypergraph H with m hyperedges, we immediately obtain an unsatisfiable k-CNF formula $F(H)$ with $2m$ clauses

Linear Formulas

- Let $f_{lin}(k)$ be the largest integer so that every linear $(k, f_{lin}(k))$-CNF formula is satisfiable. $f_{lin} \geq f(k) \geq \lfloor 2^k/(ek) \rfloor$

Theorem 6 *Any unsatisfiable linear k-CNF formula has at least*

$$\frac{1}{k}(1 + f_{lin}(k-1))^2 > \frac{4^k}{4e^2k^3}$$

clauses. There exists an unsatisfiable linear k-CNF formula with at most $8k^3 4^k$ clauses.

Remark. $\frac{1}{k}(1 + f_{lin}(k-1)^2) \leq 8k^3 4^k$ follows thus $f_{lin}(k-1) \in O(k^2 2^k)$.

Proof. Similar to the proof for the size of non-2-colourable linear k-uniform hypergraphs in "Problems and results on 3-chromatic hypergraphs and some related questions" (Erdős, Lovász).

Lemma 5 *Let F be a linear k-CNF formula. If there are at most $f_{lin}(k-1)$ variables of degree exceeding $f_{lin}(k-1)$, then F is satisfiable.*

Let X be the set of variables x with $deg_F(x) > f_{lin}(k-1)$. If F is unsatisfiable $|X| > f_{lin}(k-1)$. Therefore the lower bound follows from

$$|F| = \sum_{x \in vbl(F)} deg_F(x) \geq \frac{1}{k}(1 + f_{lin}(k-1))|X| \geq \frac{1}{k}(1 + f_{lin}(k-1))^2$$

Proof of Lemma 5

- For a literal u let $deg_F(u)$ be the degree of the variable underlying u in F.
- First we construct a linear $(k-1)$-CNF formula F' as follows:
- For every clause $C \in F$, let u_C be a literal of C that maximises $deg_F(u_C)$. We write $C' := C \backslash \{u_C\}$, $F' := \{C' | C \in F\}$
- We claim that $deg_{F'}(x) \leq f_{lin}(k-1)$ for all variables x, thus F' and therefore F is satisfiable
- Consider a variable x. Clearly $deg_{F'}(x) \leq deg_F(x)$ and so if $deg_F(x) \leq f_{lin}(k-1)$ we are done.
- Otherwise let $C'_1, \dots, C'_t$, $t = deg_{F'}(x)$ be clauses in F' containing x or $\overline{x}$. There are clauses $C_i, \dots, C_t$ in F such that $C'_i = C_i \backslash \{u_{C_i}\}$, $1 \leq i \leq t$.
- By choice of u_{C_i}, $deg_F(u_{C_i}) \geq deg_F(x) > f_{lin}(k-1)$. Since F is linear, the u_{C_i}'s have to be distinct, thus $t \leq f_{lin}(k-1)$ $\square$

Proof of the upper bound: There exists an unsatisfiable linear k-CNF formula with at most $8k^3 4^k$ *clauses.*

- Take a linear k-uniform hypergraph $H = (V, E)$ with n vertices and m edges.
- We now replace each literal in each clause by its complement with probability $\frac{1}{2}$, independently on each clause. Let F denote the resulting (random) formula.
- Any fixed assignment α has a $1 - 2^{-k}$ chance of satisfying a given clause of F, and thus:
  $Pr[[]\alpha \quad satisfies \quad F] = (1 - 2^{-k})^m \leq e^{-m2^{-k}}$
  There are 2^n distinct assignments, hence by the union bound
  $Pr[[]some \quad \alpha \quad satisfies \quad F] < 2^n e^{-m2^{-k}} = c^{ln(2)n - m2^{-k}}$
- If $\frac{m}{n} \geq ln(2)2^k$, the second expression is at most 1,
  $\Rightarrow$ with positive probability no assignment satisfies F.

