Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Theoretical Computer Science 321 (2004) 347–370

Theoretical Computer Science

www.elsevier.com/locate/tcs

On the complexity of resolution with bounded conjunctions☆

Juan Luis Esteban[a,1], Nicola Galesi[a,*,2], Jochen Messner[b]

[a] *Department Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, Pau Gargallo 5, E-08028 Barcelona, Spain*

[b] *Abt. Theoretische Informatik, Universität Ulm, D-89069 Ulm, Germany*

Received 16 April 2003; received in revised form 15 March 2004; accepted 19 April 2004
Communicated by A. Razborov

Abstract

We analyze size and space complexity of $Res(k)$, a family of propositional proof systems introduced by Krajíček in (Fund. Math. 170 (1–3) (2001) 123) which extend Resolution by allowing disjunctions of conjunctions of up to $k \geqslant 1$ literals. We show that the treelike $Res(k)$ proof systems form a strict hierarchy with respect to proof size and also with respect to space. Moreover Resolution, while simulating treelike $Res(k)$, is almost exponentially separated from treelike $Res(k)$. To study space complexity for general $Res(k)$ we introduce the concept of dynamical satisfiability which allows us to prove in a unified way all known space lower bounds for Resolution and to extend them to $Res(k)$.

Keywords: Resolution; Complexity of Proofs

1. Introduction

A central theme in computational complexity is whether there is an *efficient* propositional proof system, i.e. a proof system that for any tautology provides a proof

☆ A preliminary version appeared in the Conference Proceedings of ICALP 2002 [15]. The authors were partially supported by the DAAD project *Acciones* integradas Hispano-Alemanas no. 70116.

* Corresponding author.
E-mail address: galesi@lsi.upc.es (N. Galesi).

[1] Partially supported by MEC through grant PB 98-0937-C04 (FRESCO Project).

[2] Part of this work was done while the author was at the Department of Computer Science of the University of Toronto, supported by a CSERC Research grant. The author was also partly supported by grant CICYT TIC2001-1577-C03-02.

doi:10.1016/j.tcs.2004.04.004

polynomial in the size of the tautology. As observed in [14] this corresponds to the question whether $NP = coNP$. Hence the investigation of the complexity of proof systems can be seen as a way to tackle $NP \neq coNP$: prove that for every propositional proof system there are tautologies that require superpolynomial proofs. Also in [14] it was proposed what is now known as Cook's program: try to find families of tautologies hard to prove for progressively more powerful propositional proof systems until having sufficient knowledge to prove $NP \neq coNP$, which implies $P \neq NP$.

Among the most studied proof systems are those related to resolution. While there are several lower bounds for the complexity of proofs in propositional resolution [19,33,7,11], resolution-based proof systems are still a subject of research [22,4,3]. On the one hand, one is interested in finding more and more powerful combinatorial lower bounds techniques that hopefully can be applied to stronger systems [27,28]. On the other hand, resolution-based proof systems are of practical interest in the field of automated theorem proving.

Given that no polynomial time algorithm can exist to find proofs for a non-efficient proof system like resolution, Bonet et al. [13] proposed the following approach: for a proof system P, find algorithms running in time polynomial in the size of the shortest P-proof of the formula we are seeking proofs for. If such an algorithm exists we say that P is *automatizable*. Despite that many proof systems, including resolution, are not automatizable conditioned on plausible complexity assumptions [2,23,13], there are examples of resolution-based proof systems known to be sub-exponentially automatizable [7,11,3].

This work focuses on the family of refutation systems $Res(k)$, $k \geqslant 1$, introduced by Krajíček in [22] as a generalization of resolution. Instead of clauses, $Res(k)$ allows to infer k-clauses, i.e. disjunctions of k-bounded conjunctions. In [4] Atserias et al. gave exponential lower bounds for $Res(2)$ refutations of random formulas and the weak $2n$ to n pigeonhole principle. Moreover, generalizing the algorithm of Beame et al. in [6], Atserias and Bonet in [3] provided a quasi-polynomial automatization for treelike $Res(k)$, the restricted system in which the proof is a tree. Segerlind et al. in [30] proved exponential lower bounds for $Res(k)$ refutations of random formulas and the weak pigeonhole principle, improving the results in [4]. Finally Razborov in [29], generalizes and improves the results of [30].

In the following we say that a proof system P *dominates* a proof system Q if (1) P polynomially simulates Q, i.e. for any Q-proof there is P-proof of the same formula that is polynomially related in size. And (2) P is almost exponentially separated from Q, i.e. there exists a class of formulas having polynomial size P-proofs, but requiring Q-proofs of almost exponential size. Let us note here that throughout the paper we will consider a lower bound of $2^{\Omega(n/\log n)}$ as *almost exponential*. It is known that resolution dominates treelike resolution [12,10]. We improve this result by proving that resolution dominates treelike $Res(k)$, for k constant. In fact we prove the stronger result that treelike $Res(k+1)$ dominates treelike $Res(k)$. Segerlind et al. in [30], proved the same result also for daglike $Res(k)$, so solving a natural open problem arising from our paper. Moreover, the treelike $Res(k)$ hierarchy is below resolution, since treelike $Res(k)$ is simulated by resolution, see [21]. In fact we give a simulation of treelike *Res*(k) by resolution that only doubles the size of the proof.

To prove the separations we extend to *Res*(k) a technique based on games introduced by Pudlák and Impagliazzo in [26] and used by Ben-Sasson et al. in [10] to give lower bounds for treelike resolution. Then we define a generalization of the pebbling contradictions from [10], and we extend the technique of [10] based on pebbling of graphs, to work for *Res*(k). In particular, we show a combinatorial lemma (Lemma 11) about pebbling of graphs which might be interesting in its own right and could also be applied to other areas.

Concerning automatizability, our result and the algorithm from [3] show the quasi-polynomial automatizability for a hierarchy of almost exponentially separated proof systems. Moreover, since the generalized pebbling contradictions have resolution refutations using constant length clauses, it says that there are examples for which the algorithm of [11] can be exponentially faster in finding resolution proofs than the algorithm of [3] in finding treelike *Res*(k) proofs.

Proof size is not the only complexity measure for proof systems. Inspired by a work of Kleine Büning [20], Esteban and Torán in [16] introduced the concept of space complexity for resolution refutations. Alekhnovich et al. in [1] gave an equivalent formulation of resolution space and generalized it to other proof systems. Space lower bounds for resolution are known for several important classes of contradictions [16,1,9]. Recently, Atserias and Dalmau in [5] gave a game-theoretic combinatorial characterization of resolution width and proved that resolution space is always lower bounded by width, that allows them to obtain all previously known space lower bounds for resolution.

In this work, we introduce the concept of dynamical satisfiability which allows us to prove in an unified and generalized way space lower bounds not only for resolution but also for *Res*(k). Using dynamical satisfiability we immediately get all known space lower bounds for resolution and extend them to *Res*(k). Dynamical satisfiability and the game-theoretic characterization of width given in [5] are related and under certain aspects quite similar. It is not difficult to see that dynamical satisfiability is equivalent to a particular case of the extended existential pebble game defined in [5]. Although clearly less general than the existential pebble game of [5], dynamical satisfiability is simpler to describe because it is defined only in terms of sets of partial assignments instead that by a two-player game on structures. Informally speaking, it seems to capture the part of the characterization of [5] essential to get space lower bounds, and in fact it was obtained extrapolating the common structure from the proofs of the known space lower bounds given in [16,1,9]. Moreover, recently dynamical satisfiability was also used in [18] to get size lower bounds for a restricted form of the treelike Gentzen calculus.

Some relationships between size and space for resolution refutations are already known [16,1,8]. Here we prove a size-space relationship for treelike *Res*(k) which allows us to translate the previous size lower bounds to almost optimal space lower bounds. Moreover, we obtain that treelike *Res*(k) also forms a strict hierarchy with respect to space.

2. Preliminary definitions

A *literal* l is a variable or its negation. We denote by $\neg l$ the opposite literal. A k-*term* is a conjunction of up to $k \geqslant 1$ literals. A k-*clause* is an unbounded disjunction of k-terms. A set of k-clauses or *configuration* means the conjunction of the k-clauses contained in it. We use calligraphic letter to denote configurations and $|\mathcal{F}|$ denote the number of k-clauses in $\mathcal{F}$. Given a formula $\mathcal{F}$, we denote by $\mathcal{F} + Ax(\mathcal{F})$, the formula obtained from $\mathcal{F}$ by adding the tautological clauses $x \vee \neg x$, for any variable x appearing in $\mathcal{F}$.

Assignments (possibly total) to the variables of a formula or of a set of k-clauses are usually denoted by α. The size $|\alpha|$ of an assignment α, is the number of different variables to which α gives a truth value. We call an assignment β a *sub-assignment* of α, denoted by $\beta \sqsubseteq \alpha$, if any variable that is assigned by β is assigned by α to the same value.

Res(k) is a refutation system for CNF formulas, introduced in [22]. It is defined by the following rules: (i) weakening, (ii) $\wedge$-introduction, and (iii) k-cut.

$$(i)\ \frac{A}{A \vee \bigwedge_{l \in L} l}, \quad (ii)\ \frac{A \vee \bigwedge_{l \in L} l \quad B \vee \bigwedge_{l \in K} l}{A \vee B \vee \bigwedge_{l \in L \cup K} l},$$
$$(iii)\ \frac{A \vee \bigwedge_{l \in L} l \qquad B \vee \bigvee_{l \in L} \neg l}{A \vee B},$$

where A and B are k-clauses, and L, K are sets of literals such that $|L \cup K| \leqslant k$. Notice that *Res*(1) is resolution with a weakening rule.

A *Res*(k) *refutation* of a *CNF* formula $\mathcal{F}$ is a list of k-clauses $C_1, \ldots, C_n$ such that C_n is the empty clause, denoted by λ, and for all $i \in [n]$, C_i is either a clause in $\mathcal{F}$ or is obtained from previous k-clauses using the *Res*(k) rules. A refutation can also be viewed as a directed acyclic graph, *dag* for short. An initial clause will have no incoming edges and the node λ will have no outgoing edges. If the graph is a tree we will have a treelike refutation. Treelike *Res*(k) is restricted to treelike refutations.

The *size* of a *Res*(k) refutation is the number of k-clauses in the refutation. Given a refutation P we use $|P|$ to denote the size of P. Sometimes we use $Size_k(\mathcal{F})$ (resp. $Size_k^*(\mathcal{F})$) to denote the minimal size of a refutation of $\mathcal{F}$ in *Res*(k) (resp. in treelike *Res*(k)).

