Cliques enumeration and tree-like resolution proofs

Massimo Lauria
massimo.lauria@uniroma1.it
Dipartimento di Scienze Statistiche - Sapienza Università di Roma, Italy

Abstract

We show the close connection between the enumeration of cliques in a k-clique free graph G, the running time of DPLL-style algorithms for k-clique problem, and the length of tree-like resolution refutations for formula $\mathrm{Clique}(G, k)$, which claims that G has a k-clique. The length of any such tree-like refutation is within a "fixed parameter tractable" factor from the number of cliques in the graph. We then proceed to drastically simplify the proofs of the lower bounds for the length of tree-like resolution refutations of $\mathrm{Clique}(G, k)$ shown in [Beyersdorff et at. 2013, Lauria et al. 2017], which now reduce to a simple estimate of said quantity.

Key words: Proof Complexity, Clique, Resolution, Decision Tree

1. Introduction

The k-clique problem asks whether a graph has a set of k pairwise connected vertices, i.e., a *k-clique*. Being one of the standard NP-complete problems it seems very unlikely that it has an efficient algorithm, even in approximation [26, 24]. The brute force approach to solve the problem on a graph of n vertices is to check all of the $\approx n^k$ sets of k vertices. Even more sophisticated algorithms (for example [38]) still run in time $n^{\Omega(k)}$ in the worst case. Is this the best we can hope for? If one believes the Exponential Time Hypothesis we cannot even get down to $n^{o(k)}$ [25, 29]. The problem is so difficult that we would like to prove its hardness without using unproved hypotheses. Unfortunately we cannot do that unless we restrict the computational model. For example we know that circuits that solve k-clique must have size $n^{\Omega(k)}$ if they are restricted to be either of constant depth [36] or without negation gates [35]. These results hold even if we are satisfied with solving k-clique only asymptotically almost surely in the Erdős-Renyi model, where the edges of the graph are picked (independently) at random according to the appropriate density.

In this paper we focus on *decision trees* and DPLL-style algorithms [16, 15] for the k-clique problem, namely algorithms that explore the space of solutions by guessing possible members for the k-clique. In case a choice happens not to be fruitful, the algorithm backtracks and tries other possibilities. The execution of a DPLL-style algorithm is represented by a decision tree, therefore identify the two concepts.

Given a graph G with no k-cliques, we can build a propositional formula $\mathrm{Clique}(G, k)$ that falsely claims that G has a clique of size k. A correct algorithm that searches for a

k-clique in G must fail, and its trace be an efficiently verifiable proof that $\mathrm{Clique}(G, k)$ is unsatisfiable. When this algorithm is simple enough, as in the case of decision trees, the proof can be written down in a simple language as well. If such language does not allow for short proofs of unsatisfiability of $\mathrm{Clique}(G, k)$, the running time of the algorithm must be long too. Decisions trees produce, on failure, *tree-like resolution* proofs of unsatisfiability of formula $\mathrm{Clique}(G, k)$, for which we can prove strong lower bounds. By studying how the proof is written down, we can ignore how the algorithm finds it. A lot of those details can be abstracted away up to the point that we can see the proof as the result of a non-deterministic, i.e., "very lucky", proof search. Concretely, it means that a lower bound for the length of tree-like resolution proofs applies to all decision trees, regardless of how smart their decision strategy is.

The field of *proof complexity* [14, 6, 37] studies the length of proofs of propositional unsatisfiability. The languages in which such proofs are written down are called *proof systems*. The more general the proof system, more general is the class of algorithms that the system captures. Resolution is definitely the most famous and it is at the core of state-of-the-art SAT algorithms [4, 30, 31]. The first proof length lower bound for resolution was proven in [23], followed by more general lower bound techniques [27, 8]. In practical SAT solving memory is a resource as precious as time, hence [19, 1] developed a notion of proof space that is supposed to model memory usage. Another important proof complexity parameter of resolution proofs is the "width", i.e., the size of each proof line. Estimating the required width of a resolution proof is a proxy to estimate the required length [8] or the required space [3, 21]. The correspondence between width and space is not very tight [33], which leads to study resolution space directly, using various pebbling games to model memory allocation. While it is hard to know how much memory is needed to win these games in general [22, 13], it is possible to use and combine well understood cases of these games in order to prove resolution space lower bounds and resolution length vs space trade-offs [32].

