A super-polynomial separation between resolution and cut-free sequent calculus

Theodoros Papamakarios
Department of Computer Science, University of Chicago, Chicago

Abstract

We show a quadratic separation between resolution and cut-free sequent calculus width. We use this gap to get, for the first time, first, a super-polynomial separation between resolution and cut-free sequent calculus for refuting CNF formulas, and secondly, a quadratic separation between resolution width and monomial space in polynomial calculus with resolution. Our super-polynomial separation between resolution and cut-free sequent calculus only applies when clauses are seen as disjunctions of unbounded arity; our examples have linear size cut-free sequent calculus proofs writing, in a particular way, their clauses using binary disjunctions. Interestingly, this shows that the complexity of sequent calculus depends on how disjunctions are represented.

2012 ACM Subject Classification Theory of computation → Proof complexity

Keywords and phrases Proof Complexity, Resolution, Cut-free LK

Digital Object Identifier 10.4230/LIPIcs.MFCS.2023.75

Acknowledgements We wish to thank Alexander Razborov for numerous suggestions and remarks that greatly improved the presentation of the paper.

1 Introduction

Whether cut-free sequent calculus can polynomially simulate resolution for refuting CNF formulas is a question existing since the beginnings of proof complexity. It was first raised in [13] and iterated e.g. in [25]. Cook and Reckhow [13] show that in the tree-like case, there are examples where resolution can have exponentially smaller proofs. Arai, Pitassi and Urquhart [3] point out that the answer may heavily depend on how clauses are represented. A clause consisting of the literals, say $\ell_1, \ell_2, \ell_3, \ell_4$, can be seen as either a single disjunction of arity four, or as a series of applications of binary disjunctions, for example $(\ell_1 \vee \ell_2) \vee (\ell_3 \vee \ell_4)$, and this can have a profound impact on the complexity of sequent calculus proofs. The result of Cook and Reckhow above applies in the case where clauses are seen as single disjunctions of unbounded arity, or the case where the order in which the binary disjunctions are applied is fixed. If we are free to choose the order, then tree-like cut-free sequent calculus can quasi-polynomially simulate tree-like resolution, and this is optimal [3]. In the DAG-like case, and if we are free to choose the order in which binary disjunctions are applied, Reckhow [21] shows that cut-free sequent calculus can polynomially simulate regular resolution, and Arai [2] shows that it can polynomially simulate resolution for refuting k-CNF formulas F, where k grows at most logarithmically as a function of the size of the shortest resolution refutation of F . However, the general question has remained unresolved.

We define the width of a sequent calculus proof as the maximum number of formulas occurring in a sequent of the proof. This definition extends in a natural way the concept of the width of a resolution proof to stronger proof systems. Furthermore, it allows for a simple, abstract characterization of sequent calculus width generalizing the characterization of Atserias and Dalmau for resolution width [4]. Using this characterization, we show a quadratic gap between resolution width and cut-free sequent calculus width. Resolution is a sequent calculus system that has only atomic cuts, so this says that including atomic cuts in cut-free sequent calculus can shorten the width of proofs. Utilizing then this gap, we show a

super-polynomial separation between cut-free sequent calculus and resolution size. To put it in other words, atomic cuts can super-polynomially decrease the size of proofs. This result applies only when clauses are seen as disjunctions of unbounded arity. There is a way to write the clauses in our examples using binary disjunctions, so that the resulting formulas have linear size cut-free sequent calculus refutations. Thus, as it was already known for the tree-like case, the complexity of sequent calculus proofs can depend on how disjunctions are represented.

Several notions of width have been used to show space lower bounds in different proof systems, demonstrating a close relationship between the two measures [5, 14, 4, 11, 12, 10, 9, 17, 20]. We note that our characterization of cut-free sequent calculus width for refuting CNF formulas coincides with the concept of dynamic satisfiability, introduced by Esteban, Galesi and Messner [14] as a tool for proving space lower bounds in resolution and k-DNF resolution. It is easily seen that dynamic satisfiability is a weakened version of resolution width. How much weaker however is a question that has not been addressed. We show that it is strictly weaker, the quadratic gap between resolution and cut-free sequent calculus width being a quadratic gap between the two. Furthermore, our basic construction extends to stronger versions of dynamic satisfiability used to prove monomial space lower bounds in algebraic proof systems [11, 12], allowing us to make progress towards separating resolution width from monomial space.

To put things into perspective, Atserias and Dalmau [4] show that for a k-CNF formula F, $W(F \vdash \bot)$, the minimum width, and $\mathrm{CSpace}(F \vdash \bot)$, the minimum clause space needed to refute F in resolution satisfy

$$W(F \vdash \bot) \leq \mathrm{CSpace}(F \vdash \bot) + k, \tag{1.1}$$

and Galesi, Kołodziejczyk and Thapen [17] show a similar relation between resolution width and the minimum monomial space needed to refute F in polynomial calculus with resolution:

$$W(F \vdash \bot) \leq O\left((\mathrm{MSpace}(F \vdash \bot))^2\right) + k. \tag{1.2}$$

Ben-Sasson and Nordström [6, 7] give for every n, a formula F of size n such that $W(F \vdash \bot) = O(1)$ and $\mathrm{CSpace}(F \vdash \bot) = \Omega(n/\log n)$, rendering a relation between resolution width and clause space in the opposite direction impossible. Whether clause space can be meaningfully bounded in terms of monomial space is unknown, but the two measures are related in a more indirect way: they coincide up to polynomial and $\log n$ factors once regularized, meaning that a super-polynomial separation of them would imply a strong trade-off between monomial space and size [20]. Despite however this close relationship, no separation between width and monomial space is currently known — the techniques of [6, 7] in particular fail to generalize to the case of monomial space. Our result shows a quadratic separation between the two.

2 Sequent calculus

The sequent calculus was introduced by Gentzen to formulate and prove his famous cut-elimination theorem. Many authors describe it as the most elegant proof system, and indeed, it illustrates the symmetries of logic at the level of syntax, like no other system.

Sequent calculus's version for classical logic is often denoted by LK. We shall use LK to denote its propositional part. LK operates with *sequents*. A sequent is a tuple of the form (Γ, Δ), where Γ and Δ are finite sets of formulas. Traditionally, a sequent (Γ, Δ) is written

as $\Gamma \to \Delta$. This is to remind us its semantic interpretation: $\Gamma \to \Delta$ is to be read as "if all formulas in Γ are true, then at least one formula in Δ is true".

