Space characterizations of complexity measures and size-space trade-offs in propositional proof systems

Theodoros Papamakarios[a,*], Alexander Razborov[b,c]

[a]*Department of Computer Science, University of Chicago, 5730 S Ellis Ave, Chicago, 60637, Illinois, USA*
[b]*University of Chicago, 1100 E 58th St, Chicago, 60637, Illinois, USA*
[c]*Steklov Mathematical Institute, Gubkina 8, Moscow, 117966, Russia*

Abstract

We identify two new clusters of proof complexity measures equal up to polynomial and $\log n$ factors. The first cluster contains the logarithm of tree-like resolution size, regularized clause and monomial space, and clause space, ordinary and regularized, in regular and tree-like resolution. Consequently, separating clause or monomial space from the logarithm of tree-like resolution size is equivalent to showing strong trade-offs between clause space and length, and equivalent to showing super-critical trade-offs between clause space and depth. The second cluster contains width, Σ_2 space (a generalization of clause space to depth 2 Frege systems), ordinary and regularized, and the logarithm of tree-like $R(\log)$ size. As an application, we improve a known size-space trade-off for polynomial calculus with resolution. We further show a quadratic lower bound on tree-like resolution size for formulas refutable in clause space 4, and introduce a measure intermediate between depth and the logarithm of tree-like resolution size.

Keywords: Proof Complexity, Resolution, Size-Space Trade-offs

1. Introduction

With the rise of computer science, questions like "can we solve a problem?" got a quantitative counterpart: "how hard is it to solve a problem?". Proof complexity deals with the quantitative version of "can we prove a the-

*Corresponding author

Email addresses: `papamakarios@uchicago.edu` (Theodoros Papamakarios), `razborov@uchicago.edu` (Alexander Razborov)

orem?", namely, the question "how hard is it to prove a theorem?". The systematic study of the latter question for propositional proof systems started with Cook and Reckhow [1], where its fundamental role in complexity theory was identified.

The most natural, arguably also the most important, measure of the complexity of a proof is its size, and indeed, much of the research in propositional proof complexity has concentrated on proof size lower bounds. But given in particular their role in proof systems of practical significance, several other natural complexity measures have been considered, and that has led to a considerable line of study about relations between them (*simulations*), lack of relations thereof (*separations*) and the inherent impossibility of optimizing two different measures at once (*trade-offs*). To aid further discussion, let us review those measures and previous results that are most pertinent to this work.

A measure that directly emerged from the study of proof size lower bounds is width; the width of a resolution proof is the number of literals in the largest clause occurring in the proof. Its importance was accentuated by Ben-Sasson and Wigderson [2], who, building on the earlier works of Clegg et al. [3] and Impagliazzo et al. [4] showing an analogous result for polynomial calculus, showed that a short resolution proof can be transformed into a narrow one. Namely, we have

$$W(F \vdash \bot) \leq \log S_T(F \vdash \bot) + W(F), \tag{1}$$

$$W(F \vdash \bot) \leq O\left(\sqrt{n \log S_R(F \vdash \bot)}\right) + W(F). \tag{2}$$

Here $W(F \vdash \bot)$, $S_T(F \vdash \bot)$ and $S_R(F \vdash \bot)$ stand for the minimum width, tree-like size and DAG-like size respectively of refuting an unsatisfiable CNF F in resolution;[1] similar notation is employed throughout the paper. $W(F)$ is the maximum width of a clause in F.

Space complexity for propositional proofs was introduced in [5, 6]. Esteban and Torán [5] showed that a short tree-like resolution proof can be transformed into a resolution proof of small clause space:

$$\mathrm{CSpace}(F \vdash \bot) \leq \log S_T(F \vdash \bot). \tag{3}$$

Atserias and Dalmau [7] demonstrated the first instance of the relationship between space and width, showing that a resolution proof having small

[1] Table 1 in Section 2 contains precise definitions of all complexity measures used in our paper.

clause space can be transformed into a narrow one:

$$W(F \vdash \bot) \leq \mathrm{CSpace}(F \vdash \bot) + W(F). \tag{4}$$

Constructive versions of their result were given by Filmus et al. [8] and Razborov (unpublished), see also Krajíček [9, Theorem 5.5.5]. It is worth noting that (3) and (4) taken together provide a refinement of (1) and that, viewed this way, we relate two *sequential* measures (tree-like size and width) with a *space* measure as an intermediate. We will see more examples of such an interplay in this paper.

More recently, Bonacina [10] showed that for total space in resolution (measured as the sum of widths of clauses in a configuration) we have

$$W(F \vdash \bot) \leq O\left(\sqrt{\mathrm{TSpace}(F \vdash \bot)}\right) + W(F), \tag{5}$$

and Galesi et al. [11] showed a weakened version of (4), but for the analogue of clause space in stronger proof systems operating with polynomials (or in fact even arbitrary Boolean functions of monomials):

$$W(F \vdash \bot) \leq O\left((\mathrm{MSpace}(F \vdash \bot))^2\right) + W(F). \tag{6}$$

Regularized[2] versions μ^* of space complexity measures are defined by multiplying the measure in question μ by the logarithm of the proof length; these were considered e.g. by Ben-Sasson [13] and Razborov [12]. The latter paper also contains the suggestion that the "right" level of precision when comparing measures of this kind is up to polynomial and $\log n$ factors. We will henceforth say that, for complexity measures μ_1, μ_2, μ_1 *simulates* μ_2, and write $\mu_1 \preceq \mu_2$, if $\mu_1(F \vdash \bot) \leq (\mu_2(F \vdash \bot) \log n)^{O(1)}$ for any CNF F in n variables.[3] We call μ_1 and μ_2 *equivalent*, and write $\mu_1 \approx \mu_2$, if $\mu_1 \preceq \mu_2$ and $\mu_2 \preceq \mu_1$. Clearly $\preceq$ is transitive, and this implies that $\approx$ is an equivalence relation and $\preceq$ imposes a partial order on its equivalence classes.

This notion of simulation brings structure to the framework of comparing proof complexity measures that is faithful to both previous research (say, results (1)-(6) look quite natural) and open problems (cf. Section 5). Regarding the choice to include $\log n$ factors in $\preceq$, $\log n$ factors appear naturally e.g. in simulations involving regularized space measures, so that we

[2]The paper [12] used the word "amortized" but Sam Buss pointed out to us that it is somewhat misleading in this context.

[3]Note that the size/length measures appear in this set-up under a logarithm. Hence this corresponds to *quasi*-polynomial simulations in the Cook-Reckhow framework.

wouldn't want to consider for example a separation of $O(1)$ vs. $\Omega(\log n)$ involving such measures to be a true separation. We also note that this aligns well with standard practice in *computational* complexity, where in most major results about space, a $\log n$ lower bound on it is included as an assumption.

The paper [12] identified a big cluster of ordinary and regularized space complexity measures, including *total space* $\mathrm{TSpace}(F \vdash \perp)$ and *variable space* $\mathrm{VSpace}(F \vdash \perp)$, that are all equivalent to proof depth in resolution:

$$D(F \vdash \perp) \approx \mathrm{TSpace}(F \vdash \perp) \approx \mathrm{TSpace}^*(F \vdash \perp) \approx VSpace^*(F \vdash \perp). \quad (7)$$

One notable measure that defied this classification was (regularized) clause space.[4]

Our contributions

In this paper we identify two other big clusters of equivalent complexity measures not covered by the results in [12]. The cumulative picture combining both previously known and new results is summarized in Figure 1. There, an arrow from μ_1 to μ_2 means that $\mu_1 \preceq \mu_2$. A solid arrow from μ_1 to μ_2 indicates that a separation between μ_1 and μ_2 is known, that is, it additionally indicates that there exists a sequence $\{F_k(x_1, \ldots, x_{n_k})\}$ of unsatisfiable CNFs such that $\mu_2(F_k \vdash \perp) \geq (\mu_1(F_k \vdash \perp) \log n_k)^{\omega(1)}$.

Let us briefly explain this picture. The first new cluster is centered around the logarithm of tree-like resolution size $\log S_T$. Given the proof method of the simulation (3) in [5], it can be obviously strengthened in two directions: by replacing the left-hand side with clause space in *tree-like* resolution or by replacing it with *regularized* clause space. Tree-like clause space in resolution (TCSpace in Figure 1, see also Table 1) was shown to be equivalent to the logarithm of tree-like size in the same paper [5, Corollary 5.1]; in other words, after this replacement in the left-hand side, the bound (3) becomes tight, within the precision we are tolerating.

We show that the second variant, that is *regularized* clause space, is also equivalent to the logarithm of tree-like resolution size, and this result extends to also include regularized *monomial* space to the same cluster. Given that [5, Corollary 5.1] also holds for (ordinary) clause space in *regular* resolution $\mathrm{RCSpace}(F \vdash \perp)$ [5, Corollary 4.2], this means that all these space measures turn out to be equivalent to each other and to the log of tree-like resolution

[4] A technical remark: [12, Theorem 3.2] does not apply to clause space as it is not bounded from below by the number of variables.

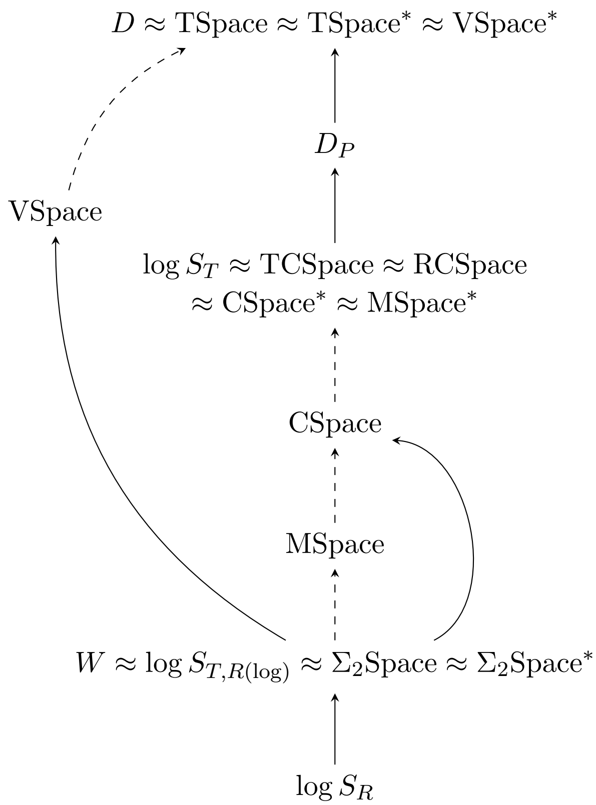

Figure 1: Simulations

size. We also remark (given the results above, this readily follows from definitions) that *regularized* versions of clause space in tree-like or regular resolution are also in this cluster.

