Lecture 8— Space complexity in proof complexity

Massimo Lauria — lauria.massimo@gmail.com

Office 1107, Ookayama West 8th Building

Friday — November 20th, 2015 (This document was updated on June 21, 2017)

In the class about SAT solving we mention how big as a problem is the memory is limited, and that CDCL algorithm requires so much of it. In this lecture we discuss the study of space in proof complexity. We first introduce appropriate variants of the proof systems of interest, and we prove space lower bounds for them.

We consider a model that takes in account memory when it comes to proof verification. The study of this model has been initiated by Jacobo Toran and by Alekhnovich et al. [1].

Definition 1. *The blackboard model Consider a proof system which is a proof is a sequence of line and such that each line is either*

- *an axiom* $\frac{}{A_i}$ *;*
- *the derivation of a new line from a constant number of previous lines* $\frac{L_{j_1} \quad L_{j_2} \quad \cdots \quad L_{j_k}}{L}$.

In the blackboard model a derivation is a sequence of memory configurations

$$(M_0, M_1, \ldots, M_T) \tag{1}$$

each of which is a set of proof lines. $M_0 = \emptyset$*, and each* M_{i+i} *is either:*

Axiom download $M_i \cup A_i$ *where* A_i *is an axiom; or*

Inference step $M_i \cup L$ *where* $\frac{L_1 \quad L_2 \quad \cdots \quad L_k}{L}$ *and* $\{L_1, L_2, \ldots, L_k\} \subseteq M_i$*; or*

Erasure *an arbitrary subset of* M_i*.*

The length of such a proof is T*, the* space *of such a proof is the maximum, among all memory configurations, of their space. The space cost of a configuration depends on the nature of the proof lines, and therefore of on the original proof system.*

This model only makes sense of proof system in which a proof is a sequence of proof lines. For each proof system we have to take in account a different way to apppropriately measure the cost of line.

Definition 2. *The space of a memory configuration is measured as*

- *the number of the clauses (for resolution);*
- *the total number of monomials in all polynomials (for polynomial calculus);*

[1] Jacobo Torán. Lower bounds for space in resolution. In *CSL*, pages 362–373, 1999; and Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Space complexity in propositional calculus. *SIAM J. Comput.*, 31(4):1184–1211, 2002

- *here the situation is less clear: any CNF can be refuted having only five inequalities in memory, but of course the coefficients in these inequalities are very large therefore it may make sense to consider the sum of the log of all coefficients in memory at each step.*[2]

The study of space for resolution is naturally the most developed. Recently there has been a string of progress related to polynomial calculus, but there is still a lot to understand. For cutting planes we really don't know anything.

If this and next lecture we will only deal with resolution. We already argued that memory it is very important for CDCL SAT solver. I would say that it is not clear yet how much resolution space correlate with actual memory usage in solvers.

An easy case: tree-like resolution.

The space complexity for tree like resolution it is earies to understand. Consider this complexity measure $\mathcal{C}$ on binary trees with in-degree either zero or two,

- if t is a single leaf then $\mathcal{C}(t) := 0$;
- otherwise the root of t has left and right subtrees t_0 and t_1, respectively, and we define

$$\mathcal{C}(t) := \max\{1 + \min\{\mathcal{C}(t_0), \mathcal{C}(t_1)\}, \mathcal{C}(t_0), \mathcal{C}(t_1)\} \ .$$

This complexity measure $\mathcal{C}(t)$ corresponds to the height of the largest complete binary tree embeddable in t. The definition of embedding is given below.

Definition 3. *Consider two rooted binary trees t_a and t_b. We say that t_a embeds into t_b if t_a can be transformed into t_b by a sequence of the following operations;*

- *splitting an edge (u, v) into the path of length two made by edges (u, w), (w, v) where w is a new vertex;*
- *adding edges and vertices.*

Exercise 4. Show that $\mathcal{C}(t) \geq h$ if and only if the complete binary tree of height h can be embedded into t.

Exercise 5. Consider a tree-like resolution refutation such that its underlying structure is represented by tree t. Show that when $\mathcal{C}(t) \leq h$ if and only if the proof can be represented (in the blackboard memory model) in space $h + 2$.

Corollary 6. *If a formula has a tree-like refutation of size S then it has a tree-like refutation of size S and space $O(\log S)$.*

Space lower bounds for hard formulas

Space in resolution is essentially the number of clauses that are kept in memory. If you think about CDCL solver, this roughly correspond to the size of the clause database.