Let us present the rules of the system. In what follows, A, B represent arbitrary formulas, and $\Gamma, \Delta, \Gamma', \Delta'$ represent finite sets of formulas. Sets are written in a quite plain manner: We write Γ, A instead of $\Gamma \cup \{A\}$, A instead of $\{A\}$, A, B instead of $\{A, B\}$ and so on. The *axioms* of LK are all sequents of the form $A \to A$. Of the inference rules, first we have a rule, which allows us to add formulas to the left or right part of a sequent. This rule is called the *thinning* or *weakening* rule and has the form

$$\frac{\Gamma \to \Delta}{\Gamma' \to \Delta'},$$

i.e., from $\Gamma \to \Delta$ derive $\Gamma' \to \Delta'$, where $\Gamma \subseteq \Gamma'$ and $\Delta \subseteq \Delta'$. Next, we have rules for each connective. These come in pairs; a connective is treated differently according to which side of its sequent it appears. The rules for the connectives $\wedge$, $\vee$ and $\neg$ are shown in Table 1. These rules are called *analytic*, and they already form a complete proof system for proving

$$\neg\mathrm{L}: \frac{\Gamma, \neg A \to \Delta, A}{\Gamma, \neg A \to \Delta} \qquad \neg\mathrm{R}: \frac{\Gamma, A \to \Delta, \neg A}{\Gamma \to \Delta, \neg A}$$

$$\wedge\mathrm{L}_1: \frac{\Gamma, A \wedge B, A \to \Delta}{\Gamma, A \wedge B \to \Delta} \qquad \wedge\mathrm{L}_2: \frac{\Gamma, A \wedge B, B \to \Delta}{\Gamma, A \wedge B \to \Delta} \qquad \wedge\mathrm{R}: \frac{\Gamma \to \Delta, A \wedge B, A \qquad \Gamma \to \Delta, A \wedge B, B}{\Gamma \to \Delta, A \wedge B}$$

$$\vee\mathrm{R}_1: \frac{\Gamma \to \Delta, A \vee B, A}{\Gamma \to \Delta, A \vee B} \qquad \vee\mathrm{R}_2: \frac{\Gamma \to \Delta, A \vee B, B}{\Gamma \to \Delta, A \vee B} \qquad \vee\mathrm{L}: \frac{\Gamma, A \vee B, A \to \Delta \qquad \Gamma, A \vee B, B \to \Delta}{\Gamma, A \vee B \to \Delta}$$

Table 1 The analytic LK rules

tautologies; we shall call this system *cut-free* LK or LK^-. Finally, there is the *cut rule*:

$$\frac{\Gamma, A \to \Delta \qquad \Gamma \to \Delta, A}{\Gamma \to \Delta}. \tag{2.1}$$

So, already having a proof of, say $\to B$, we may use it to prove $\to A$: $\to A$ can be derived from $B \to A$ and $\to B$ via the cut rule, and now to prove $\to A$, we need to prove the weaker formula $B \to A$. B can be anything. It doesn't need to have any intuitive relation to A, but even as such, it might be the case that a proof of $B \to A$ is much shorter than a proof of $\to A$. Gentzen's cut elimination theorem says that there is always an effective procedure of eliminating all applications of the cut rule from a proof, making it purely analytic. We refer to the formula A in applications of rule (2.1) as the formula being cut, or as the *cut formula*.

An LK proof of a sequent σ is a derivation of σ starting with the axioms and applying the rules of LK. We imagine starting with the axioms at the bottom, and going to the top by applying the LK rules. More formally, an LK proof of σ is a sequence consisting of sequents that ends with σ, in which every sequent is either an axiom, or results from previous sequents by one of the LK rules. An LK^- proof is an LK proof that never uses the cut rule. We may

view proofs as DAGs, by drawing edges from premises to conclusions in applications of the inference rules. If the DAG corresponding to a proof is a tree, we shall refer to the proof as being *tree-like*.

3 Sequent calculus as a satisfiability algorithm

It will be particularly convenient to consider the following view of LK. Following Smullyan [23], let us write a sequent $A_1, \ldots, A_k \to B_1, \ldots, B_\ell$ as $T\,A_1, \ldots, T\,A_k, F\,B_1, \ldots, F\,B_\ell$. That is, we annotate the formulas appearing on the left side of a sequent by T, the formulas appearining on its right side by F, and conjoin the two sides to form a single set. T and F stand for true and false respectively — $T\,A$ should be thought of as asserting that A is true and $F\,A$ as asserting that A is false.

Annotated formulas that are not annotated variables are naturally divided into two groups: those of a conjunctive and those of a disjunctive type. Formulas of the form $T\,A \wedge B$, $F\,A \vee B$, $T\,\neg A$ or $F\,\neg A$ belong to the former group, and those of the form $T\,A \vee B$ or $F\,A \wedge B$ to the latter. We use the letter "α" to stand for an arbitrary annotated formula of conjunctive type, and the letter "β" to stand for an arbitrary annotated formula of disjunctive type. We define the components α_i of a formula α and the components β_i of a formula β as shown in Table 2.

α	α_1	α_2	β	β_1	β_2
$T\,A \wedge B$	$T\,A$	$T\,B$	$F\,A \wedge B$	$F\,A$	$F\,B$
$F\,A \vee B$	$F\,A$	$F\,B$	$T\,A \vee B$	$T\,A$	$T\,B$
$T\,\neg A$	$F\,A$				
$F\,\neg A$	$T\,A$				

Table 2 Smullyan's notation

These provisions allow on the one hand for an extremely concise description of the rules of Table 1; they can be written as:

$$\frac{S, \alpha, \alpha_1}{S, \alpha}, \qquad \frac{S, \alpha, \alpha_2}{S, \alpha}, \qquad \frac{S, \beta, \beta_1 \qquad S, \beta, \beta_2}{S, \beta}.$$

More importantly, they reveal an algorithmic interpretation of LK. An LK proof, seen from the top to the bottom, i.e. from the sequent $\sigma \coloneqq A_1, \ldots, A_k \to B_1, \ldots, B_\ell$ it is proving to the axioms, describes the execution of an algorithm that tries to find a truth assignment (or more generally a model) that falsifies σ. The algorithm begins with σ written as $T\,A_1, \ldots, T\,A_k, F\,B_1, \ldots, F\,B_\ell$, asserting that there is an assignment that makes all A_i true and all B_i false, or equivalently, an assignment that falsifies σ. Then it keeps expanding this set, by applying the LK rules in reverse, that is from the conclusion to the premises. This expansion takes the form of a tree (or a DAG if we identify nodes labelled by the same set). At any point, we may choose a leaf labelled by S, α and add to it a single child labelled by S, α, α_i, as any assignment satisfying α must also satisfy every α_i. Or we may choose a leaf labelled by S, β and add to it two children, one labelled by S, β, β_1 and the other by S, β, β_2, as any assignment satisfying β must either satisfy β_1 or β_2. The thinning rule allows the algorithm to forget information: We may add to a leaf labelled by S, a child labelled by a subset of S. Finally, the cut rule allows us to add to a leaf labelled by S, two children, one labelled by $S, T\,A$ and the other by $S, F\,A$ for any formula A, as every assignment must satisfy either A or $\neg A$. This may greatly facilitate the search procedure. If at any point a set

of the form $T\ A, F\ A$ is reached, then the search process may terminate at that particular branch, as no assignment can set A to both true and false. Notice that the contradiction $T\ A, F\ A$ corresponds to the axiom $A \to A$. A tree (or DAG) constructed this way, every branch of which ends with a leaf labelled by a set of the form $T\ A, F\ A$, is an LK proof of σ.

