Cook and Reckhow are Wrong:
Subexponential Tableau Proofs for Their Family of Formulae

Fabio Massacci[1]

Abstract. It is widely believed that a family Σ_n of unsatisfiable formulae defined by Cook and Reckhow [*Proc. of the ACM Symp. on Theory of Comp. 1974*] gives a lower bound of $O(2^{2^n})$ on the proof size with analytic tableaux.

This claim plays a key role in the proof that tableaux cannot polynomially simulate tree resolution.

We show that it is wrong by exhibiting an analytic tableau proof whose size has an *upper bound* of $O(n \times 2^{n^2})$, which, although not polynomial in the size (2^n) of the input formula, is exponentially shorter than the claimed lower bound.

We claim that the pitfall, in that and other papers, is due to the blurring of n-ary and binary versions of tableaux.

1 Introduction

The study of upper and lower bounds on the proof size of propositional tautologies using resolution and tableaux played a major role in computer science since the ground breaking papers by Cook & Reckhow [1, 2]. This line of research has been quite fruitful in providing a sound computational basis for ranking variants of resolution and tableaux.

The key tool is polynomial simulation. Informally, a proof system Π^+ is more powerful than Π if we can map every proof of a formula A in the system Π into a proof of A in the system Π^+, using a polynomial function (in the size of the proof with Π) but the converse does not hold. The last step is usually proved by exhibiting a family of formulae for which there is an exponential lower bound (in the size of the formula) on every proof in Π whereas there are short polynomial proofs in Π^+. We refer to Urquhart [6] for formal definitions.

For instance, the claim that tableaux cannot polynomially simulate tree-resolution is based on the fact that the family of formulae Σ_n by Cook & Reckhow [1, 2] has only exponential size tableau proofs but a polynomial resolution proof.

In the rest of the paper we show that this claim is wrong. We recall the family Σ_n (§2) and analytic tableaux (§3), and discuss the construction of a particular tableau proof of size exponentially shorter than the claimed lower bound (§4). Finally (§5) we discuss why so many papers seem to have "proved" this results and show that the problem lies in the seemingly trivial generalization of results valid for clausal tableaux to the class of ordinary (binary) tableaux.

Throughout the paper we assume a basic knowledge of propositional logic. For an introduction see Smullyan [5].

[1] Dipartimento di Informatica e Sistemistica, Università di Roma I "La Sapienza", via Salaria 113, Roma, Italy email: `massacci@dis.uniroma1.it`

ECAI 98. *13th European Conference on Artificial Intelligence*
Young Researcher Paper
Edited by Henri Prade
Published in 1998 by John Wiley & Sons, Ltd.

2 Cook and Reckhow Family of Formulae

The family of Cook and Reckhow formulae [1] is constructed by associating a set of clauses Σ_n to a binary tree of depth n:

- the tree with one node is associated to the empty clause;
- each internal node is associated to a different variable;
- each leaf is associated to a clause whose literals are the atoms of the internal nodes, considered positive if the path (from the root to the leaf) continues at the left of the node and negative if it is at the right.

We represent it as follows [6, 3]:

$$\Sigma_n = \left\{\pm A \vee \pm A_{\pm} \vee \pm A_{\pm\pm} \vee \ldots \vee \pm A_{\pm\langle n-1\rangle\pm}\right\}$$

where the string $\pm \cdots \pm$ is determined by the signs of the previous literals. For instance we have

$$\begin{aligned}\Sigma_1 &= \{A, \neg A\}\\ \Sigma_2 &= \{A \vee A_+, A \vee \neg A_+, \neg A \vee A_-, \neg A \vee \neg A_-\}\end{aligned}$$

To represent clauses using binary connectives we follow Urquhart [6] and assume that $\vee$ associates to the left, so that $L_1 \vee L_2 \ldots \vee L_{n-1} \vee L_n = L_1 \vee (L_2 \vee \ldots (L_{n-1} \vee L_n) \cdots))$.

3 Analytic Tableaux

The original definition of analytic tableaux given by Beth and systematized by Smullyan [5] is simple: a *tableau for a finite set of formulae* $\Sigma = \{A_1, \ldots A_n\}$ is an ordered (binary) tree, whose nodes are labelled by formulae according the following rules. We start by placing A_1 at the origin then we add a successor node with A_2 and so on until A_n labels the last node of the sequence. Then we may extend $\mathcal{T}$ as follows:

- If $A \wedge B$ labels a node on the path down to a leaf N, then we may add below N a sole successor node with A and then below A another node labelled with B
- If $A \vee B$ occurs on the path down to a leaf N, then we may add below N a left successor node with A and a right successor node with B.