We consider a well-known *pebbling game* on dags. In a dag a node is called *source* if it has no predecessor, and *target* if it has no successor. The aim of the pebbling game is to put a pebble on a target node of the dag using the following rules:

1. a pebble can be put on any source node;
2. a pebble can be taken away from any node;
3. a pebble can be put on any node, provided all its predecessors are pebbled.

The *pebbling number* of a dag G, denoted by $pn(G)$, is the minimal number of pebbles needed to pebble a target node in G, following the rules of the game. Using the pebbling game, in [16] the space of a resolution refutation is defined as the minimal number of pebbles needed to pebble its underlying graph. Obviously this definition can be extended to $Res(k)$ refutations. The space of a $Res(k)$ refutation is the minimum number of pebbles needed to pebble its underlying graph.

Using the equivalent formulation for space from [1], we can view a $Res(k)$ refutation of a formula $\mathcal{F}$ as a set of configurations $\mathcal{C}_0,\dots,\mathcal{C}_s$ such that $\mathcal{C}_0=\emptyset$, $\mathcal{C}_s$ is the empty k-clause and each $\mathcal{C}_t$ for $t = 1,\dots,s$ is obtained from $\mathcal{C}_{t-1}$ by one of the following rules:

1. *Axiom download*: $\mathcal{C}_t := \mathcal{C}_{t-1} \cup \{C\}$ for some clause $C \in \mathcal{F}$.
2. *Memory erasing*: $\mathcal{C}_t := \mathcal{C}_{t-1} - \{C\}$ for some $C \in \mathcal{C}_{t-1}$.
3. *Inference adding*: $\mathcal{C}_t := \mathcal{C}_{t-1} \cup \{C\}$, for some C obtained by one of the rules of $Res(k)$ applied to clauses in $\mathcal{C}_{t-1}$.

Given a refutation P as a set of configurations, the *space* of P is the maximal size of a configuration in P. The *space* of refuting an unsatisfiable formula $\mathcal{F}$ in $Res(k)$ (resp. in treelike $Res(k)$), denoted by $Space_k(\mathcal{F})$ (resp. $Space^*_k(\mathcal{F})$), is the minimal space of a $Res(k)$ refutation (resp. treelike refutation) of $\mathcal{F}$.

3. The pebbling contradictions

Ben-Sasson et al. considered in [10] pebbling contradictions associated to the pebbling game on dags G. They proved that treelike resolution refutations of these formulas require exponential size in $pn(G)$, which gives a $2^{\Omega(n/\log n)}$ lower bound using a family of graphs from [25] with pebbling number $\Omega(n/\log n)$.

We prove that these formulas have $O(n)$ size refutations in treelike $Res(2)$ (Theorem 2). Therefore they give an almost exponential separation between treelike resolution and treelike $Res(2)$. In this section, we define a generalization of the pebbling contradictions to extend the separation to successive levels of treelike $Res(k)$.

3.1. The basic pebbling contradictions

For any node w of a given dag $G=(V,E)$, let $x(w)$ mean that node w can be pebbled. The pebbling came is described using a Horn formula that for any node w contains the clause $\neg x(v_1) \vee \cdots \vee \neg x(v_k) \vee x(w)$ where $v_1,\dots,v_k$ $(k \geqslant 0)$ are all the predecessors of w. If w is a source, the clause is just $x(w)$ and we call it a *source clause*, otherwise it is called a *pebbling clause*. In order to obtain a contradiction we add for each target node $t \in V$ the *target clause* $\neg x(t)$. We denote this contradiction by Peb_G. For our purpose it actually suffices to consider dags G where every non-source node has in-degree 2, as it is the case for the family of graphs from [25] which we will use in Corollary 14. For such a graph G any pebbling clause in G is of the form $\neg x(u) \vee \neg x(v) \vee x(w)$ where u and v are the parents of w. Since Peb_G is a Horn formula it has a very simple treelike resolution refutation.

Lemma 1. *Let $\mathcal{F}$ be an unsatisfiable Horn formula. Then there is a treelike resolution refutation of $\mathcal{F}$ that in any step involves an initial clause from $\mathcal{F}$ and, moreover, uses any initial clause at most once.*

Proof. We will show how to construct the treelike refutation for $\mathcal{F}$. It is well known [20] that the following method can be used to decide the unsatisfiability of a Horn formula $\mathcal{F}$:

> Let $M_0 = \emptyset$. The set M_{d+1} is obtained from M_d by adding some atom $x \notin M_d$ from $\mathcal{F}$ such that there is a clause A_d in F where $A_d = \neg x_1 \vee \cdots \vee \neg x_l \vee x$ with $x_1, \ldots, x_l \in M_d$ and $l \geqslant 0$. If no more atoms can be added to M_d according to the above rule, then $\mathcal{F}$ is unsatisfiable iff there is a clause $\neg x_1 \vee \cdots \vee \neg x_l$ in $\mathcal{F}$ such that $x_1, \ldots, x_l \in M_d$.

Notice that in the above construction the clauses A_d and $A_{d'}$ are distinct for $d' > d$ since each variable is added at most once to $M_{d'}$.

Actually, when $\mathcal{F}$ is unsatisfiable, the above method can be understood as a resolution refutation of $\mathcal{F}$ where in each resolution step a clause that consists of one variable is involved. The treelike form of this refutation may however be of exponential size. Now using the sets M_i from the above construction (in decreasing order of i) the following algorithm produces a treelike resolution refutation of $\mathcal{F}$ that in each step involves an initial clause.

> Start with the clause $C_1 = \neg x_1 \vee \cdots \vee \neg x_l$ such that $x_1, \ldots, x_l \in M_{d_1}$, where d_1 is the final index. Now we will subsequently derive clauses C_i and indices d_i for $i = 1, 2, \ldots$ such that C_i is a disjunction of some negated variables from M_{d_i}. Obtain d_i and C_i from d_{i-1} and C_{i-1} as follows: let $d_i < d_{i-1}$ be the minimal index such that all variables in C_{i-1} are fully contained in M_{d_i+1}. This means that in order to construct M_{d_i+1} from M_{d_i} a variable x from C_{i-1} had been added such that there is a clause in $\mathcal{F}$ of the form $A_{d_i} = \neg x_1 \vee \cdots \vee \neg x_l \vee x$. C_i is obtained from C_{i-1} by resolving with A_{d_i} on x. Notice that all variables in C_i are contained in M_{d_i}. Continue like this until C_i is the empty clause.

Since $M_1 = \emptyset$ and $d_{i+1} < d_i$ the empty clause will be derived in at most d_1 steps. Moreover, since all clauses A_{d_i} are different, each input clause is used at most once. □

3.2. Generalized pebbling contradictions

The contradiction $Peb_{G,l,k}$ $(k, l \geqslant 1)$ is obtained from Peb_G by introducing $k \cdot l$ variables $x(v)_{i,j}$, $i \in [l]$, $j \in [k]$ for each propositional variable $x(v)$ in Peb_G. Each variable $x(v)$ is replaced by

$$\bigwedge_{i \in [l]} \bigvee_{j \in [k]} x(v)_{i,j}.$$

The resulting formula is then transformed into CNF using de Morgan's laws, and distributivity. Hence, each source clause $x(s)$ in Peb_G will correspond to the $Peb_{G,l,k}$-*source clauses*

$$x(s)_{i,1} \vee \cdots \vee x(s)_{i,k}$$

for $i \in [l]$. Each target clause $\neg x(t)$ in Peb_G will correspond to the $Peb_{G,l,k}$*-target clauses*

$$\neg x(t)_{1,j_1} \vee \cdots \vee \neg x(t)_{l,j_l}$$

for $j_1,\ldots,j_l \in [k]$. And each pebbling clause $\neg x(u) \vee \neg x(v) \vee x(w)$ in Peb_G will correspond to the $Peb_{G,l,k}$*-pebbling clauses*

$$\neg x(u)_{1,j_1} \vee \cdots \vee \neg x(u)_{l,j_l} \vee \neg x(v)_{1,m_1} \vee \cdots \vee \neg x(v)_{l,m_l} \vee x(w)_{i,1} \vee \cdots \vee x(w)_{i,k}$$

for $j_1,\ldots,j_l,m_1,\ldots,m_l \in [k]$, $i \in [l]$. Clearly, $Peb_{G,l,k}$ is a contradiction since Peb_G is. Moreover, $Peb_{G,l,k}$ has a small resolution refutation in treelike $Res(k)$.

Theorem 2. *There is a treelike $Res(k)$ refutation of $Peb_{G,l,k}$ that involves less than twice the number of clauses in $Peb_{G,l,k}$.*

Proof. From $Peb_{G,l,k}$, using $\wedge$-introductions and each clause only once, we derive in a treelike fashion the formula $Peb_{G,l,1}$, where each literal $x(v)_i$ is substituted by the clause $x(v)_{i,1} \vee \cdots \vee x(v)_{i,k}$, and each literal $\neg x(v)_i$ by the k-term $\neg x(v)_{i,1} \wedge \cdots \wedge \neg x(v)_{i,k}$. Then we apply the refutation used to prove Lemma 1 on this formula. Observe that this refutation corresponds to a treelike $Res(k)$ refutation, since a resolution step involving $x(v)_i$ and $\neg x(v)_i$ corresponds now to a cut in $Res(k)$. Combining both derivations we obtain a treelike $Res(k)$ refutation of $Peb_{G,l,k}$ that uses each input clause at most once. Since we did not use weakenings, every inner node in the refutation tree has two children. Now in such a binary tree the number of inner nodes is one less than the number of leaves. Hence the stated bound follows.