A depth-first implementation of the algorithm described above is shown as Algorithm 1 below. Algorithm 1 is called on a sequent represented as a set of annotated formulas. It

Algorithm 1 The LK algorithm

procedure LK(S)
  if S contains both $T\ A$ and $F\ A$ for some formula A **then**
    return false
  if for every $\alpha \in S$, all $\alpha_i \in S$ and for every $\beta \in S$, there is a $\beta_i \in S$ **then**
    return true
  go to either **1**, **2** or **3**
  1. select an $S' \subseteq S$ and **return** LK(S')
  2. select an arbitrary formula A and **return** LK($S, T\ A$) **or** LK($S, F\ A$)
  3. select an $A \in S$
  if $A = \alpha$ **then**
    select a component α_i and **return** LK(S, α_i)
  if $A = \beta$ **then**
    return LK(S, β_1) **or** LK(S, β_k)

chooses at each recursive call non-deterministically what rule to apply and which formula to apply it to. If false is returned then there is an LK proof of our initial sequent; if no such proof exists, then there is an assignment that falsifies our initial sequent and Algorithm 1 is able to find it, returning true. As presented, line 1, corresponding to the thinning rule, is redundant. However, incorporating memoization, that is the ability to stop the search when a set S has already been encountered in a previous recursive call that has returned, effectively identifying nodes labelled by the same set, this line makes it possible to greatly prune the search for a falsifying assignment. In terms of proofs, DAG-like proofs may be shorter than tree-like proofs. The key point in analyzing the correctness of Algorithm 1 (or equivalently the completeness of LK), is that when at the base case true is returned, we can create an assignment consistent with S by setting for each formula A, A to true if $T\ A \in S$, and false otherwise.

4 The width of sequent calculus proofs

We define the *width* of a sequent as the number of formulas it contains, and the width of a sequent calculus proof as the maximum of the widths of the sequents it contains. It is not hard to see that for any provable sequent S_0, there is an LK proof of S_0 of width a constant plus the width of S_0. The concept of the minimum width needed to prove a sequent becomes non-trivial only if we restrict the class of cut formulas we are allowed to use. We shall be mainly interested in the minimum width over all LK^- proofs of S_0, which we denote by $W_{\mathrm{LK}^-}(\vdash S_0)$.

We are going to give a characterization of $W_{\mathrm{LK}^-}(\vdash S_0)$ in terms of the definition below. In what follows, sequents are viewed as in the above section, viz. as sets of annotated formulas.

▶ **Definition 4.1.** Following the terminology of [23], let us call a sequent S_0 *analytically k-consistent* if there is a set of sequents $\mathcal{S}$ containing S_0 and such that for each $S \in \mathcal{S}$:

1. for any formula A, S does not contain both $T\,A$ and $F\,A$;
2. $S' \subseteq S \implies S' \in \mathcal{S}$;
3. $|S| < k \;\&\; \alpha \in S \implies S, \alpha_i \in \mathcal{S}$ for every component α_i of α;
4. $|S| < k \;\&\; \beta \in S \implies S, \beta_i \in \mathcal{S}$ for some component β_i of β.

If the following condition is also satisfied, then we call S_0 *synthetically k-consistent* with respect to the set $\mathcal{C}$:

5. $|S| < k \implies S, T\,A \in \mathcal{S}$ or $S, F\,A \in \mathcal{S}$ for any formula $A \in \mathcal{C}$.

It is often helpful to see definitions such as the above, as describing a strategy for the adversary, in a game between a prover and an adversary played on a formula/sequent/set of formulas. In this case, the game is as follows: The configurations of the game are sequents. The initial configuration is S_0. In every round, the prover either deletes some formulas in the current sequent S, or selects an α-formula in S and adds a component of it to S, or selects a β formula, in which case the adversary adds a component of it to S. Allowing condition 5, the prover may choose an arbitrary formula $A \in \mathcal{C}$ and the adversary must respond by adding either $T\,A$ or $F\,A$ to S. The game ends with prover winning once S contains $T\,A$ and $F\,A$ for some formula A. The prover can always win provided that S_0 is provable. The question is: given a bound k, can she win always maintaining that the width of S is at most k? Definition 4.1 describes a strategy for the adversary, permitting the prover from winning when she maintains that bound.

▶ **Theorem 4.1.** *Suppose that $|S_0| \leq k$. Then S_0 is analytically k-consistent if and only if $W_{\mathrm{LK}^-}(\vdash S_0) > k$. It is synthetically k-consistent with respect to $\mathcal{C}$ if and only if every LK proof of S_0 in which every cut formula belongs to $\mathcal{C}$ has width more than k.*

Proof. Let us only show the former sentence. Suppose first that S_0 is analytically k-consistent, and let $\mathcal{S}$ be the set of sequents witnessing this. We will show that in every tree-like LK^- derivation (not necessarily beginning with axioms) τ of S_0, of width at most k, there is an initial sequent (i.e. one appearing as a leaf) in $\mathcal{S}$. From the first condition of Definition 4.1 that sequent is not an axiom, thus τ is not a proof. It is enough to show this for tree-like derivations, since a DAG-like derivation can be transformed into a tree-like one without increasing the width.

Base case. If τ contains just S_0, then we are done since $S_0 \in \mathcal{S}$.
Inductive step. Take some initial sequents $S_1, \ldots, S_r$ from which a sequent S is derived via an inference rule ρ, and remove them to get the derivation τ'. From the induction hypothesis, there is an initial sequent in τ' that belongs to $\mathcal{S}$. If that sequent is not S, then it also appears in τ and we are done. Otherwise, we have the following cases according to what rule ρ is:

Case 1. If it is the weakening rule, and thus $r = 1$ and $S_1 \subseteq S$, then from the second condition of Definition 4.1, $S_1 \in \mathcal{S}$.
Case 2. If ρ is the α-rule, and thus $r = 1$, $\alpha \in S$ and $S_1 = S, \alpha_i$ for some α_i, then since τ has width at most k, $|S| < k$, and hence from the third condition of Definition 4.1, $S_1 \in \mathcal{S}$.
Case 3. If ρ is the β-rule, and thus $\beta \in S$ and each S_i is of the form S, β_i, then again $|S| < k$, and from the fourth condition of Definition 4.1, some S_i belongs to $\mathcal{S}$.

Now suppose that $W_{\mathrm{LK}^-}(\vdash S_0) > k$. Set

$$\mathcal{S} \coloneqq \{S \mid |S| \leq k \;\&\; W_{\mathrm{LK}^-}(\vdash S) > k\}.$$

Clearly $S_0 \in \mathcal{S}$. We will show that $\mathcal{S}$ satisfies the conditions 1–4 of Definition 4.1. For each $S \in \mathcal{S}$, first S cannot contain $T\,A$ and $F\,A$ for some A. This is so, because such a sequent is a weakening of an axiom, and having size at most k, it has a proof of width at most k. For the closure under subsets, if $S' \subseteq S$, then $W_{\mathrm{LK}^-}(\vdash S') > k$, for otherwise $W_{\mathrm{LK}^-}(\vdash S) \leq k$ since S follows from S' via the weakening rule. For the α condition, if $\alpha \in S$ and $|S| < k$, then for each α_i it must be that $S, \alpha_i \in \mathcal{S}$, for otherwise $W_{\mathrm{LK}^-}(\vdash S) \leq k$ since S follows from S, α_i via the α-rule. Finally, if $\beta \in S$ and $|S| < k$, then there must be a β_i such that $S, \beta_i \in \mathcal{S}$, otherwise $W_{\mathrm{LK}^-}(\vdash S) \leq k$, since S follows from all S, β_i via the β-rule. ◀

5 LK$^-$ for refuting CNF formulas and resolution

A *literal* is a propositional variable x, or the negation of propositional variable $\neg x$. We let $\overline{x} \stackrel{\text{def}}{=} \neg x$ and $\overline{\neg x} \stackrel{\text{def}}{=} x$. A *clause* is a disjunction (possibly empty) of literals, and a *CNF formula* is a conjunction of clauses. The ordering of literals in a clause does not matter, so that the clause $x \vee y$ is considered to be the same as $y \vee x$. The *width*, $W(F)$, of a CNF formula F is the number of literals in the largest clause of F. A CNF formula of width at most k is called a k-CNF formula.