We now go back to the problem of determining the resolution proof complexity of $\mathrm{Clique}(G, k)$. For $k \approx n$, super polynomial lower bounds have been proved using the size-width relation [5, 8], which is by now a standard technique in proof complexity. For $k \ll n$ the problem is still open. Neither the size-width relation nor interpolation [27], which is the other main tool used in literature to prove resolution lower bounds, give anything for $k \ll n$. New techniques may be necessary to solve this problem.

In this paper we focus on the *tree-like resolution*, which is a restricted form of resolution that captures algorithms based on decision trees. Tree-like resolution is weaker than general resolution [12], but we understand better its proof length and space [9, 20]. There are three types of graphs for which the clique formula is known to require tree-like resolution refutations of length $n^{\Omega(k)}$. In [10] the lower bound is proved for the complete $(k-1)$-partite graph, as well as for Erdős-Renyi random graph with appropriate edge density. A lower bound of $n^{\Omega(\log(n))}$ holds for Ramsey graphs, i.e., graphs of n vertices that have neither a $2\log(n)$-clique nor a $2\log(n)$ independent set [28]. In this paper we do not improve on these results, but we drastically simplify them by showing a connection between the length of tree-like resolution refutations of $\mathrm{Clique}(G, k)$ and the number of cliques in G.

The paper is organized as follows. In Section 2 we give the necessary definitions and notations. In Section 3 we show the close correspondence between the number of cliques in a graph and the length of tree-like resolution refutations for the clique formulas on that graph. In Section 4 we show classes of k-clique free graphs with many cliques, hence they need large refutations for the corresponding clique formulas.

2. Preliminaries

In this paper we consider simple undirected loop-less graphs $G = (V, E)$. We write $\Gamma(v)$ to denote the set of vertices in V that are neighbors of a vertex $v \in V$. For an arbitrary set of vertices $U \subseteq V$ we denote as $\Gamma(U)$ the set of vertices $\bigcap_{u \in U} \Gamma(u)$, i.e., the common neighbors of U in G. A clique of G is a set of vertices so that there is an edge between any two of them. We denote as $\mathcal{C}(G)$ the set of cliques of G. We denote as $[m]$ the set of integers from 1 to m.

A CNF formula over a set of variables is a conjunction of distinct clauses, each of them being the disjunction of distinct literals. A literal is either an occurrence of a variable or its negation. D is a subclause of a clause C when D is a disjunction of literals contained in C. We indicate that D is a subclause of C with notation $C \subseteq D$.

The *k-clique formula* $\mathrm{Clique}(G, k)$ is a CNF formula over variables $x_{i,v}$ for every $v \in V$ and $i \in [k]$, where the boolean variable $x_{i,v}$ indicates whether the i-th vertex of the clique is v. The formula is the conjunction of clauses

$$\bigvee_{v \in V} x_{i,v} \qquad \forall i \in [k] \; , \tag{1a}$$

$$\neg x_{i,u} \vee \neg x_{j,v} \qquad \forall i, j \in [k], i \neq j, \forall u, v \in V, \{u, v\} \notin E \; , \tag{1b}$$

$$\neg x_{i,u} \vee \neg x_{i,v} \qquad \forall i \in [k], \forall u, v \in V, u \neq v \; . \tag{1c}$$

Clauses (1a) are called clique axioms, clauses (1b) are called edge axioms, and clauses (1c) are called functionality axioms. Clearly $\mathrm{Clique}(G, k)$ is satisfiable if and only if G contains a clique of k vertices, and this holds even without the functionality axioms.

Tree-like resolution and Decision trees. The proofs of unsatisfiability of $\mathrm{Clique}(G, k)$ formula that we consider in the paper are sequences of logical inference steps obtained using the *resolution rule*, an inference rule that derives a clause from two clauses, called premises, as follows

$$\frac{A \vee x \qquad B \vee \neg x}{A \vee B} \; . \tag{2}$$

To apply rule (2) the two premises must contain, respectively, the positive and negative literal of some variable x, and thus we say that we *resolve* the two clauses over variables x. The derived clause is their *resolvant*, and it is true whenever both premises are true.

A *tree-like resolution proof* of a clause C from some CNF formula F is a rooted binary tree, directed from the leaves to the root, where each node in the tree is labeled by a clause over the variables of F. The clauses labeling the nodes in the proof must have the following properties:

- no clause contains both a literal and its negation;
- the clause labeling an internal node is the resolvant of the clauses labeling its two immediate predecessors;

- the clause at the root is a subclause of C;
- any clause labeling a leaf is a subclause of some clause of F.

