Author's personal copy

Information Processing Letters 109 (2009) 1030–1035

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

# A simplified way of proving trade-off results for resolution ☆

Jakob Nordström [1,2]

*Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

A R T I C L E   I N F O

*Article history:*
Received 13 January 2009
Received in revised form 21 May 2009
Accepted 14 June 2009
Available online 17 June 2009
Communicated by L.A. Hemaspaandra

*Keywords:*
Computational complexity
Automatic theorem proving
Proof complexity
Resolution
Trade-offs
Length
Width
Space

A B S T R A C T

We present a greatly simplified proof of the length-space trade-off result for resolution in [P. Hertel, T. Pitassi, Exponential time/space speedups for resolution and the PSPACE-completeness of black-white pebbling, in: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS '07), Oct. 2007, pp. 137–149], and also prove a couple of other theorems in the same vein. We point out two important ingredients needed for our proofs to work, and discuss some possible conclusions. Our key trick is to look at formulas of the type $F = G \wedge H$, where $G$ and $H$ are over disjoint sets of variables and have very different length-space properties with respect to resolution.

© 2009 Elsevier B.V. All rights reserved.

In these notes, we present a simplification of the length-space trade-off result for resolution in Hertel and Pitassi [9], and show how the same ideas can be used to prove other related theorems. After some brief preliminaries in Section 1, the simplified proof is given in Section 2. In Section 3 we prove two other trade-off results of a similar flavor. We point out two key ingredients needed for our proofs to work in Sections 4 and 5, and discuss possible conclusions to be drawn regarding proving trade-off results for resolution. Finally, in Section 6 we mention a few related open problems.

## 1. Definitions, notation and some useful facts

This is a very brief summary of the background material needed for this paper. We refer the reader to, for instance, [10, Chapter 4] for more details.

A *literal* is either a propositional logic variable $x$ or its negation $\bar{x}$. A *clause* $C = a_1 \vee a_2 \vee \cdots \vee a_k$ is a set of literals with *width* $W(C) = k$ equal to the number of literals appearing in it. A *CNF formula* $F = C_1 \wedge \cdots \wedge C_m$ is a set of clauses. A *k-CNF formula* is a CNF formula with all clauses of width at most $k$. We let $Vars(C)$ denote the set of variables in a clause $C$, and extend this notation to formulas by taking unions over clauses. The size $S(F)$ of a CNF formula $F$ is the total number of literals in $F$ (counted with repetitions). Also, the width $W(F)$ of $F$ is the width of its largest clause.

As in [9], we use the "configuration-style" definition of resolution. We employ the standard notation $[n] = \{1, 2, \ldots, n\}$.

☆ These results have previously been reported in [J. Nordström, A simplified way of proving trade-off results for resolution, Technical Report TR07-114, Electronic Colloquium on Computational Complexity (ECCC), Sept. 2007] and [J. Nordström, J. Håstad, Towards an optimal separation of space and length in resolution (Extended abstract), in: Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC '08), May 2008, pp. 701–710].

*E-mail address:* jakobn@mit.edu.

[1] Research supported in part by the Ericsson Research Foundation, the Foundation Olle Engkvist Byggmästare, and the Foundation Blanceflor Boncompagni-Ludovisi, née Bildt.

[2] This work performed while at the Royal Institute of Technology (KTH) in Stockholm, Sweden.

0020-0190/$ – see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.ipl.2009.06.006

Author's personal copy



**Definition 1.1** *(Resolution).* (See [1].) A *clause configuration* $\mathbb{C}$ is a set of clauses. A sequence of clause configurations $\{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ is a *resolution derivation* from a CNF formula $F$ if $\mathbb{C}_0 = \emptyset$ and for all $t \in [\tau]$, $\mathbb{C}_t$ is obtained from $\mathbb{C}_{t-1}$ by one of the following rules:

***Axiom download***. $\mathbb{C}_t = \mathbb{C}_{t-1} \cup \{C\}$ for a clause $C \in F$ (an *axiom*).
***Erasure***. $\mathbb{C}_t = \mathbb{C}_{t-1} \setminus \{C\}$ for some clause $C \in \mathbb{C}_{t-1}$.
***Inference***. $\mathbb{C}_t = \mathbb{C}_{t-1} \cup \{C \vee D\}$ for a clause $C \vee D$ inferred by the *resolution rule* from clauses $C \vee x, D \vee \bar{x} \in \mathbb{C}_{t-1}$.