Refuting a CNF formula $F = C_1 \wedge \cdots \wedge C_m$ means proving that the clauses C_i cannot be simultanesouly satisfied, that is, it means proving $C_1, \ldots, C_m \to$. LK$^-$ for proving such sequents has the following form. Of the rules in Table 1, the only one that is relevant is $\vee$L, which now, seeing clauses as disjunctions of unbounded arity, has as many premises as the number of literals in the clause it is deriving. Moreover, there is no reason to always carry the clauses C_i in sequents. We may as well delete them from every sequent, but keep in our mind that they are implicitly there. What remains are sequents of the form $\ell_1, \ldots, \ell_r \to$, where the ℓ_i's are literals, and such sequents are nothing other than clauses. To be explicit, the axioms of the resulting system are clauses of the form $x \vee \neg x$, and the inference rules are the weakening rule, from C infer $C \vee D$, and

$$\frac{C \vee \overline{\ell_1} \quad \cdots \quad C \vee \overline{\ell_r}}{C}, \tag{5.1}$$

where C and D are clauses and $\ell_1 \vee \cdots \vee \ell_r$ is a clause of the formula we are refuting. A proof of $C_1, \ldots, C_m \to$, in other words a *refutation* of $F = C_1 \wedge \cdots \wedge C_m$, in LK$^-$, is a derivation of the empty clause using the above rules. The *size* of such a derivation is the number of clauses it contains, and its width is the size of the largest clause occurring in it. We shall denote by $S_{\mathrm{LK}^-}(F \vdash \bot)$ and $W_{\mathrm{LK}^-}(F \vdash \bot)$ and the minimum size and the minimum width respectively over all LK$^-$ refutations of F.

Resolution is the system we get by adding to the above system the cut rule (2.1), where the cut formula A is restricted to be a propositional variable:

$$\frac{C \vee x \qquad C \vee \neg x}{C}. \tag{5.2}$$

We may make a proof in this system "cut-only", by pushing all applications of the rule (5.1) at the bottom levels. Namely, we can simulate rule (5.1) by (5.2) as follows: Start with $\ell_1 \vee \cdots \vee \ell_r$, derive from it and $C \vee \overline{\ell_1}$, $C \vee \ell_2 \vee \cdots \vee \ell_r$, then derive from $C \vee \ell_2 \vee \cdots \vee \ell_r$ and $C \vee \overline{\ell_2}$, $C \vee \ell_3 \vee \cdots \vee \ell_r$, and so on, until C is derived. Now the leaves containing clauses of F and these can be derived from axioms by (5.1). Deleting all axioms, and incorporating

the weakening rule into (5.2), writing it as

$$\frac{C \vee x \qquad D \vee \neg x}{C \vee D}, \tag{5.3}$$

we get the usual presentation of resolution, where, instead of deriving $F \to$, the empty clause is derived taking the clauses of F as axioms: A *resolution refutation* of a CNF formula F is a derivation of the empty clause from the clauses of F, using only the rule (5.3). We shall denote by $W_R(F \vdash \bot)$ and $S_R(F \vdash \bot)$ the minimum width and minimum size respectively, over all resolution refutations of F, and by $S_{\mathrm{TR}}(F \vdash \bot)$ the minimum size, over all tree-like resolution refutations of F.

6 Dynamic satisfiability

Adapting Definition 4.1 for resolution we get the characterization of [4] for resolution width. Adapting it for LK$^-$ restricted to refuting CNF formulas, we get the definition of dynamic satisfiability from [14]. Namely, let us call sets of literals that do not contain contradictory literals *partial assignments*. We think of the assignment, say $\{x, \neg y, z\}$, as making x true, y false and z true. A partial assignment satisfies a clause C, if it contains a literal of C. It falsifies C if it contains $\overline{\ell}$ for every ℓ in C. We get:

▶ **Definition 6.1** [14]. Let F be a CNF formula, and let k be a natural number. F is said to be *k-dynamically satisfiable* is there is a non-empty set $\mathcal{A}$ of partial assignments to its variables such that for every assignment $\alpha \in \mathcal{A}$,

1. if $\alpha' \subseteq \alpha$ then $\alpha' \in \mathcal{A}$;
2. if $|\alpha| < k$ and C is a clause of F, then there is an $\alpha' \supseteq \alpha$ in $\mathcal{A}$ that satisfies C.

Theorem 4.1 in particular, becomes:

▶ **Theorem 6.1.** *A CNF F is k-dynamically satisfiable if and only if $W_{\mathrm{LK}^-}(F \vdash \bot) > k$.*

In the game corresponding to Definition 6.1, prover chooses in each round a clause of F, and the adversary responds by choosing a literal in that clause, which adds to the current assignment. Again, the closure under subsets condition corresponds to the ability of the prover to delete at any round literals from the current assignment. The prover wins once the current assignment falsifies a clause of F.

We get a characterization of resolution width by having the prover selecting variables instead of clauses, and the adversary responding by giving values to them. More specifically, in every round the prover selects a variable x of F. Then the adversary selects either x or $\neg x$, and the prover updates the current assignment α by deleting (if she wants) literals and adding the choice of the adversary. Again the prover wins once α falsifies a clause of F. She can win always maintaining $|\alpha| < k$ if and only if $W_R(F \vdash \bot) \leq k$ [4].

Notice that, if $W(F)$ is small, the prover in the second game is more powerful. Namely, we have $W_R(F \vdash \bot) \leq W_{\mathrm{LK}^-}(F \vdash \bot) + W(F) - 1$. We already saw this when we explained how the resolution rule can simulate (5.1). In terms of games, the argument goes as follows: When the prover in the first game selects a clause C, the prover in the second game can start selecting, one by one the variables of C. If the game does not end, then the current assignment satisfies C, and then the prover can delete literals to match the assignments in the two games.

Definition 6.1 was introduced in [14] as a tool for proving space lower bounds in resolution and k-DNF resolution. The following definition is from [15, 1]. A *memory configuration* in

resolution, is a set of clauses. A resolution refutation of a CNF formula F, in configurational form, is a sequence $\mathcal{M}_1, \ldots, \mathcal{M}_t$ of configurations where $\mathcal{M}_1$ is empty, $\mathcal{M}_t$ contains the empty clause and for $i > 1$, $\mathcal{M}_i$ is obtained from $\mathcal{M}_{i-1}$ by one of the following rules:

Axiom download: $\mathcal{M}_i = \mathcal{M}_{i-1} \cup \{C\}$, where C is a clause of F.
Inference: $\mathcal{M}_i = \mathcal{M}_{i-1} \cup \{C\}$, where C is derived from clauses in $\mathcal{M}_{i-1}$ by the resolution rule.
Erasure: $\mathcal{M}_i \subseteq \mathcal{M}_{i-1}$.