A tree-like resolution *refutation* is a proof of the empty clauses. The *length* of a tree-like resolution proof is the number of its leaves.[1]

A *decision tree* that computes a function over some variables $Y = \{y_1, \ldots, y_n\}$ with values in some set $\mathcal{R}$ is a rooted binary tree, directed from the root to the leaves. Every internal node is labeled by a *query*, i.e., some variable in Y. The two directed edges going out from an internal node are labeled by 0 and by 1 and correspond to the two possible answers to the query. Leaves are labeled by values in $\mathcal{R}$. Given an assignment $\rho : \{y_i\}_{i=1}^n \to \{0, 1\}$, we define $\mathrm{path}(\rho)$ as follows: start at the root and whenever at an internal node ν labeled by a variable y_i, add to the path the outgoing edge (ν, ν') labeled by $\rho(y_i)$, and move to ν'. The process eventually ends up at a leaf, where it stops. The value of the function for the assignment ρ is the label of the leaf reached by $\mathrm{path}(\rho)$. The *size* of a decision tree is the number of its leaves.

It is useful to associate partial assignments for the variables $y_1, \ldots, y_n$ to the paths in the decision tree. A directed edge corresponds to a query y_i and an answer b, hence we associate it to the partial assignment $\{y_i \to b\}$. The partial assignment associated to a path is the union of the assignments associated to the edges in it. Since no variable occurs twice in any path, the assignment is well defined. For each node ν of the tree we denote as ρ_ν the partial assignment corresponding to the path from the root to ν.

In this paper we are interested in decision trees that solve the *search problem* for an unsatisfiable CNF formula. The search problem, given a formula F over variables Y and a total assignment ρ on them, asks to output a clause of F falsified by ρ. A decision tree that solves the search problem is a decision tree over the variables of F, where each leaf ℓ is labeled by a clause of F that is falsified by the partial assignment ρ_ℓ.

Decision trees and tree-like resolution refutations are tightly connected.

Fact ([7]). *Any tree-like resolution refutations of length s for F can be efficiently transformed into a decision tree of size s for the search problem over F. The opposite transformation exists and is efficient as well.*

In this paper we use a version of Markov's inequality that claims that for any positive random variable X with expectation $\mathbb{E}X$, it holds that $\Pr[X \geq \delta] \leq \mathbb{E}X/\delta$. A direct corollary is that an integer positive random variables with expected value $o(1)$ is zero with probability $1 - o(1)$.

3. Cliques and tree-like refutations

In this section we show that for a k-clique free graph G the length of a tree-like refutation of $\mathrm{Clique}(G, k)$ is roughly the number of cliques in G, when $k \ll |V(G)|$. We first show the lower bound, which holds for any k.

Lemma 1. *Let G be a k-clique free graph. The length of any tree-like resolution refutation of $\mathrm{Clique}(G, k)$ is at least $|\mathcal{C}(G)|$.*

[1] This definition of tree-like resolution differs a bit from others in literature. Our definition does not need a weakening rule and it is still complete for clauses with no opposite literals.

Proof. As we already discussed in the preliminaries, the length of the shortest tree-like resolution refutation for $\mathrm{Clique}(G, k)$ is the same as the size of the smallest decision tree for the search problem on $\mathrm{Clique}(G, k)$. We fix T to be such decision tree, and for any fixed clique $K \in \mathcal{C}(G)$ we define a walk through T starting at the root and reaching some leaf ℓ_K. To prove the lemma we show that for distinct cliques in $\mathcal{C}(G)$ the corresponding walks reach distinct leaves, and therefore T must have at least $|\mathcal{C}(G)|$ leaves. When the walk is at some node ν labeled by query $x_{i,v}$ the answer (i.e., the direction to take) is determined according to these rules:

(i) answer 0 if ρ_ν contains some $\{x_{j,w} \to 1\}$ where either $w = v$ or $j = i$; otherwise

(ii) answer 0 if $v \notin K$; otherwise

(iii) answer 1.

There is always a well defined answer for any query at any internal node of T, therefore the walk reaches one of the leaves, which we call ℓ_K, labeled by some clause of $\mathrm{Clique}(G, k)$ which is falsified by the answers along the walk.

