Electronic Colloquium on Computational Complexity, Report No. 44 (2003)

# A Combinatorial Characterization of Treelike Resolution Space

Juan Luis Esteban
Dept. L.S.I.
U. Politècnica de Catalunya
c/ Jordi Girona Salgado 1–3
E08034 Barcelona, Spain
esteban@lsi.upc.es

Jacobo Torán
Theoretische Informatik
Universität Ulm
Oberer Eselsberg
D-89069 Ulm, Germany
toran@informatik.uni-ulm.de

May 12, 2003

**Abstract**

We show that the Player-Adversary game from [PI00] played over CNF propositional formulas gives an exact characterization of the space needed in treelike resolution refutations. This characterization is purely combinatorial and independent of the notion of resolution. We use this characterization to give for the first time a separation between the space needed in tree-like and general resolution.

## 1 Introduction

Robinson introduced in [Rob65] the concept of *resolution*, a refutation proof system for propositional formulas in conjunctive normal form (CNF). The only inference rule in this proof system is the resolution rule:

$$\frac{C \vee x \qquad D \vee \bar{x}}{C \vee D} .$$

Cutting variable $x$ from clauses $C \vee x$ and $D \vee \bar{x}$ we get the *resolvent* clause $C \vee D$. A resolution refutation of a CNF formula $\varphi$ is a sequence of clauses $C_1, \ldots, C_s$ where each $C_i$ is either a clause from $\varphi$ or is inferred from earlier clauses by the resolution rule, and $C_s$ is the empty clause. We will denote the empty clause by $\lambda$. A resolution refutation can be seen as directed acyclic graph, a *dag*, in which the clauses are the vertices, and if two clauses are resolved then there is a directed edge going from each one of the two clauses to the resolvent. If the underlying graph in a refutation happens to be a tree, we talk about *treelike* resolution. It is known that for certain formulas general resolution can produce shorter refutations than treelike resolution [BEGJ01, BSIW00]. The reason for this is that, contrary to general resolution, in



ISSN 1433-8092

treelike resolution if a clause is needed more than once it must be re-derived from the initial clauses each time.

Due to its simplicity and to its relevance in automatic theorem proving and logic programming systems, resolution is one of the best studied refutation systems and several ways to measure the complexity of a resolution refutation have been proposed. The best studied complexity measure is the *size*. The size of a refutation is the number of clauses it contains. It is well known that certain families of propositional formulas need resolution refutations with a number of clauses that is exponential in the formula size [Hak85, Urq87, CS88, BP96].

Because of the importance of resolution, other measures for the complexity of such refutations have been introduced. Ben-Sasson and Wigderson [BSW01], building on previous work [BP96, CEI96] defined the concept of *width*. The width of a resolution refutation is the maximal number of literals in any clause of the refutation. The resolution width of a formula is the minimal width among all refutations of the formula. Ben-Sasson and Wigderson show that lower bound on the width can be used for proving lower bounds on the resolution size of certain formulas.

Another natural complexity measure is the *space*. Intuitively the resolution space of a CNF formula is the minimal number of clauses that must be kept simultaneously in order to refute a formula. The formal definition [ET01],[ABSRW02] is the following:

**Definition 1.1** *Let $k \in \mathbb{N}$, we say that an unsatisfiable CNF formula $\varphi$ has resolution refutation bounded by space $k$ if there is a series of CNF formulas $\varphi_1, \ldots, \varphi_s$, such that $\varphi_1 \subseteq \varphi$, $\lambda \in \varphi_s$, in any $\varphi_i$ there are at most $k$ clauses, and for each $i < s$, $\varphi_{i+1}$ is obtained from $\varphi_i$ by:*

*1) Deleting a clause from $\varphi_i$.*

*2) Adding the resolvent of two clauses from $\varphi_i$.*

*3) Adding a clause from $\varphi$ (initial clause).*

The space needed for the resolution of an unsatisfiable formula is the minimum $k$ for which the formula has a refutation bounded by space $k$. Note that initial clauses do not take much space because they can be added at any moment and at most two of them are needed simultaneously. The only clauses that consume space are the ones derived at intermediate stages. In [ET01, ABSRW02] it is shown that resolution refutations for certain families of formulas need linear space. It was observed in [ET01] that the space required for the resolution refutation of a CNF formula $\varphi$, corresponds to the minimum number of pebbles needed in the following game played on the graph of a refutation of $\varphi$.