The question of whether (ordinary) clause space also belongs here is what we consider to be a major, and most likely very difficult, open problem. But since it has turned out to be closely related to several other threads in proof complexity, we prefer to keep the momentum and defer further discussion to the concluding Section 5.

Our second cluster is presided over by resolution width. First, we introduce a natural analogue of clause space in DNF resolution that we call Σ_2 *space*. This can be seen as an extension of clause space to depth 2 Frege systems; indeed, the restriction of Σ_2 space to depth 1 Frege is precisely clause space, and its restriction to k-DNF resolution, for constant k, coincides, up to a constant factor, with the concept of space that has been studied before for such systems (see e.g. [14, 15]). In our model, configurations are *arbitrary* sets of DNFs, and we charge k for every individual k-DNF in the memory. Clearly, $\Sigma_2\text{Space} \leq \text{CSpace}$ and $\Sigma_2\text{Space}^* \leq \text{CSpace}^*$. Then we strengthen the Atserias-Dalmau bound (4) by replacing CSpace with Σ_2Space and continue to show that *both* ordinary and regularized versions of Σ_2 space are actually equivalent to resolution width.

Thus, remarkably, the difficult open question on whether we have a strong trade-off between space and length for clause space gets a relatively easy negative solution for a stronger proof system. We have also been able to locate in this cluster another interesting size measure: the size of tree-like proofs in the system $R(\log)$, which gives a somewhat unexpected generalization of (1). We have not been able to retrieve the equivalence of width and tree-like size in $R(\log)$ from the literature in exactly this form but it is implicit in Lauria [16] and, with a bit of effort, can be traced back as far as Krajíček [17].

It is worth noting that some of the simulations in this cluster work only in the syntactical setting. This comes in contrast with what happens with the other two clusters: all simulations involving clause, monomial, variable and total space, also work in a purely semantic setting. For example, in the case of monomial space we can allow arbitrary Boolean functions of monomials as memory configurations and allow any number of sound inferences to be performed at once in each step.

We use (some of) these simulations to prove:

1. *There are unsatisfiable CNF formulas F of size $O(n)$ with $S(F \vdash \bot) \leq O(n)$, $W(F \vdash \bot) \leq O(1)$ and* $\text{MSpace}^*(F \vdash \bot) \geq \Omega(n/\log n)$ (Theorem 4.1).

   This is an improvement on the previously known bounds $\text{MSpace}^*(F \vdash \bot) \geq \Omega(n^{2/11})$ [18], $\text{MSpace}^*(F \vdash \bot) \geq \Omega(n^{1/4})$ [19] and $\text{MSpace}^*(F \vdash \bot) \geq n^{1/2}/(\log n)^{O(1)}$ [20]. Unlike these previous results, our proof is remarkably simple.

2. *There are unsatisfiable CNF formulas F of size $O(n)$ with* $\text{CSpace}(F \vdash \bot) = 4$ *and* $S_T(F \vdash \bot) \geq \Omega(n^2/\log n)$ (Theorem 4.6).

   This is a first, admittedly modest, step toward separating clause space and, say, tree-like size; as we already said, we will discuss this question in more details in Section 5. It is for this proof that we need the last unexplained entry D_P on Figure 1: it stands for *positive depth*, and it is a one-sided version of depth. We also remark that the space bound in this result is optimal. More precisely, we make a relatively simple observation (Theorem 4.2) that $\mathrm{CSpace}(F \vdash \bot) \leq 3$ if and only if F is "essentially Horn" in which case it will possess a linear size tree-like resolution refutation.

Finally, let us briefly summarize what is known (to the best of our knowledge) in terms of separating the measures in Figure 1. Let us start with "true" separations, i.e. separations that work modulo polynomial overheads and $\log n$ factors.

Bonet and Galesi [21] proved that $W \not\preceq \log S_R$. More precisely, there are constant width formulas F of size $O(n^3)$ such that $S_R(F \vdash \bot) \leq O(n^3)$ and $W(F \vdash \bot) \geq \Omega(n)$. Ben-Sasson [13] proved that $\mathrm{VSpace} \not\preceq \mathrm{CSpace}$, and after negating the variables in his formulas, this works two more levels up on Figure 1. Namely, there are constant-width formulas F of size $O(n)$ such that $\mathrm{VSpace}(F \vdash \bot) \geq \Omega(n/\log n)$ while $D_P(F \vdash \bot) \leq O(1)$. This also provides a separation between D_P and D that, though, is much easier to prove directly [22, Theorem 4.6]. Without negating the variables, it is easy to see that Ben-Sasson's proof actually gives $D_P(F \vdash \bot) \geq \Omega(n/\log n)$, thus separating D_P from $\log S_T$ and hence from the whole middle cluster. Again, it is also easy to see this directly. Ben-Sasson, Håstad and Nordström [23, 24] separated clause space from width; while it is believed that their formulas should also have large monomial space complexity, the questions of separating clause space from monomial space, as well as monomial space from width are widely open.

Separating space complexity measures from their own regularized versions appears to be a very daunting task in general. As follows from Figure 1, for variable space this is equivalent to separating it from depth [22]. A quadratic separation between VSpace and VSpace^* was proved in [12, Section 6], with a disappointingly elaborate proof. Nothing is known in terms of separating CSpace from (the cluster of) CSpace^*: Theorem 4.6 makes a progress in that direction, but it is admittedly rather modest. Nothing seems to be known for CSpace vs. MSpace, and our structural picture provides a good heuristic explanation of the difficulty of this question: it would also separate MSpace from MSpace^*. Finally, in [25] a quadratic separation between width and monomial space has been established using methods very different from those in [24].

The paper is organized as follows. After giving the necessary definitions in Section 2, in Section 3 we refine (many simulations do not actually involve a polynomial overhead or extra $\log n$ factors) and prove the relations of Figure 1. In Section 4 we prove items 1 and 2 above. The paper is concluded with a few remarks and open problems in Section 5.

2. Preliminaries

A *literal* is a propositional variable x or its negation $\overline{x}$. We let $\overline{\overline{x}} \stackrel{\text{def}}{=} x$. A *clause* is a disjunction (possibly empty) of literals over distinct variables, and a *term* is a conjunction (possibly empty) of such literals. For a clause $C = \ell_1 \vee \cdots \vee \ell_w$, we define the term $\overline{C} \stackrel{\text{def}}{=} \overline{\ell_1} \wedge \cdots \wedge \overline{\ell_w}$; similarly for a term $t = \ell_1 \wedge \cdots \wedge \ell_w$, $\bar{t} \stackrel{\text{def}}{=} \overline{\ell_1} \vee \cdots \vee \overline{\ell_w}$. The *width* of a clause or a term is the number of literals it contains. A *CNF formula* is a conjunction of clauses, and a *DNF formula* is a disjunction of terms. The *width*, $W(F)$, of a CNF or DNF formula F is the width of the largest clause or term in it. A CNF or DNF formula of width at most w is called w-*CNF* or w-*DNF* respectively. Clauses may be alternatively viewed as 1-DNFs, but the latter class is slightly larger as tautological 1-DNFs like $x \vee \overline{x}$ *are* allowed.

A *partial (truth) assignment* (often called *restriction*) is a mapping from a subset V of all propositional variables to $\{0, 1\}$; it is naturally extended to the negations of the variables in V by $\alpha(\overline{x}) \stackrel{\text{def}}{=} \overline{\alpha(x)}$. The result of applying a partial assignment α to a CNF formula F is another CNF formula $F|_\alpha$, obtained by deleting from F all literals ℓ such that $\alpha(\ell) = 0$ and deleting all clauses containing a literal ℓ such that $\alpha(\ell) = 1$. Similarly for DNF formulas. For a formula F, we write $\alpha \models F$ if every total extension of α satisfies F or, in other words, if $F|_\alpha$ is *semantically* equal to 1. For a set of formulas S, $\alpha \models S$ means $\alpha \models \bigwedge_{F \in S} F$, and for two sets of formulas S and T, we write $S \models T$ if all total assignments satisfying every formula in S also satisfy every formula in T. For a clause C, we denote by α_C the minimal partial assignment such that $\alpha_C \models \overline{C}$. Dually, for a partial assignment α let C_α be the maximal clause with the property $\alpha \models \overline{C}_\alpha$.

Resolution is a proof system operating with clauses. Its inference rules are:

$$\frac{C}{C \vee D}, \quad \frac{C \vee x \qquad D \vee \overline{x}}{C \vee D}. \tag{8}$$

The leftmost one is called the *weakening rule*; the rightmost one is called the *resolution rule*. We refer to the variable x in an application of the resolution rule as the variable being *resolved*. One of the reasons to include the (redundant) weakening rule is that it makes resolution proofs closed under restricting by a partial assignment.

The *width* $W(\pi)$ of a resolution proof π is defined as the maximum width of a clause in it. $W(F \vdash \bot)$ is usually defined as the minimum width $W(\pi)$

of a resolution refutation π of F. This definition, however, is ill-suited for those CNFs that themselves have large width, like the pigeonhole principle. We have found it way more natural and convenient to work with its slightly modified version used in [26] (definitions of similar flavor have been given in [27, 28], see also [29]) that we will denote by $W(\vdash_F \bot)$. It is defined as follows.

Instead of just allowing the clauses of F as axioms, we allow them to participate in the following more general F-*cut rule*:

$$\frac{D \vee \overline{\ell_1} \qquad \dots \qquad D \vee \overline{\ell_r}}{D}, \tag{9}$$

where $\ell_1 \vee \ldots \vee \ell_r$ is a clause of F. In case some $D \vee \overline{\ell_j}$ contains contradictory literals, it is removed from the premises. In particular, when $D = \ell_1 \vee \cdots \vee \ell_r$, the list of premises becomes empty so the clauses of F are still available as axioms. $W(\vdash_F \bot)$ is the minimum width of refutations that, along with the resolution rule, also use the F-cut rule.

It is easy to see that

$$W(\vdash_F \bot) \leq W(F \vdash \bot) \leq W(\vdash_F \bot) + W(F) - 1,$$

hence the difference between the standard definition and ours becomes immaterial when $W(F)$ is small, and it does not have any noticeable impact on the *size* of a refutation.

One immediate advantage of this definition is that if we replace $W(F \vdash \bot)$ with $W(\vdash_F \bot)$ in (1), (2), (4), (5) or (6), we need not keep the annoying terms $W(F)$ any more, they just disappear. Simulations on Figure 1 will work without any restrictions on the width of the refuted CNF. More advantages of a similar flavor will become clear later, see Theorems 3.4 and 4.2 in particular.