Let us finally show how to derive the substituted $Peb_{G,l,1}$ formula. First observe that each $Peb_{G,l,k}$ source clause $x(s)_{i,1} \vee \cdots \vee x(s)_{i,k}$ substitutes the source clause $x(s)_i$ in $Peb_{G,l,1}$ for $i \in [l]$. From the target clauses

$$\neg x(t)_{1,j_1} \vee \cdots \vee \neg x(t)_{l,j_l}$$

of $Peb_{G,l,k}$ with $j_1,\ldots,j_l \in [k]$, we derive by solely using the $\wedge$-introduction rule, and using each of these clauses once, the k-clause

$$(\neg x(t)_{1,1} \wedge \cdots \wedge \neg x(t)_{1,k}) \vee \cdots \vee (\neg x(t)_{l,1} \wedge \cdots \wedge \neg x(t)_{l,k}),$$

which substitutes the target clause $\neg x(t)_1 \vee \cdots \vee \neg x(t)_l$ of $Peb_{G,l,1}$. In a similar way we derive from the pebbling clauses

$$\neg x(u)_{1,j_1} \vee \cdots \vee \neg x(u)_{l,j_l} \vee \neg x(v)_{1,m_1} \vee \cdots \vee \neg x(v)_{l,m_l} \vee x(w)_{i,1} \vee \cdots \vee x(w)_{i,k}$$

for $j_1,\ldots,j_l,m_1,\ldots,m_l \in [k]$ the following k-clause

$$\bigwedge_{j\in[k]} \neg x(u)_{1,j} \vee \cdots \vee \bigwedge_{j\in[k]} \neg x(u)_{l,j} \vee \bigwedge_{m\in[k]} \neg x(v)_{1,m} \vee \cdots \vee \bigwedge_{m\in[k]} \neg x(v)_{l,m} \vee \bigvee_{j\in[k]} x(w)_{i,j}$$

which substitutes the $Peb_{G,l,1}$ pebbling clause

$$\neg x(u)_1 \vee \cdots \vee \neg x(u)_l \vee \neg x(v)_1 \vee \cdots \vee \neg x(v)_l \vee x(w)_i. \qquad \square$$

Note that $Peb_{G,1,2}$ are the pebbling contradictions in [10]. By Theorem 2 and the lower bound in [10] we get an almost exponential separation between treelike resolution and treelike $Res(2)$.

Corollary 3. *Treelike $Res(2)$ dominates treelike resolution.*

4. Lower bounds for generalized pebbling contradictions

In this section, we show that any treelike $Res(k)$ refutation of $Peb_{G,l,k+1}$ with $l \geqslant k$ is of size at least $2^{(pn(G)-3)/k}$. To obtain the lower bound we generalize a game introduced in [26] to prove lower bounds for treelike resolution. It is a 2-player game where the two players build a partial assignment, one variable per round. Here, we extend the rules of this game in order to allow the use of up to k variables at each round.

4.1. A game on contradictions

The game $G_k(\mathcal{F})$ is a 2-player game played on the unsatisfiable CNF formula $\mathcal{F}$. It is played by constructing a partial assignment to the variables in $\mathcal{F}$. The game starts with the empty assignment. The partial assignment is built by both players and it is known to both of them. When a variable is set, it cannot be changed later. The aim of the first player, the Prover, is to falsify a clause in $\mathcal{F}$, which ends the game. Clearly the Prover always will win the game. So the aim of the second player, the Delayer, is to delay as much as possible the end of the game. The delay will be measured in terms of the number of points the Delayer can score. In each round the Prover asks for a k-term C. The Delayer assigns values to some (possibly all) yet unassigned variables in C. If the constructed assignment either falsifies or satisfies C, then the round is over. Otherwise, the Prover assigns values to all of the remaining unassigned variables in C and the Delayer scores one point.

We show that each treelike $Res(k)$ refutation yields a strategy for the Prover in which the Delayer scores a number of points at most logarithmic in the size of the refutation. Actually already a special type of decision tree (called k-decision tree, here) for $\mathcal{F}$ can be used by the Prover to obtain a good strategy.

It is well known, see [10], that a treelike resolution refutation of a CNF formula $\mathcal{F}$ can be transformed into a binary decision tree T of the same size such that for any assignment to $\mathcal{F}$, T yields a falsified clause of $\mathcal{F}$. In T each inner node is labeled by a variable and the decision how to continue the path at an inner node is determined by the assignment to its variable. So any total assignment will lead to a leaf node of T associated with a clause that is falsified by that assignment. Here, we consider binary decision trees where each inner node is labeled by a k-term. The decision how to continue a path at an inner node is determined by the value of its k-term under the assignment. We call such a tree a k-decision tree for $\mathcal{F}$. Similar to the well known result for $k=1$ one obtains the following result for any $k \geqslant 1$, where by the size of a k-decision tree we mean the number of its nodes.

Proposition 4. *If $\mathcal{F}$ has a treelike $Res(k)$ refutation of size S, then $\mathcal{F}$ has k-decision tree with $\leqslant S$ nodes.*

Proof. We will describe a recursive procedure, called DT that for any refutation tree T for $\mathcal{F}$ yields a decision tree $DT(T)$ for $\mathcal{F}$ that does not have more nodes than T.

For subtree T' of T, we obtain $DT(T')$ as follows: If T' consists of one leaf node (labeled by an initial clause) then $DT(T') = T'$. Otherwise let D denote the clause labeling the root of T' and consider three cases:

1. If $D = A \vee B$ is obtained by a k-cut from the clauses $A \vee \bigwedge_{l \in L} l$ and $B \vee \bigvee_{l \in L} \neg l$ labeling the roots of the two direct subtrees T_1 and T_2 (respectively) of T', then the root of $DT(T')$ is labeled by the k-term $C = \bigwedge_{l \in L} l$ and $DT(T')$ consists of the two direct subtrees $DT(T_1)$, $DT(T_2)$, such that any assignment satisfying (falsifying) C leads to $DT(T_2)$ (resp., $DT(T_1)$).
2. If D is obtained by $\wedge$-introduction, involving the k-terms C_1, C_2 such that $C_1 \wedge C_2$ is in D, then label the root of $DT(T')$ by C_1 and branch to $DT(T_1)$ (resp. $DT(T_2)$) if C_1 is falsified (satisfied).
3. If D is obtained by weakening and T'' is the direct subtree of T' then let $DT(T') = DT(T'')$.

The correctness of the transformation is proved by observing that the following invariant is maintained: any complete assignment α that leads to the root of $DT(T')$ through $DT(T)$, falsifies the clause D labeling the root of T'. □

Observe that the previous proposition also holds for formulas with tautological clauses. In fact any k-decision tree for $\mathcal{F} + Ax(\mathcal{F})$ can be pruned to a k-decision tree for $\mathcal{F}$. Since an axiom $x \vee \neg x$ cannot be falsified, it can be cut from the leaves.

Corollary 5. *If $\mathcal{F} + Ax(\mathcal{F})$ has a treelike $Res(k)$ refutation of size S, then $\mathcal{F}$ has k-decision tree with $\leqslant S$ nodes.*

For $k=1$ also the reverse inequality holds, i.e. a 1-decision tree for $\mathcal{F}$ of size S can be transformed into a treelike resolution refutation for $\mathcal{F}$ of size $\leqslant S$, see [10]. But for $k \geqslant 3$ there are formulas $\mathcal{F}$ with k-decision trees of size $O(n)$ such that any treelike $Res(k)$ refutation for $\mathcal{F}$ has size $2^{\Omega(n)}$. Notice that any contradiction $\mathcal{F}$ in k-CNF has a trivial linear k-decision tree which for each clause C has an inner node labeled with the negation of C. On the other hand, there are contradictions in 3-CNF with $O(n)$

clauses that require a resolution refutation of size $2^{\Omega(n)}$, see [33]. Since, as we will see in Theorem 16, treelike $Res(k)$ is simulated by resolution, $2^{\Omega(n)}$ is also a lower bound for treelike $Res(k)$. We obtain the following.

Corollary 6. *There is a family of contradictions $\mathcal{F}_n$ with 3-decision trees of size $O(n)$ that require refutations of size $2^{\Omega(n)}$ in treelike $Res(k)$.*

The following proposition provides a useful relation between the size of a k-decision tree for a contradiction $\mathcal{F}$ and the number of points the Delayer can score in $G_k(\mathcal{F})$.

Proposition 7. *If $\mathcal{F}$ has a k-decision tree of size S, then the Prover has a strategy for $G_k(\mathcal{F})$ such that the Delayer scores at most $\lfloor \log S \rfloor$ points.*

Proof. Let T be a k-decision tree of size S. The Prover's strategy will maintain the following invariant: if the Delayer has scored p points, then the currently constructed partial assignment α will lead to a node in T such that the subtree T_α rooted at this node is of size at most $S/2^p$.

At the beginning the invariant holds since T is by assumption of size S. Now assume that the partial assignment α constructed so far is such that T_α is of size at most $S/2^p$. Let C be the k-term labeling the root of T_α. In the next round the Prover asks for C. Now α is extended in this round to an assignment α' that will assign a value to C. Hence, α' will lead to a subtree $T_{\alpha'}$ of T_α. If the Delayer scores a point the Prover is able to guarantee that $T_{\alpha'}$ is of at most half the size of T_α: since the assignment that had been chosen by the Delayer left C unassigned, the Prover is able to choose α' such that it leads into the smaller one of the both direct subtrees of T_α. Hence $T_{\alpha'}$ has a size less than half of the size of T_α, in this case. This shows that the invariant can be maintained. □

As a consequence we obtain the following corollary that we will use to prove lower bounds for treelike $Res(k)$ refutations.

Corollary 8. *If the Delayer in $G_k(\mathcal{F})$ has a strategy that yields at least p points, then any k-decision tree for $\mathcal{F}$, and any treelike $Res(k)$ refutation for $\mathcal{F}$ and $\mathcal{F} + Ax(\mathcal{F})$ as well, is of size at least 2^p.*

Notice however that Corollary 8 will not provide lower bounds for treelike $Res(k)$ refutations of formulas in k-CNF due to the trivial k-decision tree that exists for any formula in k-CNF.

4.2. The Delayer's strategy

This subsection is devoted to the proof of Theorem 9. It is the key step in the proof of a lower bound for treelike $Res(k)$ refutations of $Peb_{G,k,k+1}$.

Theorem 9. *If G is a dag where any non-source node has in-degree 2, and $l \geqslant k \geqslant 1$, then the Delayer can score at least* $(pn(G)-3)/k$ points in the game $G_k(Peb_{G,l,k+1})$.

Let us in the following fix a dag $G = (V,E)$ where each non-source node has in-degree 2, fix further constants k, l with $l \geqslant k \geqslant 1$. We will describe a strategy for the Delayer that yields at least $(pn(G)-3)/k$ points in the game $\mathrm{G}_k(Peb_{G,l,k+1})$.