**Definition 1.2** *Given a connected directed acyclic graph with one sink the aim of the pebble game is to put a pebble on the sink of the graph, the only node with no outgoing edges, following this set of rules:*

1) *A pebble can be placed in any initial node, that is, a node with no predecessors.*

2) *Any pebble can be removed from any node at any time.*

3) *A node can be pebbled provided all its parent nodes are pebbled.*

3') *If all the parent nodes of node are pebbled, instead of placing a new pebble on it, one can shift a pebble from a parent node.*

**Lemma 1.3** *([ET01]) Let $\varphi$ be an unsatisfiable CNF formula. The space needed in a resolution refutation of $\varphi$ coincides with the number of pebbles needed for the pebble game played on the graph of a resolution refutation of $\varphi$.*

In this paper we consider the restricted case of space in treelike resolution refutations and show that this complexity measure can be exactly characterized in terms of a two-person combinatorial game introduced by Impagliazzo and Pudlák in [PI00]. This game was used for proving lower bounds on the size of treelike resolution refutations [PI00, BSIW00]. We then use the characterization to give a separation between the space needed in treelike and general resolution. Although it is known that families of formulas exist for which there is an exponential separation between the sizes of their general and treelike resolution refutations [BEGJ01, BSIW00], a separation between these two types of resolution for the space measure was not known. We present in Section 3 the first such separation. We give a family $\{F_n\}$ of formulas satisfying that $F_n$ requires treelike resolution refutations of space $n-2$ but has general refutation of space at most $\frac{2}{3}n+3$.

**The combinatorial game:**

The game is played in rounds on an unsatisfiable formula $\varphi$ in $CNF$ by two players: Prover and Delayer. Prover wants to falsify some initial clause and Delayer tries to retard this as much as possible. In each round Prover chooses a variable in $\varphi$ and asks Delayer for a value for this variable. Delayer can answer either 0,1 or $*$. In this last case Prover can choose the truth value (0 or 1) for the variable and Delayer scores one point. The variable is set to the selected value and the next round begins. The game ends when a clause in $\varphi$ is falsified (all its literals are set to 0) by the partial assignment constructed this way. The goal of Delayer is to score as many points as possible and Prover tries to prevent this. The outcome of the game is the number of points scored by Delayer.

**Definition 1.4** *Let $\varphi$ be an unsatisfiable formula in CNF. We denote by $g(\varphi)$ the maximum number of points that Delayer can score while playing the game on $\varphi$ with an optimal strategy of Prover.*

Our main result shows that for an unsatisfiable CNF formula $\varphi$, the space needed in a treelike resolution refutation of $\varphi$ is exactly $g(\varphi)+1$. Observe that the outcome of the combinatorial game depends only on the structure of $\varphi$. This characterization of treelike resolution space is therefore completely independent of the notion of resolution. We use the characterization and the relations from space and size in treelike resolution refutation to slightly improve a lower bound for the treelike resolution size in terms of the points scored in the combinatorial game from [PI00].

Atserias and Dalmau have given recently [AD02] a combinatorial characterization of resolution width that also depends only on the structure of the formula being considered. These two results point out the naturalness of resolution and its space and width complexity measures.

## 2 The Characterization

We show that for an unsatisfiable CNF formula $\varphi$, the number of points that Delayer can score while playing the game on $\varphi$ provides both an upper and a lower bound on the treelike resolution space of $\varphi$.

We show first that $g(\varphi)+1$ is an upper bound for the treelike resolution space.

