A Lower Bound for the Pigeonhole Principle in Tree-like Resolution by Asymmetric Prover-Delayer Games

Olaf Beyersdorff [1] * Nicola Galesi [2] † Massimo Lauria [2]

[1] *Institut für Theoretische Informatik, Leibniz Universität Hannover, Germany*
[2] *Dipartimento di Informatica, Sapienza Università di Roma, Italy*

Abstract

In this note we show that the asymmetric Prover-Delayer game developed in (ECCC, TR10–059) for Parameterized Resolution is also applicable to other tree-like proof systems. In particular, we use this asymmetric Prover-Delayer to show a lower bound of the form $2^{\Omega(n \log n)}$ for the pigeonhole principle in tree-like Resolution. This gives a new and simpler proof of the same lower bound established by Dantchev and Riis (CCC, 2001).

1 Introduction

Proving lower bounds by games is a very fruitful technique in proof complexity [1,8–10]. In particular, the Prover-Delayer game of Pudlák and Impagliazzo [10] is one of the canonical tools to study lower bounds in tree-like Resolution [2, 10] and tree-like $Res(k)$ [6]. The Prover-Delayer game of Pudlák and Impagliazzo arises from the well-known fact [7] that a tree-like Resolution proof for a formula F can be viewed as a decision tree which solves the search problem of finding a clause of F falsified by a given assignment. In the game, Prover queries a variable and Delayer either gives it a value or leaves the decision to Prover and receives *one* point. The number of Delayer's points at the end of the game is then proportional to the height of the proof tree. It is easy to argue that showing lower bounds by this game only works if (the graph of) every tree-like Resolution refutation contains a balanced sub-tree as a minor, and the height of that sub-tree then gives the size lower bound.

In [3] we developed a new asymmetric Prover-Delayer game which extends the game of Pudlák and Impagliazzo to make it applicable to obtain lower bounds to tree-like proofs when the proof trees are very unbalanced. In [3] we used the new asymmetric

*This work was done while the first author was visiting Sapienza University of Rome under support of DFG grant KO 1053/5–2.

†Supported by grant "Limiti di compressione in combinatoria e complessità computazionale" by Sapienza University Rome.

game to obtain lower bounds in tree-like Parameterized Resolution, a proof system in the context of parameterized proof complexity recently introduced by Dantchev, Martin, and Szeider [4]. The lower bounds we obtain in [3] for tree-like Parameterized Resolution are of the form $\Omega(n^k)$ (n is the formula size and k the parameter), but the tree-like Parameterized Resolution refutations of the formulas in question only contain balanced sub-trees of height k.

The aim of this note is to show that the asymmetric Prover-Delayer game is also applicable to other (non-parameterized) tree-like proof systems. One of the best studied principles is the pigeonhole principle. Dantchev and Riis [5] show that the pigeonhole principle requires tree-like Resolution refutations of size roughly $n!$ while its tree-like Resolution proofs only contain balanced sub-trees of height n. Therefore the game of Pudlák and Impagliazzo only yields a $2^{\Omega(n)}$ lower bound which is weaker than the optimal bound $2^{\Omega(n \log n)}$ established by Dantchev and Riis. Here we provide a new and easier proof of this lower bound by our asymmetric Prover-Delayer game.

2 Preliminaries

A *literal* is a positive or negated propositional variable and a *clause* is a set of literals. A clause is interpreted as the disjunctions of its literals and a set of clauses as the conjunction of the clauses. Hence clause sets correspond to formulas in CNF. The *Resolution system* is a refutation system for the set of all unsatisfiable CNF. Resolution uses as its only rule the *Resolution rule*

$$\frac{\{x\} \cup C \qquad \{\neg x\} \cup D}{C \cup D}$$

for clauses C, D and a variable x. The aim in Resolution is to demonstrate unsatisfiability of a clause set by deriving the empty clause. If in a derivation every derived clause is used at most once as a prerequisite of the Resolution rule, then the derivation is called *tree-like*, otherwise it is *dag-like*. The *size* of a Resolution proof is the number of its clauses. Undoubtedly, Resolution is the most studied and best-understood propositional proof system (cf. [11]).

It is well known (cf. [7]) that a tree-like refutation of F can equivalently be described as a *boolean decision tree*. A boolean decision tree for F is a binary tree where inner nodes are labeled with variables from F and leafs are labeled with clauses from F. Each path in the tree corresponds to a partial assignment where a variable x gets value 0 or 1 according to whether the path branches left or right at the node labeled with x. The condition on the decision tree is that each path α must lead to a clause which is falsified by the assignment corresponding to α. Therefore, a boolean decision tree solves the *search problem* for F which, given an assignment α, asks for a clause from F falsified by α. It is easy to verify that each tree-like Resolution refutation of F yields a boolean decision tree for F and vice versa, where the size of the Resolution proof equals the number of nodes in the decision tree. In the sequel, we will therefore concentrate on boolean decision trees to prove our lower bound to tree-like Resolution.

