Electronic Colloquium on Computational Complexity, Report No. 114 (2007)

A Simplified Way of Proving Trade-off Results for Resolution

Jakob Nordström*

Royal Institute of Technology (KTH)
SE-100 44 Stockholm, Sweden
jakobn@kth.se

September 28, 2007

Abstract

We present a greatly simplified proof of the length-space trade-off result for resolution in Hertel and Pitassi (2007), 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 possible conclusions to be drawn regarding proving trade-off results for resolution. 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. This trick is not present in the proof of Hertel and Pitassi, and thus their techniques can likely be used to prove results not obtainable by our methods.

In these notes, we present a simplification of the length-space trade-off result for resolution in [9] (soon to appear together with [8] as [10]), and show how the same ideas can be used to prove other related theorems. The simplified proof is given in Section 1. In Section 2 we prove two other trade-off results of a similar flavour. We point out two key ingredients needed for our proofs to work in Sections 3 and 4, and discuss possible conclusions to be drawn regarding proving trade-off results for resolution. Finally, in Section 5 we mention a couple of open problems that seem both natural and interesting in light of the preceding discussion. Definitions, notation and some previously known results used are given in Appendix A for reference.

1 A Proof of Hertel and Pitassi's Trade-off Result

Using the notation of Appendix A, the length-variable space trade-off theorem of Hertel and Pitassi can be expressed as follows.

Theorem 1.1 ([9]). *There is a family of CNF formulas* $\{F_n\}_{n=1}^{\infty}$ *of size* $\Theta(n^3)$ *such that:*

- *The minimal variable space of refuting* F_n *in resolution is* $\mathit{VarSp}(F_n \vdash 0) = \Theta(n)$.
- *Any resolution refutation* $\pi : F_n \vdash 0$ *in minimal variable space has length* $\exp(\Omega(n))$.
- *Adding just* 3 *extra units of storage, it is possible to obtain a resolution refutation* π' *in variable space* $\mathit{VarSp}(\pi') = \mathit{VarSp}(F_n \vdash 0) + 3 = \Theta(n)$ *and length* $L(\pi') = \mathrm{O}(n^3)$, *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 $\Theta(n)$.

The idea behind our simplified proof is as follows. Take formulas G_n that are really hard for resolution and formulas H_m which have short refutations but require linear variable space, and set $F_n = G_n \wedge H_m$ for m chosen so that $\mathit{VarSp}\big(H_m \vdash 0\big)$ is just a small constant larger

*Research supported in part by *Johan och Jakob Söderbergs stiftelse* and *Sven och Dagmar Saléns stiftelse*.

ISSN 1433-8092

than $VarSp\big(G_n \vdash 0\big)$. Then refutations in minimal variable space will have to take care of G_n, which requires exponential length, but adding one or two literals to the memory we can attack H_m instead in linear length. We write down the formal details for completeness.

Proof of Theorem 1.1. Let G_n be CNF formulas as in Theorem A.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. We know that $\Omega(n) = g(n) = \mathrm{O}\big(n^2\big)$.

Let H_m be the formulas

$$H_m = y_1 \wedge \ldots \wedge y_m \wedge (\overline{y}_1 \vee \ldots \vee \overline{y}_m) \ . \tag{1}$$

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

$$F_n = G_n \wedge H_{\lfloor g(n)/2 \rfloor + 1} \tag{2}$$

where G_n and $H_{\lfloor g(n)/2 \rfloor + 1}$ have disjoint sets of variables. By Observation A.4, any resolution refutation of F_n refutes either G_n or $H_{\lfloor g(n)/2 \rfloor + 1}$. We have

$$VarSp\big(H_{\lfloor g(n)/2 \rfloor + 1} \vdash 0\big) = 2 \cdot (\lfloor g(n)/2 \rfloor + 1) > g(n) = VarSp(G_n \vdash 0) \ , \tag{3}$$

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

Thus, we have a formula family $\{F_n\}_{n=1}^{\infty}$ of size $\Omega(n) = S(F_n) = \mathrm{O}\big(n^2\big)$ 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 1.1 (or actually slightly stronger). ☐