A *resolution refutation* of $F$ is a derivation $\pi : F \vdash 0$ of the empty clause 0 (the clause with no literals) from $F$.

We are interested in the following complexity measures:

- The *length* $L(\pi)$ of a resolution derivation $\pi$ is the number of clauses in $\pi$, i.e., the number of axiom downloads and inference steps.
- The *width* $W(\pi)$ of a derivation $\pi$ is the number of literals in the largest clause in $\pi$.
- The *clause space* $Sp(\pi)$ of a derivation $\pi$ is the maximal number of clauses in any clause configuration $\mathbb{C}_t \in \pi$.
- The *variable space* $VarSp(\pi)$ of a derivation $\pi$ is the maximal number of literals, counted with repetitions, in any clause configuration $\mathbb{C}_t \in \pi$.

The length of refuting $F$ is $L(F \vdash 0) = \min_{\pi : F \vdash 0}\{L(\pi)\}$, where the minimum is taken over all resolution refutations of $F$. The width $W(F \vdash 0)$, clause space $Sp(F \vdash 0)$ and variable space $VarSp(F \vdash 0)$ of refuting $F$ are defined completely analogously.

It is easy to see that any CNF formula $F$ over $n$ variables is refutable in length $\exp(\mathrm{O}(n))$ and width $\mathrm{O}(n)$. In [8] it was proved that the clause space of refuting $F$ is upper-bounded by the formula size. We will need the fact that there are polynomial-size $k$-CNF formulas that are very hard with respect to length, width and clause space, essentially meeting the upper bounds just stated.

**Theorem 1.2.** *(See [1,3,6,16,17].) There are arbitrarily large unsatisfiable* 3*-CNF formulas $F_n$ of size $\Theta(n)$ with $\Theta(n)$ clauses and $\Theta(n)$ variables for which it holds that $L(F_n \vdash 0) = \exp(\Theta(n))$, $W(F_n \vdash 0) = \Theta(n)$ and $Sp(F_n \vdash 0) = \Theta(n)$.*

One example of such formulas are random 3-CNF formulas, which are almost surely unsatisfiable if one picks $Kn$ clauses over $n$ variables for some suitably large $K$. Another, more explicit, example is provided by formulas encoding (the negation of) the fact that if a function $f : V \mapsto \{0,1\}$ assigns values to the vertices $V$ in an undirected graph $G$ in such a way that $f(v)$ is equal to the parity of the number of edges incident to $v$, then $\sum_{v \in V} f(v)$ cannot be odd (since each edge is counted twice). Such formulas are hard if $G$ is an expander with suitable parameters.

Clearly, for formulas $F_n$ as in Theorem 1.2 it also holds that $\Omega(n) = VarSp(F_n \vdash 0) = \mathrm{O}(n^2)$, since the clause space is a lower bound for the variable space, which is in turn upper-bounded by the clause space multiplied by the number of variables in the formula (which is the maximum size of any clause). We note in passing that determining the exact variable space complexity of any of these formula families was mentioned as an open problem in [1], and to the best of our knowledge this problem is still unsolved.

We will also need that there are formulas that are easy with respect to length but moderately hard with respect to width and clause space. These are formulas encoding (the negation of) the fact that any strict order over a finite set must have at least one minimal element.

**Theorem 1.3.** *(See [1,7,15].) There are arbitrarily large unsatisfiable* 3*-CNF formulas $F_n$ of size $\Theta(n^3)$ with $\Theta(n^3)$ clauses and $\Theta(n^2)$ variables such that $W(F_n \vdash 0) = \Theta(n)$ and $Sp(F_n \vdash 0) = \Theta(n)$, but for which there are resolution refutations[3] $\pi_n : F_n \vdash 0$ in length $L(\pi_n) = \mathrm{O}(n^3)$, width $W(\pi_n) = \mathrm{O}(n)$ and clause space $Sp(\pi_n) = \mathrm{O}(n)$.*

Finally, we will use the following easy observation.

**Observation 1.4.** *Suppose that $F = G \wedge H$ where $G$ and $H$ are unsatisfiable CNF formulas over disjoint sets of variables. Then any resolution refutation $\pi : F \vdash 0$ must contain a refutation of either $G$ or $H$.*

**Proof.** By induction, we can never resolve a clause derived from $G$ with a clause derived from $H$, since the sets of variables of the two clauses are disjoint. □

## 2. A proof of Hertel and Pitassi's trade-off result

We show the following version of the length-space trade-off theorem of Hertel and Pitassi [9], with somewhat improved parameters[4] and a very much simpler proof.