For sets $S,T \subseteq V$ let us denote by $pn(S,T)$ the pebbling number of the graph $G' = (V,E')$ where $E' = E \setminus ((V \times S) \cup (T \times V))$. In other words, we obtain G' from G by additionally making each node in S a source node, and each node in T a target node.

To describe the strategy of the Delayer we will need Lemma 11. It is a generalization of the following lemma from [10].

Lemma 10 (Ben-Sasson [10]). *For any node v in G and any subsets $S,T \subseteq V$*

$$pn(S,T) \leqslant \max\{pn(S,T \cup \{v\}), pn(S \cup \{v\},T)+1\}.$$

Lemma 11. *For any disjoint sets $W,S,T \subseteq V$, there exists a partition X,Y of W ($X \cup Y = W$ and $X \cap Y = \emptyset$) such that*

$$pn(S,T) \leqslant |X| + pn(S \cup X, T \cup Y).$$

Proof. We proceed by induction on $|W|$. If $|W| = 1$, the claim follows by Lemma 10. For the inductive step consider a partition of W into two nonempty sets W' and W''. By applying the inductive hypothesis to W', there is a partition X', Y' of W' such that $pn(S,T) \leqslant |X'| + pn(S \cup X', T \cup Y')$.

Let now $S' = S \cup X'$ and $T' = T \cup Y'$. By the inductive hypothesis applied to W'', there is a partition X'', Y'' of W'' such that $pn(S'T') \leqslant |X''| + pn(S' \cup X'', T' \cup Y'')$. Define $X = X' \cup X''$ and $Y = Y' \cup Y''$. All together we have

$$\begin{aligned} pn(S,T) &\leqslant |X'| + pn(S \cup X', T \cup Y') \\ &\leqslant |X'| + |X''| + pn(S \cup X' \cup X'', T \cup Y' \cup Y'') \\ &= |X| + pn(S \cup X, T \cup Y). \qquad \square \end{aligned}$$

Now we are ready to describe the strategy of the Delayer for the game $\mathrm{G}_k(Peb_{G,l,k+1})$. She keeps two sets of source and target nodes that she (possibly) modifies at each round. At the beginning $S_0 = T_0 = \emptyset$. Let S_r and T_r be the sets built after round r. Assume that at round $r+1$ the Prover asks for a term C of at most k literals. Let us denote by W the set of nodes associated with the variables in C. W is divided into the four sets $W \cap S_r$, $W \cap T_r$, $W_=$, and $W_> = W \setminus (S_r \cup T_r \cup W_=)$, where $W_= \subseteq W \setminus (S_r \cup T_r)$ is a maximal set with the property that $pn(S_r, T_r \cup W_=) = pn(S_r, T_r)$. Now the Delayer assigns 1 to every unassigned variable in C that is associated with a node in $W \cap S_r$, and she assigns 0 to every unassigned variable in C associated with a node in $(W \cap T_r) \cup W_=$. If now C is either satisfied or falsified by the constructed assignment, the round is over, and the Delayer sets $T_{r+1} = T_r \cup W_=$, and $S_{r+1} = S_r$. In this case the pebbling number remains the same, $pn(S_r, T_r) = pn(S_{r+1}, T_{r+1})$; otherwise the Prover assigns a value to the remaining unassigned variables in C, the Delayer scores one point and defines S_{r+1} and T_{r+1} as follows: by Lemma 11, she chooses a partition X, Y of $W_>$ s.t.

$$pn(S_r, T_r \cup W_=) \leqslant pn(S_r \cup X, T_r \cup W_= \cup Y) + |X|.$$

Now $S_{r+1} = S_r \cup X$, and $T_{r+1} = T_r \cup W_= \cup Y$. In this case the pebbling number decreases by at most $|X| \leqslant k$.

Lemma 12. *Assuming that the Delayer follows this strategy, she maintains the following invariants*:

(I1) *If a variable $x(v)_{i,j}$ is assigned a value in round r or before then the associated node v is in $S_r \cup T_r$.*

(I2) *If $v \in S_r$ then there are at most k associated variables $x(v)_{i,j}$ that are assigned to* 0.

(I3) *If $v \in T_r$ then there are at most $k-1$ associated variables $x(v)_{i,j}$ that are assigned to* 1.

(I4) $pn(G) \leqslant pn(S_r, T_r) + |S_r|$.

(I5) *At the end of round r the Delayer achieved at least $\lceil |S_r|/k \rceil$ points.*

Proof. (I1) is obvious since if $x(v)_{i,j}$ is assigned a value in some round p then $v \in S_p \cup T_p$, moreover $S_p \subseteq S_{p'}$, and $T_p \subseteq T_{p'}$ for $p \leqslant p'$.

To see that (I2) holds, notice that for any node v the Prover is allowed to assign at most k of its associated variables, and if $v \in S_r$ then Delayer will assign 1 to any associated variable in later rounds, hence the Prover can assign at most k associated variables to 0. (I3) follows by a similar argument. Observe that if the Prover was allowed to assign a variable in some round $r+1$ then $W_>$ was not empty in that round. Moreover when partitioning $W_>$ into X, Y, the set X has to be non-empty, since otherwise $pn(S_r, T_r \cup W_=) = pn(S_r, T_r \cup W_= \cup W_>)$ and therefore $W_>$ could have been added to $W_=$ which would contradict the maximality of $W_=$. Hence in this round, there are at most $k-1$ variables associated to a node added to T_{r+1}. Since in later rounds the Delayer will assign 0 to each variable associated to a node $v \in T_{r+1}$ the Prover can assign 1 to at most $k-1$ variables associated to v.

To see invariant (I4), first notice that (I4) holds for $r = 0$, since $pn(G) = pn(S_0, T_0)$. Now assume that (I4) holds at round $r \geqslant 0$. Then at round $r+1$ either $S_{r+1} = S_r$ and $pn(S_{r+1}, T_{r+1}) = pn(S_r, T_r)$ (in which case it is obvious that (I4) holds at round $r+1$), or there is a set X with $X \cap S_r = \emptyset$, $S_{r+1} = S_r \cup X$ and $pn(S_r, T_r) = pn(S_r, T_r \cup W_=)$ $\leqslant pn(S_{r+1}, T_{r+1}) + |X|$. Hence

$$\begin{aligned} pn(G) &\leqslant pn(S_r, T_r) + |S_r| \\ &\leqslant pn(S_{r+1}, T_{r+1}) + |S_r| + |X| = pn(S_{r+1}, T_{r+1}) + |S_{r+1}|, \end{aligned}$$

which shows that (I4) holds at round $r+1$.

(I5) follows since in case the Delayer scores no point in round $r+1$ then $S_{r+1} = S_r$, and otherwise if she scores a point, $|S_{r+1}| \leqslant |S_r| + k$. $\square$

Now observe that at the end of the game $G_k(Peb_{G,l,k+1})$, say at round e, the pebbling number is considerably reduced. Namely we have:

Lemma 13. *Let e be the last round, then $pn(S_e, T_e) \leqslant 3$.*

Proof. Let $G' = (V, E')$ where $E' = E \setminus ((V \times S_e) \cup (T_e \times V))$. Remember that $pn(S_e, T_e)$ was defined to be the pebbling number of G'. Notice also that any source node in G is a source in G', and that any target node in G is a target in G'.

The game ends when the constructed partial assignment falsifies a clause of $Peb_{G,l,k+1}$. If a source clause $x(s)_{i,1} \vee \cdots \vee x(s)_{i,k+1}$ associated to a source s in G is falsified then $s \in T_e$ due to (I1) and (I2). Hence s is both, a source in G' and a target node in G', which shows that one pebble suffices for a pebbling of G'. Similarly, when a target clause $\neg x(t)_{1,j_1} \vee \cdots \vee \neg x(t)_{l,j_l}$ is falsified then $t \in S_e$ by (I1) and (I3) (since $l \geqslant k$) and the pebbling number of G' is one. Finally assume that a pebbling clause associated to a node w with predecessors u and v is falsified. Similar to the previous considerations we obtain that $u, v \in S_e$, and $w \in T_e$. Hence, for a pebbling of G' it suffices to use three pebbles. □

Due to invariant (I4) this implies that $|S_e| \geqslant pn(G) - 3$. Moreover, due to (I5), the Delayer scores at least $\lceil |S_e|/k \rceil$ points. This concludes the proof of Theorem 9.

4.3. Almost exponential separations for treelike $Res(k)$

It is shown in [25] that there is an infinite family of graphs G_n, where each non-source node in G_n has in-degree 2, such that $pn(G_n) = \Omega(n/\log n)$, and n is the number of nodes in G. Moreover these graphs are uniform, where we call a family $\mathcal{F}_n$ *uniform* if there is an algorithm that on input n produces $\mathcal{F}_n$ in time polynomial in n.

Combining Theorem 9 with Corollary 8 shows that for such a graph G_n, any treelike $Res(k)$ refutation for $Peb_{G_n,k,k+1}$ has size $2^{\Omega(n/\log n)}$. On the other hand, $Peb_{G_n,k,k+1}$ consists of at most $O(n)$ clauses, hence by Theorem 2 there is a treelike $Res(k+1)$ refutation of $Peb_{G,k,k+1}$ of size $O(n)$. This yields an almost exponential separation between treelike $Res(k)$ and $Res(k+1)$.

Corollary 14. *Let $k > 0$. There is a uniform family of formulas $\mathcal{F}_n = Peb_{G_n,k,k+1}$ with a treelike $Res(k+1)$ refutation of size $O(n)$ such that any treelike $Res(k)$ refutation of $\mathcal{F}_n$ and $\mathcal{F}_n + Ax(\mathcal{F}_n)$ as well, has size $2^{\Omega(n/\log n)}$.*

Corollary 15. *Treelike $Res(k+1)$ dominates treelike $Res(k)$.*

Moreover, resolution simulates treelike $Res(k)$. The following theorem follows by adapting to $Res(k)$ the proof sketch in [21] that treelike depth $d+1$ LK is simulated by depth d LK, see [21] for a definition of these notions, and see e.g. [24] for a more detailed proof of the simulation from [21].

Theorem 16. *If $\mathcal{F}$ is a formula in CNF that has a treelike $Res(k)$ refutation of size S then $\mathcal{F}$ has a resolution refutation of size $2S$.*

Proof. For a $Res(k)$ derivation P (treelike or daglike) let $s(P)$ denote the number of occurrences of k-clauses in P that are not obtained by the weakening rule, and let $a(P)$ denote the number of occurrences of k-clauses in P that are obtained by $\wedge$-introduction. Below we will prove the following statement by induction on $a(T)$: for all formulas $\mathcal{F}$ in CNF, and for all clauses C, if T is a treelike $Res(k)$ derivation of C from $\mathcal{F}$ then there is a resolution derivation P of C from $\mathcal{F}$ with $s(P)=s(T)+a(T)$. Since weakenings can be removed in resolution refutations the theorem follows.

