ELSEVIER

Theoretical Computer Science 243 (2000) 477–487

Theoretical Computer Science

www.elsevier.com/locate/tcs

Note

The proof complexity of analytic and clausal tableaux

Fabio Massacci[a,b,*]

[a] *Dipartimento di Informatica e Sistemistica, Università di Roma I "La Sapienza", via Salaria 113, 00198 Roma, Italy*

[b] *Dipartimento di Ingegneria dell'Informazione, Università di Siena, via Roma 56, 53100 Siena, Italy*

Received November 1998; revised February 2000; accepted February 2000

Abstract

It is widely believed that a family Σ_n of unsatisfiable formulae proposed by Cook and Reckhow in their landmark paper (Proc. ACM Symp. on Theory of Computing, 1974) can be used to give a lower bound of $2^{\Omega(2^n)}$ on the proof size with analytic tableaux. This claim plays a key role in the proof that tableaux cannot polynomially simulate tree resolution. We exhibit an analytic tableau proof for Σ_n for whose size we prove an *upper bound* of $O(2^{n^2})$, which, although not polynomial in the size $O(2^n)$ of the input formula, is exponentially shorter than the claimed lower bound. An analysis of the proofs published in the literature reveals that the pitfall is the blurring of n-ary (clausal) and binary versions of tableaux. A consequence of this analysis is that a second widely held belief falls too: clausal tableaux are not just a more efficient notational variant of analytic tableaux for formulae in clausal normal form. Indeed clausal tableaux (and thus model elimination without lemmaizing) cannot polynomially simulate analytic tableaux.

Keywords: Analytic tableaux; Automated reasoning; Clausal tableaux; Proof complexity; Tree resolution

1. Introduction

The study of upper and lower bounds on the proof size of propositional tautologies using different proof systems has played a major role in computer science since the ground breaking papers by Cook and Reckhow [4, 5]. This line of research has been quite fruitful in providing a sound computational basis for ranking variants of proof systems.

* Correspondence address: Dipartimento di Informatica e Sistemistica, Università di Roma I "La Sapienza", via Salaria 113, 00198 Roma, Italy.

E-mail address: massacci@dis.uniroma1.it, massacci@dii.unisi.it (F. Massacci).

0304-3975/00/$ - see front matter
PII: S0304-3975(00)00148-1

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 system Π into a proof of A in 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 formulae) on every proof in Π whereas there are short polynomial proofs in Π^+. We refer to the comprehensive survey by Urquhart [14] for formal definitions.

Among the various proof systems which have been ranked, tableaux [12] and resolution [11] have received considerable attention. For instance, the claim that tableaux cannot polynomially simulate tree-resolution is based on a family of formulae Σ_n proposed by Cook and Reckhow [4, 5] which is claimed to have only exponential size tableau proofs but a polynomial resolution proof.

Although this claim is widely believed (see e.g. [1, 6, 9, 10, 14]), we show in this note that there is a tableaux proof for this family of formulae that is exponentially shorter than the claimed lower bound.

The interest of this result is twofold: first it reopens the question of the relative proof strength of tree-resolution and analytic tableau. The second interesting point is the nature of the error, which lies in the seemingly trivial generalization of results valid for clausal tableaux to the class of (binary) analytic tableaux. This is not a simple point: most research in automatic deduction has focused on clausal normal form and most provers work with clausal normal form. For instance, the winners of the past automatic theorem prover competitions at the International Conference on Automated Deduction work with clauses: one prover uses resolution [13] and the other clausal tableaux [7].

So, it is not a surprise that in the automatic reasoning community it is widely believed that the clausal version of tableau [7, 3], a variant of model elimination without lemmaizing [8], is just a notational (possibly more efficient) variant of the classical analytic tableaux calculus proposed by Smullyan [12].

In consequence of our analysis, this belief falls too. Combining the result proved in the literature about Cook and Reckhow family and (clausal) tableaux with the existence of a quasi-polynomial analytic tableau proof shown here, we can conclude that clausal tableaux are far from being a notational variant of analytic tableaux: they cannot even polynomially simulate analytic tableaux.