**Theorem 2.1.** *There is a family of CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that:*

[3] Note that [7], where an explicit resolution refutation upper-bounding the proof complexity measures is presented, does not talk about clause space, but it is straightforward to verify that the refutation there can be carried out in length $\mathrm{O}(n^3)$ and clause space $\mathrm{O}(n)$.

[4] Although the exact parameter values are not the focus of this paper, for the interested reader we give a short comparison to the result of Hertel and Pitassi in this footnote. Letting $n$ denote the formula size, in [9] the formulas require variable space $\Omega(\sqrt[3]{n})$, refutations in minimal space must have length $\exp(\Omega(\sqrt[3]{n}))$, but allowing 3 more literals in memory brings the length down to (sub)linear in $n$. Our formulas require space $\Omega(\sqrt{n})$, refutations in minimal space must have length $\exp(\Omega(\sqrt{n}))$, and 2 more literals are sufficient to get down to linear length. By padding, it is easy to go from our parameters to those in [9] but not the other way round. Also, our trade-offs hold for resolution *refutations* of unsatisfiable formulas, whereas [9] deals with *derivations* of a specific unit clause $x$ from satisfiable formulas. Again, it is easy to go from the refutation setting to the derivation setting. In the other direction, the natural thing to do would be to add the clause $\bar{x}$ to the formulas and argue that the trade-off should still hold. Such a modification introduces nontrivial technical problems, however, and it is unclear to us whether the (already very intricate) proof in [9] can be extended to handle this case. As a final comment, let us note that a drawback of our result is that our formulas are non-explicit while those in [9] are explicitly constructible.

Author's personal copy



- *The variable space required to refute $F_n$ in resolution is $VarSp(F_n \vdash 0) = \Theta(n)$.*
- *Any refutation $\pi : F_n \vdash 0$ in minimal variable space has length $L(\pi) = \exp(\Omega(\sqrt{n}))$.*
- *Adding at most 2 extra units of storage, one can obtain a refutation $\pi'$ in space $VarSp(\pi') = VarSp(F_n \vdash 0) + 2 = \Theta(n)$ and length $L(\pi') = O(n)$, i.e., linear in the formula size.*

We note that the CNF formulas used by Hertel and Pitassi, as well as those in our proof, have clauses of width $\Theta(n)$.

**Proof.** Let $G_n$ be CNF formulas as in Theorem 1.2 having size $\Theta(n)$, refutation length $L(G_n \vdash 0) = \exp(\Omega(n))$, and refutation clause space $Sp(G_n \vdash 0) = \Theta(n)$. Let us define $g(n) = VarSp(G_n \vdash 0)$ to be the refutation variable space of the formulas. Then it holds that $\Omega(n) = g(n) = O(n^2)$.

Let $H_m$ be the formulas $H_m = y_1 \wedge \cdots \wedge y_m \wedge (\bar{y}_1 \vee \cdots \vee \bar{y}_m)$. It is not hard to see that there are resolution refutations $\pi : H_m \vdash 0$ in length $L(\pi) = 2m + 1$ and variable space $VarSp(\pi) = 2m$, and that $L(H_m \vdash 0) = 2m + 1$ and $VarSp(H_m \vdash 0) = 2m$ are also the lower bounds (all clauses must be used in any refutation, and the minimum space refutation must start by downloading the wide clause and some unit clause, and then resolve).

Now define $F_n = G_n \wedge H_{\lfloor g(n)/2 \rfloor + 1}$, where $G_n$ and $H_{\lfloor g(n)/2 \rfloor + 1}$ have disjoint sets of variables. By Observation 1.4, any resolution refutation of $F_n$ refutes either $G_n$ or $H_{\lfloor g(n)/2 \rfloor + 1}$. We have

$$VarSp(H_{\lfloor g(n)/2 \rfloor + 1} \vdash 0) = 2 \cdot \left(\lfloor g(n)/2 \rfloor + 1\right) > g(n) = VarSp(G_n \vdash 0), \qquad (1)$$

so a resolution refutation in minimal variable space $g(n)$ must refute $G_n$. This requires length $\exp(\Omega(n))$. However, by construction $H_{\lfloor g(n)/2 \rfloor + 1}$ is refutable in variable space $2(\lfloor g(n)/2 \rfloor + 1) \leqslant g(n) + 2$, so if we allow at most two more literals in memory, the resolution refutation can instead disprove the formula $H_{\lfloor g(n)/2 \rfloor + 1}$ in length linear in the (total) formula size.