If $a(T)=0$ then T is already a resolution derivation. Now assume $a(T)>0$, and consider the last k-cut in T where a k-term $\bigwedge_{l\in L} l$ with $|L|\geqslant 2$ is involved, say

$$\frac{A\vee\bigwedge_{l\in L} l \quad B\vee\bigvee_{l\in L}\neg l}{A\vee B}.$$

Since this was a last cut, $A\vee B$, and $B\vee\bigvee_{l\in L}\neg l$ are clauses. Let T_1, T_2 denote subtrees deriving $A\vee\bigwedge_{l\in L} l$ and $B\vee\bigvee_{l\in L}\neg l$, respectively. Since T_1 must contain some $\wedge$-introduction to produce the term $\bigwedge_{l\in L} l$ we have that $a(T_2)<a(T)$ and we conclude by the inductive hypotheses that there is a resolution derivation P_2 of $B\vee\bigvee_{l\in L}\neg l$ from $\mathcal{F}$ of size $s(P_2)=s(T_2)+a(T_2)$. Consider also the rest of the derivation $T'=T\backslash(T_1\cup T_2)$. T' derives C from $\mathcal{F}\wedge(A\vee B)$. By the inductive hypothesis we obtain a resolution derivation P' of C from $\mathcal{F}\wedge(A\vee B)$ with $s(P')=s(T')+a(T')=s(T)+a(T)-\sum_{i=1,2} s(T_i)+a(T_i)$.

Now we add $B\vee\bigvee_{l\in L}\neg l$ to the initial clauses and show how to transform T_1 to a derivation tree T_1' of $A\vee B$ from $\mathcal{F}\wedge(B\vee\bigvee_{l\in L}\neg l)$ with $s(T_1')=s(T_1)+r$, and $a(T_1')=a(T_1)-r$ for some $r\geqslant 1$. Note that $\bigwedge_{l\in L} l$ can arrive to $A\vee\bigwedge_{l\in L} l$ through several paths, say r. Now, trace in T_1 the occurrence of the term $\bigwedge_{l\in L} l$ towards the leaves until one encounters a k-clause in which this term is introduced by $\wedge$-introduction. Denote these k-clauses by $C_i\vee\bigwedge_{l\in L} l$ for $i=1,\dots,r$, and denote the clauses from which they are derived by $A_i\vee\bigwedge_{l\in L_i} l$ and $B_i\vee\bigwedge_{l\in L_i'} l$ with $L=L_i\cup L_i'$, and $C_i=A_i\vee B_i$. Now replace for $i=1,\dots,r$ the $\wedge$-introduction

$$\frac{A_i\vee\bigwedge_{l\in L_i} l \quad B_i\vee\bigwedge_{l\in L_i'} l}{C_i\vee\bigwedge_{l\in L} l}$$

by two k-cuts (and eventually one weakening)

$$\frac{\dfrac{\dfrac{B\vee\bigvee_{l\in L}\neg l \quad A_i\vee\bigwedge_{l\in L_i} l}{A_i\vee B\vee\bigvee_{l\in L\backslash L_i}\neg l}}{A_i\vee B\vee\bigvee_{l\in L_i'}\neg l} \qquad B_i\vee\bigwedge_{l\in L_i'} l}{C_i\vee B}.$$

Further replace on the path towards the root of T_1 the term $\bigwedge_{l\in L} l$ by B. To obtain the derivation tree T_1' one may again need to add some weakenings on this path.

Applying the inductive hypothesis to T_1' we obtain a resolution derivation P_1 of $A\vee B$ from $\mathcal{F}\wedge(B\vee\bigvee_{l\in L}\neg l)$ with $s(P_1)=s(T_1')+a(T_1')=(s(T_1)+r)+(a(T_1)-r)=s(T_1)+a(T_1)$.

Now combine the resolution derivations P_2, P', and P_1 to obtain the resolution derivation $P = P_2, P_1, P'$ of C from $\mathcal{F}$ with size $s(P) = S(P_2) + s(P_1) + s(P') = s(T) + a(T)$. □

Let us note that Theorem 16 still holds when allowing in $Res(k)$ a $\wedge$-introduction rule that involves up to $j \leqslant k$ premises. See for example [30], or [21] for the unbounded $\wedge$-introduction for LK.

$$\frac{A_1 \vee l_1 \quad \cdots \quad A_j \vee l_j}{A_1 \vee \cdots \vee A_j \vee (l_1 \wedge \cdots \wedge l_j)}.$$

To see this, notice that in the proof of Theorem 16 we just need to introduce one additional cut for each premise of a $\wedge$-introduction.

Moreover, since the increase in size does not depend on k, the simulation holds also for $Res(\infty)$ where we allow unbounded conjunctions, i.e. a refutation in $Res(\infty)$ is defined to be a refutation in $Res(k)$ for some k.

Corollary 17. *If $\mathcal{F}$ has a treelike $Res(\infty)$ refutation of size S then $\mathcal{F}$ has a resolution refutation of size $2S$.*

From Theorem 16 and Corollary 14 we get:

Corollary 18. *Resolution dominates treelike $Res(k)$ for $k \geqslant 1$.*

5. Space complexity in *Res*(*k*)

We consider now the space complexity of $Res(k)$. We show that, as happened for size, treelike $Res(k)$ for constant k, also forms a strict hierarchy for space. Moreover, we extend all known resolution space lower bounds to $Res(k)$.

5.1. *Space separations for the treelike Res(k) hierarchy*

Consider the following definition from [3]. Given a formula $\mathcal{F}$ over variables in X, and a $k \geqslant 1$, define the formula $\mathcal{F} + Ex_k(\mathcal{F})$ as follows: for any set L of at most k literals over X introduce a new variable z_L meaning $\bigwedge_{l \in L} l$. Now $Ex_k(\mathcal{F})$ contains for each of these variables z_L the clauses $\neg z_L \vee l$, for $l \in L$, and $z_L \vee \bigvee_{l \in L} \neg l$. Then $\mathcal{F} + Ex_k(\mathcal{F})$ is obtained by adding these clauses to $\mathcal{F}$.

Atserias and Bonet proved in [3], that resolution refutations (treelike resolution refutations, resp.) of $\mathcal{F} + Ex_k(\mathcal{F})$ can be easily converted into $Res(k)$ refutations (treelike $Res(k)$ refutations, resp.) of $\mathcal{F}$ and vice-versa incrementing the size by only factor k.

Lemma 19 (Atserias and Bonet [3]). *For any formula $\mathcal{F}$ in CNF and $k \geqslant 1$, if $\mathcal{F}$ has a (treelike) Res(k) refutation of size S, then $\mathcal{F} + Ex_k(\mathcal{F})$ has a resolution refutation (treelike resolution refutation, resp.) of size* $O(kS)$.

Lemma 20 (Atserias and Bonet [3]). *For any formula $\mathcal{F}$ in CNF and $k \geqslant 1$, if $\mathcal{F} + Ex_k(\mathcal{F})$ has a (treelike) resolution refutation of size S, then $\mathcal{F} + Ax(\mathcal{F})$ has a Res(k) refutation (treelike Res(k) refutation, resp.) of size* $O(kS)$.

Using the same simulations employed in [3] to prove Lemmas 19 and 20, it is easy to see that similar relations hold for the space as well. We sketch the proofs of the next two lemmas.

Lemma 21. *For any $\mathcal{F}$ in CNF and $k \geqslant 1$, if $\mathcal{F}$ has a (treelike) Res(k) refutation in space S, then $\mathcal{F} + Ex_k(\mathcal{F})$ has a resolution refutation (treelike resolution refutation, resp.) in space $S + 2$.*

Proof sketch. Each step in the *Res*(k) proof of $\mathcal{F}$ involving a conjunction of the form $l_1 \wedge \cdots \wedge l_r$, $r \leqslant k$, is simulated in resolution using the variable $z_{l_1,\dots,l_r}$ and the corresponding axioms of $Ex_k(\mathcal{F})$. For instance, in the case of the r-Cut there is a straightforward space 3, treelike proofs of $\neg z_{l_1,\dots,l_r}$ from $\neg l_1 \vee \cdots \vee \neg l_r$ and the clauses of $Ex_k(\mathcal{F})$, $\neg z_{l_1,\dots,l_r} \vee l_i$, for $i = 1, \dots, r$. Hence the space of the original proof is incremented by at most 2. □

Lemma 22. *For any $\mathcal{F}$ and $k \geqslant 1$, if $\mathcal{F} + Ex_k(\mathcal{F})$ has a (treelike) resolution refutation in space S, then $\mathcal{F} + Ax(\mathcal{F})$ has a Res(k) refutation (treelike Res(k) refutation, resp.) in space $S + 2$.*

Proof sketch. For each clause of $Ex_k(\mathcal{F})$ including a literal $z_{l_1,\dots,l_r}$, we derive in *Res*(k) a k-clause in which $z_{l_1,\dots,l_r}$ has been replaced by $l_1 \wedge \cdots \wedge l_r$ and $\neg z_{l_1,\dots,l_r}$ by $(\neg l_1 \vee \cdots \vee \neg l_r)$. For instance, there is a straightforward space 3 treelike *Res*(k) proof deriving $\neg l_1 \vee \cdots \neg l_r \vee (l_1 \wedge \cdots \wedge l_r)$, using the tautological axioms $l_i \vee \neg l_i$ and $\wedge$-introductions. The space is incremented by at most 2. □

Esteban and Torán [16] proved the following relation between size and space, for treelike resolution.

Lemma 23 (Esteban and Torán [16]). *If a formula over n variables has a treelike resolution refutation in space S, then it has a treelike resolution refutation of size at most* $\binom{n+S}{S}$.

We extend the previous lemma to treelike *Res*(k), obtaining the following.