The clause space of such a refutation is $\max_{1 \leq i \leq t} |\mathcal{M}_i|$. The clause space of a CNF formula F, denoted by $\mathrm{CSpace}(F \vdash \bot)$, is the minimum clause space, over all refutations, in configurational form, of F.

▶ **Theorem 6.2** [14]. *If F is k-dynamically satisfiable, then* $\mathrm{CSpace}(F \vdash \bot) \geq k$.

We thus have

$$W_R(F \vdash \bot) - W(F) + 1 \leq W_{\mathrm{LK}^-}(F \vdash \bot) \leq \mathrm{CSpace}(F \vdash \bot). \tag{6.1}$$

It is shown in [6] that there are 6-CNF formulas F of size $O(n)$ such that $W_R(F \vdash \bot) = O(1)$ and $\mathrm{CSpace}(F \vdash \bot) = \Omega(n/\log n)$. It is easy to show that $W_{\mathrm{LK}^-}(F \vdash \bot) = O(1)$, thus these formulas in fact provide a gap between $W_{\mathrm{LK}^-}(F \vdash \bot)$ and $\mathrm{CSpace}(F \vdash \bot)$. The question of whether there is a gap between $W_R(F \vdash \bot)$ and $W_{\mathrm{LK}^-}(F \vdash \bot)$ has not been addressed, and it is what we will deal with next.

7 A quadratic gap between LK$^-$ and resolution width

Let $F = \bigwedge_{i=1}^{s} C_i$ and $G = \bigwedge_{i=1}^{t} D_i$ be unsatisfiable CNF formulas. We define

$$F \times G \stackrel{\text{def}}{=} \bigwedge_{i=1}^{s} \bigwedge_{j=1}^{t} (C_i \vee D_j).$$

$F \times G$ is the CNF expansion of the formula $F \vee G$, which is also unsatisfiable.

Remarkably, LK$^-$ width and resolution width exhibit a different behavior with respect to this construction. This disparity ultimately relies on the fact that the cut rule gives us the ability to combine given proofs into a more complicated proof.

On one hand, we have:

▶ **Lemma 7.1.** *If F and G are over disjoint sets of variables, then*

$$W_{LK^-}(F \times G \vdash \bot) \geq W_{LK^-}(F \vdash \bot) + W_{LK^-}(G \vdash \bot) - 1.$$

Proof. Suppose that F is k-dynamically satisfiable, G is ℓ-dynamically satisfiable, and let $\mathcal{A}$ and $\mathcal{B}$ respectively be sets witnessing this. We need to show that $F \times G$ is $(k+\ell)$-dynamically satisfiable, that is we need to find a set satisfying the conditions of Definition 6.1 for the parameter $k + \ell$. We claim that

$$\mathcal{C} \coloneqq \{\alpha \cup \beta \mid \alpha \in \mathcal{A} \ \ \& \ \ \beta \in \mathcal{B}\}$$

is such a set. Closure under subsets immediately follows from the fact that $\mathcal{A}$ and $\mathcal{B}$ are closed under subsets. For the second condition, suppose that $\gamma \in \mathcal{C}$, $|\gamma| < k + \ell$, and let $C_i \vee D_j$ be a clause of $F \times G$, where C_i is a clause of F and D_j is a clause of G. Since $\gamma \in \mathcal{C}$, there is an $\alpha \in \mathcal{A}$ and a $\beta \in \mathcal{B}$ such that $\gamma = \alpha \cup \beta$. Moreover, since $|\gamma| < k$ and F and

G do not share variables, either $|\alpha| < k$ or $|\beta| < \ell$. In the first case there is an $\alpha' \supseteq \alpha$ in $\mathcal{A}$ satisfying C_i, and thus $\alpha' \cup \beta$ is an assignment in $\mathcal{C}$ satisfying $C_i \vee D_j$. In the second case there is a $\beta' \supseteq \beta$ in $\mathcal{B}$ satisfying D_j, and thus $\alpha \cup \beta'$ is an assignment in $\mathcal{C}$ satisfying $C_i \vee D_j$. ◀

For resolution on the other hand, we have:

▶ **Lemma 7.2.** $W_R(F \times G \vdash \bot) \leq \max\{W_R(F \vdash \bot) + W(G), W_R(G \vdash \bot)\}$.

Proof. Let π and ρ be resolution refutations of F and G respectively, both of minimum width. Replacing every clause C in π with $C \vee D_i$ we get a resolution proof π_i of D_i from $F \times G$. π_i has width at most $W_R(F \vdash \bot) + W(F)$. Replacing then every clause D_i in ρ with π_i we get a resolution refutation of $F \times G$ with the stated width. ◀

Choosing an appropriate seed and iterating, we get our result.

▶ **Theorem 7.1.** *There are CNF formulas G with n^2 variables, size $O(n)^n$, and such that $W_R(G \vdash \bot) = O(n)$ and $W_{\mathrm{LK}^-}(G \vdash \bot) = \Omega(n^2)$.*

Proof. Let F be a CNF formula with n variables, width $O(1)$, size $\Theta(n)$, and such that $W_R(F \vdash \bot) = \Theta(n)$. Such formulas exist from e.g. [8]. Consider the formula $F^n := F_1 \times \cdots \times F_n$, where the F_i's are copies of F over mutually disjoint sets of variables. From Lemma 7.2, $W_R(F^n \vdash \bot) = O(n)$. On the other hand $W_{\mathrm{LK}^-}(F \vdash \bot) = \Omega(n)$ from (6.1), and hence from Lemma 7.1, $W_{\mathrm{LK}^-}(F^n \vdash \bot) = \Omega(n^2)$. ◀

8 Separating resolution width from monomial space

Monomial space is a generalized version of clause space. While configurations in the case of clause space are sets of clauses, for monomial space, arbitrary linear combinations, over a field $\mathbb{F}$, of clauses are allowed as the contents of a configuration, where such a linear combination P is interpreted as the asserting that $P = 0$. As a matter of fact, all known lower bounds for monomial space even hold in the case where arbitrary Boolean functions of clauses are allowed. The term monomial space comes from the fact that this concept captures space in proof systems employing algebraic reasoning.

Namely, seeing clauses as monomials — a clause $\ell_1 \vee \cdots \vee \ell_r$ is seen as the monomial $\overline{\ell_1} \ldots \overline{\ell_r}$ — the question of whether a set of clauses over the variables $x_1, \ldots, x_n$ is unsatisfiable, becomes the question of whether the polynomial 1 belongs to the ideal generated by those clauses and the clauses $x_i^2 - x_i$ and $x_i + \overline{x_i} - 1$ in $\mathbb{F}[x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}]$. A systematic way of generating this ideal, in a space oriented model, is the following [1]. Configurations are sets of polynomials over $\mathbb{F}[x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}]$. A refutation of a CNF formula F, in configurational form, is a sequence $\mathcal{M}_1, \ldots, \mathcal{M}_t$ of configurations where $\mathcal{M}_1$ is empty, $\mathcal{M}_t$ contains the empty clause and for $i > 1$, $\mathcal{M}_i$ is obtained from $\mathcal{M}_{i-1}$ by one of the following rules:

Axiom download: $\mathcal{M}_i = \mathcal{M}_{i-1} \cup \{C\}$, where C is either a clause of F, $x_i^2 - x$, or $x_i + \overline{x_i} - 1$.
Inference: $\mathcal{M}_i = \mathcal{M}_{i-1} \cup \{P\}$, where P is either a linear combination of polynomials in $\mathcal{M}_{i-1}$ or a literal multiplied by some polynomial in $\mathcal{M}_{i-1}$.
Erasure: $\mathcal{M}_i \subseteq \mathcal{M}_{i-1}$.