Thus, we have a formula family $\{F_n\}_{n=1}^{\infty}$ of size $\Omega(n) = S(F_n) = O(n^2)$ refutable in length and variable space both linear in the formula size, but where any minimum variable space refutation must have length $\exp(\Omega(n))$. Adjusting the indices as needed, we get a formula family with a trade-off of the form stated in Theorem 2.1. □

We note that the trick of "gluing together" two unrelated formulas with very different length-space properties in resolution is not present in the proof of Hertel and Pitassi, and thus it might be possible to use their techniques to prove results not obtainable by our methods.

## 3. Some other trade-off results for resolution

Using a similar trick as in the previous section, we can prove the following length-clause space trade-off.

**Theorem 3.1.** *There is a family of k-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that*:

- *The minimal clause space of refuting $F_n$ in resolution is $Sp(F_n \vdash 0) = \Theta(\sqrt[3]{n})$.*
- *Any refutation $\pi : F_n \vdash 0$ in minimal clause space must have length $L(\pi) = \exp(\Omega(\sqrt[3]{n}))$.*
- *There are resolution refutations $\pi' : F_n \vdash 0$ in asymptotically minimal clause space $Sp(\pi') = O(Sp(F_n \vdash 0))$ and length $L(\pi') = O(n)$, i.e., linear in the formula size.*

The same game can be played with refutation width as well.

**Theorem 3.2.** *There is a family of k-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta(n)$ such that*:

- *The minimal width of refuting $F_n$ is $W(F_n \vdash 0) = \Theta(\sqrt[3]{n})$.*
- *Any resolution refutation $\pi : F_n \vdash 0$ in minimal width must have length $L(\pi) = \exp(\Omega(\sqrt[3]{n}))$.*
- *There are refutations $\pi' : F_n \vdash 0$ in width $W(\pi') = O(W(F_n \vdash 0))$ and length $L(\pi') = O(n)$.*

We only present the proof of Theorem 3.1, as Theorem 3.2 is proved in exactly the same manner.

**Proof of Theorem 3.1.** Let $G_n$ be a 3-CNF formula family as in Theorem 1.2 having size $\Theta(n)$, refutation length $L(G_n \vdash 0) = \exp(\Theta(n))$, and refutation clause space $Sp(G_n \vdash 0) = \Theta(n)$. Let $H_m$ be a 3-CNF formula family as in Theorem 1.3 of size $\Theta(m^3)$ such that $L(H_m \vdash 0) = O(m^3)$ and $Sp(H_m \vdash 0) = \Theta(m)$. Define $g(n) = \min\{m \mid Sp(H_m \vdash 0) > Sp(G_n \vdash 0)\}$. Note that since $Sp(H_m \vdash 0) = \Omega(m)$ and $Sp(G_n \vdash 0) = O(n)$, we know that $g(n) = O(n)$.

Now as before let $F_n = G_n \wedge H_{g(n)}$, where $G_n$ and $H_{g(n)}$ have disjoint sets of variables. By Observation 1.4, any resolution refutation of $F_n$ is a refutation of either $G_n$ or $H_{g(n)}$. Since $g(n)$ has been chosen so that $Sp(H_{g(n)} \vdash 0) > Sp(G_n \vdash 0)$, a refutation in minimal clause space has to refute $G_n$, which requires exponential length. However, since $g(n) = O(n)$, Theorem 1.3 tells us that there are refutations of $H_{g(n)}$ in length $O(n^3)$ and clause space $O(n)$. □

## 4. Making the main trick explicit

The proofs of the theorems in Sections 2 and 3 come very easily; in fact almost *too* easily. What is it that makes this possible? In this and the next section, we want to highlight two key ingredients in the constructions.