Theorem 24. *For any formula $\mathcal{F}$ in CNF over n variables and $k \geqslant 1$, if $Size_k^*(\mathcal{F} + Ax(\mathcal{F})) \geqslant S$, then $Space_k^*(\mathcal{F}) \geqslant \Omega(\log S / \log n)$.*

Proof. Let $Size^*_k(\mathcal{F} + Ax(\mathcal{F})) \geqslant S$ for some formula $\mathcal{F}$ in CNF with n variables. We have that $Size^*_1(\mathcal{F} + Ex_k(\mathcal{F})) \geqslant \Omega(S/k)$ by Lemma 20. Since it is known, see [16], that the number of variables is an upper bound for treelike resolution space, it is easy to see that Lemma 23 in turn implies that $Space^*_1(\mathcal{F}_k + Ex_k(\mathcal{F})) \geqslant \Omega(\log S/\log n)$, which implies the claim by Lemma 21. □

As a corollary of the previous theorem and the size lower bound of Corollary 14, we obtain a space lower bound for $Peb_{G_n,k,k+1}$.

Corollary 25. *Let $k \geqslant 1$. There is a uniform family of formulas $\mathcal{F}_n = Peb_{G_n,k,k+1}$ with* $O(n)$ *variables such that any treelike $Res(k)$ refutation for $\mathcal{F}_n$ needs space $\Omega(n/\log^2 n)$.*

On the other hand, the treelike $Res(k+1)$ refutation of $Peb_{G,k,k+1}$ we constructed in the proof of Theorem 2 needs only constant space, since it essentially consists of a linear tree in which the leaves are replaced by subtrees of constant size.

Theorem 26. *There are* $O(1)$ *space refutations for $Peb_{G,k,k+1}$ in treelike $Res(k+1)$.*

Therefore, the treelike $Res(k)$ space hierarchy is strict.

Corollary 27. *For $k \geqslant 1$ there is a uniform family of formulas $\mathcal{F}_n$ over* $O(n)$ *variables that have constant space refutations in treelike $Res(k+1)$ but require space $\Omega(n/\log^2 n)$ in treelike $Res(k)$.*

5.2. Space lower bounds for Res(k)

A general way to obtain space lower bounds for resolution was given in [5] as a consequence of width lower bounds. We define the concept of μ-dynamical satisfiability for CNF formulas which provides a direct way to obtain space lower bounds, not only for resolution but also for $Res(k)$. It is not difficult to see that dynamical satisfiability is equivalent to a particular case of the extended existential pebble game defined in [5].

Definition 28. Let $\mathcal{F}$ be a CNF over n variables and let $1 \leqslant \mu \leqslant n$. $\mathcal{F}$ is *μ-dynamically satisfiable* if there is a class $\mathcal{R}_\mathcal{F}$ of partial assignments such that the following properties hold:

1. *Closure under inclusion*: if $\alpha \in \mathcal{R}_\mathcal{F}$ and $\beta \sqsubseteq \alpha$, then $\beta \in \mathcal{R}_\mathcal{F}$;
2. *Extendibility*: if $\alpha \in \mathcal{R}_\mathcal{F}$ and $|\alpha| < \mu$ and C is a clause in $\mathcal{F}$, then there is a partial assignment $\beta \in \mathcal{R}_\mathcal{F}$ such that $\beta \sqsupseteq \alpha$, $\beta(C) = 1$.

We show that dynamical satisfiability implies space lower bounds for $Res(k)$. Obviously when k is 1, the result is valid for resolution.

Theorem 29. *Let $\mathcal{F}$ be an unsatisfiable formula in CNF, which moreover is μ-dynamically satisfiable. Then $Space_k(\mathcal{F}) > \mu/k$.*

Proof. Let $\mathcal{R}_{\mathcal{F}}$ be the class of partial assignments that makes $\mathcal{F}$ μ-dynamically satisfiable. Let $\mathcal{C}_0,\ldots,\mathcal{C}_s$ be a set of configurations representing a refutation of $\mathcal{F}$ in *Res*(k). Assuming by contradiction that $Space_k(\mathcal{F})\leqslant\mu/k$, we build by induction a sequence of partial assignments, α_i, to the variables of $\mathcal{F}$, where $i=0,\ldots,s$. These assignments have the following three properties: $\alpha_i\in\mathcal{R}_{\mathcal{F}}$, $\mathcal{C}_i|_{\alpha_i}\equiv 1$ and $|\alpha_i|\leqslant k|\mathcal{C}_i|$. The contradiction is reached since no partial assignment can satisfy $\mathcal{C}_s$ which includes the empty clause, so $Space_k(\mathcal{F})>\mu/k$.

Since $\mathcal{C}_0=\emptyset$, α_0 can be set as the empty assignment. Given α_i, we build α_{i+1} according to the rule used to produce $\mathcal{C}_{i+1}$ from $\mathcal{C}_i$.

- *Axiom download*: Let C be the down-loaded clause of $\mathcal{F}$. If a clause can be downloaded, then $|\mathcal{C}_i|\leqslant\mu/k-1$, hence $|\alpha_i|\leqslant\mu-k\leqslant\mu-1$, since $k\geqslant 1$. Since $\mathcal{F}$ is μ-dynamically satisfiable and $|\alpha_i|<\mu$, by the extendibility of $\mathcal{R}_{\mathcal{F}}$, there is a $\beta\in\mathcal{R}_{\mathcal{F}}$ such that $\beta\sqsupseteq\alpha$ and $C|_{\beta}\equiv 1$. Notice that by the closure property of $\mathcal{R}_{\mathcal{F}}$ and the fact that C is a clause, we can assume that β is setting to 1 at most one literal in C. Setting $\alpha_{i+1}=\beta$ it follows that $\alpha_{i+1}\in\mathcal{R}_{\mathcal{F}}$ and $\mathcal{C}_{i+1}|_{\alpha_{i+1}}\equiv 1$. As $|\beta|\leqslant|\alpha|+1$ and $|\mathcal{C}_{i+1}|=|\mathcal{C}_i|+1$, then $|\alpha_{i+1}|\leqslant k|\mathcal{C}_{i+1}|$.
- *Inference adding*: Set $\alpha_{i+1}=\alpha_i$. The derived k-clause is satisfied from soundness of *Res*(k) and $\alpha_{i+1}\in\mathcal{R}_{\mathcal{F}}$ because $\alpha_i\in\mathcal{R}_{\mathcal{F}}$.
- *Memory erasing*: Let C be the k-clause deleted from $\mathcal{C}_i$ to get $\mathcal{C}_{i+1}$. Clearly $\mathcal{C}_{i+1}|_{\alpha_i}\equiv 1$. For every k-clause C_j in $\mathcal{C}_{i+1}$ let $\beta_j\sqsubseteq\alpha_i$ be minimal (with respect to $\sqsubseteq$) such that $C_j|_{\beta_j}\equiv 1$. Define $\alpha_{i+1}=\bigsqcup_j\beta_j$. As $\alpha_{i+1}\sqsubseteq\alpha_i$ and $\alpha_i\in\mathscr{R}_{\mathcal{F}}$ then by the closure property $\alpha_{i+1}\in\mathcal{R}_{\mathcal{F}}$. By construction $\mathcal{C}_{i+1}|_{\alpha_{i+1}}\equiv 1$. Finally, as at most k variables are needed to satisfy a k-clause, $|\alpha_{i+1}|\leqslant k|\mathcal{C}_{i+1}|$. □

It is easy to prove size lower bounds for treelike *Res*(k) from Theorem 29 and a lemma from [16] which states that if a formula $\mathcal{F}$ requires resolution space S, then $\mathcal{F}$ requires treelike resolution size 2^S.

Since $Space^*_k(\mathcal{F})\geqslant Space_k(\mathcal{F})$, a space lower bound for resolution also yields a size lower bound for treelike resolution.

Corollary 30. *If $\mathcal{F}$ is μ-dynamically unsatisfiable, then* $Size^*_k(\mathcal{F})\geqslant 2^{\Omega(\mu/k)}$.

The rest of this section will be devoted to prove space lower bounds for *Res*(k) using μ-dynamical satisfiability.

5.2.1. Semiwide formulas

We show that the concept of semiwideness, introduced in [1], implies dynamical satisfiability.

Definition 31 (Alekhnovich et al. [1]). A partial assignment α for a satisfiable CNF $\mathcal{F}$ is *$\mathcal{F}$-consistent* if α does not falsify $\mathcal{F}$ and can be extended to an assignment satisfying $\mathcal{F}$.

The notion of consistency is used to define semiwideness for a CNF $\mathcal{F}$.

Definition 32 (Alekhnovich et al. [1]). A CNF $\mathcal{F}$ is *μ-semiwide* if and only if there exists a partition $\mathcal{F}',\mathcal{F}''$ of $\mathcal{F}$ such that $\mathcal{F}'$ is satisfiable and for any clause C in $\mathcal{F}''$, any $\mathcal{F}'$-consistent assignment α, with $|\alpha|<\mu$, can be extended to an $\mathcal{F}'$-consistent assignment satisfying C.

Now we prove that semiwideness is a particular case of dynamical satisfiability.

Lemma 33. *Let $\mathcal{F}$ be an unsatisfiable CNF over n variables. If $\mathcal{F}$ is μ-semiwide, then $\mathcal{F}$ is μ-dynamically satisfiable.*

Proof. Let $\mathcal{F}',\mathcal{F}''$ be the partition of $\mathcal{F}$ guaranteed by μ-semiwideness of $\mathcal{F}$. Fix

$$\mathcal{R}_{\mathcal{F}} = \{\alpha \mid \alpha \text{ is } \mathcal{F}'\text{-consistent}\}$$

If α is $\mathcal{F}'$-consistent, any β such that $\beta \sqsubseteq \alpha$ is $\mathcal{F}'$-consistent, so $\mathcal{R}_{\mathcal{F}}$ has the closure property. Finally to show that $\mathcal{R}_{\mathcal{F}}$ has the extendibility property, we prove that for any clause C in $\mathcal{F}$ and any $\alpha \in \mathcal{R}_{\mathcal{F}}$, such that $|\alpha|<\mu$, there is an extension β of α in $\mathcal{R}_{\mathcal{F}}$ that satisfies C. If $C \in \mathcal{F}'$, by $\mathcal{F}'$-consistency of α, there is a $\mathcal{F}'$-consistent β extending α that satisfies C. Hence $\beta \in \mathcal{R}_{\mathcal{F}}$.