The rules ensure that neither edge nor functionality axioms get falsified, therefore the assignment ρ_{ℓ_K} must falsify a clique axiom, i.e., must contains $\{x_{i,v} \to 0\}_{v \in V(G)}$ for some $i \in [k]$. This implies that for any walk and any $v \in V(G)$, there is some query $x_{i,v}$ along the walk which has been answered according to either rule (ii) or rule (iii). Now consider the two walks that correspond to two distinct clique K_1 and K_2 in $\mathcal{C}(G)$. The walks proceed identically along the same branch of T until they meet for the first time a node with query $x_{j,w}$ where w belongs to either $K_1 \setminus K_2$ or $K_2 \setminus K_1$ and either rule (ii) or rule (iii) applies. In that case the walks diverge, reaching two distinct leaves ℓ_{K_1} and ℓ_{K_2}. □

It would be nice to prove the converse of Lemma 1, namely to build a tree-like resolution refutation of length at most $\mathrm{poly}(|V(G)|) \cdot |\mathcal{C}(G)|$. This is not possible: consider a graph with n vertices which is made by the union of an isolated vertex and a complete graph of $n-1$ vertices. The $\mathrm{Clique}(G, n)$ formula for this graph is essentially a pigeonhole principle formula and therefore is hard for resolution [23]. There are two possible workarounds.

- The approach of [5] is to add *monotonicity axioms* to the formula to enforce that the k indexes point to vertices in an increasing fashion. Vertices of the graph are enumerated as $v_1, v_2, \ldots v_n$, and monotonicity axioms are

$$\neg x_{j_1,v_{i_1}} \vee \neg x_{j_2,v_{i_2}} \qquad \forall\, 1 \le i_2 < i_1 \le n, \quad \forall\, 1 \le j_1 < j_2 \le k\ . \tag{3}$$

- We treat k-clique as a parameterized problem [17], for parameter $k \ll |V(G)|$.

In the latter case we can show a tree-like resolution refutation of $\mathrm{Clique}(G, k)$ of length at most $f(k) \cdot n \cdot |\mathcal{C}(G)|$ for any k-clique free graph G of n vertices.

Theorem 2. *Let G be a k-clique free graph over n vertices. There is a tree-like refutation of* $\mathrm{Clique}(G, k)$ *of length $f(k) \cdot n \cdot |\mathcal{C}(G)|$, for some function f.*

Proof. We argue the existence of such refutation by building the corresponding decision tree. Consider the order $v_1, v_2, \ldots, v_n$ of the vertices of G and fix $\mu = \emptyset$. We use μ to keep track of the partial mapping between $[k]$ and the vertices of the clique identified by the answers to the queries. The tree queries the variables x_{1,v_j} for j going from 1 to n in order, and stops when one query is answered 1. If none is, then the clique axiom for $i = 1$ is falsified. Otherwise we add $\{1 \to v_{j_1}\}$ to μ, where $x_{1,v_{j_1}}$ is the query that has been answered 1. The tree repeats this process for $i > 1$ up to k, querying the variables x_{i,v_j} for j from 1 to n in order, until one of these variables in answered 1. If none is, then the i-th clique axiom is falsified. If instead some $x_{i,v_{j_i}}$ is answered 1 then either $v_{j_i} \notin \Gamma(\mathrm{rng}(\mu))$ and then we are at a leaf because an edge axiom has been falsified, or we add $\{i \to v_{j_i}\}$ to μ and continue with index $i+1$. At every node in the tree there is a corresponding value of μ which identifies a clique of G. In particular we can associate a specific value of μ to every leaf.

To bound the size of the tree observe that if the last mapping added to μ is $\{i \to v_{j_i}\}$, then the tree queries all variables x_{i+1,v_j}. The branch reaches a leaf as soon as the answers are all 0 or when one answer is 1 and violates an edge axiom. In all other cases μ gets extended. Hence the branching reaches at most $n + 1$ leaves with each specific value of μ. How many values of μ occur in total? Recall that μ is a mapping from some indices in $[k]$ to the vertices of some clique in G, therefore they are at most $f(k) \cdot |\mathcal{C}(G)|$. $\square$

4. Graphs with many cliques

In this section we use Lemma 1 to obtain much easier proofs of the lower bounds shown in [10, 28]. The first example is the complete $(k-1)$-partite graph of n vertices, where we assume that $k-1$ divides n. The graph is made by $(k-1)$ blocks of $\frac{n}{k-1}$ vertices each. There are no edges within a block and there is an edge between any two vertices in different blocks. The graph has obviously $n^{\Omega(k)}$ cliques, so the next theorem follows immediately from Lemma 1.