The monomial space of such a refutation is the maximum number of distinct monomials occurring in a configuration. The monomial space, $\mathrm{MSpace}(F \vdash \bot)$, of F, is the mimimum monomial space over all refutations of F.

The gap shown in the previous section can be extended to stronger versions of dynamic satisfiability that have been used to show monomial space lower bounds, thus showing a gap between resolution width and monomial space. The configurations in those are not assignments as in Definition 6.1, but sets of assignments. They will not be arbitrary sets however; they will have a certain structure. Namely, we call a set H of assignments *admissible*, if it is of the form

$$H = H_1 \times \cdots \times H_r \stackrel{\text{def}}{=} \{\alpha_1 \cup \cdots \cup \alpha_r \mid \alpha_i \in H_i\},$$

where each H_i is a non-empty set of non-empty assignments, for any two assignments $\alpha_i \in H_i$ and $\alpha_j \in H_j$ for $i \neq j$, the domains of α_i and α_j do not intersect, and moreover, if an assignment $\alpha \in H_i$ gives the value ϵ to a variable x, then there is also an assignment $\alpha' \in H_i$ giving to x the value $1 - \epsilon$. The H_i's are called the *factors* of H; we write $\|H\|$ for their number. We write $H' \sqsubseteq H$ if every factor of H' is a factor of H.

▶ **Definition 8.1** [11, 12]**.** Let F be a CNF formula and let k be a natural number. We say that F is *k-extendible* if there is a non-empty set of admissible configurations $\mathcal{H}$ such that for each $H \in \mathcal{H}$,

1. if $H' \sqsubseteq H$, then $H' \in \mathcal{H}$;
2. if $\|H\| < k$ and C is a clause of F, then there is an $H' \sqsupseteq H$ in $\mathcal{H}$, such that every $\alpha \in H'$ satisfies C.

▶ **Theorem 8.1** [11, 12]**.** *If F is k-extendible, then* $\text{MSpace}(F \vdash \bot) \geq \lfloor k/4 \rfloor$.

Lemma 7.1 with the same proof applies here as well.

▶ **Lemma 8.1.** *Let F and G be CNF formulas over disjoint sets of variables. If F is k-extendible and G is ℓ-extendible, then $F \times G$ is $(k + \ell)$-extendible.*

Proof. Let $\mathcal{H}$ and $\mathcal{I}$ be sets of admissible configurations witnessing the k and ℓ-extendibility of F and G. Since F and G are over disjoint sets of variables, we may assume that the domains for any of two assignments in $\mathcal{H}$ and $\mathcal{I}$ do not intersect. Set

$$\mathcal{J} \coloneqq \{H \times I \mid H \in \mathcal{H} \ \ \& \ \ I \in \mathcal{I}\}.$$

Clearly, $\mathcal{J}$ is a set of admissible configurations. We claim that it satisfies the conditions of Definition 8.1 for the parameter $k + \ell$. Closure under $\sqsubseteq$ immediately follows from the fact that $\mathcal{H}$ and $\mathcal{I}$ are closed under $\sqsubseteq$. For the second condition, suppose that $J = H \times I \in \mathcal{J}$, $\|J\| < k + \ell$, and let $C_i \vee D_j$ be a clause of $F \times G$, where C_i is a clause of F and D_j is a clause of G. Since $\|J\| < k + \ell$, either $\|H\| < k$ or $\|I\| < \ell$. In the first case, there is an $H' \sqsupseteq H$ in $\mathcal{H}$ such that all assignments in H' satisfy C_i. Then $H' \times I$ is an admissible configuration in $\mathcal{J}$ such that all assignments in it satisfy $C_i \vee D_j$. The second case is analogous. ◀

Therefore, we get:

▶ **Theorem 8.2.** *There are CNF formulas G with n^2 variables, size $O(n)^n$, and such that $W_R(G \vdash \bot) = O(n)$ and* $\text{MSpace}(G \vdash \bot) = \Omega(n^2)$.

Proof. Again, let F be a CNF formula with n variables, width $O(1)$ and size $\Theta(n)$, that is $\Omega(n)$-extendible. Such formulas exist, see [11, 16, 12, 9]. The formulas $F^n \coloneqq F_1 \times \cdots \times F_n$, where the F_i's are copies of F over mutually disjoint sets of variables, have resolution width $O(n)$, and from Lemma 8.1 they are $\Omega(n^2)$-extendible, thus from Theorem 8.1 require $\Omega(n^2)$ monomial space. ◀

9 A super-polynomial separation between resolution and LK$^-$ size

Many of the relations in resolution involving width, can be as well stated for LK$^-$. In fact, they seem to be better suited for LK$^-$; there, the additive $W(F)$ factor that naturally comes with resolution width disappears. We have already seen that $W_{\mathrm{LK}^-}(F \vdash \bot) \leq \mathrm{CSpace}(F \vdash \bot)$, refining the relation between clause space and width of [4]. But let us give an alternative, constructive proof, here. For sets S and T of formulas, we write $S \models T$ if every total assignment satisfying every formula in S, also satisfies every formula in T.

▶ **Theorem 9.1.** *For any unsatisfiable CNF formula F, $W_{\mathrm{LK}^-}(F \vdash \bot) \leq \mathrm{CSpace}(F \vdash \bot)$.*

Proof. Let $\mathcal{M}_1, \ldots, \mathcal{M}_t$ be a refutation of F, of clause space s. We shall construct a sequence $\mathbf{T}_1, \ldots, \mathbf{T}_t$ of trees, the vertices of which are labelled by sets of literals, such that for every set S labelling a leaf of $\mathbf{T}_i$, $S \models \mathcal{M}_i$ and $|S| \leq |\mathcal{M}_i|$.

We set $\mathbf{T}_1$ to be a tree with one vertex labelled by the empty set. Now, suppose we have constructed $\mathbf{T}_{i-1}$. If $\mathcal{M}_i$ results from $\mathcal{M}_{i-1}$ via an inference step, we set $\mathbf{T}_i := \mathbf{T}_{i-1}$. If $\mathcal{M}_i \subseteq \mathcal{M}_{i-1}$, then we add to every leaf of $\mathbf{T}_{i-1}$ labelled by a satisfiable set S, a child labelled by a subset $S' \subseteq S$ such that $|S'| \leq |\mathcal{M}_i|$ and $S' \models \mathcal{M}_i$. Finally, if $\mathcal{M}_i = \mathcal{M}_{i-1} \cup \{C\}$, for a clause $C = \ell_1 \vee \cdots \vee \ell_r$ of F, we add to every child of $\mathbf{T}_i$ labelled by a satisfiable set S, r children labelled by the sets $S \cup \{\ell_j\}$.

Replacing each set $\{\ell_1, \ldots, \ell_k\}$ occurring in $\mathbf{T}_t$, by the clause $\overline{\ell_1} \vee \cdots \vee \overline{\ell_k}$, we get an LK$^-$ refutation of F of width at most s. It is clear from the construction of $\mathbf{T}_t$ that every clause has width at most s and every clause not at a leaf, results from the clauses at its children via either the weakening or the $\vee$L rule. Moreover, since $\mathcal{M}_t$ is unsatisfiable, no set labelling a leaf of $\mathbf{T}_i$ is satisfiable, that is sets at the leaves become weakenings of axioms. ◀

Next, we have the size-width relations of [8]. Here the proofs are the same as those in [8].