Let us also remark that resolution with the F-cut rule is nothing else but Gentzen's sequent calculus with only atomic cuts, restricted to proving sequents of the form $C_1, \ldots, C_m \to$, where $C_1, \ldots, C_m$ are clauses (see [25]).

DNF resolution, or depth 2 Frege, is the straightforward extension of resolution where we allow, apart from variables in the resolution rule, also formulas of depth 1[5] to be resolved. DNF resolution operates with DNF

[5] For this reason, some authors use the term "depth 1 Frege" for DNF resolution; we prefer to stick to the convention under which depth refers to *lines* in a proof.

formulas. Its axioms and inference rules are:

$$\frac{}{x \vee \overline{x}}, \quad \frac{G}{G \vee H}, \quad \frac{G \vee t_1 \qquad H \vee t_2}{G \vee H \vee (t_1 \wedge t_2)}, \quad \frac{G \vee t \qquad H \vee \overline{t}}{G \vee H},$$

where G and H are DNF formulas and t, t_1, t_2 and $t_1 \wedge t_2$ are terms. The leftmost rule is the weakening rule in this context, and the rightmost rule is called the *cut rule*. The remaining rule allows us to deal with $\wedge$ connectives, and is called $\wedge$-*introduction*.

For a non-decreasing function $f : \mathbb{N} \to \mathbb{N}$, $R(f)$ is the subsystem of DNF resolution where each DNF in a proof of size s is required to have width at most $f(s)$. $R(k)$ for k a constant is usually denoted by $\mathrm{Res}(k)$ (thus, resolution is $\mathrm{Res}(1)$). DNF resolution and $R(f)$ were first introduced in [30].

Next, we would like to consider systems for manipulating terms. The syntactic details of such systems will not matter for our results, but for concreteness, let us present a prominent system of algebraic flavor originally introduced in [3]. We will actually use an extension, proposed in [6], called *polynomial calculus with resolution* and abbreviated as *PCR*. PCR works with a fixed field $\mathbb{F}$. Clauses/terms are represented as *monomials*. The syntactic objects PCR operates with are *polynomials* in $\mathbb{F}[x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}]$, represented as linear combinations over $\mathbb{F}$ of monomials, and a proof line P is to be interpreted as asserting that $P = 0$. The axioms and inference rules of the system are:

$$\frac{}{\ell^2 - \ell}, \quad \frac{}{\ell + \overline{\ell} - 1}, \quad \frac{P \qquad Q}{\alpha P + \beta Q}, \quad \frac{P}{\ell P},$$

where $\ell \in \{x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}\}$, $P, Q \in \mathbb{F}[x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}]$ and $\alpha, \beta \in \mathbb{F}$.

In each of the above systems, non-logical axioms are given as a set of clauses S, viewed as a CNF formula F (in PCR, a clause $C = \ell_1 \vee \cdots \vee \ell_k \in S$ is represented as the monomial $\overline{\ell_1} \ldots \overline{\ell_k}$). A proof of the unsatisfiability of F, or a *refutation* of F, is a derivation of a syntactic contradiction, denoted by $\perp$, from the clauses of F. In resolution and DNF resolution $\perp$ is the empty clause; in PCR, it is the polynomial 1.

We can view proofs as DAGs, by drawing edges from premises to conclusions in applications of the inference rules. If a proof DAG is a tree, that is every formula or polynomial in it is used as a premise at most once, then we say that the proof is *tree-like*. The *size* of a tree-like proof is the number of its leaves, and its *depth* is the length of its longest root-to-leaf path. We denote tree-like size and depth by S_T and D respectively. Let us note that one could define the size of a tree-like proof in a couple of different ways: as the number of nodes of the proof tree, as the total number of symbols in the proof in the case of resolution proofs, or the total number of monomials occurring in the proof in the case of PCR proofs, to name some reasonable choices. For the tree-like versions of the systems we are considering (warning: this is no longer true for DAG-like PCR), the particular choice is inessential, as all these measures are polynomially related.

We will also consider a one-sided version of depth, which we call *positive depth*. (The analogue of this notion in the context of computational complexity was recently defined in [31].) The positive depth of a tree-like resolution proof is the maximum number of *negative* literals introduced along a root-to-leaf path: With a slight abuse of notation,[6] the positive depth of the root clause C is $D_P(C) \stackrel{\text{def}}{=} 0$, and for each occurrence $C \vee x, D \vee \overline{x} \vdash C \vee D$ of the resolution rule in the proof, $D_P(C \vee x) \stackrel{\text{def}}{=} D_P(C \vee D)$ and $D_P(C \vee \overline{x}) \stackrel{\text{def}}{=} D_P(C \vee D) + 1$. The positive depth of the proof itself is the maximum of $D_P(C)$ over leaves C in the proof.

To define space complexity measures, we need to consider a different topology, namely view a proof as a sequence of *memory configurations* [5, 6]. A memory configuration will be a set of clauses in resolution, a set of DNF formulas in DNF resolution, or a set of polynomials in PCR. In a proof from a CNF F then, to go from a memory configuration to the next we may do one of the following:

Axiom Download: add a clause of the formula F, or a logical axiom of the system we are working with;

Erasure: delete a clause/DNF formula/polynomial, or

Inference: add the result of applying an inference rule to formulas in the current configuration.

A proof in configurational form is said to be *tree-like* if, whenever a formula is used as a premise in an inference rule, it is immediately erased from the memory.

The *clause space* of a configuration in resolution is the number of clauses it contains, its *variable space* the number of *distinct* variables it contains, and its *total space* the total number of literals, counting repetitions, it contains.

[6] Strictly speaking D_P is a function of a node in the proof, not of the clause sitting at it.

For DNF resolution, we will be interested in what we call Σ_2 *space* of a configuration. The Σ_2 space of a configuration $\mathcal{M} = \{G_1, \ldots, G_s\}$ is defined as the sum of widths: $\Sigma_2\mathrm{Space}(\mathcal{M}) \stackrel{\mathrm{def}}{=} W(G_1) + \ldots + W(G_s)$. (Recall that the width of a DNF G is the number of literals in its largest term.) For PCR, we will consider the *monomial space* of a configuration, which is the number of *distinct* monomials in it.

For a space measure μ on configurations and a proof $\pi = \mathcal{M}_1, \ldots, \mathcal{M}_t$, we naturally let $\mu(\pi) \stackrel{\mathrm{def}}{=} \max\{\mu(\mathcal{M}_i) \mid 1 \leq i \leq t\}$. As in [12], we will also consider *regularized* versions μ^* defined as $\mu^*(\pi) \stackrel{\mathrm{def}}{=} \mu(\pi) \cdot \log|\pi|$, where $|\pi| \stackrel{\mathrm{def}}{=} t$ is the *length*, that is the number of configurations, of π. All logarithms in this paper have base 2.

Finally, for a complexity measure μ on proofs, we write $\mu(F \vdash G)$ for the minimum value of $\mu(\pi)$, taken over all proofs of G from F; if such a proof does not exist, we set $\mu(F \vdash G)$ to be ∞. In most cases, the measure μ clearly suggests what the underlying proof system should be. For example, $W(F \vdash \bot)$ is the minimum width of a resolution refutation of F, and $\mathrm{MSpace}^*(F \vdash \bot)$ is the minimum regularized monomial space of a PCR refutation (in configurational form) of F. $S_T(F \vdash \bot)$ shall mean the minimum size of a tree-like resolution refutation of F. We shall use the notation $S_{T,R(f)}(F \vdash \bot)$ to mean the minimum size of a tree-like $R(f)$-refutation of F. $\mathrm{TCSpace}(F \vdash \bot)$ is the minimum clause space taken over all tree-like configurational refutations of F in resolution. Likewise, $\mathrm{RCSpace}(F \vdash \bot)$ stands for the clause space in *regular* resolution, i.e. the subsystem of resolution where we require that a variable cannot be resolved more than once on any path in (the DAG resulting from the expansion of) the configurational proof π.

Table 1 summarizes the complexity measures considered in this paper.

3. Simulations

3.1. Tree-like resolution size and regularized monomial space

First we show that $\log S_T$ in resolution, TCSpace, RCSpace, CSpace* and MSpace*, are all equivalent. Our main new contribution is the following simulation.

Theorem 3.1. *For any unsatisfiable CNF formula F over n variables,*

$$\log S_T(F \vdash \bot) \leq 2\mathrm{MSpace}^*(F \vdash \bot)\log(n+1),$$
$$\mathrm{TCSpace}(F \vdash \bot) \leq 2\left(\mathrm{MSpace}^*(F \vdash \bot) + 1\right).$$

Tree-like resolution	$S_T(\pi) =$ number of leaves in π; $\mathrm{TCSpace}(\pi) = \max\{\lvert\mathcal{M}\rvert \mid \mathcal{M} \in \pi\}$
Regular resolution	$\mathrm{RCSpace}(\pi) = \max\{\lvert\mathcal{M}\rvert \mid \mathcal{M} \in \pi\}$
Resolution	$W(\pi) = \max\{W(C) \mid C \in \pi\}$; $\mathrm{CSpace}(\pi) = \max\{\lvert\mathcal{M}\rvert \mid \mathcal{M} \in \pi\}$; $\mathrm{VSpace}(\pi) = \max\{\lvert\mathrm{Vars}(\mathcal{M})\rvert \mid \mathcal{M} \in \pi\}$; $\mathrm{TSpace}(\pi) = \max\{\sum_{C\in\mathcal{M}} W(C) \mid \mathcal{M} \in \pi\}$; $\mathrm{C/V/TSpace}^*(\pi) = \mathrm{C/V/TSpace}(\pi)\log\lvert\pi\rvert$
DNF resolution	$\Sigma_2\mathrm{Space}(\pi) = \max\{\sum_{G\in\mathcal{M}} W(G) \mid \mathcal{M} \in \pi\}$; $\Sigma_2\mathrm{Space}^*(\pi) = \Sigma_2\mathrm{Space}(\pi)\log\lvert\pi\rvert$
PCR	$\mathrm{MSpace}(\pi) = \max\{$number of monomials in $\mathcal{M} \mid \mathcal{M} \in \pi\}$; $\mathrm{MSpace}^*(\pi) = \mathrm{MSpace}(\pi)\log\lvert\pi\rvert$

Table 1: Complexity measures in different proof systems

Proof. The proof is analogous to the construction in [12] showing that depth is upper bounded by regularized variable space. Let $\mathcal{M}_1, \ldots, \mathcal{M}_t$ be a refutation of F in configurational form, of monomial space s. We show, by induction on d, that for every interval $[i..j] \subseteq [1..t]$ with $j > i$, $j - i \leq 2^d$, and for every clause D such that $\alpha_D \models \mathcal{M}_i$ and $\alpha_D \models \neg\mathcal{M}_j$, it holds that $S_T(F \vdash D) \leq (n+1)^{ds}$ and, moreover, the assumed tree-like resolution proof can be carried out in clause space at most $ds + 2$. The theorem follows by taking $[i..j] := [1..t]$, $d := \lceil \log t \rceil$ and $D := \bot$.