[2] Pavel Pudlák, Nicola Galesi, and Neil Thapen. The space complexity of cutting planes refutations. In *CCC*, 2015

Exercise 7. Show that for every unsatisfiable CNF with n variables there is a resolution refutation of length $2^{O(n)}$ and space $n + O(1)$.

The previous exercise can legitimately cause some doubt about the necessity of studying space. After all $n + O(1)$ space requirements is not too bad when it comes to SAT solving. It means that we need to keep more or less all formula in memory. There are some catches

- A concrete SAT solver needs to know which clauses in the database need to be preserved. A refutation that requires, say, $O(\log n)$ space may be easier to find than one that requires $\Omega(n)$ space, because there is less risk to purge a useful clause;

- the proof in the above exercise has exponential size, therefore it makes sense to understand whether to obtain polynomial length it is necessary to inccrease space usage.

Atserias and Dalmau (2008)[3] prove that resolution width essentially lower bound resolution space. This immediately induces tight bounds for formulas which require large width.

Theorem 8. *Atserias and Dalmau, 2008 Consider any unsatisfiable k-CNF with a resolution refutation of space S. Then the formula has a refutation of width at most $S + k - 3$.*

Proof. Proof outline In class we will discuss the proof as presented in Filmus et al. [4].

The main idea is that we start from a refutation

$$(M_0, M_1, \ldots, M_T) \tag{2}$$

of space S and we get an "inverse" refutation of the same formula

$$(M'_T, M'_{T-1}, \ldots, M'_0) \tag{3}$$

where M'_i is the negation of M_i, espressed as a CNF. Since M_i has at most S clauses, then M'_i can be expressed as a CNF in which each clause has width at most S. Notice that since M_0 is the empty CNF (i.e. the true one), its negation is the empty DNF (i.e. the false one), which is represented in CNF form by an empty clause. Vice versa M_T contains the empty clause, so its DNF representation is the empty DNF, and its negation is the empty CNF. Therefore the M'_T and M'_0 are indeed the starting and ending configuration of a refutation, respectively.

The new sequence is not a refutation per se. We need to build connective configurations between M'_{i+1} and M'_i. These connective configuration increase the width during axiom download of at most $k - 2$, and do not increase it otherwise. Therefore the final width seems to be $S + k - 2$. Actually, thought, we can use the following observation to get the $S + k - 3$ upper bound.

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

[4] Yuval Filmus, Massimo Lauria, Mladen Mikša, Jakob Nordström, and Marc Vinyals. From small space to small width in resolution. *ACM Trans. Comput. Logic*, 16(4):28:1–28:15, July 2015

Claim 9. *It is always possible to process a resolution derivation so that the length does not change, the space does not increase, and so that after every axiom download the space is at most $S - 1$.*

Proof. Observe that after every axiom download that pushes the space to S there must be an erasure. In such case we can change the proof to do the erasure first and, if the erasure didn't delete the axiom itself from the configuration, the axiom download. □

Using the previous claim it is possible to save one from the width. □

Corollary 10. *Random k-CNF on n variables and Tseitin formulas on k-regular expander graphs over n vertices have, when unsatisfiable, resolution space complexity $\Theta(n)$.*

Exercise 11. Show that if a k-CNF formula has a tree like refutation of length S, then it is possible to find a resolution refutation in time $n^{O(\log S+k)}$.

Exercise 12. More difficult Show that if a k-CNF formula has a tree like refutation of length S, then it is possible to find a tree-like resolution refutation in time $n^{O(\log S+k)}$.

Space lower bounds and trade-offs for easy formulas

This is actually the most interesting and developed part of the theory. We won't cover much of it in this lecture, but we will spend next lecture on it. The theory is based on the concept of pebbling of a graph. This concept has been used to model space in deterministic and non-deterministic computation.

The *pebbling tautologies* are strongly connected with this concept. While so far we prove space lower bounds only for formula that require large width (and therefore that are hard), pebbling tautologies have proofs that are simultaneously

- short;
- small width;
- and nevertheless may require large space;
- or exhibit trade-off behaviors between proof length and proof space.

For an overview of this theory I suggest the survey of Nordström.[5] In next lecture we will illustrate the main ideas of the theory, and we will show some results in it.

Trade-off on Tseitin grids.

All trade-off results for pebbling formula will always go under the linear space regime. If we allow unlimited space to refute the formula, then linear space is sufficient. Nevertheless it is possible to prove trade-offs in the superpolynomial regime using Tseitin formulas over grid graphs of length ℓ and width w, where all edges are doubled edges (see Figure 1).