▶ **Theorem 9.2.** *For any unsatisfiable CNF formula F, $W_{\mathrm{LK}^-}(F \vdash \bot) \leq \log S_{\mathrm{TR}}(F \vdash \bot)+1$.*

Proof. This is in fact a weakened version of Theorem 9.1, as $\mathrm{CSpace}(F \vdash \bot) \leq \log S_{\mathrm{TR}}(F \vdash \bot) + 1$ [15]. But let us give a direct construction instead. We shall construct, by induction on s, for every tree-like resolution refutation $\mathbf{T}$ of F of size s, an LK$^-$ refutation of F of width at most $\log s + 1$.

If $\mathbf{T}$ has size 1, then it has width 0; if it has size 3 then it has width 2. For the inductive step, suppose that $\mathbf{T}$ has size $s > 3$. Let $\mathbf{T}_1$ and $\mathbf{T}_2$ be the subproofs of $\mathbf{T}$, deriving $\neg x$ and x respectively, for some variable x. One of $\mathbf{T}_1$ and $\mathbf{T}_2$, say $\mathbf{T}_1$, must have size at most $s/2$. $\mathbf{T}_1|_x$ is a refutation of $F|_x$ of size at most $s/2$, and $\mathbf{T}_2|_{\overline{x}}$ is a refutation of $F|_{\overline{x}}$ of size less than s. From the induction hypothesis, there are LK$^-$ refutations π_1 and π_2 of $F|_x$ and $F|_{\overline{x}}$ of width $\log s$ and $\log s + 1$ respectively. Start with π_2. To every application of Rule (5.1), add the extra premise $C \vee x$. But x can be derived from π_1, and hence all those $C \vee x$ can be derived via the weakening rule from x: We can do the same to π_1, now adding $C \vee \overline{x}$ as the extra premise, and moreover add to every clause the variable x. The refutation obtained when we combine π_1 and π_2 is a valid LK$^-$ refutation of F of width at most $\log s + 1$. ◀

▶ **Theorem 9.3.** *For any unsatisfiable CNF formula F over n variables, $W_{\mathrm{LK}^-}(F \vdash \bot) = O\left(\sqrt{n \log S_{\mathrm{LK}^-}(F \vdash \bot)}\right)$.*

Proof. The construction will be the same as that of Theorem 9.2. The problem here is that it is not clear what variable to choose to recurse on. The trick is to choose the variable that appears more often in the clauses of the proof.

For an LK$^-$ refutation π of a CNF and a $d \geq 0$, let π^* be the set of clauses in π of width greater than d. We call the clauses in π^* the *fat* clauses of π. We show, by induction on n, that for any CNF F in n variables, any LK$^-$ refutation π of F and any integers $d, b \geq 0$,

$$|\pi^*| < a^b \implies W_{\text{LK}^-}(F \vdash \bot) \leq d + b,$$

where $a = (1 - d/(2n))^{-1}$. The theorem follows taking π to be of minimum size and $d \coloneqq \left\lceil \sqrt{n \log S(\pi)} \right\rceil$.

If $n = 1$, then there is an LK$^-$ refutation of F of width 2, and in the case that $b + d < 2$ the implication becomes trivially true. For the inductive step, suppose that $n > 1$, let $d, b \geq 0$, and let π be an LK$^-$ refutation of F with $|\pi^*| < a^b$. If $b = 0$, then π itself is a refutation of width at most $d + b$. Suppose $b > 0$. There are $2n$ literals, so there must be some literal, say the variable x, appearing in at least $d|\pi^*|/(2n)$ clauses in π^*. $\pi|_x$ is an LK$^-$ refutation of $F|_x$ with at most $|\pi^*|\,(1 - d/(2n)) < a^{b-1}$ fat clauses, so by the induction hypothesis (notice that a is a decreasing function of n), there is an LK$^-$ refutation π' of $F|_x$ of width at most $d + b - 1$. Furthermore, $\pi|_{\overline{x}}$ is an LK$^-$ refutation of $F|_{\overline{x}}$ with less than a^b clauses, so by the induction hypothesis, there is an LK$^-$ refutation π'' of $F|_{\overline{x}}$ of width at most $d + b$. Combining π' and π'' as in the proof of Theorem 9.2, we get an LK$^-$ refutation of F of width at most $d + b$. ◀

Notice that in Theorem 9.2 and the relation $W_{\text{LK}^-}(F \vdash \bot) \leq \text{CSpace}(F \vdash \bot)$, we have LK$^-$ in the left hand side and resolution in the right hand side. That is to say, cuts are eliminated when constructing the small width proofs. It is tempting to speculate on whether the same is also true for Theorem 9.3, that is whether we can replace $S_{\text{LK}^-}(F \vdash \bot)$ with $S_R(F \vdash \bot)$. After all, the only place in the proof of Theorem 9.3 where we need LK$^-$ in the right hand side is the case $b = 0$. Theorem 7.1 says that this cannot be true. In fact, the formulas of Theorem 7.1 give the main theorem of this section, which is:

▶ **Theorem 9.4.** *There is a CNF formula F with n^2 variables and size $O(n)^n$, such that $S_R(F \vdash \bot) = O(n)^n$ but $S_{\text{LK}^-}(F \vdash \bot) \geq \exp(\Omega(n^2))$.*

Proof. The formulas of Theorem 7.1 are such. The upper bound $S_R(F \vdash \bot) = O(n)^n$ follows from the construction of Lemma 7.2. The lower bound follows from the fact that $W_{\text{LK}^-}(F \vdash \bot) = \Omega(n^2)$ and Theorem 9.3. Namely, Theorem 9.3 gives

$$S_{\text{LK}^-}(F \vdash \bot) \geq \exp\left(\Omega\left(\frac{(W_{\text{LK}^-}(F \vdash \bot))^2}{n^2}\right)\right) = \exp\left(\Omega\left(n^2\right)\right). \quad \blacktriangleleft$$

It is important to note that the $\exp\left(\Omega\left(n^2\right)\right)$ lower bound in Theorem 9.4 holds for the version of LK$^-$ operating on clauses, where the clauses of the CNF formula F to be refuted are viewed as disjunctions of unbounded arity. It does not hold when the clauses of F are made up from binary disjunctions and moreover we are free to choose the order in which they are applied. If $\vee$ in the definition of of $F \times G$, is seen as a binary disjunction, then having derived $C_1, \ldots, C_n \to$ and $D_1, \ldots, D_t \to$, it is easy to see that we may derive from these sequents $C_1 \vee D_1, \ldots, C_1 \vee D_t, \ldots, C_s \vee D_1, \ldots, C_s \vee D_t \to$ in $s \cdot t$ steps, and in this case F^n in Theorem 9.4 has an LK$^-$ refutation of size $n^{O(n)}$. An analogous situation occurs between the tree-like versions of LK$^-$ and resolution [3]. But let us notice, concluding, that with binary disjunctions, LK$^-$ cannot be seen as a system operating on clauses, and it becomes rather unnatural to compare it with resolution — it is not even clear, in this case, whether resolution can polynomially simulate LK$^-$. LK$^-$ for clauses consisting of binary disjunctions is closer to resolution with limited extension, in which case resolution does polynomially simulate it [24].