Suppose that $d = 0$, so that $j = i + 1$. The statement is vacuously true except when the step consists in downloading an axiom C from F, simply because in all other cases we have $\mathcal{M}_i \models \mathcal{M}_{i+1}$ and hence D with the specified properties does not even exist. Let D be a clause for which $\alpha_D \models \mathcal{M}_i$ and $\alpha_D \models \neg(\mathcal{M}_i \cup \{C\})$. Then we necessarily must have $\alpha_D \models \neg C$, which is equivalent to saying that D is a weakening of C.

For the inductive step, suppose that $d > 0$, let $[i..j] \subseteq [1..t]$ be any interval with $j - i \leq 2^d$, $j > i+1$, and let D be a clause such that $\alpha_D \models \mathcal{M}_i$ and $\alpha_D \models \neg\mathcal{M}_j$. Set $k := i + \lceil (j-i)/2 \rceil$, so that $k - i \leq 2^{d-1}$ and $j - k \leq 2^{d-1}$. Let the list $m_1, \ldots, m_s$ contain all monomials occurring in $\mathcal{M}_k$. For a clause A and a monomial $m = \ell_1 \ldots \ell_r$, consider the tree shown in Figure 2 designating a derivation of A in resolution; it is obtained from the obvious decision tree deciding m by reversing edges and weakening the result by A. Denote this tree by $\mathbf{T}_{A;m}$.

Let us now describe the required tree-like resolution proof of D. Start with $\mathbf{T}_{D;m_1}$. To every leaf of $\mathbf{T}_{D;m_1}$ labelled by a clause D', append the tree

$$
\begin{tikzcd}
 & A & & & \\
A \vee \ell_1 \arrow[ur] & & A \vee \overline{\ell_1} \arrow[ul] & & \\
 & A \vee \ell_2 \arrow[ur] & & A \vee \overline{\ell_1} \vee \overline{\ell_2} \arrow[ul] & \\
 & & & \vdots & \\
 & & & A \vee \overline{\ell_1} \vee \cdots \vee \overline{\ell_{r-1}} & \\
 & & A \vee \ell_r \arrow[ur] & & A \vee \overline{\ell_1} \vee \cdots \vee \overline{\ell_r} \arrow[ul]
\end{tikzcd}
$$

Figure 2: The tree $\mathbf{T}_{A;m}$

$\mathbf{T}_{D';m_2}$. Continue this process for all $m_1, \ldots, m_s$. If at any point during this construction, a forbidden disjunction containing a variable and its negation occurs, then we delete that node and merge its sibling with its parent. The resulting tree $\mathbf{T}$ has at most $(n+1)^s$ leaves, and each of its leaves is labelled by a clause E such that $\alpha_E \models \mathcal{M}_k$ or $\alpha_E \models \neg\mathcal{M}_k$. From the induction hypothesis, there are tree-like resolution proofs of all those clauses from F, of size $(n+1)^{(d-1)s}$. Therefore, there is a tree-like resolution proof of D from F of size $(n+1)^{ds}$.

To see that this proof can be carried out in clause space at most $ds+2$, notice that the proof designated by $\mathbf{T}$ can be carried out in clause space $s+2$. Proceed with this proof, and whenever a clause at its leaves is downloaded, keep all current clauses in memory (there are at most s of them — the maximum clause space is hit when the parent of two leaves is brought to memory), and derive it in clause space at most $(d-1)s+2$. The fact that such a derivation exists is guaranteed by the induction hypothesis. The resulting proof has clause space at most $s + (d-1)s + 2 = ds + 2$. $\square$

Remark 3.1. As we already remarked in the introduction, the above construction works for any *sound* system whose configurations are Boolean functions of monomials. In particular, it works for the purely semantic system called functional calculus [6]. Even more generally, it works for any proof system in which small configurations have low decision tree complexity.

For the rest of the relations, we claim that for an unsatisfiable CNF formula F over n variables, we have

$$\begin{aligned}
\mathrm{RCSpace}(F \vdash \bot) &\leq \mathrm{TCSpace}(F \vdash \bot) \\
&\leq \log S_T(F \vdash \bot) + 2 \\
&\leq 2\left(\mathrm{MSpace}^*(F \vdash \bot)\log(n+1) + 1\right) \\
&\leq 2\left(\mathrm{CSpace}^*(F \vdash \bot)\log(n+1) + 1\right) \\
&\leq 2\left(\mathrm{RCSpace}^*(F \vdash \bot)\log(n+1) + 1\right) \\
&\leq 2\Big((\mathrm{RCSpace}(F \vdash \bot))^2 \log(n+1)\log(2n) + 1\Big).
\end{aligned}$$

The first inequality follows from the observation that every tree-like refutation can be pruned to the regular form, and this operation does not increase its space. The second inequality is [5, Theorem 2.1], and the third is Theorem 3.1. The fourth and the fifth inequalities are obvious. Finally, the last inequality follows from [5, Corollary 4.2]. Namely, Esteban and Torán showed that if π is a resolution refutation, in configurational form, of clause space s and depth d, then

$$|\pi| \leq \binom{d+s}{s}. \tag{10}$$

Taking π to be a regular resolution refutation of minimum clause space, we get, since a regular refutation must have depth at most n,

$$\mathrm{RCSpace}^*(F \vdash \bot) \leq (\mathrm{RCSpace}(F \vdash \bot))^2 \log(2n).$$

As a byproduct, we get that TCSpace $\approx$ RCSpace. This comes in sharp contrast with the situation for size, where there is an exponential separation between tree-like and regular resolution [32].

We also see from (10) that, somewhat surprisingly, instead of regularizing clause space by multiplying it by the logarithm of *size*, we could have as well used a much weaker regularization multiplying by the logarithm (!) of depth, and the resulting measure would still be in this cluster. This allows us to rephrase the main open problem of whether CSpace $\approx$ CSpace* as whether there exists a *super-critical* (in the sense of [33]) trade-off between clause space and depth, that is, whether restricting clause space must necessarily result in proofs of depth $\gg n$. See Section 5 for more details.

The remaining (non-trivial) simulation on Figure 1 involving this cluster is:

Theorem 3.2. *For any unsatisfiable CNF formula F,*

$$\mathrm{TCSpace}(F \vdash \bot) \leq D_P(F \vdash \bot) + 2.$$

Proof. The argument is a refinement of the argument in [5] showing that tree-like clause space is bounded by depth. We show, by induction on $\mathbf{T}$, that if $\mathbf{T}$ is a tree-like resolution proof of a clause E from F of positive depth d, then there is a tree-like resolution proof, in configurational form, of E from F of clause space at most $d+2$.

If $\mathbf{T}$ has size at most 2, then $d \leq 1$, and $\mathrm{TCSpace}(F \vdash \perp) \leq 3$. Otherwise, let $\mathbf{T}_1$ and $\mathbf{T}_2$ be the subproofs of $\mathbf{T}$ proving the two clauses E_1 and E_2 respectively from which E is derived via an application of the resolution rule and possibly applications of the weakening rule. One of $\mathbf{T}_1$ and $\mathbf{T}_2$, say $\mathbf{T}_1$, must have positive depth at most $d-1$. From the induction hypothesis, there is a tree-like proof π_1 of E_1 of clause space at most $d+1$, and a tree-like proof π_2 of E_2 of clause space at most $d+2$. Deriving first E_2 using π_2, and then, keeping E_2 in memory, deriving E_1 using π_1, we get a proof of E of clause space at most $d+2$. □

3.2. Resolution width and Σ_2 space

The simulations for our second cluster will depend upon the following "locality" property of DNF resolution.

Lemma 3.3. *Let α be a partial assignment. For each of the inference rules of DNF resolution, if both premises contain a term satisfied by α, then α satisfies some term in the conclusion.*

The main theorem of this section says that as long as we transition from depth 1 Frege to depth 2 Frege, then not only width continues to be smaller than space, but in fact it becomes (almost) equal to it. As a historical remark, an extension of the Atserias-Dalmau bound (4) for the case of $\mathrm{Res}(k)$ is sketched in [8], and, although it is not stated explicitly, it is also apparent in [14].

Theorem 3.4. *For any unsatisfiable CNF formula F,*

$$\frac{1}{5}\Sigma_2\mathrm{Space}(F \vdash \perp) \leq W(\vdash_F \perp) \leq \Sigma_2\mathrm{Space}(F \vdash \perp).$$

Proof. Let $\mathcal{M}_1, \ldots, \mathcal{M}_t$ be a DNF resolution refutation of F, of Σ_2 space s. We will construct a sequence $\mathbf{T}_1, \ldots, \mathbf{T}_t$ of derivations in the system "resolution plus the F-cut rule (9)". The property we are going to maintain is that for every clause D labelling a leaf of $\mathbf{T}_i$, either D is a weakening of a clause C in F (call such a leaf an *axiom leaf*) or the following hold:

1. for every $G \in \mathcal{M}_i$, α_D satisfies some term of G;

2. $W(D) \leq \Sigma_2\text{Space}(\mathcal{M}_i)$.

$\mathbf{T}_1$ has one vertex labelled by the empty clause. Now suppose we have constructed $\mathbf{T}_{i-1}$ such that 1 and 2 hold for all non-axiom leaves. For every such leaf v labelled by a clause D, do the following.

- *Axiom Download*: Suppose that $\mathcal{M}_i = \mathcal{M}_{i-1} \cup \{C\}$, where $C = \ell_1 \vee \cdots \vee \ell_r$ is either a clause of F (viewed as a 1-DNF) or a logical axiom $x \vee \overline{x}$. If C and D contain conflicting literals, then item 1 is automatically satisfied and we do nothing at this leaf. Next, $C \subseteq D$ would have implied that C is a clause of F which is impossible since we have assumed that the leaf is non-axiom. Thus, there must exist a $j \in [r]$ such that $\ell_j \notin D$. For each such j, we add to v a child labelled by $D \vee \overline{\ell_j}$. This will be an application of the F-cut rule if C is a clause or of the resolution rule if C is $x \vee \overline{x}$.

- *Erasure*: Suppose that $\mathcal{M}_i \subseteq \mathcal{M}_{i-1}$. Add to v a single child labelled by a clause $E \subseteq D$ such that $W(E) \leq \Sigma_2\text{Space}(\mathcal{M}_i)$ and for every $G \in \mathcal{M}_i$, α_E satisfies some term of G.