If $C \in \mathcal{F}''$, since $|\alpha|<\mu$, then by semiwideness of $\mathcal{F}$, there is a $\mathcal{F}'$-consistent β extending α that satisfies C. Hence $\beta \in \mathcal{R}_{\mathcal{F}}$. □

We will consider now two semiwide formulas, namely Graph Tautologies and Pigeonhole Principle. Alekhnovich et al. [3] proved that the class of contradictions GT_n is $n/2$-semiwide. Hence by Lemma 33 and Theorem 29:

Corollary 34. *GT_n is $n/2$-dynamically satisfiable and $Space_k(GT_n) > n/2k$.*

Besides, these formulas provide another example that separates resolution from tree-like $Res(k)$. In [31] it is proved that GT_n has polynomial size resolution refutations, hence also $Res(k)$ polynomial size refutations. This along with Corollaries 34 and 30 gives another proof for Corollary 18.

Alekhnovich et al. prove in [3] that for $m>n$, PHP^m_n is n-semiwide, so we have by Lemma 33 and Theorem 29:

Corollary 35. *For any $m>n$, the formula PHP^m_n is n-dynamically satisfiable and $Space_k(PHP^m_n) > n/k$.*

5.2.2. *Random formulas*

Let $\mathbf{F}^n_m$ be the probability distribution obtained by selecting m clauses of size exactly 3 independently, uniformly at random from the set of all $2^3 \cdot \binom{n}{3}$ clauses of size 3 built on n distinct variables. $\mathcal{F} \sim \mathbf{F}^n_m$, means that $\mathcal{F}$ is selected at random from this distribution. A random 3-CNF formula is a formula $\mathcal{F} \sim \mathbf{F}^n_m$. In this subsection, we prove that random 3-CNF with clause/variable ratio $\Delta > 4.6$ requires $\Omega(n/k\Delta^{1+\varepsilon})$ space in $Res(k)$. Our result can be extended to any l-CNF.

We need some preliminary definitions from [9].

The *matching game* is a two-player game defined on bipartite graphs $G=(U,V,E)$. For a node $u\in U$, let $N(u)=\{v\in V\mid(u,v)\in E\}$.

The first player, Pete, is looking for a subset $U'\subseteq U$ unmatchable into V, downloading vertices of U into U' or removing vertices from U', one at time. The second player, Dana, tries to delay Pete as long as she can, forcing a matching of U' into V. During the game the players will build a set of edges $m\subseteq E$ and the set U' as follows:

Initially $m=\emptyset=U'$. At each round only one of the following occurs:

1. Pete downloads a $u\in U$ into U', and Dana, if possible, answers by $v_u\in N(u)$ such that v_u is not a vertex of any edge in m. Then (u,v_u) is added to m;
2. Pete removes a u from U'. Then (u,v_u) is also removed from m, releasing v_u for a future use by Dana.

Pete wins when no answer is possible for Dana in case 1. Dana wins the game when she can force a matching of the whole U into V. The set m defines a partial matching in G. The complexity of the game, *MSpace*(G), is the cardinality of the smallest U' Pete has to produce in any strategy to win. Notice that when $|U|>|V|$ Pete can always win and *MSpace*$(G)\leqslant|V|+1$. Moreover, if *MSpace*$(G)>k$, then there is strategy for Dana such that for any $U'\subseteq U$, $|U|\leqslant k$, and for any $u\in U\backslash U'$ she can always find a v_u to match u.

Given a CNF $\mathcal{F}$, the bipartite graph $G_{\mathcal{F}}=(U,V,E)$ associated to $\mathcal{F}$ is defined this way: U is the set of clauses of $\mathcal{F}$, V is the set of variables of $\mathcal{F}$ and $(C,x)\in E$ iff the variable x appears (negated or not) in C. It is clear that any partial matching m in $G(\mathcal{F})$, defines an assignment α_m that satisfies all clauses mentioned in m and such that $|\alpha_m|=|m|$.

Lemma 36. *Let $\mathcal{F}$ be a CNF formula. If MSpace$(G(\mathcal{F}))\geqslant\mu$, then $\mathcal{F}$ is μ-dynamically satisfiable.*

Proof. Let $\mathcal{F}$ be formed by the clauses $C_1,\dots,C_t$. Since *MSpace*$(G(\mathcal{F}))\geqslant\mu$, there is a strategy $\mathcal{S}$ for Dana such that as long as $|U'|<\mu$, she can always extend the matching m built so far, to any other possible clause still not in U' (cf. definition of *MSpace*).

Let $I=\{i_1,\dots,i_l\}\subseteq[t]$ be a set of indices. We need the order of the indices in I to be meaningful. Therefore, any set J obtained permuting the elements of I will be considered different from I. For $I\subseteq[t]$, let $P_I=\{J\mid J$ is a permutation of $I\}$. Given an ordered set $I\subseteq[t]$, let $\mathcal{F}_I=\{C_i\in\mathcal{F}\mid i\in I\}$, where the order of I is inherited in $\mathcal{F}_I$. Let moreover m_I the matching built by Dana following the strategy $\mathcal{S}$ when the clauses in $\mathcal{F}_I$ are put by Pete into U' in the order inherited from I. Let α_I be the assignment associated to the matching m_I. We define $\mathcal{R}_{\mathcal{F}}$ as follows:

$$\mathcal{R}_{\mathcal{F}}=\{\alpha_J\mid J\in P_I \text{ for some } I\subseteq[t],|I|\leqslant\mu\}.$$

$\mathcal{R}_{\mathcal{F}}$ is clearly closed under inclusion by definition. Let $\alpha\in\mathcal{R}_{\mathcal{F}}$, with $|\alpha|<\mu$ and let C_l be a clause in $\mathcal{F}$. There is a $I\subseteq[t]$, and a $J\in P_I$, such that $\alpha=\alpha_J$. Since there is a 1-1 correspondence between m_J and the domain of α_J, then $|J|<\mu$. If $l\in J$, then C_l is satisfied by α_J and we have nothing to prove. Otherwise let $J'=J\cup\{l\}$ and l is the last element in the order of J'. $|J'|\leqslant\mu$ and hence $\alpha_{J'}\in\mathcal{R}_{\mathcal{F}}$. Moreover $\alpha_{J'}$ clearly satisfies C_l, $\alpha_{J'} \sqsupseteq \alpha_J$ since l is defined as last element in the order of J'. Hence $\mathcal{R}_{\mathcal{F}}$ verifies extendibility. □

When $\mathcal{F}$ is a random k-CNF, Ben-Sasson and Galesi in [9] proved the following result

Lemma 37 (Ben-Sasson and Galesi [9]). *Let* $\mathcal{F} \sim \mathbf{F}^n_{\Delta \cdot n}$, $\Delta > 4.6$. *For any* $\varepsilon < 1$, $MSpace(G_{\mathcal{F}}) > n/\Delta^{1+\varepsilon}$.

Which, by Lemma 36, implies

Corollary 38. *If* $\mathcal{F} \sim \mathbf{F}^n_{\Delta \cdot n}$, $\Delta > 4.6$, *then* $\mathcal{F}$ *is* $n/\Delta^{1+\varepsilon}$*-dynamically satisfiable*.

Which by Theorem 29 and Corollary 30 in turns implies:

Corollary 39. *If* $\mathcal{F} \sim \mathbf{F}^n_{\Delta \cdot n}$, $\Delta > 4.6$, *then for each* $k \geqslant 1$, $Space_k(\mathcal{F}) \geqslant \Omega(n/k\Delta^{1+\varepsilon})$ *and* $Size^*_k(\mathcal{F}) \geqslant 2^{\Omega(n/k\Delta^{1+\varepsilon})}$.

5.2.3. *Tseitin contradictions*

Tseitin contradictions are unsatisfiable sets of clauses associated to undirected graphs. Let $G = (V, E)$ be a graph and let $w : V \to \{0, 1\}$ be an odd weight function, that is, a function such that $\bigoplus_{v \in V} w(v) = 1$. The Tseitin contradiction $T(G, w)$ is defined as follows. Consider variables x_e associated to each edge e of G. For all $v \in V$, let $Par(v)$ be the CNF expansion of $\bigoplus_{e \ni v} x_e \equiv w(v)$. Then

$$T(G, w) := \bigwedge_{v \in V} Par(v).$$

Observe that if the maximal degree $d(G)$ of G is constant, the number of clauses defining $T(G, w)$ is linear in the number of nodes in G.

Definition 40. Let G be a connected graph over n nodes. The *connectivity expansion* $c(G)$ of a connected graph G is the minimal number of edges to remove from G to obtain a graph in which the largest connected component is of size at most $n/2$.

Let $G = (V, E)$ be a constant degree connected graph and w an odd weight function. Let α be a partial assignment on variables of $T(G, w)$. $E(\alpha)$ is the subset of E corresponding to the variables assigned by α, and $G_{\max}(\alpha) = (V_{\max}(\alpha), E_{\max}(\alpha))$ is the maximal connected component in $(V, E - E(\alpha))$. Note that by Definition 40, for any set $E' \subseteq E$ of edges, such that $|E'| < c(G)$ there is a maximal connected component in the graph $(V, E - E')$ of size bigger than $n/2$, which moreover is unique.

Following [1], we define a special class of assignments.

Definition 41. An assignment α with $|\alpha| < c(G)$ is *admissible* for $T(G, w)$ if there exists an assignment β such that (1) $\alpha \sqsubseteq \beta$, and (2) for all $v \notin V_{\max}(\alpha)$, β satisfies $Par(v)$.

Notice that the class of admissible assignments is clearly closed under inclusion. This follows since for any $\alpha' \sqsubseteq \alpha$ where α is admissible it holds $V_{\max}(\alpha) \subseteq V_{\max}(\alpha')$.

The following lemma was proved in [1].

Lemma 42. *Assume that α is admissible for $T(G,w)$. Then for any $v_0 \in V_{\max}(\alpha)$ there exists an assignment β such that $\alpha \sqsubseteq \beta$ and for each vertex $v \neq v_0$, β satisfies $Par(v)$.*

We are ready to prove dynamical satisfiability of Tseitin formulas. The proof is very similar to the resolution space lower bounds for Tseitin formulas in [1].

Theorem 43. *Let $G=(V,E)$ be a graph with $d(G) \geqslant 1$ and let $w: V \to \{0,1\}$ be such that $\bigoplus_{v\in V} w(v)=1$. Then $T(G,w)$ is $(c(G)-d(G))$-dynamically satisfiable.*

Proof. We prove that $\mathcal{R}$, the class of admissible assignments for $T(G,w)$, witnesses the $(c(G)-d(G))$-dynamical satisfiability of $T(G,w)$.