2 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 2.1. *There is a family of k-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta\big(n^3\big)$ such that:*

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

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

Theorem 2.2. *There is a family of k-CNF formulas $\{F_n\}_{n=1}^{\infty}$ of size $\Theta\big(n^3\big)$ such that:*

- *The minimal width of refuting F_n is $W(F_n \vdash 0) = \mathrm{O}(n)$.*
- *Any resolution refutation $\pi : F_n \vdash 0$ in minimal width must have length $L(\pi) = \exp(\Omega(n))$.*
- *There are refutations $\pi' : F_n \vdash 0$ in width $W(\pi') = \mathrm{O}\big(W(F_n \vdash 0)\big)$ and length $L(\pi') = \mathrm{O}\big(n^3\big)$, i.e., linear in the formula size.*

We only present the proof of Theorem 2.1. Theorem 2.2 is proved in exactly the same manner.

Proof of Theorem 2.1. Let G_n be a 3-CNF formula family as in Theorem A.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 A.3 of size $\Theta\bigl(m^3\bigr)$ such that $L(H_m \vdash 0) = \mathrm{O}\bigl(m^3\bigr)$ and $Sp(H_m \vdash 0) = \Theta(m)$. Define

$$g(n) = \min\bigl\{m \mid Sp(H_m \vdash 0) > Sp(G_n \vdash 0)\bigr\} \ . \tag{4}$$

Note that since $Sp(H_m \vdash 0) = \Omega(m)$ and $Sp(G_n \vdash 0) = \mathrm{O}(n)$, we know that $g(n) = \mathrm{O}(n)$.

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 A.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 $Sp\bigl(H_{g(n)} \vdash 0\bigr) > Sp(G_n \vdash 0)$, a refutation in minimal clause space has to refute G_n, which requires exponential length. However, since $g(n) = \mathrm{O}(n)$, Theorem A.3 tells us that there are refutations of $H_{g(n)}$ in length $\mathrm{O}\bigl(n^3\bigr)$ and clause space $\mathrm{O}(n)$. $\square$

3 Making the Main Trick Explicit

The proofs of the theorems in Sections 1 and 2 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 1.1, 2.1 and 2.2 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 $M_2\bigl(G_n\bigr)$ 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 $M_2\bigl(H_m\bigr)$ of the second resource,
- The index $m = m(n)$ is chosen so as to minimize $M_2\bigl(H_{m(n)}\bigr) - M_2\bigl(G_n\bigr) > 0$, i.e., so that $H_{m(n)}$ requires *just a little bit* more of the second resource than G_n.

Then for $F_n = G_n \wedge H_{m(n)}$, if we demand that a resolution refutation π 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 $M_2\bigl(H_{m(n)}\bigr) - M_2\bigl(G_n\bigr)$, we can get a refutation π' 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 $C \in F$, the remaining subformula $F \setminus \{C\}$ is satisfiable. We note that if we would add the requirement in Sections 1 and 2 that the formulas under consideration should be minimally unsatisfiable, the proof idea outlined above fails completely. In contrast, the result in [9] seems to be independent of any such conditions. What conclusions can be drawn from this?

On the one hand, trade-off results for minimally unsatisfiable formulas 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.[1] So in practice, trade-off results for non-minimal formulas might be just as interesting.

[1] 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 $L_1 \in \mathsf{NP}$ and $L_2 \in \mathsf{co\text{-}NP}$ such that $L = L_1 \cap L_2$ [12]. MINIMAL UNSATISFIABILITY is DP-complete [13], and it seems to be commonly believed that $\mathsf{DP} \nsubseteq \mathsf{NP} \cup \mathsf{co\text{-}NP}$.

4 An Auxiliary Trick for Variable Space

A second important reason why our proof of Theorem 1.1 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 fixed, it becomes much more difficult to find such examples. Although the formulas in Theorem A.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 1.1. (Note, however, that the formula families used in the proofs of Theorems 2.1 and 2.2 consist of k-CNF formulas).