- The case of an *inference* is immediately taken care of by Lemma 3.3, D does not change.

Since $\perp \in \mathcal{M}_t$, $\mathbf{T}_t$ cannot contain any non-axiom leaves and hence defines a refutation of F. Also, it is clear from the construction and property 2 above that any clause D appearing in it must satisfy $W(D) \leq \max_{1 \leq i \leq t} \Sigma_2\text{Space}(\mathcal{M}_i) = s$. Hence $W(\vdash_F \perp) \leq s$.

For the converse inequality, suppose that $C_1, \ldots, C_t$ is a refutation in the system "resolution plus the F-cut rule", of width w. For every $i \in [t-1]$, set $G_i \coloneqq \bigvee_{j=1}^{i} \overline{C_j}$. Each G_i is a w-DNF. For our small space refutation, we start by deriving G_{t-1} and $G_{i-1} \vee C_i$ for each $i \in [t-1]$. Having derived these formulas, we can use a series of cuts to derive the empty clause: From $G_{t-1} = G_{t-2} \vee \overline{C_{t-1}}$ and $G_{t-2} \vee C_{t-1}$ we can derive G_{t-2} by cutting on C_{t-1}, then from G_{t-2} and $G_{t-3} \vee C_{t-2}$ we can derive G_{t-3}, and so on. Notice that G_{t-1} contains a tautology of size $O(w)$, and hence has a derivation whose complexity depends only on w; one can sees in particular that G_{t-1} has a tree-like proof of Σ_2 space at most $3w$. To derive $G_{i-1} \vee C_i$, notice that either C_i is a clause of F, in which case we can immediately derive $G_{i-1} \vee C_i$, or $G_{i-1} \vee C_i$ is a tautology. In the case $G_{i-1} \vee C_i$ is a tautology, then C_i will be the result of applying either the resolution rule or the weakening rule or F-cut rule to some clauses among $C_1, \ldots, C_{i-1}$. In either case, it can be checked that $G_{i-1} \vee C_i$ has a tree-like proof of Σ_2 space at most $3w$. We therefore see that that the overall refutation of F can be carried in Σ_2 space at most $5w$. $\square$

Remark 3.2. In the refutation of Σ_2 space at most $5w$ constructed in the second part of the proof above, there is a constant number of formulas in every configuration. This implies that *a posteriori*, DNF resolution will retain its power in terms of space even if we restrict the *formula space* (the maximum number of DNFs in a configuration) to a constant. This in turn immediately implies, also a posteriori, that we can balance our definition of Σ_2 space replacing in it $W(G_1) + \ldots + W(G_s)$ with $s \cdot \max(W(G_1), \ldots, W(G_s))$ (since s is a constant, these expressions differ by at most a constant multiplicative factor), and the resulting measure will still be equivalent to Σ_2 space. We are not aware of a direct proof of this simulation by-passing width.

We get from Theorem 3.4 that strong length-space trade-offs conjectured for variable, clause and monomial space, are ruled out for DNF resolution. In particular, we get:

Corollary 3.5. *For any unsatisfiable CNF formula F with n variables,*

$$\Sigma_2\mathrm{Space}^*(F \vdash \bot) \leq O\left((\Sigma_2\mathrm{Space}(F \vdash \bot))^2 \log n \right).$$

Proof. Let $s := \Sigma_2\mathrm{Space}(F \vdash \bot)$. By the first part of Theorem 3.4, F has a width $O(s)$ resolution refutation with the additional F-cut rule. We apply to this refutation the construction from the second part of Theorem 3.4 in which we can clearly assume $t \leq n^{O(s)}$ (since all clauses in the sequence can be assumed to be different). By an easy inspection, the length of the resulting refutation will still be $n^{O(s)}$. Therefore,

$$\Sigma_2\mathrm{Space}^*(F \vdash \bot) \leq O(s^2 \log n). \qquad \square$$

Corollary 3.6. *If F has a constant Σ_2 space refutation, then it has a refutation of constant Σ_2 space and polynomial length.*

Proof. The refutation constructed in the proof of Corollary 3.5 will in our case also have constant Σ_2 space. $\square$

Let us finally deal with the remaining measure, tree-like proofs in $R(\log)$.

Theorem 3.7. *Let F be an unsatisfiable CNF formula over n variables. Then*

$$\Sigma_2\mathrm{Space}(F \vdash \bot)^{1/2} \leq \log S_{T,R(\log)}(F \vdash \bot) \leq O(W(\vdash_F \bot) \log n).$$

Proof. For the upper bound, let π be a resolution refutation of F of width $w := W(\vdash_F \bot)$. Apply to it the construction in the second part of the proof of Theorem 3.4 once again. By inspection (cf. the proof of Corollary 3.5), this refutation is tree-like, has size $n^{O(w)}$ and every term occurring in it has width at most w. Padding it with dummy formulas if necessary, we can assume that it has size $\geq 2^w$ which makes it into a tree-like $R(\log)$ refutation of the required size.

For the lower bound, the argument is an adaptation of the argument in [5] showing (3). Namely, by pebbling, a tree-like proof $\mathbf{T}$ of size $s > 1$ can be turned into a proof in configurational form, where each configuration contains at most $\log s$ formulas occurring in $\mathbf{T}$. If $\mathbf{T}$ is a refutation in $R(\log)$, then all terms occuring in $\mathbf{T}$ have width at most $\log s$, so the resulting refutation has Σ_2 space $(\log s)^2$. □

Remark 3.3. For the more conventional system $\mathrm{Res}(\log n)$, the subsystem of DNF resolution where each DNF in a refutation of F is required to have width $O(\log n)$, n being the number of variables of F, the second inequality in Theorem 3.7 is false (see Figure 4 in the concluding remarks section). This follows from an easy adaptation of the proof of [14, Corollary 14].

4. Size-space trade-offs and tree-like size lower bounds

4.1. A lower bound on regularized monomial space

One application of the results of the previous section is that they easily allow us to show trade-offs[7] between regularized clause or monomial space and size.

It is known [19, 20] that there are formulas F of size $\Theta(n)$ that have a resolution refutation of size $O(n)$ (and thus a $O(n)$ refutation in the stronger system PCR), but $\mathrm{MSpace}^*(F \vdash \bot) \geq n^{1/2}/(\log n)^{O(1)}$. Theorem 3.1, combined with the lower bounds of [32] and [34] on $\log S_T$ and TCSpace immediately gives the following improvement.

Theorem 4.1. *For every $n \geq 0$, there is a formula F of size $\Theta(n)$ that has a resolution refutation of size $O(n)$, width $O(1)$, and such that* $\mathrm{MSpace}^*(F \vdash \bot) \geq \Omega\left(n/\log n\right)$.

[7] We would like to stress that, following the (perhaps, unfortunate) convention established in the previous papers, we mean *potential* trade-offs. In other words, we prove lower bounds on the regularized space and we only know that *our method fails* to extend them to the ordinary monomial space. As we explained in Section 1 and will further elaborate in Section 5, proving *actual* trade-offs in this setting is a major and difficult open problem.

Proof. [32] demonstrates the existence of an $O(1)$-CNF F that has resolution refutations of size $O(n)$, width $O(1)$, and such that $\log S_T(F \vdash \bot) \geq \Omega(n/\log n)$. In fact, [32] shows that $\Omega(n/\log n)$ is also the lower bound on the number of points the Delayer can score in the Prover-Delayer game of [35] played on F. Now, it is proved in [34] that this number of points is precisely equal to $\text{TCSpace}(F \vdash \bot)$ and then the result immediately follows from the second inequality in Theorem 3.1. $\square$

4.2. *Trade-offs between positive depth and tree-like size for Horn formulas and tree-like size lower bounds*

We would like next to focus on tree-like size lower bounds for resolution attained for formulas with small clause space. We will show that a tree-like resolution refutation of a Horn formula actually describes a pebbling strategy, the space and time of the strategy being the positive depth and size respectively of the proof. This gives a more transparent version of the result of [32] used in the proof of Theorem 4.1, which moreover has a natural generalization allowing us to prove some tree-like lower bounds for formulas of small clause space.

4.2.1. *Horn formulas — basics*

Horn formulas, that include pebbling formulas, have seen a plethora of applications in proof complexity over the past two decades, including separating resolution size from tree-like resolution size [32], separating width from variable space and clause space [13, 24, 15], separating depth from tree-like clause space [22], and giving trade-offs [13, 15, 19, 18], to name a few.

A CNF formula is called *Horn* if every clause in it has at most one non-negated variable. Equivalently, a Horn formula is a set of implications involving variables, with at most one variable at the right hand side of the implication. An implication of the form $x_1, \ldots, x_k \to y$ is asserting that if all the x_i's are true, then y is true; $x_1, \ldots, x_k \to$ is asserting that one of the x_i's is false, $\to y$ is asserting that y is true, and $\to$ is a contradiction.

The following result states that Horn formulas make up, in a certain sense, the easiest class of formulas for proof complexity. For its purposes, it is convenient to define a slightly modified version $\text{CSpace}(\vdash_F \bot)$ of the clause space, in the same vein we defined $W(\vdash_F \bot)$ above. Namely, we replace the three standard rules with the following

Three-in-one rule: from a configuration $\mathcal{M}$, infer any configuration $\mathcal{M}^* \subseteq \mathcal{M} \cup F \cup \{C\}$, where C is obtained from clauses in $\mathcal{M}, F$ via a single application of the resolution or weakening rule.

Theorem 4.2. *Let F be a CNF formula. The following are equivalent:*

1. *F contains an unsatisfiable CNF sub-formula resulting from a Horn formula by negating some of its variables;*
2. $\mathrm{CSpace}(F \vdash \bot) \leq 3$*;*
3. $\mathrm{CSpace}(\vdash_F \bot) \leq 1$*;*
4. $W(\vdash_F \bot) \leq 1$*.*

Note that in order for the statement $W(\vdash_F \bot) \leq 1$ to make sense, the weakening rule has to be incorporated to the F-cut rule. That is, we use the F-cut rule in this theorem in the form:

$$\frac{D \vee \bar{\ell}_1 \qquad \ldots \qquad D \vee \bar{\ell}_r}{D \vee E}.$$

Proof. For $1 \implies 2$, we can w.l.o.g. assume that F itself is an unsatisfiable Horn formula. We show, by induction on the number of variables n, that it can be refuted in clause space 3. The base case is trivial. Now, suppose that $n > 0$, and let y be a variable such that F contains the clause $\to y$. Such a clause must exist, for if every clause contained a negated variable, then we could satisfy F by setting every variable to false. Setting $y := 1$, we get an unsatisfiable Horn formula $F|_{y:=1}$ with $n-1$ variables. From the induction hypothesis, there is a clause space 3 refutation of $F|_{y=1}$. Weakening every clause in it by $\overline{y}$ gives us a space 3 proof of $\overline{y}$ from F. Now we only have to download y and infer $\bot$.