The common paradigm for the proofs of Theorems 2.1, 3.1, and 3.2 is as follows. We are given two complexity measures $M_1$ and $M_2$ that we want to trade off against one another. We do this by finding formulas $G_n$ and $H_m$ such that:

- The formulas $G_n$ are very hard with respect to the first resource measured by $M_1$, while $M_2(G_n)$ is at most some (more or less trivial) upper bound.
- The formulas $H_m$ are very easy with respect to $M_1$, but there is some nontrivial *lower* bound on the usage $M_2(H_m)$ of the second resource.
- The index $m = m(n)$ is chosen so as to minimize $M_2(H_{m(n)}) - M_2(G_n) > 0$, i.e., so that $H_{m(n)}$ requires *just a little bit* more of the second resource than $G_n$.

Author's personal copy



Then for $F_n = G_n \wedge H_{m(n)}$, if we demand that a resolution refutation $\pi$ must use the minimal amount of the second resource, it will have to use a large amount of the first resource. However, relaxing the requirement on the second resource by the very small expression $M_2(H_{m(n)}) - M_2(G_n)$, we can get a refutation $\pi'$ using small amounts of both resources.

Clearly, the formula families $\{F_n\}_{n=1}^{\infty}$ that we get in this way are "redundant" in the sense that each formula $F_n$ is the conjunction of two formulas $G_n$ and $H_m$ which are themselves already unsatisfiable. Formally, we say that a formula $F$ is *minimally unsatisfiable* if $F$ is unsatisfiable, but removing any clause $C \in F$ makes the remaining subformula $F \setminus \{C\}$ satisfiable. We note that if we would add the requirement in Sections 2 and 3 that the formulas under consideration should be minimally unsatisfiable, the proof idea outlined above fails completely. What conclusions can be drawn from this?

On the one hand, trade-off results for minimally unsatisfiable formulas would seem more interesting, since they tell us something about a property that some natural formula family has, rather than about some funny phenomena arising because we glue together two totally unrelated formulas.

On the other hand, one could argue that the main motivation for studying space is the connection to memory requirements for proof search algorithms, for instance, algorithms using clause learning. And for such algorithms, a minimality condition might appear somewhat arbitrary. There are no guarantees that "real-life" formulas will be minimally unsatisfiable, and most probably there is no efficient way of testing this condition. More precisely, the problem of deciding minimal unsatisfiability is NP-hard but not known to be in NP. Formally, a language $L$ is in the complexity class DP if and only if there are two languages $L_1 \in \mathrm{NP}$ and $L_2 \in \text{co-NP}$ such that $L = L_1 \cap L_2$ [13]. MINIMAL UNSATISFIABILITY is DP-complete [14], and it seems to be commonly believed that $\mathrm{DP} \nsubseteq \mathrm{NP} \cup \text{co-NP}$.[5] Therefore, in practice trade-off results for non-minimal formulas might be just as interesting.

## 5. An auxiliary trick for variable space

A second important reason why our proof of Theorem 2.1 gives sharp results is that we are allowed to use CNF formulas of growing width. It is precisely because of this that we can easily construct the needed formulas $H_m$ that are hard with respect to variable space but easy with respect to length. If we would have to restrict ourselves to $k$-CNF formulas for $k$ constant, it would be much more difficult to find such examples. Although the formulas in Theorem 1.3 could be plugged in to give a slightly weaker trade-off, we are not aware of any family of $k$-CNF formulas that can provably give the very sharp result in Theorem 2.1.

This is not the only example of a space measure behaving badly for formulas of growing width. Another example of this is the relationship between clause space and width. When space began to be studied in the late 1990s, it was soon noted in several papers (for instance [1,3,16]) that the lower bounds on refutation width and refutation space for different formula families coincided. In [2], it was shown that this was not a coincidence, but that the minimal refutation clause space upper-bounds the minimal refutation width by

$$Sp(F \vdash 0) \geqslant W(F \vdash 0) - F + 3, \qquad (2)$$

but it remained open whether space and width could be separated or the two measures were asymptotically the same. In the sequence of works [11,12,4] jointly with Håstad and Ben-Sasson we proved that the inequality is asymptotically strict in the sense that there are $k$-CNF formula families $F_n$ with $W(F_n \vdash 0) = \mathrm{O}(1)$ but $Sp(F_n \vdash 0) = \Theta(n/\log n)$.