**Theorem 2.1** *If a $CNF$ formula $\varphi$ requires treelike resolution space $S$, then Delayer has a strategy in which at least $S-1$ points can be scored, that is, $S-1 \leq g(\varphi)$.*

**Proof.** Let be $S$ the minimum space needed in any treelike resolution refutation of $\varphi$. We give a strategy for Delayer for playing the combinatorial game on $\varphi$ that scores at least $S-1$ points with any strategy of Prover. We prove the result by induction on the number of variables in $\varphi$, $n$.

For the base case $n=1$, $\varphi$ contains just one variable and therefore $S \leq 2$. Delayer just needs to answer $*$ to the only variable asked by Prover.

For $n>1$, let $x$ be the first variable asked by Prover and let $\varphi_{x=1}$ and $\varphi_{x=0}$ the $CNF$ formulas obtained after given value 1 and 0 respectively to variable $x$ in $\varphi$. Any treelike refutation of $\varphi$ requires $S$ pebbles and therefore either

i) any treelike space for refuting each of $\varphi_{x=1}$ and $\varphi_{x=0}$ is at least $S-1$ or

ii) for one of the formulas (say $\varphi_{x=1}$) the treelike resolution space is at least $S$.

Any other possibility would imply that $\varphi$ could be refuted in space less than $S$.

In the first case Delayer can answer $*$ and scores one point. By induction hypothesis Delayer can score $S-2$ more points playing the game in any of the formulas $\varphi_{x=1}$ or $\varphi_{x=0}$. In the second case Delayer answers the value leading to the formula that requires treelike resolution space $S$ ($x=1$ in this case) and the game is played on $\varphi_{x=1}$ in the next round. ■

On the other hand $g(\varphi)$ is also a lower bound for the treelike resolution space. Let us consider a resolution refutation of $\varphi$, $R$, and suppose that Prover and Delayer play the game on $\varphi$. Delayer follows a strategy scoring at least $g(\varphi)$ points and Prover chooses the variables in an order induced by the refutation in the following way: Prover starts at the empty clause in $R$ and in general at the end of a round moves to a clause $C$. In the next round Prover chooses the resolved variable $x$ from the two parent clauses of $C$. If Delayer assigns to $x$ a value 0 or 1 then Prover moves to the parent clause that is falsified by the partial assignment and the new round starts. If Delayer assigns $x$ value $*$ then Prover can choose value 0 or 1 for $x$ and moves to the parent clause falsified by the chosen partial assignment. In this case we mark the clause with $*$. The game ends when Prover can move to an initial clause.

For a refutation $R$ let us denote by game($R$) the subgraph of $R$ formed by all the clauses that can be visited by Prover and the edges joining them in the described game (with a strategy from Delayer scoring at least $g(\varphi)$ points). We show that the pebble game played on game($R$) needs at least $g(\varphi)+1$ pebbles. Since game($R$) is a subgraph of $R$, by Lemma 1.3 this implies that treelike space for $\varphi$ is at least $g(\varphi)+1$.

**Theorem 2.2** *The treelike space needed for refuting a CNF $\varphi$ is at least $g(\varphi)+1$.*

**Proof.** Let $R$ be a treelike resolution refutation of $\varphi$. game($R$) is also a tree and in any path from the empty clause to an initial clause in game($R$) there are at least $g(\varphi)$ nodes marked with $*$ (branching nodes). We will show that game($R$) requires at least $g(\varphi)+1$ pebbles. This implies the result since game($R$) is a subgraph of $R$.