- We construct a linear k-uniform hypergraph with few hyperedges, but with a large hyperedge-vertex ratio. Let $q \in \{k, \ldots, 2k\}$ be a prime power.
- Choose $d \in \mathbb{N}$ such that $q^2 ln(2)2^k \leq q^d < q^3 ln(2)2^k$ and set $n := q^d$.
- Consider the d-dimensional vector space $\mathbb{F}_q^d$. By choice of d we have $nln(2)2^k \leq \frac{n^2}{q^2}$, hence we can choose $m := nln(2)2^k$ distinct lines in $\mathbb{F}_q^d$.
- Form each such line arbitrarily select k points and form a hyperedge.
- Let E be the set of all m hyperedges formed this way. Thus, $H = (\mathbb{F}_q^d, E)$ is a k-uniform hypergraph. It is a linear hypergraph, since any pair of distinct lines intersect in at most one point.
- By construction, $\frac{m}{n} = ln(2)2^k$, and $m = nln(2)2^k \leq q^3 ln(2)^2 4^k \leq ln(2)^2 8k^3 4^k$, which proves the upper bound. $\square$

A Sudden Jump in Complexity

- Tovey (1984): For 3-CNF formulas with maximum variable degree $f(3) + 1 = 4$ satisfiability is NP-complete.
- Kratochvíl, Savický and Tuza (1993) generalised this sudden jump: For every fixed $k \geq 3$, satisfiability of $(k, f(k) + 1)$-CNF formulas is NP-complete.
- Berman, Karpinski and Scott (2003) showed that for $(k, f(k) + 1)$-CNF formulas it is even hard to approximate the maximum number of clauses that can be simultaneously satisfied

Theorem 9 *Let $k \geq 3$. Then,*
(1) deciding satisfiability of k-CNF formulas with variable degrees at most $f(k)+1$ is NP-complete
(2) deciding satisfiability of k-CNF formulas with clause neighbourhoods of size at most $max\{k+3, l(k)+2\}$ is NP-complete
(3) deciding satisfiability of k-CNF formulas with clause conflict-neighbourhoods of size at most $lc(k)+1$ is NP-complete

General contruction of F̂ for F so that F̂ is satisfiable iff F is satisfiable

- For a set of $j \geq 2$ variables, $U = \{x_0, x_1, \ldots, x_{j-1}\}$, the 2-CNF formula
$$\{\{x_0, \overline{x_1}\}, \{x_1, \overline{x_2}\}, \ldots, \{x_{j-2}, \overline{x_{j-1}}\}, \{x_{j-1}, \overline{x_0}\}\}$$
is called an equaliser of U.
- Let F be a k-CNF formula, $k \geq 3$. For each variable $x \in vbl(F)$, we replace every occurrence by a new variable inheriting the sign of x in this occurrence.
- This yields a k-CNF formula F' with $|F|$ clauses over a set of $k|F|$ variables.
- For each $x \in vbl(F)$ we add an equaliser for the set of variables that have replaced occurrences of x.
- This gives a set F'' of at most $k|F|$ 2-clauses.
- $\hat{F} := F' \cup F''$ is satisfiable iff F is satisfiable.

- every variable of $\mathrm{vbl}(\hat{\mathrm{F}})$ occurs at most 3 times in $\hat{\mathrm{F}}$
- each k-clause in F' does not share variables with any other clause in F' and the number of its neighbouring 2-clauses in F'' is at most $2k$; at most k of the 2-clauses are in the conflict-neighbourhood
- each 2-clause in F'' neighbours two k-clauses in F' and at most two 2-clauses in F''

Proof of (1) (variable degrees)

- Let $k \geq 3$ and fix some minimal unsatisfiable $(k, f(k)+1)$-CNF formula G.
- Choose some clause C in G and replace one of its literals by $\overline{x}$ for a new variable x to get $G(x)$.
- $G(x)$ is satisfiable, every satisfying assignment has to set x to 0, all variables have degree at most $f(k)+1$ and $deg_{G(x)}(x) = 1$
- Given a k-CNF formula F we first generate $\hat{\mathrm{F}}$. Then we augment each 2-clause in $\hat{\mathrm{F}}$ by $(k-2)$ positive literals of new variables so that it becomes a k-clause.
- For each new variable x we add a copy of $G(x)$ to our formula. By renaming variables in G these copies are chosen so that their variable sets are pairwise disjoint.
- The new formula is satisfiable iff $\hat{\mathrm{F}}$ is satisfiable.
- The maximum variable degree is $max\{3, f(k)+1\}$, which is $f(k)+1$
- This constitutes a polynomial reduction of satisfiability of general k-CNF formulas to satisfiability of k-CNF formulas with maximum variable degree $f(k)+1$. $\square$