However, if we are allowed to consider formulas of growing width, the fact that the inequality (2) is not tight is entirely trivial. Namely, let us say that a CNF formula $F$ is *k-wide* if all clauses in $F$ have size at least $k$. In [8], it was proven that for $F$ a $k$-wide unsatisfiable CNF formula it holds that $Sp(F_n \vdash 0) \geqslant k + 2$. So in order to get a formula family $F_n$ such that $W(F_n \vdash 0) - W(F_n) = \mathrm{O}(1)$ but $Sp(F_n \vdash 0) = \omega(1)$, just pick some suitable formulas $\{F_n\}_{n=1}^{\infty}$ of growing width.

In our opinion, these phenomena are clearly artificial. Since every CNF formula can be rewritten as an equivalent $k$-CNF formula without increasing the size more than linearly (using extension variables), the right approach when studying space measures in resolution seems to be to require that the formulas under study should have constant width.

## 6. Conclusion and some open problems

We have established that for all the measures clause space, variable space and width, there are nontrivial trade-offs with respect to length in resolution. However, our trade-off results apply only for a very carefully selected ratio of space/width-to-formula-size and display a abrupt decay of proof length when the space/width is increased even by very small amounts. It would be desirable to obtain a clearer and more complete view of what kind of trade-off phenomena can occur if we are interested in a wider range of parameters and also in trade-offs which are more robust in the sense that they are not sensitive to small changes in the proof complexity measures.

For width, [6] showed that given a resolution refutation $\pi$ of a $k$-CNF formula $F$ in length $L(\pi) = L$, there exists a refutation in width $\mathrm{O}(\sqrt{n \log L})$, where $n$ is the number of variables in $F$. However, the refutation resulting from the proof is not the same $\pi$, but another refutation $\pi'$ which is potentially exponentially longer than $\pi$. A very natural question is whether this increase in length is necessary or

[5] It should be pointed out, however, that although finding a minimal unsatisfiable subformula is presumably hard, partitioning a formula into subformulas over distinct variable sets is easy. For instance, a SAT solver which runs a preprocessor looking for distinct variable sets will easily find the partition of $F$ into $G$ and $H$. Nevertheless, the time–space requirements of the SAT solver will still be subject to the trade-offs shown above, and there are also (artificial) ways to construct formulas in such a way that the subformulas do not have distinct variable sets (by adding small sets of dummy variables that overlap in $G$ and $H$).

Author's personal copy



whether the exponential blow-up is just an artifact of the proof.

**Open Problem 1.** If $F$ is a $k$-CNF formula over $n$ variables refutable in length $L$, can one always find a refutation $\pi$ of $F$ in width $W(\pi) = \mathrm{O}(\sqrt{n \log L})$ with length no more than, say, $L(\pi) = \mathrm{O}(L)$ or at most $\mathrm{poly}(L)$? Or is there a formula family which necessarily exhibits a length-width trade-off in the sense that there are short refutations and narrow refutations, but all narrow refutations have a length blow-up (polynomial or superpolynomial)?

Note that here we do not ask that the resolution refutations should attain the *exact* minimum with respect to refutation length or width, but rather that it should be within a constant factor (or for length possibly within a polynomial factor). Therefore, the ideas in Sections 2 and 3 can no longer be used.

Similar questions can be posed for clause space and variable space. Suppose that we have a formula that is refutable in both small space and short length. Is it then possible to refute the formula in small space and short length *simultaneously*, possibly increasing the space by a constant factor or the length by some polynomial factor? In recent joint work with Ben-Sasson [5], we give a very strong negative answer to this question by proving superpolynomial or even exponential trade-offs for length with respect to both variable space and clause space for space in almost the whole range between constant and linear in the formula size. However, it still remains open what kind of trade-offs are possible at the extremal points of the space interval, i.e., for constant and (super)linear space.

Given a formula refutable in constant space, we know that there must exist a refutation in polynomial length as well. This follows by applying the upper bound (2) on width in terms of clause space, and then noting that narrow proofs are trivially short (for width $w$, $(2 \cdot \#\text{variables})^w$ is an upper bound on the total number of distinct clauses). But as in [6], the refutation we end up with is not the same as that with which we started. This leads to the following question.

**Open Problem 2.** Given a family of polynomial-size $k$-CNF formulas $\{F_n\}_{n=1}^{\infty}$ with refutation clause space $Sp(F_n \vdash 0) = \mathrm{O}(1)$, are there refutations $\pi : F_n \vdash 0$ simultaneously in length $L(\pi) = \mathrm{poly}(n)$ and clause space $Sp(\pi) = \mathrm{O}(1)$? Or is it possible that restricting the space to constant can force the length to be superpolynomial?