For $2 \implies 3$, let $\mathcal{M}_1, \ldots, \mathcal{M}_t$ be a space 3 refutation of the formula F. We can assume that this refutation does not contain applications of the weakening rule (if it does we can simply erase them; it is easy to see that after this we will still get a legitimate refutation). Consider a path in the corresponding refutation tree of *maximum* possible length, say $C_i \in \mathcal{M}_{t_i}$ $(0 \leq i \leq h)$ are such that $t_0 < \ldots < t_h = t$, C_0 is an axiom, $C_h = \bot$ and for $i \geq 1$, C_i is obtained by resolving C_{i-1} with some $D_{i-1} \in \mathcal{M}_{t_i - 1}$. It remains to show that D_{i-1} is actually an axiom for any $i \geq 1$. For $i = 1$ this follows from the maximality of the chosen path. For $i \geq 2$, we have $\mathcal{M}_{t_{i-1}} = \{C_{i-2}, D_{i-1}, C_{i-1}\}$. Therefore C_{i-1} is consistent (and hence not resolvable) with the two other clauses in $\mathcal{M}_{t_{i-1}}$. All clauses that may have been inferred in $\mathcal{M}_{t_{i-1}+1}, \ldots, \mathcal{M}_{t_i}$ must have C_{i-1} as one of their premises and, as a consequence, are also not resolvable with C_{i-1}. Hence the only clauses in those configurations that may be resolvable with C_{i-1} (in particular, D_{i-1}) are the axioms.

The implication $3 \implies 4$ is proven by an argument similar to the first part of the proof of Theorem 3.4. Namely, a space 1 refutation of minimum length in the three-in-one model must necessarily be a sequence $\{D_1\}, \dots, \{D_t\}$, where D_{i+1} is obtained by resolving D_i with a clause C_i in F. We construct a sequence $\mathbf{T}_1, \dots, \mathbf{T}_t$ of width 1 derivations using *only* the F-cut rule, such that the non-axiom leaves of $\mathbf{T}_i$ are all those literals among $\overline{\ell_{i,1}}, \dots, \overline{\ell_{i,r_i}}$, where $D_i = \ell_{i,1} \vee \dots \vee \ell_{i,r_i}$, that are not axioms of F. To get $\mathbf{T}_{i+1}$ from $\mathbf{T}_i$, we add to every non-axiom leaf v of $\mathbf{T}_i$ a child labelled by ℓ for every literal ℓ of C_i that does not conflict with the label of v.

Finally, for $4 \implies 1$, we again proceed by induction on the number of variables n of F. The base case is trivial. Suppose that $n > 0$. The fact that there is a width 1 refutation of F, forces F to have a one literal clause (since the refutation must start somewhere), say ℓ. Setting $\ell \coloneqq 1$, we get a width 1 refutation of $F|_{\ell:=1}$. From the induction hypothesis, a sub-formula G of $F|_{\ell:=1}$ is unsatisfiable Horn up to negating some variables. Let $\widehat{G}$ be the corresponding sub-formula of F; $\widehat{G}$ is obtained from G by restoring $\overline{\ell}$ to some of its clauses. Then $\widehat{G} \wedge \ell$ is an unsatisfiable Horn sub-formula of F. $\square$

4.2.2. Tree-like resolution proofs as pebbling strategies

The paper [13] shows that a configurational resolution refutation π of the so-called *pebbling contradiction* Peb_G on a graph G defines a pebbling strategy on G, of time at most $|\pi|$ and space equal to the variable space $\mathrm{VSpace}(\pi)$. These are strategies in the so-called black-white game of [36]. We shall show that a tree-like resolution proof $\mathbf{T}$ of any Horn formula H defines a pebbling strategy of time equal to the size of $\mathbf{T}$ and space essentially equal to the *positive* depth of $\mathbf{T}$. These are strategies in the more basic black-only pebbling game that in the case $H = \mathrm{Peb}_G$ corresponds to the black-only pebbling game on G. Urquhart [22] showed how to relate them to ordinary depth. In a sense, our Proposition 4.3 below can be viewed as a (far-reaching) refinement of his result.

The rules of the black-only pebbling game, played on a Horn formula H, are as follows. There is a limited amount of pebbles. Pebbles are placed on the variables of H according to the rules:

1. A pebble can be placed on a variable y if $x_1, \dots, x_k \to y$ is a clause of H, and all $x_1, \dots, x_k$ have pebbles on them. In particular, a pebble can be always placed on any variable y such that $\to y$ is a clause of H.
2. A pebble can be removed from a variable at any time.

A *configuration* of the pebbling game is a set of the variables of H. A *pebbling strategy* is a sequence of configurations, each resulting from the previous one by one of the rules above. We say that a pebbling strategy *refutes* H if it ends with a configuration where for some clause $x_1, \ldots, x_k \rightarrow$ of H, all variables $x_1, \ldots, x_k$ are pebbled. Note that if H is unsatisfiable, then such a clause $x_1, \ldots, x_k \rightarrow$ must exist.

Proposition 4.3. *Let H be an unsatisfiable Horn formula. A tree-like resolution refutation* $\mathbf{T}$ *of H of size s and positive depth d can be converted into a pebbling strategy that, starting with the empty configuration, refutes H in at most s steps and using at most $d+1$ pebbles.*

Proof. We begin by modifying the original refutation in such a way that as a first step, suitable weakenings of the clauses in H are derived so that the original refutation can be carried out from these using the symmetric resolution rule:

$$\frac{C \vee x \qquad C \vee \overline{x}}{C}. \tag{11}$$

More precisely, every leaf of $\mathbf{T}$ naturally corresponds to a partial assignment α, and we replace the axiom sitting at that leaf with its weakening C_α.

This refutation need not necessarily consist of Horn formulas even if the original one did so. Nonetheless we will still represent clauses in the sequential form $S \rightarrow T$, where S, T are disjoint sets of variables. Note that due to the symmetry of the rule (11), the positive depth $D_P(S \rightarrow T)$ as defined in Section 2 (that is, relative to $\mathbf{T}$) is equal to $|S|$. In particular, $|S| \leq d$ for any clause $S \rightarrow T$ appearing in $\mathbf{T}$.

We shall now show by induction that every subtree of $\mathbf{T}$ deriving a clause $S \rightarrow T$, describes a pebbling strategy that, starting with pebbles on all variables of S and using at most $d+1$ pebbles, either refutes H, or ends with a configuration which has pebbles on all variables of S and on one variable of T. Thus, if T is empty then the former must occur and, in particular, the strategy corresponding to the empty sequent $\rightarrow$ will start with no pebbles on the variables of H and will refute H.

Suppose that $S \rightarrow T$ is at a leaf. If there are variables $x_1, \ldots, x_k$ in S such that $x_1, \ldots, x_k \rightarrow$ is a clause of H, then that leaf describes a strategy that, starting with pebbles on all variables in S, immediately refutes H. Otherwise, since H is Horn, there must be variables $x_1, \ldots, x_k$ in S and a variable y in T such that $x_1, \ldots, x_k \rightarrow y$ is a clause of H. Then the strategy of that leaf is to put a pebble on y. Since $|S| \leq d$, the number of pebbles used is at most $d+1$, as required.

If $S \to T$ is not at a leaf, then consider its left and right subtrees $\mathbf{T}_1$ and $\mathbf{T}_2$ proving $S, x \to T$ and $S \to T, x$ respectively (cf. (11)). The strategy corresponding to $S \to T$ is defined as follows. First follow $\mathbf{T}_2$'s strategy. If that strategy either refutes H or places a pebble on one of T's variables, then we are done. Otherwise, when the strategy of $\mathbf{T}_2$ is concluded, there are pebbles on S and x. Remove all other pebbles and follow the strategy of $\mathbf{T}_1$. The bound $d+1$ on the number of pebbles used at any moment follows from the same bound for $\mathbf{T}_1$ and $\mathbf{T}_2$.

Clearly, the number of steps of the pebbling strategy corresponding to $\to$ is at most the size of $\mathbf{T}$, and the required bound on the number of pebbles was already noticed. ☐

Remark 4.1. The proof of Proposition 4.3 relies on an intuitionistic interpretation of the resolution rule. In the intuitionistic tradition, the denotational view of assigning truth values is, philosophically, nonsense. A proposition is "true" if it is provable, and a proof of e.g. a formula $S \to T$ is a construction that given proofs of all the elements of S produces a proof of some element in T. What Proposition 4.3 says is that a tree-like resolution derivation of $S \to T$ precisely describes such a construction, assuming that proofs of all the clauses of H are known. Moreover this construction will be economical in the number of steps and memory if the size and the positive depth respectively of the proof are small. Let us further notice, that although Proposition 4.3 is stated for Horn formulas, it really is general; it could be stated, with minimal modifications, for arbitrary CNFs.

4.2.3. Tree-like size lower bounds

The following theorem turns pebbling time-space trade-offs for a Horn formula H into tree-like size lower bounds for its substituted version $H[\vee_2]$. We formulate it in a somewhat general form, to account for various kinds of pebbling trade-offs in the literature. The substituted version $F[\vee_2]$ of a CNF $F(x_1, \ldots, x_n)$ is defined by replacing x_i with $y_i \vee z_i$ for mutually distinct variables $\{y_1, z_1, \ldots, y_n, z_n\}$, followed by converting the result back to the CNF form in the straightforward way.

Theorem 4.4. *Let H be an unsatisfiable Horn formula on n variables. Suppose that every pebbling strategy that refutes H in s steps and using d pebbles, starting with no pebbles on its variables, satisfies $(d-1)f(s) \geq g(n)$ for non-decreasing positive functions f, g. Then $f(t)\log t \geq g(n)$, where $t := S_T(H[\vee_2] \vdash 0)$.*

Proof. Let $\mathbf{T}$ be a tree-like resolution refutation of $H[\vee_2]$ of size t represented as in the proof of Proposition 4.3. That is weakenings appear at the leaves only and are omitted from the resolution rule, so that all applications of the resolution rule have the form (11). Recall that this implies that the positive depth of a clause in this proof is exactly the number of negated variables in it.