Consider any strategy for pebbling the tree game($R$), and consider the first moment $s$ in which all the paths going from an initial clause to the empty clause contain a pebble. After moment $s-1$ a pebble has to be placed on an initial clause $C$, and before that, the path going from $C$ to the empty clause is the only path without pebbles. This path contains at least $g(\varphi)$ nodes marked with *. In each one of these nodes starts a path going to an initial clause. All these paths are disjoint and they all contain a pebble at instant $s-1$ (otherwise there would be at moment $s$ a path from the empty clause to some initial clause without any pebble). Together with the pebble at moment $s$, this makes at least $g(\varphi)+1$ pebbles. ■

As mentioned in the introduction, the combinatorial game was defined in [PI00] as a tool for proving lower bound for the size of treelike resolution refutation. Impagliazzo and Pudlák prove the following result:

**Theorem 2.3** *[PI00] If Delayer has a strategy on a formula $\varphi$ which scores $r$ points then any treelike resolution refutation of $\varphi$ has size at least $2^r$.*

Based on the relations between size and space in treelike resolution refutations and the above characterization, we can slightly improve this result by a factor of two. For this the following result from [ET01] is needed:

**Theorem 2.4** *If a CNF formula requires space $s$ then it requires treelike resolution refutations of size at least $2^s-1$.*

Together with the combinatorial characterization of treelike resolution space this implies:

**Corollary 2.5** *For any unsatisfiable CNF formula $\varphi$, if Delayer has a strategy on $\varphi$ which scores $r$ points then any treelike resolution refutation of $\varphi$ has size at least $2^{r+1} - 1$.*

# 3 A separation between treelike and general resolution space

We present in this section a family of formulas that require more space when refuted using treelike resolution than when this is done with general resolution. The formulas are a particular case of the the pebbling contradictions introduced in [BSIW00]. These are based on the pebbling game and are defined in the following way:

**Definition 3.1** *Let $G = (V, E)$ be a directed acyclic graph in which every node has in-degree 0 or 2 and has a unique node with out-degree 0. $P(G)$ denotes the pebbling formula based on $G$. For every node $v \in V$ $P(G)$ contains the variables $v_0$ and $v_1$. $P(G)$ defined as the conjunction of the following clauses:*

*i) A source node $s$ in $G$ (a node with no incoming edges) has associated the* source clause $s_0 s_1$.

*ii) The target node $t$ (the node without outgoing edges) has the two* target clauses *$\bar{t}_0$ and $\bar{t}_1$ associated to it.*

*iii) Any nonsource node $w$ with parent nodes $u$ and $v$ has four* pebbling clauses *associated: $\bar{u}_0\bar{v}_0 w_0 w_1$, $\bar{u}_0\bar{v}_1 w_0 w_1$, $\bar{u}_1\bar{v}_0 w_0 w_1$ and $\bar{u}_1\bar{v}_1 w_0 w_1$.*

It is not hard to see that for any directed acyclic graph $G$ (with the required degree condition) $P(G)$ is a contradiction.

Let $T_n$ denote the complete binary tree with $n$ levels. We give an upper bound for the space required to resolve $P(T_n)$ in general resolution.

For the proof of this result we use the following notation: for a formula $\varphi$ and a clause $C$ $\varphi \vdash^s C$ means that $C$ can be derived from $\varphi$ using resolution space at most $s$.

**Lemma 3.2** *For $n \geq 5$, if $P(T_{n-3}) \vdash^{s-2} \lambda$, $P(T_{n-2}) \vdash^{s-1} \lambda$ and $P(T_{n-1}) \vdash^s \lambda$ then $P(T_n) \vdash^s \lambda$.*

**Proof.** We give a resolution strategy for refuting $P(T_n)$ measuring the space needed. The set of clauses kept at each stage in the refutation can be seen in Tables 2 and 3. The variables names follow the schematic representation of $T_n$ in Figure 1. Since $P(T_{n-1}) \vdash^s \lambda$ it follows that $P(T_n) \vdash^s b_0 b_1$. This is because all the clauses in $P(T_{n-1})$ occurs in $P(T_n)$ except for clauses $\bar{b}_0$ and $\bar{b}_1$. Similarly, since $P(T_{n-2}) \vdash^{s-1} \lambda$ it is also clear that $P(T_n) \vdash^{s-1} d_0 d_1$. So we can derive the two clauses $b_0 b_1$ and $d_0 d_1$ using space $s$ by first deriving $b_0 b_1$ in space $s$, keeping it, and then deriving $d_0 d_1$. h The maximum amount of space used until this point is $s$.