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 clause space began to be studied in the late 1990s, it was soon noted in several papers (for instance [1, 4, 15]) 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

$$Sp(F \vdash 0) \geq W(F \vdash 0) - W\big(F\big) + 3 \ , \tag{5}$$

but it remained open whether space and width could be separated or the two measures were asymptotically the same. In [11], 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 $Sp(F_n \vdash 0) = \Theta(\log n)$.

However, if we are allowed to consider formulas of growing width, the fact that the inequality (5) is not tight is entirely trivial (as was remarked in [11]). Namely, let us say that a CNF formula F is *k-wide* if all clauses in F have size at least k. In [7], it was proved that for F a k-wide unsatisfiable CNF formula it holds that $Sp\big(F \vdash 0\big) \geq k + 2$. So in order to get a formula family F_n such that $W(F_n \vdash 0) - W\big(F_n\big) = \mathrm{O}(1)$ 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, the right thing to do when studying space measures in resolution seems to be to require that the formulas under study should have constant width.

In the next and final section, we propose two slightly different trade-off problems, which are phrased so as to circumvent the technical problems discussed above.

5 Two Open Trade-off Problems

In [5] it was shown that given a resolution refutation π of a k-CNF formula F in length $L(\pi) = L$, there exists a refutation in width $\mathrm{O}\big(\sqrt{n \log L}\big)$, where n is the number of variables in F. However, the refutation resulting from the proof is not the same π, but another refutation π' which is potentially exponentially longer than π. It would be interesting to know whether this increase in length is necessary, i.e., whether there is a length-width trade-off, or 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, is it true that there is always a refutation π of F in width $W(\pi) = \mathrm{O}\big(\sqrt{n \log L}\big)$ 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)?*

A similar question can be posed for clause space. Given a refutation in small space, we know that there must exist a refutation in short length. This follows by applying the upper bound (5) 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 again, the refutation we end up with is not the same as that with which we started.

For concreteness, let us fix the space to be constant. If a polynomial-size k-CNF formula has a refutation in constant clause space, then by the above reasoning it must be refutable in polynomial length. But can we get a refutation which is both short and tight simultaneously?

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 can it be that restricting the space, we end up with really long refutations? It would be interesting to know what holds in this case, and how it relates to the trade-off results for variable space in [9].

Acknowledgements

We are most grateful to Philipp Hertel and Toniann Pitassi for patiently answering several technical questions about Theorem 1.1. Also, we want to thank Jan Krajíček, Johan Håstad, Per Austrin and Mikael Goldmann for useful discussions and insightful comments.

A Definitions, Notation and Some Useful Facts

A *literal* is either a propositional logic variable x or its negation $\overline{x}$. We define $\overline{\overline{x}} = x$. A *clause* $C = a_1 \vee \ldots \vee a_k$ is a set of literals. The *width* $W(C)$ of a clause C is the number of literals appearing in it. A clause containing at most k literals is called a *k-clause*. A *CNF formula* $F = C_1 \wedge \ldots \wedge C_m$ is a set of clauses. A *k-CNF formula* is a CNF formula consisting of k-clauses. We let $Vars(C)$ denote the set of variables in a clause C, and extend this notation to formulas by taking unions over clauses. Also, the width $W(F)$ of a CNF formula F is the width of its largest clause.

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

Definition A.1 (Resolution [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 $C \vee x, D \vee \overline{x} \in \mathbb{C}_{t-1}$.

A resolution derivation $\pi : F \vdash A$ of a clause A from a CNF formula F is a derivation $\{\mathbb{C}_0, \ldots, \mathbb{C}_\tau\}$ such that $A \in \mathbb{C}_\tau$. 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 derivation π is the number of distinct clauses in π.
- The *width* $W(\pi)$ of a derivation π is the number of literals in the largest clause in π.
- The *clause space* $Sp(\pi)$ of a resolution derivation π is the maximal number of clauses in any clause configuration $\mathbb{C}_t \in \pi$.
- The *variable space* $VarSp(\pi)$ of a resolution derivation π 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 is defined wholly analogously.