The closure property is immediate by the closure property of the class of admissible assignments.

Let C be a clause in $T(G,w)$ and $\alpha \in \mathcal{R}$, $|\alpha| < c(G) - d(G)$. Let v be the node of G such that $C \in Par(v)$. We will show an assignment β in $\mathcal{R}$, extending α and satisfying C. This proves the extendibility property of $\mathcal{R}$.

If v is not in the maximal connected component of $G(\alpha)$, i.e. $v \notin V_{\max}(\alpha)$, then, by admissibility, there is a β extending α and satisfying $Par(v)$, in particular C. Observe that $V_{\max}(\alpha) = V_{\max}(\beta)$, so β is admissible and hence in $\mathcal{R}$.

Assume now that $v \in V_{\max}(\alpha)$. Let $E(v)$ be the set of edges adjacent to v, clearly $|E(v)| \leqslant d(G)$. Let $E(\alpha)$ be the edges assigned in α. Let $E' = E(\alpha) \cup E(v)$. Notice also that, since $|\alpha| = |E(\alpha)| < c(G) - d(G)$ and $|E(v)| \leqslant d(G)$, then $|E'| < c(G)$.

Let $V'_{\max}$ be the maximal connected component of $(V, E \backslash E')$. Since $|E'| < c(G)$ we have that $|V'_{\max}| > n/2$ and therefore unique, which means that $V'_{\max} \subseteq V_{\max}(\alpha) \backslash \{v\}$. Fix $v_0 \in V'_{\max}$ and let β' be the extension of α from Lemma 42, such that for all $v' \neq v_0$, β' satisfies $Par(v')$. Clearly C is satisfied by β', say by setting the variable x_e to the truth value x_e^{ε}.

Let $\beta = \alpha \cup \{x_e := x_e^{\varepsilon}\}$. We prove that β is admissible and hence in $\mathcal{R}$. Clearly $|\beta| < c(G)$ since $|\alpha| < c(G) - d(G)$ and $d(G) \geqslant 1$. Observe that $V'_{\max} \subseteq V_{\max}(\beta)$ and therefore $v_0 \in V_{\max}(\beta)$. Now notice that β' is the required extension of β that satisfies $Par(v')$ for any node $v' \notin V_{\max}(\beta)$. $\square$

Linear lower bounds for Tseitin contradictions are a consequence of the following lemma which uses expander graphs.

Lemma 44 (Urquhart [33], Alekhnovich [1]). *There exists a family of constant degree connected graphs $G=(V,E)$ with connectivity expansion $\Omega(|V|)$.*

Theorem 45. *Let $G=(V,E)$ be connected graph over n vertices provided by Lemma* 44, *and let w be an odd weight function. Then for any $k \geqslant 1$, $Space_k(T(G,w)) \geqslant \Omega(n)$.*

6. Discussion and open problems

One interesting open problem regarding space complexity is to find out the space for refuting $Peb_{G,1,2}$ in resolution. This problem was posed in [8] and mentioned as an open problem in several other papers. Notice that Theorem 26, giving constant space $Res(2)$ refutations for $Peb_{G,1,2}$, answers to this question for $Res(2)$.

As suggested by Ben-Sasson, a problem arising from this work is to know whether $Space_k(F) \geqslant Space_1(F)/k$, for all unsatisfiable CNF F and for all k. Notice that, again by Theorem 26, a positive answer to this question would solve the resolution space complexity of $Peb_{G,1,2}$, giving a constant upper bound. On the other hand, proving a non-constant lower bound for resolution space of $Peb_{G,1,2}$, would give a counter-example to the previous inequality. Moreover, since $Peb_{G,1,2}$ has a constant-width resolution refutation, see [10], this lower bound would also separate the space complexity measure from the width complexity measure for general resolution, solving a problem arising in [5].

There are purely combinatorial characterizations of width [5], and of treelike space [17]. So, one natural question to ask is whether dynamical satisfiability could provide a combinatorial characterization of $Res(k)$ space. The constant-space $Res(2)$ refutations of Theorem 26 prove that $Peb_{G,1,2}$ is $O(1)$-dynamical satisfiable. Therefore a super-constant lower bound for resolution space of $Peb_{G,1,2}$ formulas cannot be proved using the dynamical satisfiability property. On the other hand, proving that dynamical satisfiability is equivalent to $Res(k)$ space, would give constant space refutations of $Peb_{G,1,2}$ in resolution.

Acknowledgements

We would like to thank Jan Krajíček for some remarks about consequences of our separations in Bounded Arithmetic, Maria Luisa Bonet, Albert Atserias and Neil Thapen for helpful discussions.

References

[1] M. Alekhnovich, E. Ben-Sasson, A. Razborov, A. Wigderson, Space complexity in propositional calculus, SIAM J. Comput. 31 (4) (2002) 1184–1211.

[2] M. Alekhnovich, A.A. Razborov, Resolution is not automatizable unless $W[P]$ is not tractable, Fortysecond IEEE Symp. on Foundations of Computer Science, FOCS 2001, pp. 210–219.

[3] A. Atserias, M.L. Bonet, On the automatizability of resolution and related propositional proof systems, Computer Science Logic, CSL 2002, Lecture Notes in Computer Science, Vol. 2471, Springer, Berlin, pp. 569–583.

[4] A. Atserias, M.L. Bonet, J.L. Esteban, Lower bounds for the weak pigeonhole principle beyond resolution, Inform. and Comput. 176 (2) (2002) 136–152.

[5] A. Atserias, V. Dalmau, A combinatorial characterization of resolution width, Proc. 18th Annual IEEE Conf. on Computational Complexity, CCC 2003, pp. 239–247.

[6] P. Beame, R.M. Karp, T. Pitassi, M.E. Saks, On the complexity of unsatisfiability proofs for random k-CNF formulas, SIAM J. Comput. 31 (4) (2002) 1048–1075.

[7] P. Beame, T. Pitassi, Simplified and improved resolution lower bounds, 37th IEEE Symp. on Foundations of Computer Science, FOCS 1996, pp. 274–282.
[8] E. Ben-Sasson, Size-space tradeoffs for resolution, 34th Annual ACM Symp. on Theory of computing, STOC 2002, pp. 457–464.
[9] E. Ben-Sasson, N. Galesi, Space complexity of random formulae in resolution, Random Structures Algorithms 23 (1) (2003) 92–109.
[10] E. Ben-Sasson, R. Impagliazzo, A. Wigderson, Near optimal separation of treelike and general resolution, Electronic Colloquium on Computational Complexity (ECCC) TR00-005, 2000, Combinatorica, to appear.
[11] E. Ben-Sasson, A. Wigderson, Short proofs are narrow—resolution made simple, J. ACM 48 (2) (2001) 149–168.
[12] M.L. Bonet, J.L. Esteban, N. Galesi, J. Johannsen, On the relative complexity of resolution refinements and cutting planes proof systems, SIAM J. Comput. 30 (5) (2000) 1462–1484.
[13] M.L. Bonet, T. Pitassi, R. Raz, On interpolation and automatization for Frege systems, SIAM J. Comput. 29 (6) (2000) 1939–1967.
[14] S. Cook, R. Reckhow, The relative efficiency of propositional proof systems, J. Symbolic Logic 44 (1979) 36–50.
[15] J.L. Esteban, N. Galesi, J. Messner, On the complexity of resolution with bounded conjunctions, International Colloquium on Automatas, Languages and Programming, ICALP 2002, Lecture Notes in Computer Science, Springer, Berlin, 2380, pp. 220–231.
[16] J.L. Esteban, J. Torán, Space bounds for resolution, Inform. and Comput. 171 (1) (2001) 84–97.
[17] J.L. Esteban, J. Torán, A combinatorial characterization of treelike resolution space, Inform. Process. Lett. 87 (6) (2003) 295–300.
[18] N. Galesi, N. Thapen, The complexity of treelike systems over λ-local formulae, 19th IEEE Conf. on Computational Complexity, 2004, to appear.
[19] A. Haken, The intractability of resolution, Theoret. Comput. Sci. 39 (1985) 297–308.
[20] H. Kleine Büning, T. Lettmann, Aussagenlogik: Deduktion und Algorithmen, B.G. Teubner, Stuttgart, 1994 (English trans. Propositional Logic: Deduction and Algorithms. Cambridge Tracts in Theoretical Computer Science 1999).
[21] J. Krajíček, Lower bounds to the size of constant-depth propositional proofs, J. Symbolic Logic 59 (1) (1994) 73–86.
[22] J. Krajíček, On the weak pigeonhole principle, Fund. Math. 170 (1–3) (2001) 123–140.
[23] J. Krajíček, P. Pudlák, Some consequences of cryptographical conjectures for S_2^1 and EF, Inform. and Comput. 140 (1) (1998) 82–94.
[24] A. Maciel, T. Pitassi, A.R. Woods, A new proof of the weak pigeonhole principle, 32nd Annual ACM Symp. on Theory of Computing, STOC 2000, pp. 368–377.
[25] W.J. Paul, R.E. Tarjan, J.R. Celoni, Space bounds for a game on graphs, Math. Systems Theory 10 (1977) 239–251.
[26] P. Pudlák, R. Impagliazzo, Lower bounds for DLL algorithms, Eleventh Annual ACM-SIAM Symp. on Discrete Algorithms, SODA 2000, pp. 128–136.
[27] R. Raz, Resolution lower bounds for the weak pigeonhole principle, 34th ACM Symp. on Theory of computing, STOC 2002, pp. 553–562.
[28] A. Razborov, Resolution lower bounds for perfect matching principles, Seventeenth IEEE Annual Conf. on Computational Complexity, CCC 2001, pp. 17–26.
[29] A. Razborov, Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution, Manuscript, 2003.
[30] N. Segerlind, S. Buss, R. Impagliazzo, A switching lemma for small restrictions and lower bounds for k-DNF resolution, 43th IEEE Symp. on Foundations of Computer Science, FOCS 2002, pp. 604–614.
[31] G. Stålmarck, Short resolution proofs for a sequence of tricky formulas, Acta Inform. 33 (3) (1996) 277–280.
[32] G.S. Tseitin, On the complexity of derivation in propositional calculus, Studies in Constructive Mathematics and Mathematical Logic, Part 2. Consultants Bureau, New York, London, 1968, pp. 115–125.
[33] A. Urquhart, Hard examples for resolution, J. ACM 34 (1) (1987) 209–219.