Proof of (2) (neighbourhoods)

- Let $k \geq 3$. Fix some minimal unsatisfiable $k - CNF$ formula G where all neighbourhoods have size at most $l(k) + 1$.
- We choose some clause C and replace one of its lieterals by $\overline{x}$ for a new variable x, resulting in a k-CNF formula $G(x)$ that forces x to 0 in every satisfying assignment.
- Starting from a 3-CNF formula F we proceed as before:
- We produce $\hat{\mathrm{F}}$ consisting of 3- and 2-clauses, we augment all clauses to k-clauses with disjoint copies of $G(x)$ for each new variable x.
- A 3-clause in F' had 6 neighbours in $\hat{\mathrm{F}}$ and gained $k - 3$ new neighbours, so there are at most $k + 3$.
- A 2-clause had 4 neighbours and gets an extra neighbour for each of the $k - 2$ new literals, which makes $k + 2$ neighbours.
- In a copy $G(x)$ all clauses stay with a neighbourhood of size at most $l(k) + 1$ except for the special clause C where we have planted the new literal $\overline{x}$. This clause may now have $l(k) + 2$ neighbours.
- $\Rightarrow$ bound of $max\{k + 3, l(k) + 2\}$ and the polynomial reduction from satisfiability of general 3-CNF formulas is completed. $\square$

- Given a variable set $U = \{x_0, x_1, \ldots, x_{j-1}\}, j \geq 2$, let $W = \{z_0, z_1, \ldots, z_{j-1}\}$ be a set of variables disjoint from U. The $(U \cup W)$-equaliser
  $\{\{x_0, \overline{z_0}\}, \{z_0, \overline{x_1}\}, \{x_1, \overline{z_1}\}, \{z_1, \overline{x_2}\}, \ldots,$
  $\{z_{j-2}, \overline{x_{j-1}}\}, \{x_{j-1}, \overline{z_{j-1}}\}, \{z_{j-1}, \overline{x_0}\}\}$

  is called a stretched equaliser of U.
- the 2-clauses in stretched equalisers have a conflict with two other 2-clauses but to at most one of the k-clauses in F'

Proof of (3) (conflict-neighbourhoods)

- $k \geq 3$, fix minimal unsatisfiable k-CNF formula G with conflict-neighbourhood size at most $lc(k)+1$
- Recall from Lemma 4: G must have a pair of clauses C and D which share a unique variable, say y, in a conflicting manner.
- Choose a new variable x and replace y in C by $\overline{x}$. This the building block $G(x)$ forcing x to be 0. The clause C' containing $\overline{x}$ has a conflict-neighbourhood of size at most $lc(k)$.
- Given F, a k-CNF formula, we move on to $\hat{\mathrm{F}}$ and then expand 2-clauses with the help of new variables that are forced to 0 by disjoint copies of $G(x)$.
- In the final product k-clauses in F' have at most k conflict-neighbours, k-clauses obtained from augmenting 2-clauses have at most $3+(k-2)=k+1$ conflict-neighbours, and clauses in copies of $G(x)$ have conflict-neighbourhoods of size at most $lc(k+1)$.
- The maximum size of a conflict neighbourhood is $max\{k+1, lc(k)+1\}$ which equals $lc(k)+1$. $\square$

Open Problems

Open Problem 1. *Is it possible to improve any of the known lower bounds on $f(k)$, $l(k)$, and $lc(k)$ by a constant factor?*

Open Problem 2. *Is there a constant $c_0 > 1$ with $f(k) \geq c_0 l(k)/k$ for k large enough?*

Open Problem 3. *Is there a constant $c_1 > 1$ such that $l(k) \geq c_1 lc(k)$ for k large enough?*

Open Problem 4. *Are the functions $f(k)$, $l(k)$ and $lc(k)$ computable?*

Thank you!