If we only have clauses, we can also use *clausal tableau*:

- if $A_1 \vee A_2 \ldots \vee A_n$ occurs on the path down to a leaf N, then we may extend the tableau by adding n successor nodes $N_1, \ldots N_n$, each N_i labelled by A_i.

A *tableau proof* is a tableau where for every path from the root to a leaf there is a formula A such that both A and $\neg A$ occurs along the path.

The *size of a tableau proof* for a set of formulae Σ is usually the number of *internal nodes* of a tableau proof of Σ.

4 A Subexponential Tableau Proof for Σ_n

It is (wrongly) believed that tableau proofs for Σ_n have at least size $O(2^{2^n})$. Since Σ_n has size $O(2^n)$, where the size is measured as the number of symbols, all tableau proofs should have size larger than $O(2^{|\Sigma|})$, if claimed upper bounds were true. On the contrary, our proof has size $O(2^{\log^2|\Sigma|})$.

Due to lack of space we only give the rules we applied.

Our tableau proof starts with the initial segment Σ_n:

$$\begin{array}{l} P_n^+ \left\{ \begin{array}{l} A \vee A_+ \vee A_{++} \vee \cdots \vee A_{+\langle n-1\rangle+} \\ \vdots \\ A \vee \neg A_+ \vee \neg A_{+-} \vee \cdots \vee \neg A_{+-\langle n-2\rangle-} \end{array} \right. \\ P_n^- \left\{ \begin{array}{l} \neg A \vee A_- \vee A_{-+} \vee \cdots \vee A_{-+\langle n-2\rangle+} \\ \vdots \\ \neg A \vee \neg A_- \vee \neg A_{--} \vee \cdots \vee \neg A_{-\langle n-1\rangle-} \end{array} \right. \end{array} \quad (0)$$

We start by reducing all formulae in the upper initial prefix P_n^+ in sequence, using rule β. We only branch on the first literal A, splitting the tree into A and $A_+ \vee A_{++} \vee \cdots \vee A_{++\langle n-2\rangle+}$. Then in the right subtree, that labelled with $A_+ \vee A_{++} \vee \cdots \vee A_{++\langle n-2\rangle+}$, we split again on the second formula of P_n^+ obtaining again A on the left subtree and $A_+ \vee A_{++} \vee \cdots \vee \neg A_{++\langle n-2\rangle+}$ on the right. We continue until we have reduced all clauses in P_n^+.

Notice that we do *not* reduce $A_+ \vee \cdots \vee A_{++\langle n-2\rangle+}$ in the right subtree before having reduced all formulae in P_n^+. This is the key step that explains why claimed lower bounds on tableau proofs fail (see §5).

In this way we obtain a sort of comb, with 2^{n-1} nodes on the left (the "teeth"), each labelled with A, and a "spine" whose nodes are labelled with $A_+ \vee A_{++} \vee \cdots \vee A_{+\langle n-1\rangle+}$, then $\neg A_+ \vee A_{+-} \vee \cdots \vee A_{+-+\langle n-2\rangle+}$, and so on until we get $\neg A_+ \vee \neg A_{+-} \vee \cdots \vee \neg A_{+-\langle n-1\rangle-}$. Now we can observe that these 2^{n-1} nodes correspond exactly to the initial segment of the tableau for Σ_{n-1} if we replace systematically $A_{+\sigma}$ by A_σ. So, to continue the proof below the spine of the comb, we simply use the recursive construction of the tableau for $n-1$.

Next, we start the construction of the subtrees starting with A (the teeth of the comb). For each subtree we work as follows: apply rule β to the formulae of P_n^- in sequence, splitting on $\neg A$ on one side and $A_- \vee A_{-+} \vee \ldots A_{-+\langle n-2\rangle+}$ on the other side and repeat the modus operandi we have followed above for the construction of the spine until we have exhausted all formulae of P_n^-. Again we obtain another comb where the spine is the initial sequence of the tableau for Σ_{n-1} if we replace syntactically $A_{-\sigma}$ with A_σ.