10 Conclusion

We showed a quadratic gap between resolution and cut-free sequent calculus width. In terms of the sequent calculus, this says that atomic cuts can shorten the width of proofs. It is well known that cuts can make proofs exponentially shorter. Allowing arbitrary cuts we get a system polynomially equivalent with any Frege system. These are very powerful; proving non-trivial lower bounds for them is completely out of reach of current methods. But even allowing cuts of depth $d+1$ in an LK system that has cuts of depth d for any constant $d \geq 0$, gives exponentially shorter proofs [19]. And this goes lower: For any constant $k \geq 0$, allowing as cut formulas conjunctions and disjunctions of arity $k+1$ in an LK system that has as cuts conjunctions and disjunctions of arity at most k, again gives exponentially shorter proofs [22]. We show in this paper that even allowing propositional variables as cuts, gives super-polynomially shorter proofs.

Cut-free sequent width for refuting CNF formulas naturally compares to well studied complexity measures related to resolution: it sits between resolution width and clause space. Our quadratic gap in particular, provides a separation between resolution width and clause space. Stronger such separations are known [6, 7]. Nontheless, our basic construction extends to provide a quadratic gap between resolution width and monomial space. This is to be seen in conjunction with relation (1.2) showing that monomial space provides an upper bound to resolution width.

Several questions remain open:

1. Can cut-free sequent calculus width for refuting CNF formulas be bounded in terms of resolution width? Given the similarity between the two measures, the combination of Lemmas 7.1 and 7.2 giving a quadratic separation might come as a surprise. Can this separation be improved? A strong separation in particular, would give an exponential separation between resolution and cut-free sequent calculus.
2. Our super-polynomial separation of resolution and cut-free sequent calculus on the one hand applies only when clauses are seen as disjunctions of unbounded arity. On the other hand, it involves formulas whose size grows exponentially. Can there be a separation independent of the representation of clauses? Can there be a separation for formulas whose size grows polynomially?
3. Cut-free sequent calculus width is bounded by clause space. Can it be bounded in terms of monomial space in a relation similar to (1.2)? This is a good point to also mention that whether (1.2) can be improved to a linear inequality or there are examples where it is tight is unknown as well, and there do not seem to be strong indications for which case is true.
4. We show that resolution width and monomial space cannot coincide. Whether they coincide up to polynomial factors however remains open, although it is speculated (cf. [18]) that this is not the case, and moreover, as it is the case for resolution width and clause space [6, 7], there is an $O(1)$ vs $\Omega(n/\log n)$ separation.

References

1 Michael Alekhnovich, Eli Ben-Sasson, Alexander Razborov, and Avi Wigderson. Space complexity in propositional calculus. *SIAM J. of Computing*, 31:1184–1211, 2002.

2 Noriko Arai. Relative efficency of propositional proof systems: resolution vs. cut-free LK. *Annals of Pure and Applied Logic*, 104:3–16, 2000.

3 Noriko Arai, Toniann Pitassi, and Alasdair Urquhart. The complexity of analytic tableaux. *J. of Symbolic Logic*, 71:777–790, 2006.

4 Albert Atserias and Victor Dalmau. A combinatorial characterization of resolution width. *J. of Computer and System Sciences*, 74:323–334, 2008.

5 Eli Ben-Sasson and Nicola Galesi. Space complexity of random formulae in resolution. *Random Structures & Algorithms*, 23:92–109, 2003.

6 Eli Ben-Sasson and Jakob Nordström. Short proofs may be spacious: An optimal separation of space and length in resolution. In *Pr. of the 49th Annual IEEE Symp. on Foundations of Computer Science*, pages 709–718, 2008.

7 Eli Ben-Sasson and Jakob Nordström. Understanding space in proof complexity: Separations and trade-offs via substitutions. In *Pr. of the 2nd Symp. on Innovations in Computer Science*, pages 401–416, 2011.

8 Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *J. of the ACM*, 48:149–169, 2001.

9 Patrick Bennett, Ilario Bonacina, Nicola Galesi, Tony Huynh, Mike Molloy, and Paul Wollan. Space proof complexity for random 3-cnfs. *Information and Computation*, 255:165–176, 2017.

10 Ilario Bonacina. Total space in resolution is at least width squared. In *Pr. of the 43rd International Colloquium on Automata, Languages, and Programming*, pages 56:1–56:13, 2016.

11 Ilario Bonacina and Nicola Galesi. Pseudo-partitions, transversality and locality: a combinatorial characterization for the space measure in algebraic proof systems. In *Pr. of the 4th Conf. on Innovations in Theoretical Computer Science*, pages 455–472, 2013.

12 Ilario Bonacina and Nicola Galesi. A framework for space complexity in algebraic proof systems. *J. of the ACM*, 62:23:1–23:20, 2015.

13 Stephen Cook and Robert Reckhow. On the lengths of proofs in the propositional calculus (pr. ver.). In *Pr. of the 6th Annual ACM Symp. on Theory of Computing*, pages 135–148, 1974.

14 Juan Luis Esteban, Nicola Galesi, and Jochen Messner. On the complexity of resolution with bounded conjunctions. *Theoretical Computer Science*, 321:347–370, 2004.

15 Juan Luis Esteban and Jacobo Torán. Space bounds for resolution. *Information and Computation*, 171:84–97, 2001.

16 Yuval Filmus, Massimo Lauria, Mladen Miksa, Jakob Nordström, and Marc Vinyals. Towards an understanding of polynomial calculus: New separations and lower bounds (extended abstract). In *Pr. of the 40th International Colloquium on Automata, Languages, and Programming*, pages 437–448, 2013.

17 Nicola Galesi, Leszek Kołodziejczyk, and Neil Thapen. Polynomial calculus space and resolution width. In *Pr. of the 60th Annual IEEE Symp. on Foundations of Computer Science*, pages 1325–1337, 2019.

18 Trinh Huynh and Jakob Nordström. On the virtue of succinct proofs: amplifying communication complexity hardness to time-space trade-offs in proof complexity. In *Pr. of the 44th Annual ACM Symp. on Theory of Computing*, pages 233–248, 2012.

19 Jan Krajíček. Lower bounds to the size of constant-depth propositional proofs. *J. of Symbolic Logic*, 59:73–86, 1994.

20 Theodoros Papamakarios and Alexander Razborov. Space characterizations of complexity measures and size-space trade-offs in propositional proof systems. *J. of Computer and System Sciences*, 137:20–36, 2023.

21 Robert Reckhow. *On the lengths of proofs in the propositional calculus*. PhD thesis, Department of Computer Science, University of Toronto, 1975.

22 Nathan Segerlind, Samuel Buss, and Russell Impagliazzo. A switching lemma for small restrictions and lower bounds for k-DNF resolution. *SIAM J. of Computing*, 33:1171–1200, 2004.

23 Raymond Smullyan. *First-order Logic*. Dover, 1995.

24 Grigori Tseitin. On the complexity of derivation in propositional calculus. *Studies in constructive mathematics and mathematical logic, part 2*, pages 115–125, 1968.

25 Alasdair Urquhart. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 1:425–467, 1995.