At the other end of the space interval, one can ask whether refutations of a formula in linear clause space, which have to exist, might have to be longer than the shortest refutation of the same formula.

**Open Problem 3.** Are there formulas with trade-offs in the range space $\geqslant$ formula size? Or can every refutation be carried out in, say, at most linear clause space?

We find Open Question 3 especially intriguing. Note that all bounds on clause space proven so far are in the regime where the space is less than formula size (which is quite natural, since by [8] we know the size of the formula is an upper bound on the minimal clause space needed). It is unclear to what extent such lower bounds on space are relevant to state-of-the-art SAT solvers, however, since such algorithms will presumably use at least a linear amount of memory to store the formula to begin with. For this reason, it seems to be a highly interesting problem to determine what can be said if we allow extra clause space above linear.

## Acknowledgements

We are most grateful to Philipp Hertel and Toniann Pitassi for patiently answering several technical questions about Theorem 2.1. Also, we want to thank Per Austrin, Eli Ben-Sasson, Mikael Goldmann, Johan Håstad, and Jan Krajíček for useful discussions and insightful comments. Finally, we thankfully acknowledge the anonymous referees for their valuable suggestions that helped to improve the presentation in this paper.

## References

[1] M. Alekhnovich, E. Ben-Sasson, A.A. Razborov, A. Wigderson, Space complexity in propositional calculus, SIAM Journal on Computing 31 (4) (2002) 1184–1211. Preliminary version appeared in STOC '00.

[2] A. Atserias, V. Dalmau, A combinatorial characterization of resolution width, Journal of Computer and System Sciences 74 (3) (2008) 323–334. Preliminary version appeared in CCC '03.

[3] E. Ben-Sasson, N. Galesi, Space complexity of random formulae in resolution, Random Structures and Algorithms 23 (1) (2003) 92–109. Preliminary version appeared in CCC '01.

[4] 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 (FOCS '08), Oct. 2008, pp. 709–718.

[5] E. Ben-Sasson, J. Nordström, Understanding space in resolution: Optimal lower bounds and exponential trade-offs, Technical Report TR09-034, Electronic Colloquium on Computational Complexity (ECCC), Mar. 2009.

[6] E. Ben-Sasson, A. Wigderson, Short proofs are narrow—resolution made simple, Journal of the ACM 48 (2) (2001) 149–169. Preliminary version appeared in STOC '99.

[7] M.L. Bonet, N. Galesi, Optimality of size-width tradeoffs for resolution, Computational Complexity 10 (4) (2001) 261–276.

[8] J.L. Esteban, J. Torán, Space bounds for resolution, Information and Computation 171 (1) (2001) 84–97. Preliminary version appeared in STACS '99.

[9] P. Hertel, T. Pitassi, Exponential time/space speedups for resolution and the PSPACE-completeness of black-white pebbling, in: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS '07), Oct. 2007, pp. 137–149.

[10] J. Nordström, Short proofs may be spacious: Understanding space in resolution, PhD thesis, Royal Institute of Technology, Stockholm, Sweden, May 2008. Available at the webpage http://people.csail.mit.edu/jakobn/research/.

[11] J. Nordström, Narrow proofs may be spacious: Separating space and width in resolution, SIAM Journal on Computing 39 (1) (2009) 59–121. Preliminary version appeared in STOC '06.

[12] J. Nordström, J. Håstad, Towards an optimal separation of space and length in resolution (Extended abstract), in: Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC '08), May 2008, pp. 701–710.

[13] C.H. Papadimitriou, Computational Complexity, Addison-Wesley, 1994.

[14] C.H. Papadimitriou, D. Wolfe, The complexity of facets resolved, Journal of Computer and System Sciences 37 (1) (1988) 2–13.

Author's personal copy



[15] G. Stålmarck, Short resolution proofs for a sequence of tricky formulas, Acta Informatica 33 (3) (1996) 277–280.

[16] J. Torán, Lower bounds for space in resolution, in: Proceedings of the 13th International Workshop on Computer Science Logic (CSL '99), in: Lecture Notes in Computer Science, vol. 1683, Springer, 1999, pp. 362–373.

[17] A. Urquhart, Hard examples for resolution, Journal of the ACM 34 (1) (1987) 209–219.