Create a probability space of partial assignments by choosing independently for every variable x of H, which was substituted by $y \vee z$, one of y and z with probability $1/2$ and setting it to zero. Note that for any α from this space, $H[\vee_2]|_\alpha$ is identical to H up to renaming its variables and hence $\mathbf{T}|_\alpha$ is a refutation of H, again up to renaming variables. Let $D_1, \ldots, D_k$ be all clauses of positive depth at least $g(n)/f(t)$ occurring in $\mathbf{T}$, that is, all clauses that contain at least $g(n)/f(t)$ negated variables. We have that

$$P\left[\bigvee_{i=1}^{k} (D_i|_\alpha \neq 1)\right] \leq \sum_{i=1}^{k} P[D_i|_\alpha \neq 1] \leq k2^{-g(n)/f(t)} \leq t2^{-g(n)/f(t)}.$$

If $f(t)\log t < g(n)$, then the above probability is smaller than 1, which means that there is a point α in our sample space such that $\mathbf{T}|_\alpha$ is a tree-like resolution refutation of size at most t and positive depth $\leq g(n)/f(t)$. This, from Proposition 4.3, gives a pebbling strategy that refutes H in t steps using d pebbles, where $(d-1)f(t) < g(n)$. □

For a DAG G, the pebbling contradiction Peb_G is defined as the Horn formula consisting of all clauses $S \to x$, where $x \in V(G)$ and S is the set of all its immediate predecessors, as well as the clauses $x \to$ for any sink x. Plugging into Theorem 4.4 various DAGs from the literature with known bounds on their pebbling complexity and various functions f, we can get several corollaries. The first is a simplified proof of the separation by Ben-Sasson et al.

Corollary 4.5 [32]**.** *For every n, there is a formula F of size $O(n)$ that has DAG-like resolution refutations of size $O(n)$, and such that every tree-like resolution refutation of F requires size* $\exp(\Omega(n/\log n))$.

Proof. First notice that a constant width and linear size refutation of an unsatisfiable Horn formula always exists from Theorem 4.2, and such a refutation remains of constant width and linear size after substituting each x_i with $y_i \vee z_i$. Hence there is always a linear size refutation of $H[\vee_2]$. The lower bound follows by setting $f \coloneqq 1$ in Theorem 4.4, and using the graphs of [37] having constant in-degree and requiring $g(n) = \Omega(n/\log n)$ pebbles to pebble. □

The next result was promised in the introduction. It should be compared with Theorem 4.2.

Theorem 4.6. *For every n, there is a formula F of size $O(n)$ that has tree-like resolution refutations of clause space 4, and such that every tree-like resolution refutation of F has size $\Omega(n^2/\log n)$.*

Proof. The lower bound follows by setting $f(t) := t$ in Theorem 4.4, and using the graphs of [38, Theorem 2.3.2] having linear size and exhibiting a $ds \geq g(n) = \Omega(n^2)$ time-space trade-off. These graphs can be pebbled using 3 pebbles, and that immediately gives that $\mathrm{CSpace}(\mathrm{Peb}_{G_n}[\vee_2] \vdash \bot) \leq O(1)$. By being more careful however, we can bring the space down to the minimum possible value for which a super-linear lower bound on tree-like resolution size is possible, which is 4 by Theorem 4.2.

More precisely, the above graphs have the following form. They contain two directed vertex-disjoint paths, let us call them U and L, and there are additional edges from vertices of U to vertices of L (see Figure 3). Moreover, there are no two vertices in U both with edges to the same vertex in L.

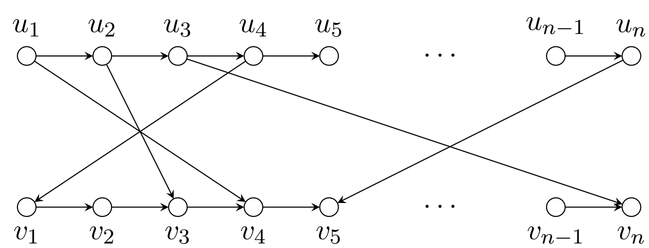

Figure 3: The form of the graphs giving the trade-off in Theorem 4.6

Let G be such a graph, and let $H \stackrel{\text{def}}{=} \mathrm{Peb}_G$ be the corresponding Horn formula. Call the variables of the path U, $u_1, \ldots, u_n$ and the variables of L, $v_1, \ldots, v_n$ as in Figure 3, and suppose that to obtain $H[\vee_2]$, u_i is substituted by $x_i \vee y_i$ and v_i by $z_i \vee w_i$. We first note that any clause $x_i \vee y_i$ can be derived in clause space 3.

Indeed, after assigning $x_i := 0$, $y_i := 0$ in $H[\vee_2]$, we will get an unsatisfiable formula F such that if we additionally negate all its variables, the result will be Horn. Hence, by Theorem 4.2, $\mathrm{CSpace}(F \vdash \bot) \leq 3$. To get the desired derivation, we simply weaken all clauses in this refutation by appending $x_i \vee y_i$.

Notice that the derivations provided by Theorem 4.2 are tree-like. To show the bound $\mathrm{CSpace}(H[\vee_2] \vdash \bot) \leq 4$, we need to show, having derived

$z_{i-1} \vee w_{i-1}$, how to derive $z_i \vee w_i$. Suppose that $\overline{u_j} \vee \overline{v_{i-1}} \vee v_i$ is a clause of H, so that $\overline{x_j} \vee \overline{z_{i-1}} \vee z_i \vee w_i$, $\overline{x_j} \vee \overline{w_{i-1}} \vee z_i \vee w_i$, $\overline{y_j} \vee \overline{z_{i-1}} \vee z_i \vee w_i$ and $\overline{y_j} \vee \overline{w_{i-1}} \vee z_i \vee w_i$ are clauses of $H[\vee_2]$. Notice that

$$\mathrm{CSpace}\left(\begin{array}{c}(z_{i-1} \vee w_{i-1}) \wedge \\ (x_j \vee y_j) \wedge \\ (\overline{x_j} \vee \overline{z_{i-1}} \vee z_i \vee w_i) \wedge \\ (\overline{y_j} \vee \overline{z_{i-1}} \vee z_i \vee w_i)\end{array} \vdash\ w_{i-1} \vee z_i \vee w_i\right) \leq 4.$$

In this derivation, all premises are immediately removed from memory after they are used as premises in an inference. Similarly, we have

$$\mathrm{CSpace}\left(\begin{array}{c}(w_{i-1} \vee z_i \vee w_i) \wedge \\ (x_j \vee y_j) \wedge \\ (\overline{x_j} \vee \overline{w_{i-1}} \vee z_i \vee w_i) \wedge \\ (\overline{y_j} \vee \overline{w_{i-1}} \vee z_i \vee w_i)\end{array} \vdash\ z_i \vee w_i\right) \leq 4.$$

Running the first derivation, deleting everything from memory except $w_{i-1} \vee z_i \vee w_i$ and then running the second derivation, deriving $x_j \vee y_j$ in clause space 3 whenever it is downloaded in these derivations, we get the desired clause space 4 derivation of $z_i \vee w_i$. □

By using the construction from [38, Theorem 4.2.6], Theorem 4.6 can be further generalized to a lower bound $(n/\log n)^{\Omega(k)}$ on the tree-like resolution size of refuting formulas with clause space k. Let us further notice that the fact that the space 4 refutation in Theorem 4.6 is tree-like might be interesting, as typically tree-like resolution size lower bounds have been proven in the literature based on the prover-delayer game of [35], which also gives a lower bound for the clause space of tree-like resolution refutations (cf. Theorem 4.1).

5. Concluding remarks

We showed that $\log S_T$, CSpace* and MSpace* are equivalent up to polynomial and $\log n$ factors, demonstrating a picture perfectly analogous to the picture involving D, VSpace* and TSpace* in [12]. The most important question remains (widely) open:

Problem 5.1. Is it true that CSpace $\approx \log S_T$ or MSpace $\approx \log S_T$? Recall for comparison that $\log S_T \approx$ CSpace$^* \approx$ MSpace*.

Equivalently, do there exist strong trade-offs between clause (or monomial) space and length? Strong trade-offs here means that restricting, let's say clause space, must necessarily give rise to refutations π of F of length $|\pi| \gg \exp(\mathrm{CSpace}(F \vdash \bot) \log n)$. Trade-offs of this kind should be contrasted with trade-off results in e.g. [15, 18]. Moreover, the existence of such trade-offs is a perfect analogue of Urquhart's question [22] about variable space vs. depth studied in [12, Section 6]. Let us make a few more remarks about this problem.

Firstly, for very small space essentially this question was already asked in the literature before. Namely (see e.g. [39, Open Problem 16]), are there formulas having constant clause space refutations and with the property that any such refutation must necessarily have super-polynomial length? Suitably adjusting it to our framework:

Problem 5.2 (small space variant). Are there formulas that have $(\log n)^{O(1)}$ clause or monomial space refutations and with the property that any such refutation must be of super-quasi-polynomial length $\exp((\log n)^{\omega(1)})$? Equivalently, any tree-like resolution refutation must have super-quasi-polynomial length.

In terms of the perceived difficulty, we do not discern too much of a difference between Problems 5.1, 5.2 and Nordström's question [39, Open Problem 16]. In fact, we would like to cautiously conjecture that there are formulas F with $\mathrm{CSpace}(F \vdash \bot) \leq 5$ and $\mathrm{CSpace}^*(F \vdash \bot) \geq \exp\left(n^{\Omega(1)}\right)$. But the only result we were able to prove in that direction is the rather weak Theorem 4.6.

Secondly, as suggested by Figure 1, any strong separation between monomial space and clause space would immediately solve Problem 5.1 for monomial space. As we consider the latter to be most likely very difficult, we take it as a good heuristic explanation of why we have not seen any progress on the former problem as well. But let us ask this, and one obviously relevant question, explicitly anyway:

Problem 5.3. Is it true that CSpace $\approx$ MSpace? Is it true that MSpace $\approx W$?

We note that by the result from [23, 24], at least one of these must be false. A quadratic separation between width and monomial space has been recently proved by the first author (manuscript in preparation). For a discussion on related topics, see also [40, Section 7.5.5].

Finally, while all these conjectured trade-offs are very strong, they are still not super-critical in the sense of [33], at least not according to our framework. A super-critical trade-off is a trade-off between two complexity measures in which restricting one measure causes an increase in the second measure that goes well beyond a worst case upper bound for it. Here, a worst case upper bound on proof length is 2^n, and every proof of clause space s can be assumed to have length $2^{O(sn)}$, as this is an upper bound on the total number of different configurations of clause space at most s. This still leaves us with a potential gap between 2^n and 2^{n^2} and, in fact, a result along these lines is known [41]. But from the perspective we are trying to advocate in this paper, their *logarithms* are polynomially related.

However, as we pointed out in Section 3.1, in all these questions refutation length can be replaced with depth. Since the depth, as a stand-alone measure, is always bounded by n, the question e.g. of whether CSpace $\approx$ CSpace* is actually the same as the question of whether there exists a *super-critical* trade-off between clause space and *depth*.