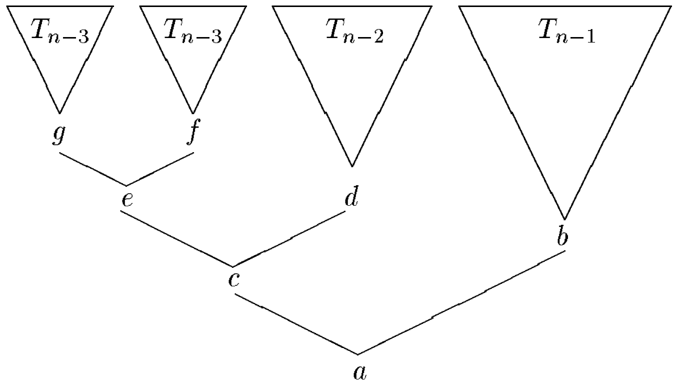


Figure 1: Complete tree $T_n$

From clauses $\bar{a}_0$, $\bar{a}_1$, the pebbling clauses for $a$ (which are initial clauses) and clause $b_0b_1$, we can derive using constant space 3 $\bar{c}_0$ and $\bar{c}_1$. This means that from the stage with the clauses $d_0d_1$ and $b_0b1$ we can derive $d_0d_1$ $\bar{c}_0$ and $\bar{c}_1$ using space 4 (Table 2).

Now from $d_0d_1$, $\bar{c}_0$, $\bar{c}_1$ and the pebbling clauses for $c$ we get in space 5 $\bar{e}_0$ and $\bar{e}_1$. The derivation is very similar to that in Table 2, but now clauses $\bar{c}_0$ and $\bar{c}_1$ must be kept in memory as they are not initial clauses. The detailed derivation is in Table 3.

Since $P(T_{n-3}) \vdash^{s-2} \lambda$ it follows that $P(T_n) \vdash^{s-2} f_0f_1$. During this derivation we have to keep $\bar{e}_0$ and $\bar{e}_1$, so the maximum amount of space used is $s$. From $f_0f_1$, $\bar{e}_0$, $\bar{e}_1$ and the pebbling clauses for $e$ we get $\bar{g}_0$ and $\bar{g}_1$ in space 5 as in Table 3. Again as $P(T_{n-3}) \vdash^{s-2} \lambda$ it follows clear that $P(T_n) \vdash^{s-2} g_0g_1$. From $g_0g_1$, $\bar{g}_0$ and $\bar{g}_1$ we derive $\lambda$ in space 3. ■

From this results follows the upper bound for the resolution space of $P(T_n)$.

**Corollary 3.3** *For every $n$, $P(T_n)$ has a resolution refutation with space at most $2n/3+3$.*

**Proof.** The result follows from the fact that for $n = 2 \mod 3$, $P(T_n)$ has a refutation with space at most $2(n+1)/3+1$. We prove this by induction on $n$. The base case $n = 2$ is clear since it is easy to check that $P(T_2)$ has resolution refutations of space 3. It also holds that for any $n$, $P(T_{n+1})$ requires space at most $s+1$ if $P(T_n)$ can be refuted using space $s$. For the induction step, let us suppose that $n = 2 \mod 3$. By induction hypothesis the space needed for $P(T_{n-3})$ is at most $2(n-2)/3+1$. Using the above property we get that the space needed for $(T_{n-2})$ and for $(T_{n-1})$ respectively at most $2(n-2)/3+2$ and $2(n-2)/3+3 = 2(n+1)/3+1$. By the above lemma $P(T_n)$ requires also at most space $2(n+1)/3+1$. ■