1.1. Plan of the paper

In the next sections, we recall the Cook and Reckhow class of formulae Σ_n (Section 2) and the proof theory of analytic and clausal tableaux (Section 3). Next we show the construction of a particular tableau proof for these formulae (Section 4) and shows that its size is exponentially shorter than the claimed lower bound (Section 5).

Then we discuss the nature of the error in the blurring of the clausal and binary version of tableaux and show that clausal tableaux cannot polynomially simulate analytic tableaux (Section 6). A brief discussion concludes the paper (Section 7).

2. Propositional logic and Cook and Reckhow formulae

Throughout the paper we assume a basic knowledge of propositional logic (for an introduction, see [12]). Propositional *formulae* denoted by A, B, X, Y, are formed from propositional variables $p \in \mathscr{P}$ as follows:

$$A, B ::= p | \neg A | A \wedge B | A \vee B.$$

A *literal*, denoted by L is either a propositional variable p or a negated propositional variable $\neg p$. The *sign* of a literal is positive if the literal is p and negative if it is $\neg p$. A *clause* is a disjunction of literals which can also be seen as a multiset of literals.

To represent clauses as formulae using binary connectives we follow Urquhart [14] and assume that $\vee$ associates to the right, so that the clause $L_1 \vee L_2 \cdots \vee L_{n-1} \vee L_n$ is a shorthand for the formula $L_1 \vee (L_2 \vee \cdots (L_{n-1} \vee L_n) \cdots))$.

The family of Cook and Reckhow formulae [4] is constructed by associating a set of clauses Σ_n to a binary tree of depth n. The construction works as follows:

- the tree with one node is associated to the empty clause;
- each internal node of a tree 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 continues at the right.

In particular, we are interested in sets of clauses corresponding to *complete binary trees of depth n*. We represent them as follows [6]:

$$\Sigma_n = \bigcup \{\pm A \vee \pm A_{\pm} \vee \pm A_{\pm\pm} \vee \cdots \vee \pm A_{\pm\langle n-1 \rangle \pm}\}, \tag{1}$$

where the string $\pm \cdots \pm$ is determined by the signs of the previous literals.

For instance, the simple families Σ_1 and Σ_2 are

$$\Sigma_1 = \{A, \neg A\},$$

$$\Sigma_2 = \{A \vee A_+, A \vee \neg A_+, \neg A \vee A_-, \neg A \vee \neg A_-\}.$$

Notice that the family Σ_n has size $O(2^n)$ (the number of leaves in a complete tree of depth n) where the size is measured as the number of symbols.

3. Analytic and clausal tableaux

As a preliminary notion we recall the classification of formulae as α and β formulae [12] in Table 1.

The original definition of *analytic tableaux* as given by Beth and systematized by Smullyan is very simple and we quote directly from Smullyan's book [12, pp. 24, 29], where X is a formula:

> An analytic tableau for X is an ordered dyadic tree, whose points are (occurrences of) formulas, which is constructed as follows. We start by placing X at the origin.

Table 1
α, β notation

α	α_1	α_2	β	β_1	β_2
$X \wedge Y$	X	Y	$X \vee Y$	X	Y
$\neg(X \vee Y)$	$\neg X$	$\neg Y$	$\neg(X \wedge Y)$	$\neg X$	$\neg Y$
$\neg\neg X$	X				

Now suppose $\mathcal{T}$ is a tableaux for X which has already been constructed; let Y be an end point. Then we may extend $\mathcal{T}$ by either of the following two operations.

(A) If some α occurs on the path P_Y, then we may adjoin either α_1 or α_2 as the sole successor of Y. (In practice, we usually successively adjoin α_1 and then α_2.)

(B) If some β occurs on the path P_Y, then we may simultaneously adjoin β_1 as the left successor of Y and β_2 as the right successor of Y.

[...] *Tableaux for finite sets*. If S is a *finite* set $\{X_1,\dots,X_n\}$ a tableau for S is meant a tableau starting with

$$\begin{array}{c} X_1 \\ X_2 \\ \vdots \\ X_n \end{array}$$

and then continued using Rules A, B.