Theorem 3 ([10])**.** *Let G be the complete $(k-1)$-partite over n vertices. Any tree-like resolution refutation for* $\mathrm{Clique}(G, k)$ *has length $n^{\Omega(k)}$.*

A more interesting example is the random graph $\mathcal{G}(n, p)$ where $p = n^{-(1+\epsilon)\frac{2}{k-1}}$ for any $\epsilon > 0$. For constant k in $G \sim \mathcal{G}(n, p)$ the expected number of k-cliques is

$$\binom{n}{k}\left(n^{-(1+\epsilon)\frac{2}{k-1}}\right)^{\binom{k}{2}} < n^k n^{-(1+\epsilon)k} = \frac{1}{n^{\epsilon k}} \ , \tag{4}$$

and by Markov's inequality there is none with high probability. Nevertheless any tree-like resolution refutation of $\mathrm{Clique}(G, k)$ has length $n^{\Omega(k)}$, again with high probability.

Theorem 4 ([10])**.** *For any constant $\epsilon > 0$, let $G \sim \mathcal{G}(n, p)$ where $p = n^{-(1+\epsilon)\frac{2}{k-1}}$. Graph G has no k-clique with high probability, and yet any tree-like resolution refutation of* $\mathrm{Clique}(G, k)$ *has length $n^{\Omega(k)}$.*

For $p = 1/2$ the interesting value of k is $2\log n + 2$. For $G \sim \mathcal{G}(n, 1/2)$ the expected number of k-cliques with $k = 2\log n + 2$ is at most

$$\binom{n}{k}\left(\frac{1}{2}\right)^{\binom{k}{2}} < \frac{1}{n^{O(1)}} \tag{5}$$

therefore by Markov's inequality G has no $(2\log n+2)$-clique with probability $1-o(1)$.

Theorem 5 ([10])**.** *Let $G \sim \mathcal{G}(n, 1/2)$. With probability $1 - o(1)$ graph G has no $(2\log n+2)$-clique, and yet any tree-like resolution refutation of* $\mathrm{Clique}(G, 2\log n+2)$ *has length $n^{\Omega(\log n)}$.*

Both Theorems 4 and 5 can be easily proved using Lemma 1. To lower bound the number of cliques we use the following extension lemma. Various versions of this extension lemma exist in literature, e.g., in [10, 28]. We formulate one that applies to our two settings of parameters.

Lemma 6. *Let $G \sim \mathcal{G}(n, p)$ with $p \leq \frac{1}{2}$. With probability $1 - o(1)$ graph G has $n^{\Omega(-\frac{\log n}{\log p})}$ distinct cliques.*

Proof. We want to find many different cliques in G. First we show that with high probability an arbitrary set of $\Omega\left(-\frac{\log n}{\log p}\right)$ vertices has a polynomial number of common neighbors. We fix a "canonical" clique size $k' = -\frac{\log n}{2\log p}$, which is at most $\frac{\log n}{2}$, and we estimate $|\Gamma(R)|$ for an arbitrary set $R \subseteq V(G)$ with $|R| < k'$. For a fixed set of vertices R of size less than k' we consider the random variables X_v for $v \in V(G) \setminus R$, where X_v is 1 if $v \in \Gamma(R)$ and 0 otherwise. Randomness of X_v is with respect of the sampling of G. These are $n - |R| = n(1 - o(1))$ independent Bernoulli variables with expectation $p^{|R|} > p^{k'} = n^{-1/2}$. The expectation $|\Gamma(R)|$ is at least $\sqrt{n}\,(1 - o(1))$. The probability that $|\Gamma(R)| < \sqrt{n}/2$ is, by Chernoff Bound [18, Theorem 1.1], at most

$$\exp\left(\frac{-(1 - o(1))\sqrt{n}}{8}\right) . \tag{6}$$

Since there are at most $n^{O(\log n)}$ possible sets R of size less than k', by union bound we get that all such R have, simultaneously, at least $\sqrt{n}/2$ common neighbors with probability $1 - o(1)$ with respect to the sampling of G.