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

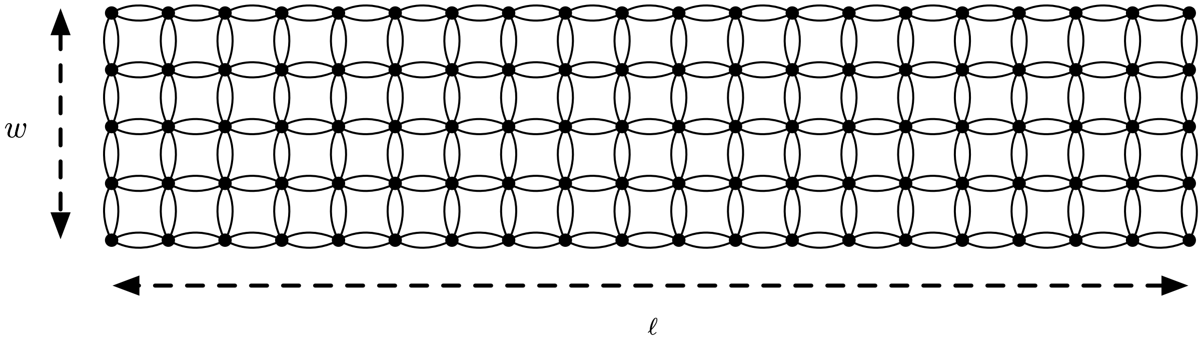

Figure 1: A grid with double edges of width w and length ℓ.

Theorem 13 (Beck, Nordström, Tang, 2013)**.** *Consider an unsatisfiable Tseitin formula over a $w \times \ell$ grid graph with double edges, with odd charge function. Let $n = O(w \times \ell)$ the size of the formula. It holds that*

- *the formula has a resolution refutation of length $O(2^w n^{O(1)})$ and clause space $2^{O(w)} + n^{O(1)}$,*
- *the formula has a tree-like resolution refutation of length $n^{O(w)}$ and clause space $O(w \log n)$,*
- *if $1 \leq w \leq n^{1/4}$, then if a resolution refutation has length L and clause space S,*

$$L = \left(\frac{2^{\Omega(w)}}{S}\right)^{\Omega\left(\frac{\log\log n}{\log\log\log n}\right)} .$$

The previous theorem was shown in Beck, Nordström, Tang (2013), is an evolution of a similar results by Beame, Beck and Impagliazzo (2012).[6]

Exercise 14. Show that the Tseitin formula over an $w \times \ell$ grid as above[7] has a resolution refutation of length $O(2^{O(w)}\ell)$ and clause space $2^{O(w)}$.
(Hint: think of summing all linear equations left to right, have at each point in time a linear equation with $O(w)$ variables in memory.)

Exercise 15. Show that the formula described above has a tree-like resolution refutation of length $\ell^{O(w)}$ and clause space $O(w \log \ell)$.
(Hint: build a decision tree of height $O(w \log \ell)$ using a divide and conquer approach. Plese provide a proof that this construction gives the desired upper bound.)

References

[ABSRW02] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Space complexity in propositional calculus. *SIAM J. Comput.*, 31(4):1184–1211, 2002.

[6] Paul Beame, Christopher Beck, and Russell Impagliazzo. Time-space tradeoffs in resolution: Superpolynomial lower bounds for superlinear space. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 213–232. ACM, 2012; and Chris Beck, Jakob Nordstrom, and Bangsheng Tang. Some trade-off results for polynomial calculus. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 813–822. ACM, 2013

[7] We can ignore the double edges in this exercise.

[AD08] Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *J. Comput. Syst. Sci.*, 74(3):323–334, 2008.

[BBI12] Paul Beame, Christopher Beck, and Russell Impagliazzo. Time-space tradeoffs in resolution: Superpolynomial lower bounds for superlinear space. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 213–232. ACM, 2012.

[BNT13] Chris Beck, Jakob Nordstrom, and Bangsheng Tang. Some trade-off results for polynomial calculus. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 813–822. ACM, 2013.

[FLM$^+$15] Yuval Filmus, Massimo Lauria, Mladen Mikša, Jakob Nordström, and Marc Vinyals. From small space to small width in resolution. *ACM Trans. Comput. Logic*, 16(4):28:1–28:15, July 2015.

[Nor13] Jakob Nordström. Pebble games, proof complexity and time-space trade-offs. *Logical Methods in Computer Science*, 9:15:1–15:63, September 2013.

[PGT15] Pavel Pudlák, Nicola Galesi, and Neil Thapen. The space complexity of cutting planes refutations. In *CCC*, 2015.

[Tor99] Jacobo Torán. Lower bounds for space in resolution. In *CSL*, pages 362–373, 1999.