We can also construct a *clausal tableau* $\mathcal{T}_C$ if the set S of input formulae is only made of clauses. We replace rule (B) with the following rule (C):

(C) If some clause $L_1 \vee L_2 \vee L_3 \vee \cdots \vee L_n$ occurs on the path P_Y, then we may simultaneously adjoin each L_i as the ith successor of Y, for $i = 1 \dots n$.

Model elimination without lemmaizing, as proposed by Loveland [8], is a notational variant of clausal tableau.

In a clausal tableau all nodes are labelled by literals, except for the initial nodes storing the set S of clauses. In an analytic tableau the internal nodes may also be clauses.

A *tableau proof* for a finite set of formulae S is a tableau for S where for every path from the root to a leaf there is a formula A such that both A and $\neg A$ occur along the path.

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

This measure is sufficiently accurate for both analytic and clausal tableau, if we are concerned with polynomial simulation. If k is the size of the largest clause in S (hence k is at worst proportional to S), counting all nodes would increase the measure by a factor of 2 for analytic tableaux and a factor of k for clausal tableau. A further refinement using the number of symbols would increase the measure by a factor of k.

4. A quasi-polynomial tableau proof for Σ_n

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

In the sequel, we use $\langle a\rangle, \langle b\rangle, \langle c\rangle \ldots$ to mark various positions of the tableau proof, and $\langle a-b\rangle$ to denote the fragment of the tableau proof going from $\langle a\rangle$ to $\langle b\rangle$, etc.

The tableau proof for Σ_n starts with the following initial segment (see Section 3):

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

We expand the tableau by reducing all formulae in the upper initial prefix P_n^+ in sequence, using rule B. We only branch on the first literal A, splitting the tree with A as left successor and $A_+ \vee A_{++} \vee \cdots \vee A_{++\langle n-2\rangle +}$ as right successor. 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 A on the left subtree and $\neg A_+ \vee A_{+-} \vee \cdots \vee A_{+-+\langle n-3\rangle +}$ 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^+. As we shall see, this is the key step that explains why claimed lower bounds on tableau proofs fail (see Section 6). The final outcome is shown in Fig. 1.

We have obtained a sort of comb, with 2^{n-1} nodes on the left (the "teeth"), each labelled with A, and a "spine" (the path from $\langle b\rangle$ to $\langle d\rangle$ 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 $\neg A_+ \vee \neg A_{+-} \vee \cdots \vee \neg A_{+-\langle n-1\rangle -}$. Observe that the 2^{n-1} nodes from $\langle b\rangle$ to $\langle d\rangle$ 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 point $\langle d\rangle$, 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) indicated by $\langle c_i\rangle$. For each subtree we work as follows: apply rule (B) to the formulae of P_n^- in sequence, splitting on $\neg A$ on one side and $A_- \vee A_{-+} \vee \cdots A_{-+\langle n-2\rangle +}$ on the other side and repeat the modus operandi we have followed above for the construction of the spine $\langle b-d\rangle$ until we have exhausted all formulae of P_n^-. The final outcome is shown in Fig. 2.

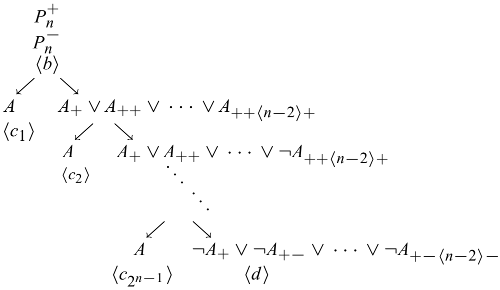

Fig. 1. The first step of the tableau proof.

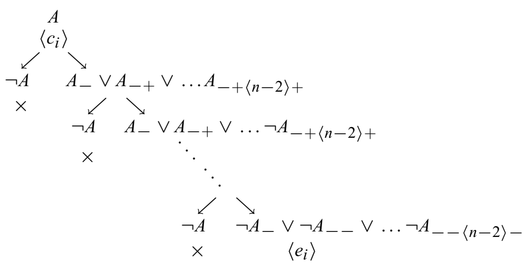

Fig. 2. The second (repeated) step of the tableau proof.