Now we can easily show that there must be many "ordered" cliques: pick a sequence of vertices $v_1, v_2, v_3, \ldots$, with each $v_i \in \Gamma(\{v_1, \ldots, v_{i-1}\})$, until the sequence cannot be extended anymore. For each v_i in the sequence there are at least $\sqrt{n}/2$ choices, as long as $i \leq k'$, therefore we produce $n^{\Omega(k')}$ distinct ordered cliques.

Different ordered cliques could potentially correspond to the same clique, therefore we must discount repetitions. We will show that with probability $1 - o(1)$ with respect to the sampling of G the latter has no clique larger than $4k'$, so no clique corresponds to more than $4k'!$ ordered cliques, and there must be at least $n^{\Omega(k')}$ cliques in the end.

To bound the size of the maximum clique in the graph we use the usual argument: the expected number of cliques of size $4k'$ in G is

$$\binom{n}{4k'} p^{\binom{4k'}{2}} \approx \frac{1}{n^{\Omega(k)}} \tag{7}$$

hence by Markov's inequality the probability that there is one is $o(1)$. $\square$

Lemma 1 and 6 immediately imply Theorems 4 and 5 by setting $p = n^{-(1+\epsilon)\frac{2}{k-1}}$ and $p = \frac{1}{2}$, respectively.

A generalization of Theorem 5, discussed in [28], is the problem of witnessing that a graph is c-Ramsey, i.e., that it has neither a $c\log n$-clique nor independent set of size

$c \log n$. The results in [28] actually show that even witnessing the absence of $c \log n$-cliques in a c-Ramsey graph is hard.

Theorem 7 ([28])**.** *Let G be a c-Ramsey graphs on n vertices. Any tree-like resolution refutation of* $\mathrm{Clique}(G, c \log n)$ *has length* $n^{\Omega(\log n)}$.

This theorem is a generalization of Theorem 5 because a random graph has neither an independent set nor a clique of size $2 \log n$, with high probability. Therefore it is c-Ramsey for any $c \geq 2$. To prove the theorem we will need some facts about the edge density of Ramsey graphs. For any two non empty disjoint sets of vertices A and B in a graph G, the *edge density* $d(A, B)$ is the ratio between the number of edges with one end in A and the other in B, and $|A| \cdot |B|$. It turns out that in most of a c-Ramsey graph the edge density must be balanced.

Lemma 8 ([34])**.** *There exist constants $\beta > 0$, $\delta > 0$ such that if G is a c-Ramsey graph, then there is a set $S \subseteq V(G)$ with $|S| \geq n^{\frac{3}{4}}$ such that, for all $A, B \subseteq S$, if $|A|, |B| \geq |S|^{1-\beta}$ then $\delta \leq d(A, B) \leq 1 - \delta$.*

Corollary 9. *There exist constants $\beta > 0$, $\delta' > 0$ such that if G is a c-Ramsey graph, then there is a set $S \subseteq V(G)$ with $|S| \geq n^{\frac{3}{4}}$ such that, for all $A \subseteq S$ of size at least $2|S|^{1-\beta}$, at least $\delta'|A|$ vertices v in A have $|\Gamma(v) \cap A| \geq \delta'|A|$.*

Proof. Fix S, β and δ as in Lemma 8. Consider a set $A \subseteq S$ of size $2|S|^{1-\beta}$, and split arbitrarily A in two disjoint parts A_0 and A_1 with with $|A_0| = |A_1| = |A|/2$, and set U to be the set of vertices in A_0 with at least $\delta|A|/4$ neighbors in A_1. The total number of edges between A_0 and A_1 is at most $|U||A_1| + \delta|A_0 \setminus U||A|/4$, and at least $\delta|A_0||A_1|$ by Lemma 8. Therefore $|U| \geq \delta|A|/4$ and we can fix $\delta' = \delta/4$. $\square$

Proof of Theorem 7. By the Lemma 1 we just need to show that there are $n^{\Omega(\log n)}$ cliques in a c-Ramsey graph. We pick a sequence of vertices $v_1, v_2, v_3, \ldots$, with each $v_i \in \Gamma(\{v_1, \ldots, v_{i-1}\})$. We will show that there are at least $n^{\Theta(1)}$ choices for each new vertex v_i and that the sequence can be extended for at least $\Omega(\log n)$ steps.