We have (somewhat surprisingly) proved that DNF resolution behaves very differently from resolution with respect to space (Theorems 3.4 and 3.7 and Corollaries 3.5 and 3.6). Intermediate systems based on Res(k) for a constant k were studied in a similar context before, and it is very natural to wonder what is the situation for those systems.

Let us first remark that for Res(k)-refutations, the definition of space from [14, 15] (formula space) coincides with ours up to a factor of k so we need not distinguish between the two. Then Theorem 3.1 readily generalizes to this regime and gives $\log S_{T,\mathrm{Res}(k)} \approx$ Res(k)Space*, extending the bottom half of Figure 1 as shown in Figure 4. The proof of Corollary 3.5, however, fails for a constant k as badly as it fails for $k = 1$. Hence we have one more question to ask:

Problem 5.4 (Res(k)-variant). Is there a constant $k > 0$ such that $\log S_{T,\mathrm{Res}(k)} \approx$ Res(k)Space or at least $\log S_{T,\mathrm{Res}(k)} \preceq$ CSpace?

Let us also mention that as k increases, both hierarchies, $\log S_{T,\mathrm{Res}(k)}$ (and, hence, also Res(k)Space*) and Res(k)Space are proper ([14] and [15] respectively). This excludes the dual version of Remark 3.2: while the formula space of DNF resolution refutations can be reduced to constant, this is not true for the widths of individual formulas in the memory.

The relation between VSpace and CSpace is also unknown in one direction (the opposite one is taken care of by [13]). Let us re-iterate the problem posed e.g. in [12]:

Problem 5.5. Is it true that CSpace $\preceq$ VSpace?

Just as with the questions of similar nature discussed above, a negative answer would also imply a separation between VSpace and VSpace*, hence we can expect it to be very difficult.

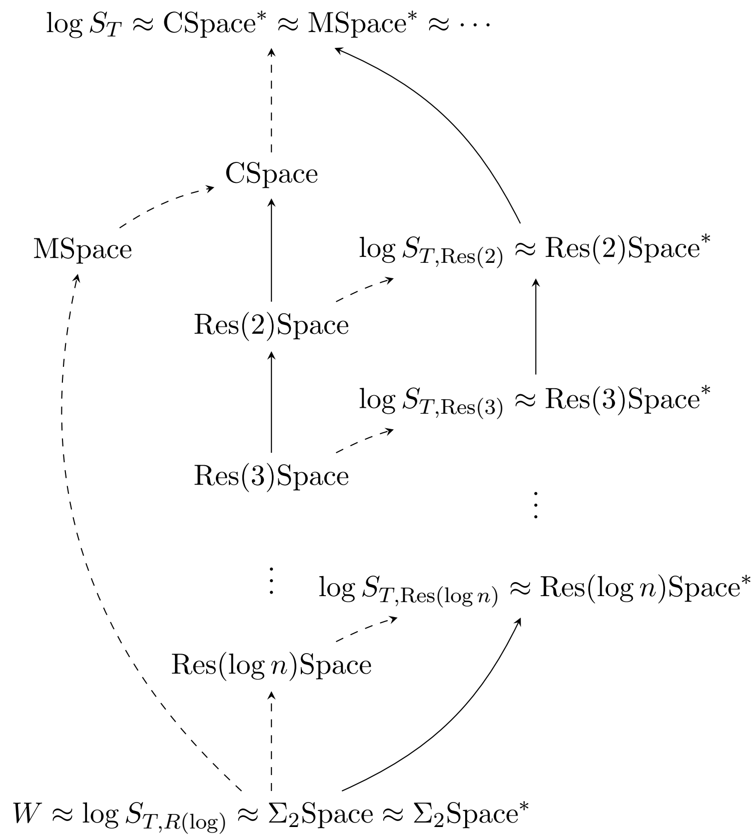

Figure 4: Σ_2 space and tree-like size for subsystems of DNF resolution

Acknowledgment

We are very grateful to the anonymous referees of our paper for many valuable remarks and suggestions.

References

[1] S. Cook, R. Reckhow, The relative efficiency of propositional proof systems, Journal of Symbolic Logic 44 (1979) 36–50.

[2] E. Ben-Sasson, A. Wigderson, Short proofs are narrow—resolution made simple, Journal of the ACM 48 (2001) 149–169.

[3] M. Clegg, J. Edmonds, R. Impagliazzo, Using the Groebner basis algorithm to find proofs of unsatisfiability, in: Proceedings of the 28th Annual ACM Symposium on Theory of Computing, 1996, pp. 174–183.

[4] R. Impagliazzo, P. Pudlák, J. Sgall, Lower bounds for the polynomial calculus and the Gröbner basis algorithm, Computational Complexity 8 (1999) 127–144.

[5] J. L. Esteban, J. Torán, Space bounds for resolution, Information and Computation 171 (2001) 84–97.

[6] M. Alekhnovich, E. Ben-Sasson, A. Razborov, A. Wigderson, Space complexity in propositional calculus, SIAM Journal of Computing 31 (2002) 1184–1211.

[7] A. Atserias, V. Dalmau, A combinatorial characterization of resolution width, Journal of Computer and System Sciences 74 (2008) 323–334.

[8] Y. Filmus, M. Lauria, M. Miksa, J. Nordström, M. Vinyals, From small space to small width in resolution, ACM Transactions of Computational Logic 16 (2015) 28:1–28:15.

[9] J. Krajíček, Proof Complexity, Cambridge University Press, 2019.

[10] I. Bonacina, Total space in resolution is at least width squared, in: Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming, Vol. 55, 2016, pp. 56:1–56:13.

[11] N. Galesi, L. Kołodziejczyk, N. Thapen, Polynomial calculus space and resolution width, in: Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science, 2019, pp. 1325–1337.

[12] A. Razborov, On space and depth in resolution, Computational Complexity 27 (2018) 511–559.

[13] E. Ben-Sasson, Size-space tradeoffs for resolution, SIAM Journal of Computing 38 (2009).

[14] J. L. Esteban, N. Galesi, J. Messner, On the complexity of resolution with bounded conjunctions, Theoretical Computer Science 321 (2004) 347–370.

[15] E. Ben-Sasson, J. Nordström, Understanding space in proof complexity: Separations and trade-offs via substitutions, in: Proceedings of the 2nd Symposium on Innovations in Computer Science, 2011, pp. 401–416.

[16] M. Lauria, A note about k-DNF resolution, Information Processing Letters 137 (2018) 33–39.

[17] J. Krajíček, Lower bounds to the size of constant-depth propositional proofs, Journal of Symbolic Logic 59 (1994) 73–86.

[18] C. Beck, J. Nordström, B. Tang, Some trade-off results for polynomial calculus: extended abstract, in: Proceedings of the 45th Annual ACM Symposium on Theory of Computing, 2013, pp. 813–822.

[19] T. Huynh, J. Nordström, On the virtue of succinct proofs: amplifying communication complexity hardness to time-space trade-offs in proof complexity, in: Proceedings of the 44th Annual ACM Symposium on Theory of Computing Conference, 2012, pp. 233–248.

[20] M. Göös, T. Pitassi, Communication lower bounds via critical block sensitivity, SIAM Journal of Computing 47 (2018) 1778–1806.

[21] M. L. Bonet, N. Galesi, Optimality of size-width tradeoffs for resolution, Computational Complexity 10 (2002) 261–276.

[22] A. Urquhart, The depth of resolution proofs, Studia Logica 99 (2011) 349–364.

[23] J. Nordström, J. Håstad, Towards an optimal separation of space and length in resolution, Theory of Computing 9 (2013) 471–557.

[24] E. Ben-Sasson, J. Nordström, Short proofs may be spacious: An optimal separation of space and length in resolution, in: Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science, 2008, pp. 709–718.

[25] T. Papamakarios, A super-polynomial separation between resolution and cut-free sequent calculus, Electronic Colloqium on Computational Complexity, Report No. 176 (2021).

[26] N. Galesi, N. Thapen, Resolution and pebbling games, in: Proceedings of the 8th Theory and Applications of Satisfiability Testing Conference, Vol. 3569, 2005, pp. 76–90.

[27] H. K. Büning, On generalized horn formulas and k-resolution, Theoretical Computer Science 116 (1993) 405–413.

[28] O. Kullmann, Investigating a general hierarchy of polynomially decidable classes of cnf's based on short tree-like resolution proofs, Electronic Colloqium on Computational Complexity, Report No. 041 (1999).

[29] O. Beyersdorff, O. Kullmann, Unified characterisations of resolution hardness measures, in: Proceedings of the 17th Theory and Applications of Satisfiability Testing Conference, 2014, pp. 170–187.

[30] J. Krajíček, On the weak pigeonhole principle, Fundamenta Mathematicae 170 (2001) 123–140.

[31] B. Loff, S. Mukhopadhyay, Lifting theorems for equality, in: Proceedings of the 36th International Symposium on Theoretical Aspects of Computer Science, Vol. 126, 2019, pp. 50:1–50:19.

[32] E. Ben-Sasson, R. Impagliazzo, A. Wigderson, Near optimal separation of tree-like and general resolution, Combinatorica 24 (2004) 585–603.

[33] A. Razborov, A new kind of tradeoffs in propositional proof complexity, Journal of the ACM 63 (2016) 16:1–16:14.

[34] J. L. Esteban, J. Torán, A combinatorial characterization of treelike resolution space, Information Processing Letters 87 (2003) 295–300.

[35] P. Pudlák, R. Impagliazzo, A lower bound for DLL algorithms for k-SAT, in: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, 2000, pp. 128–136.

[36] S. Cook, R. Sethi, Storage requirements for deterministic polynomial time recognizable languages, Journal of Computer and System Sciences 13 (1976) 25–37.

[37] W. Paul, R. Tarjan, J. Celoni, Space bounds for a game on graphs, Mathematical Systems Theory 10 (1977) 239–251.

[38] T. Lengauer, R. Tarjan, Asymptotically tight bounds on time-space trade-offs in a pebble game, Journal of the ACM 29 (1982) 1087–1130.

[39] J. Nordström, Pebble games, proof complexity, and time-space trade-offs, Logical Methods in Computer Science 9 (2013).

[40] S. Buss, J. Nordström, Proof complexity and SAT solving, in: A. Biere, M. Heule, H. van Maaren, T. Walsh (Eds.), Handbook of Satisfiability, 2nd Edition, IOS Press, 2021, Ch. 7, pp. 233–350.

[41] P. Beame, C. Beck, R. Impagliazzo, Time-space trade-offs in resolution: Superpolynomial lower bounds for superlinear space, SIAM Journal of Computing 45 (2016) 1612–1645.