We define the *size* $S(F)$ of a CNF formula F to be the numbers of literals in it, counted with repetitions.

Note that if one wanted to be really precise, the size and space measures should probably measure the number of *bits* needed rather than the number of literals. However, counting literals makes matters substantially cleaner, and the difference is at most a logarithmic factor anyway. Therefore, counting literals seems to be the established way of measuring size and variable space.

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 [7] it was proved that the clause space of refuting F is upper-bounded by the formula size. More precisely, the minimal clause space is at most the number of clauses, or the number of variables, plus a small constant, or in formal notation $Sp(F \vdash 0) \leq \min\{|F|, |\mathit{Vars}(F)|\} + \mathrm{O}(1)$.

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 A.2 ([1, 3, 4, 5, 15, 16]). *There are arbitrarily large unsatisfiable 3-CNF formulas F_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 $Sp(F_n \vdash 0) = \Theta(n)$.*

Clearly, for these formulas F_n it also holds that $\Omega(n) = \mathit{VarSp}(F_n \vdash 0) = \mathrm{O}(n^2)$. We note in passing that the exact variable space complexity 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.[2]

Theorem A.3 ([1, 6, 14]). *There are arbitrarily large unsatisfiable 3-CNF formulas F_n in 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 there are resolution refutations $\pi_n : F_n \vdash 0$ 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 A.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. □

References

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

[2] Albert Atserias and Victor Dalmau. A combinatorical characterization of resolution width. In *Proceedings 18th IEEE Annual Conference on Computational Complexity (CCC '03)*, pages 239–247, July 2003.

[3] Paul Beame, Richard Karp, Toniann Pitassi, and Michael Saks. The efficiency of resolution and Davis-Putnam procedures. *SIAM Journal on Computing*, 31(4):1048–1075, 2002.

[4] Eli Ben-Sasson and Nicola Galesi. Space complexity of random formulae in resolution. *Random Structures and Algorithms*, 23(1):92–109, August 2003.

[5] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow—resolution made simple. *Journal of the ACM*, 48(2):149–169, March 2001.

[6] Maria Luisa Bonet and Nicola Galesi. A study of proof search algorithms for resolution and polynomial calculus. In *Proceedings 40th Annual IEEE Symposium on Foundations of Computer Science (FOCS '99)*, pages 422–431, October 1999.

[7] Juan Luis Esteban and Jacobo Torán. Space bounds for resolution. *Information and Computation*, 171(1):84–97, 2001.

[2] Note that [6], 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 $\mathrm{O}(n)$.

[8] Philipp Hertel and Toniann Pitassi. *Black-White Pebbling is PSPACE-Complete*. Technical Report TR07-044, Electronic Colloquium on Computational Complexity (ECCC), May 2007.

[9] Philipp Hertel and Toniann Pitassi. *An Exponential Time/Space Speedup For Resolution*. Technical Report TR07-046, Electronic Colloquium on Computational Complexity (ECCC), May 2007.

[10] Philipp Hertel and Toniann Pitassi. Exponential time/space speedups for resolution and the PSPACE-completeness of black-white pebbling. In *Proceedings 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS '07)*, October 2007. To appear.

[11] Jakob Nordström. Narrow proofs may be spacious: Separating space and width in resolution (Extended abstract). In *Proceedings 38th Annual ACM Symposium on Theory of Computing (STOC '06)*, pages 507–516, May 2006.

[12] Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[13] Christos H. Papadimitriou and David Wolfe. The complexity of facets resolved. *Journal of Computer and System Sciences*, 37(1):2–13, 1988.

[14] Gunnar Stålmarck. Short resolution proofs for a sequence of tricky formulas. *Acta Informatica*, 33(3):277–280, May 1996.

[15] Jacobo Torán. Lower bounds for space in resolution. In *Proceedings 13th International Workshop Computer Science Logic (CSL '99)*, volume 1683 of *Lecture Notes in Computer Science*, pages 362–373. Springer, 1999.

[16] Alasdair Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, January 1987.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/