Using Corollary 9 we can pick an initial set of vertices V_1 of size $n^{3/4}$, and constants $\beta > 0$ and $\delta' > 0$ as stated in the corollary itself. We fix $t = \lfloor \frac{\log(n^{3\beta/4})}{\log 1/\delta'} \rfloor$, so that $n^{3/4} \cdot \delta'^t \geq n^{3(1-\beta)/4}$. Observe that $t = \Omega(\log n)$.

For $1 \leq i \leq t$ we pick some $v_i \in V_i$ with $|\Gamma(v_i) \cap V_i| \geq \delta'|V_i|$, and we fix V_{i+1} to be $\Gamma(v_i) \cap V_i$. This is possible because we start with $|V_1| = n^{3/4}$ and at each step $i \leq t$ we can apply Corollary 9 to ensure that $|V_{i+1}| \geq n^{3/4} \cdot (\delta')^{i+1}$, given that $|V_i| \geq n^{3/4} \cdot (\delta')^i \geq n^{3(1-\beta)/4}$.

The choice of each v_i is made among $\delta'|V_i| = n^{\Theta(1)}$ vertices therefore we produce $n^{\Omega(t)} = n^{\Omega(\log n)}$ ordered cliques of length $< c \log n$. At most $(c \log n)!$ ordered cliques correspond to a clique in the graph, hence the number of cliques in G is $n^{\Omega(\log n)}$. An application of Lemma 1 concludes the proof. $\square$

Conclusion

The performance of a DPLL algorithm for k-clique that runs on a k-clique free graph G is equivalent to the proof complexity of a tree-like resolution refutation of the corresponding $\mathrm{Clique}(G, k)$ formula. In this paper we have shown that the latter is closely related to the number of cliques in G itself. As a consequence, we get new and simpler proofs of some lower bounds in [10, 28].

The main problem left open in this work, raised for the first time in [11], is to prove these lower bounds for resolution. Since resolution has refutations of $\mathrm{Clique}(G,k)$ of length $2^k \mathrm{poly}(n)$ for any $(k-1)$-colorable graph, we already know that Theorem 3 and in particular Lemma 1 do not generalize [10]. On the other hand it is likely that Theorems 4, 5, and 7 do, even though we do not know how to prove that. Partial progress was made in the recent paper [2], which generalizes Theorems 4 and 5 to regular resolution, a subsystem of resolution much more powerful than the tree-like one.

During part of this work the author was funded by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement ERC-2014-CoG 648276 AUTAR). This work was partially done during the Special Semester on Computational and Proof Complexity organized at the Chebyshev Laboratory of St.Petersburg, in the Spring of 2016. Special thanks go to Nicola Galesi and Olaf Beyersdorff for many conversations and joint works about the k-clique problem.

References

[1] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Space complexity in propositional calculus. *SIAM J. Comput.*, 31(4):1184–1211, 2002.

[2] A. Atserias, I. Bonacina, S. F. de Rezende, M. Lauria, J. Nordström, and A. A. Razborov. Clique is hard on average for regular resolution. In *STOC 2018, 50th ACM Symposium on Th. of Computing*, 2018.

[3] Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *J. Comput. Syst. Sci.*, 74(3):323–334, 2008.

[4] Roberto J. Bayardo Jr. and Robert Schrag. Using CSP look-back techniques to solve real-world SAT instances. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI '97)*, pages 203–208, July 1997.

[5] Paul Beame, Russell Impagliazzo, and Ashish Sabharwal. The resolution complexity of independent sets and vertex covers in random graphs. *Comput. Complex.*, 16(3):245–297, 2007.

[6] Paul Beame and Toniann Pitassi. Propositional proof complexity: Past, present, and future. In *Current Trends in Theoretical Computer Science*, pages 42–70. World Scientific Publishing, 2001.

[7] Eli Ben-Sasson, Russell Impagliazzo, and Avi Wigderson. Near optimal separation of tree-like and general resolution. *Combinatorica*, 24(4):585–603, 2004.

[8] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. *J. ACM*, 48(2):149–169, 2001.

[9] O. Beyersdorff, N. Galesi, and M. Lauria. A characterization of tree-like resolution size. *Information Processing Letters*, 113(18):666–671, 2013.