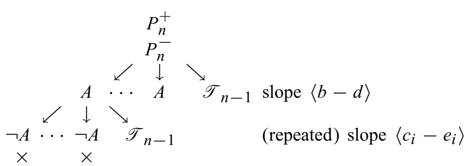

Fig. 3. The structure of the tableau proof.

Again we obtain a comb. Each left subtree contains a pair A, $\neg A$ along the path from the root to the leaf. The path from $\langle c_i \rangle$ to $\langle e_i \rangle$ is again the initial segment of the tableaux for Σ_{n-1} if we replace syntactically $A_{-\sigma}$ with A_σ.

The general structure of the tableau $\mathscr{T}_n$ for Σ_n is summarized in Fig. 3.

5. Complexity analysis

We analyze the proof size in terms of the number of nodes, and include the terminal nodes to provide a tight upper bound.

Let T_n be the size of the tableaux for Σ_n. At first note that $T_1 = 2$ because $\Sigma_1 = \{A, \neg A\}$, so that pasting A and $\neg A$ one below the other already yields a tableau proof. In the general case, we have

$$T_n = 2 \cdot 2^{n-1} + T_{n-1} + 2^{n-1} \cdot (1 + 2^{n-1} + T_{n-1}).$$

The first addendum is the size of the initial segment $\langle a - b \rangle$, the second addendum is the size of the slope $\langle b - d \rangle$ which is then completed into a subtree $\mathscr{T}_{n-1}$. The last addendum is the overall size of the subtrees $\langle c_i \rangle$: each subtree includes the root A, the 2^{n-1} leaves labelled $\neg A$ and the remaining slope $\langle c_i - e_i \rangle$ which is then completed into a subtree $\mathscr{T}_{n-1}$.

This equation boils down to the following:

$$T_1 = 2, \tag{2}$$

$$T_n = (2^{n-1} + 1) \cdot T_{n-1} + 2^{n-1} \cdot (2^{n-1} + 3). \tag{3}$$

For our purposes, a crude estimate of the proof size is sufficient. Thus, we start by observing that $T_n \geqslant 2^n$, and therefore by some simple algebraic transformation we get $T_n \leqslant 2 \cdot 2^n \cdot T_{n-1}$. Then, we proceed by defining a recursive function U_n which clearly bounds T_n from above:

$$U_1 = 2, \tag{4}$$

$$U_n = 2^{n+1} \cdot U_{n-1}. \tag{5}$$

A closed-form solution for U_n can be easily found:

$$U_n = 2 \cdot \prod_{i=2}^{n} 2^{i+1} = 2^n \cdot 2^{\sum_{i=2}^{n} i} = 2^n \cdot 2^{(n+2)(n-1)/2}$$

which gives us the desired *upper bound*

$$T_n \leqslant U_n = 2^{(n^2+3n-2)/2}. \tag{6}$$

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

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

6. On the difference between clausal and analytic tableaux

The lower bound of $2^{\Omega(2^n)}$ has appeared in a number of papers such as Cook and Reckhow [4, 5], Arai, [1], D'Agostino and Mondadori, [6], Murray and Rosenthal [9], and Urquhart [14] and we may wonder what went wrong.

A careful analysis reveals that the unsound step is the seemingly trivial extension of results from clausal tableaux to analytic tableaux.

In a nutshell, all papers above rely on variants of the following lemma:

Lemma 2. *If $t(S)$ is the size of a tableau proof for a finite set of formulae S then there is clause $A_1 \vee \cdots \vee A_n \in S$ such that $t(S) = \sum_{i=1}^{n} t(S \cup \{A_i\} \setminus \{A_1 \vee \cdots \vee A_n\})$.*

This is undoubtedly true with a clausal tableau: each time we branch the tree we add n leaves each containing a literal from the reduced clause (see again rule (C) in Section 3). One can use this lemma to prove a restricted version of Cook and Reckhow lower bound (for the proof see [14]).