We now analyze the size of the proof in term of the number of nodes, including terminal nodes and thus providing a tighter upper bound. So let S_n be the size of the tableau for Σ_n. Note that $S_1 = 2$ because $\Sigma_1 = \{A, \neg A\}$.

In the general case, we have an initial prefix of size 2^n, followed by a comb with 2^{n-1} teeth. As we have noted, the spine of the main comb is the initial prefix of the tableau proof for a syntactic variant of Σ_{n-1}; so we can directly include it in the size of the corresponding proof S_{n-1}. For each comb starting from the 2^{n-1} nodes labelled with A we have 2^{n-1} teeth, each labelled with $\neg A$, and a spine. The spine is again the initial prefix of a tableau proof of a variant of Σ_{n-1} and can be accounted for by adding S_{n-1}. Summing the various components we get

$$\begin{array}{rcl} S_1 &=& 2 \\ S_n &=& 2^n + S_{n-1} + 2^{n-1} \times (1 + 2^{n-1} + S_{n-1}) \end{array}$$

To simplify the calculus we choose the following upper bound for $n \geq 4$: $S_n \leq 2^n \times (S_{n-1} + 2^{n-1})$.

After some algebraic transformation we get

$$S_n \quad \leq \quad 2^{\frac{n(n+1)}{2}} + \sum_{i=1}^{n} 2^{\frac{(2n-i)(i+1)}{2}} \quad = O(n \times 2^{n^2})$$

Theorem 1 *The proof complexity of analytic tableaux for the Σ_n family is bounded from above by $O(n \times 2^{n^2})$.*

Although this is not a polynomial in $|\Sigma|$, the *upper bound* $\log|\Sigma| \times 2^{\log^2|\Sigma|}$ is exponentially smaller than the *claimed lower bound* $2^{|\Sigma|}$.

5 What Went Wrong in the Past?

The lower bound of 2^{2^n} has appeared in a number of papers such as Cook and Reckhow [1, 2], D'Agostino and Mondadori, [3], Murray and Rosenthal [4], and Urquhart [6].

A careful analysis reveals that the unsound step is the seemingly trivial extension of the results from clausal tableaux to "traditional" binary tableaux à la Smullyan. In a nutshell, all papers above rely on variants of the following lemma:

Lemma 2 *If $s(\Sigma)$ is the size of a tableau proof for Σ then there is clause $A_1 \vee A_2 \vee \ldots A_n \in \Sigma$ such that $s(\Sigma) = \sum_{i=1}^{n} s(\Sigma \cup \{A_i\} \setminus \{A_1 \vee \ldots A_n\})$.*

This is undoubtedly true with clausal tableau. Yet, the lemma no longer holds for traditional binary tableau. It only holds if we assume a particular proof search strategy when reducing $A_1 \vee A_2 \vee \ldots \vee A_n$:

- we first create a left successor with A_1 and a right successor with $A_2 \vee \ldots \vee A_n$
- then, we continue the reduction focusing immediately on $A_2 \vee A_3 \ldots \vee A_n$ and split the tree with A_2 and $A_3 \vee \ldots \vee A_n$, then we move down the tree and split it into A_3 and $A_4 \vee \ldots \vee A_n$, etc.

This is only a strategy, and indeed our proof does not use it.

Hence, we reopen a problem which seemed to be closed:

Question 1 *Can analytic tableaux simulate tree-resolution?*

With a sort of domino effect, many other results, such as those by D'Agostino & Mondadori [3] on the relative efficiency of tableaux and truth tables, may need to be reconsidered.

REFERENCES

[1] S. Cook and R. Reckhow, 'On the lengths of proofs in the propositional calculus', in *Proc. of STOC-74*, pp. 135–148, (1974).

[2] S. Cook and R. Reckhow, 'The relative efficiency of propositional proof systems', *JSL*, **44**, 36–50, (1979).

[3] M. D'Agostino and M. Mondadori, 'The taming of the cut', *JLC*, **4**(3), 285–319, (1994).

[4] N. Murray and E. Rosenthal, 'On the computational intractability of analytic tableau methods', *J. of the IGPL*, **2**(2), 205–228, (1994).

[5] R. Smullyan, *First Order Logic*, Springer-Verlag, 1968. Republished by Dover, New York, in 1995.

[6] A. Urquhart, 'The complexity of propositional proofs', *Bull. of Symbolic Logic*, **1**(4), 425–467.