[10] Olaf Beyersdorff, Nicola Galesi, and Massimo Lauria. Parameterized complexity of dpll search procedures. *ACM Transactions on Computational Logic (TOCL)*, 14(3):20, 2013.

[11] Olaf Beyersdorff, Nicola Galesi, Massimo Lauria, and Alexander A. Razborov. Parameterized bounded-depth frege is not optimal. *ACM Trans. Comput. Theory*, 4(3):7:1–7:16, September 2012.

[12] Maria Luisa Bonet and Nicola Galesi. Optimality of size-width tradeoffs for resolution. *Computational Complexity*, 10(4):261–276, 2001.

[13] S. M. Chan, M Lauria, J. Nordström, and M. Vinyals. Hardness of approximation in pspace and separation results for pebble games. In *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, FOCS '15, pages 466–485, Washington, DC, USA, 2015. IEEE Computer Society.

[14] Stephen A. Cook and Robert Reckhow. The relative efficiency of propositional proof systems. *Journal of Symbolic Logic*, 44(1):36–50, March 1979.

[15] M. Davis, G. Logemann, and D. Loveland. A machine program for theorem-proving. *Commun. ACM*, 5:394–397, July 1962.

[16] M. Davis and H. Putnam. A computing procedure for quantification theory. *J. ACM*, 7:201–215, July 1960.

[17] R. Downey and M. Fellows. *Fundamentals of Parameterized Complexity*. Springer-Verlag, 2013.

[18] Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009.

[19] Juan Luis Esteban and Jacobo Torán. Space bounds for resolution. *Information and Computation*, 171(1):84–97, 2001. Preliminary versions of these results appeared in *STACS '99* and *CSL '99*.

[20] Juan Luis Esteban and Jacobo Torán. A combinatorial characterization of treelike resolution space. *Information Processing Letters*, 87(6):295–300, 2003.

[21] Yuval Filmus, Massimo Lauria, Mladen Miksa, Jakob Nordström, and Marc Vinyals. From small space to small width in resolution. In *31st International Symposium on Theoretical Aspects of Computer Science, STACS*, pages 300–311, 2014.

[22] John R Gilbert, Thomas Lengauer, and Robert Endre Tarjan. The pebbling problem is complete in polynomial space. *SIAM Journal on Computing*, 9(3):513–524, 1980.

[23] A. Haken. The intractability of resolution. *Theoretical Computer Science*, 39:297–308, 1985.

[24] J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999. Preliminary version in *FOCS '96*.

[25] R. Impagliazzo and R. Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

[26] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Springer, 1972.

[27] Jan Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *The Journal of Symbolic Logic*, 62(2):457–486, 1997.

[28] M. Lauria, P. Pudlák, V. Rödl, and N. Thapen. The complexity of proving that a graph is ramsey. *Combinatorica*, 37(2):253–268, Apr 2017.

[29] Daniel Lokshtanov, Dániel Marx, Saket Saurabh, et al. Lower bounds based on the exponential time hypothesis. *Bulletin of EATCS*, 3(105), 2013.

[30] João P. Marques-Silva and Karem A. Sakallah. GRASP: A search algorithm for propositional satisfiability. *IEEE Transactions on Computers*, 48(5):506–521, May 1999. Preliminary version in *ICCAD '96*.

[31] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an efficient SAT solver. In *Proceedings of the 38th Design Automation Conference (DAC '01)*, pages 530–535, June 2001.

[32] J. Nordström. Pebble games, proof complexity and time-space trade-offs. *Logical Methods in Computer Science*, 9:15:1–15:63, September 2013.

[33] Jakob Nordström. Narrow proofs may be spacious: separating space and width in resolution. In *STOC*, pages 507–516, 2006.

[34] H.J. Prömel and V. Rödl. Non-ramsey graphs are c log n-universal. *Journal of Combinatorial Theory, Series A*, 88(2):379–384, 1999.

[35] B. Rossman. The monotone complexity of k-clique on random graphs. *SIAM Journal on Computing*, 43(1):256–279, 2014. Preliminary version in *FOCS '10*.

[36] Benjamin Rossman. On the constant-depth complexity of k-clique. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 721–730. ACM, 2008.

[37] Nathan Segerlind. The complexity of propositional proofs. *Bulletin of symbolic Logic*, 13(4):482–537, 2007.

[38] V. Vassilevska and R. Williams. Finding, minimizing, and counting weighted subgraphs. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 455–464. ACM, 2009.