| | | | |
|---|---|---|---|
| $d_0 d_1$ | $b_0 b_1$ | | |
| $d_0 d_1$ | $b_0 b_1$ | $\bar{c}_0 \bar{b}_0 a_0 a_1$ | |
| $d_0 d_1$ | $b_0 b_1$ | $\bar{c}_0 b_1 a_0 a_1$ | |
| $d_0 d_1$ | $b_0 b_1$ | $\bar{c}_0 b_1 a_0 a_1$ | $\bar{c}_0 \bar{b}_1 a_0 a_1$ |
| $d_0 d_1$ | $b_0 b_1$ | $\bar{c}_0 a_0 a_1$ | |
| $d_0 d_1$ | $b_0 b_1$ | $\bar{c}_0 a_0 a_1$ | $\bar{a}_0$ |
| $d_0 d_1$ | $b_0 b_1$ | $\bar{c}_0 a_1$ | |
| $d_0 d_1$ | $b_0 b_1$ | $\bar{c}_0 a_1$ | $\bar{a}_1$ |
| $d_0 d_1$ | $b_0 b_1$ | $\bar{c}_0$ | |
| $d_0 d_1$ | $b_0 b_1$ | $\bar{c}_0$ | $\bar{c}_1 \bar{b}_0 a_0 a_1$ |
| $d_0 d_1$ | $\bar{c}_1 b_1 a_0 a_1$ | $\bar{c}_0$ | |
| $d_0 d_1$ | $\bar{c}_1 b_1 a_0 a_1$ | $\bar{c}_0$ | $\bar{c}_1 \bar{b}_1 a_0 a_1$ |
| $d_0 d_1$ | $\bar{c}_1 a_0 a_1$ | $\bar{c}_0$ | |
| $d_0 d_1$ | $\bar{c}_1 a_0 a_1$ | $\bar{c}_0$ | $\bar{a}_0$ |
| $d_0 d_1$ | $\bar{c}_1 a_1$ | $\bar{c}_0$ | |
| $d_0 d_1$ | $\bar{c}_1 a_1$ | $\bar{c}_0$ | $\bar{a}_1$ |
| $d_0 d_1$ | $\bar{c}_1$ | $\bar{c}_0$ | |

Table 1: Clauses kept in memory during the resolution derivation of $\bar{c}_1$ and $\bar{c}_0$

On the other hand in the case of treelike resolution, the space needed in a refutation of $P(T_n)$ is at most $n-2$. This follows our characterization of resolution space in treelike resolution together with the lower bound obtained in [BSIW00] on the number on points obtained by Delayer's when playing the combinatorial game on the pebbling formulas. We just need the particular case of this result for complete trees.

**Theorem 3.4** *[BSIW00] For every $n$ Delayer has a strategy in which at least $n-2$ points can be scored, when playing the combinatorial game on $P(T_n)$.*

**Corollary 3.5** *For every $n$, the space needed in a treelike resolution refutation of $P(T_n)$ is at least $n-2$.*

# 4 Conclusions and open problems

We have given an exact characterization of the space required in resolution refutations of a CNF formula based on a purely combinatorial game and independent of the resolution method. We also have shown a separation between the space needed in treelike and general resolution of a particular class of formulas. It remains open whether the characterization can be adapted to capture the space complexity in general resolution (without the treelike restriction). This could help to answer the question of whether there are families of formulas that have resolution refutations of