3 Tree-like Lower Bounds via Asymmetric Prover-Delayer Games

We review the asymmetric Prover-Delayer game from [3]. Let F be a set of clauses in n variables $x_1, \dots, x_n$. In the asymmetric game, Prover and Delayer build a (partial) assignment to $x_1, \dots, x_n$. The game is over as soon as the partial assignment falsifies a clause from F. The game proceeds in rounds. In each round, Prover suggests a variable x_i, and Delayer either chooses a value 0 or 1 for x_i or leaves the choice to the Prover. In this last case, if the Prover sets the value, then the Delayer gets some points. The number of points Delayer earns depends on the variable x_i, the assignment α constructed so far in the game, and two functions $c_0(x_i, \alpha)$ and $c_1(x_i, \alpha)$. More precisely, the number of points that Delayer will get is

$$\begin{array}{cl} 0 & \text{if Delayer chooses the value,} \\ \log c_0(x_i, \alpha) & \text{if Prover sets } x_i \text{ to 0, and} \\ \log c_1(x_i, \alpha) & \text{if Prover sets } x_i \text{ to 1.} \end{array}$$

Moreover, the functions $c_0(x, \alpha)$ and $c_1(x, \alpha)$ are chosen in such a way that for each variable x and assignment α

$$\frac{1}{c_0(x, \alpha)} + \frac{1}{c_1(x, \alpha)} = 1 \qquad (1)$$

holds. Let us call this game the (c_0, c_1)-game on F.

The connection of this game to size of proofs in tree-like Resolution is given by Theorem 1. The theorem is essentially contained in [3], but for completeness we include the full proof.

Theorem 1 ([3]). *Let F be unsatisfiable formula in CNF and let c_0 and c_1 be two functions satisfying (1) for all partial assignments α to the variables of F. If F has a tree-like Resolution refutation of size at most S, then the Delayer gets at most* $\log S$ *points in each (c_0, c_1)-game played on F.*

As remarked in [3] we get the game of Pudlák and Impagliazzo [10] by setting $c_0(x, \alpha) = c_1(x, \alpha) = 2$ for all variables x and partial assignments α.

4 Tree-like Resolution Lower Bounds for the Pigeonhole Principle

The *weak pigeonhole principle* PHP^m_n with $m > n$ uses variables $x_{i,j}$ with $i \in [m]$ and $j \in [n]$, indicating that pigeon i goes into hole j. PHP^m_n consists of the clauses

$$\bigvee_{j \in [n]} x_{i,j} \quad \text{for all pigeons } i \in [m]$$

and $\neg x_{i_1,j} \vee \neg x_{i_2,j}$ for all choices of distinct pigeons $i_1, i_2 \in [m]$ and holes $j \in [n]$. We prove that PHP^m_n is hard for tree-like Resolution. Showing the lower bound by the asymmetric game from the last section, requires a suitable choice of the functions c_0 and c_1 and then the definition of the Delayer-strategy for the (c_0, c_1)-game.

Theorem 2. *Any tree-like Resolution refutation of* PHP^m_n *has size* $2^{\Omega(n \log n)}$.

We observe that our proof of Theorem 2 also holds for the *functional pigeonhole principle* where in addition to the clauses from PHP^m_n we also include $\neg x_{i,j_1} \vee \neg x_{i,j_2}$ for all pigeons $i \in [m]$ and distinct holes $j_1, j_2 \in [n]$.

References

[1] A. Atserias and V. Dalmau. A combinatorial characterization of resolution width. *Journal of Computer and System Sciences*, 74(3):323–334, 2008. 1

[2] E. Ben-Sasson, R. Impagliazzo, and A. Wigderson. Near optimal separation of tree-like and general resolution. *Combinatorica*, 24(4):585–603, 2004. 1

[3] O. Beyersdorff, N. Galesi, and M. Lauria. Hardness of parameterized resolution. Technical Report TR10-059, Electronic Colloquium on Computational Complexity, 2010. 1, 2, 3

[4] S. S. Dantchev, B. Martin, and S. Szeider. Parameterized proof complexity. In *Proc. 48th IEEE Symposium on the Foundations of Computer Science*, pages 150–160, 2007. 2

[5] S. S. Dantchev and S. Riis. Tree resolution proofs of the weak pigeon-hole principle. In *Proc. 16th Annual IEEE Conference on Computational Complexity*, pages 69–75, 2001. 2

[6] J. L. Esteban, N. Galesi, and J. Messner. On the complexity of resolution with bounded conjunctions. *Theoretical Computer Science*, 321(2–3):347–370, 2004. 1

[7] J. Krajíček. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*, volume 60 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, Cambridge, 1995. 1, 2

[8] P. Pudlák. On the complexity of propositional calculus. In *Sets and Proofs, Invited papers from Logic Colloquium'97*, pages 197–218. Cambridge University Press, 1999. 1

[9] P. Pudlák and S. R. Buss. How to lie without being (easily) convicted and the length of proofs in propositional calculus. In *Proc. 8th Workshop on Computer Science Logic*, volume 933 of *Lecture Notes in Computer Science*, pages 151–162. Springer-Verlag, Berlin Heidelberg, 1994. 1

[10] P. Pudlák and R. Impagliazzo. A lower bound for DLL algorithms for SAT. In *Proc. 11th Symposium on Discrete Algorithms*, pages 128–136, 2000. 1, 3

[11] N. Segerlind. The complexity of propositional proofs. *Bulletin of symbolic Logic*, 13(4):482–537, 2007. 2