| | | | | |
|---|---|---|---|---|
| $\bar{c}_1$ | $\bar{c}_0$ | $d_0 d_1$ | | |
| $\bar{c}_1$ | $\bar{c}_0$ | $d_0 d_1$ | $\bar{e}_0 \bar{d}_0 c_0 c_1$ | |
| $\bar{c}_1$ | $\bar{c}_0$ | $d_0 d_1$ | $\bar{e}_0 d_1 c_0 c_1$ | |
| $\bar{c}_1$ | $\bar{c}_0$ | $d_0 d_1$ | $\bar{e}_0 d_1 c_0 c_1$ | $\bar{e}_0 \bar{d}_1 c_0 c_1$ |
| $\bar{c}_1$ | $\bar{c}_0$ | $d_0 d_1$ | $\bar{e}_0 c_0 c_1$ | |
| $\bar{c}_1$ | $\bar{c}_0$ | $d_0 d_1$ | $\bar{e}_0 c_0$ | |
| $\bar{c}_1$ | $\bar{c}_0$ | $d_0 d_1$ | $\bar{e}_0$ | |
| $\bar{c}_1$ | $\bar{c}_0$ | $d_0 d_1$ | $\bar{e}_0$ | $\bar{e}_1 \bar{d}_0 c_0 c_1$ |
| $\bar{c}_1$ | $\bar{c}_0$ | $\bar{e}_1 d_1 c_0 c_1$ | $\bar{e}_0$ | |
| $\bar{c}_1$ | $\bar{c}_0$ | $\bar{e}_1 d_1 c_0 c_1$ | $\bar{e}_0$ | $\bar{e}_1 \bar{d}_1 c_0 c_1$ |
| $\bar{c}_1$ | $\bar{c}_0$ | $\bar{e}_1 c_0 c_1$ | $\bar{e}_0$ | |
| $\bar{c}_1$ | $\bar{e}_1 c_1$ | | $\bar{e}_0$ | |
| $\bar{e}_1$ | | | $\bar{e}_0$ | |

Table 2: Clauses kept in memory during the resolution derivation of $\bar{e}_1$ and $\bar{e}_0$

small size but require a large amount of space, a question proposed by Ben-Sasson in [Ben02]. We conjecture that the Pebbling Formulas are an example of a family with this property. These formulas have small resolution size [BSW01] and as we have seen require a large amount of space in treelike refutations.

# References


[ABSRW02] M. Alekhnovich, E. Ben-Sasson, A.A. Razborov, and A. Wigderson. Space complexity in propositional calculus. *SICOMP: SIAM Journal on Computing*, 31(4):1184–1211, 2002.

[AD02] A. Atserias and V. Dalmau. A combinatorial characterization of resolution width. Electronic Colloquium on Computational Complexity, Report TR02-035, 2002.

[BEGJ01] M.L. Bonet, J.L. Esteban, N. Galesi, and J. Johannsen. On the relative complexity of resolution refinements and cutting planes proof systems. *SIAM Journal on Computing*, 30(5):1462–1484, October 2001.

[Ben02] E. Ben-Sasson. Size space tradeoffs for resolution. In ACM, editor, *Proceedings of the thirty-fourth annual ACM Symposium on Theory of Computing: Montreal, 2002*, pages 457–464, New York, NY, USA, 2002. ACM Press.

[BP96] P. Beame and T. Pitassi. Simplified and improved resolution lower bounds. In *37th Annual Symposium on Foundations of Computer Science*, pages 274–282, Burlington, Vermont, 14–16 October 1996. IEEE.

[BSIW00] E. Ben-Sasson, R. Impagliazzo, and A. Wigderson. Near-optimal separation of treelike and general resolution. In *Electronic Colloquium on Computational Complexity, Report TR02-005*, 2000. To appear in Combinatorica.

[BSW01] E. Ben-Sasson and A. Wigderson. Short proofs are narrow — resolution made simple. *Journal of the ACM*, 48(2):149–169, 2001.

[CEI96] M. Clegg, J. Edmonds, and R. Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, pages 174–183, Philadelphia, Pennsylvania, 22–24 May 1996.

[CS88] V. Chvátal and E. Szemerédi. Many hard examples for resolution. *Journal of the ACM*, 35(4):759–768, October 1988.

[ET01] J.L. Esteban and J. Torán. Space bounds for resolution. *Information and Computation*, 171(1):84–97, 2001.

[Hak85] A. Haken. The intractability of resolution. *Theoretical Computer Science*, 39(2-3):297–308, August 1985.

[PI00] P. Pudlák and R. Impagliazzo. A lower bound for DLL algorithms for k-SAT. In *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 128–136, N.Y., January 9–11 2000. ACM Press.

[Rob65] J.A. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, January 1965.

[Urq87] A. Urquhart. Hard examples for resolution. *Journal of the ACM*, 34(1):209–219, January 1987.



ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'