Theorem 3. *The proof complexity of* clausal *tableaux for the Σ_n family is bounded from below by $2^{\Omega(2^n)}$.*

The problem is that Lemma 2 no longer holds for analytic tableau, but for a particular proof search strategy for reducing $L_1 \vee L_2 \vee \cdots \vee L_n$:

- create a left successor with L_1 and a right successor with $L_2 \vee \cdots \vee L_n$
- continue the reduction focusing on $L_2 \vee L_3 \cdots \vee L_n$ and split the tree with L_2 and $L_3 \vee \cdots \vee L_n$, then we move down the tree and split it into L_3 and $L_4 \vee \cdots \vee L_n$, etc.

This reduction strategy is not necessary, and indeed we have not used it.

With this reduction strategy we can simulate clausal tableau with analytic tableaux. Notice that for each reduced clause of length k we must add k nodes in a clausal tableau and $2k$ nodes in the corresponding analytic tableau. Using this proof search strategy, an analytic tableau proof is only twice as big as the corresponding clausal tableau proof.

Combining this observation with Theorems 1 and 3 we obtain the second surprising result.

Theorem 4. *Analytic tableaux can polynomially simulate clausal tableaux but clausal tableaux cannot polynomially simulate analytic tableaux.*

So, even on sets of clauses, clausal tableaux (and hence model elimination) are far from being a more efficient notational variant of analytic tableau.

We can give an intuitive explanation by focusing on the key Lemma 2 and observe what happens when we pass from S to $S \cup \{A\}$ in a clausal and in an analytic tableau proof. Suppose that in S there are m clauses of the form $\neg A \vee L_{1i} \vee \cdots L_{nj}$. The most simple way to work with analytic tableau is to “eliminate” $\neg A$ and shorten the size of a number of clauses in the clause set. The corresponding proof is shown in Fig. 4. This proof fragment is substantially a linear sequence of length m: we have just shortened the input clauses and eliminated the literal $\neg A$ from consideration without further branching.

Consider instead what happens with the corresponding clausal tableau proof in Fig. 5 to eliminate literal $\neg A$ from consideration we must introduce $n1$ branches each of

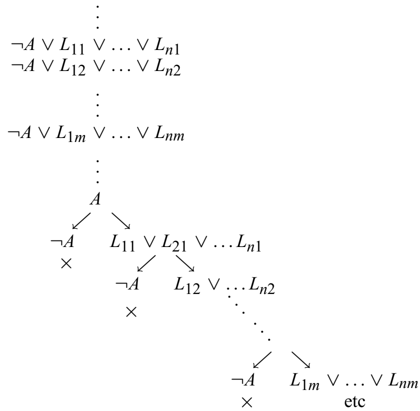

Fig. 4. An analytic tableau proof when a literal is present.

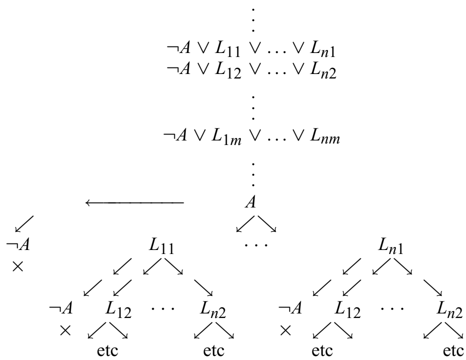

Fig. 5. A clausal tableau proof when a unit clause is present.

which must be closed and for this we need $n2$ branches, etc. To eliminate $\neg A$ from consideration we must consider a subtree of depth m.

In a nutshell, analytic tableaux can exploit unit clauses in much more effective way than clausal tableaux. A form of unit resolution can be exploited by analytic tableaux but not by clausal tableaux.

7. Conclusions

In this paper we have shown the following:

- the family of Cook and Reckhow formulae traditionally used to rank analytic tableaux and tree-resolution admits a quasi-polynomial tableau proof;
- the original exponential tableau proof was only valid for the n-ary (clausal) version of tableaux;

- clausal tableaux (and thus model elimination without lemmaizing [1]) cannot polynomially simulate analytic tableaux.

Intuitively this result can be explained in two ways: from the point of view of proof theory, analytic tableaux have one degree of freedom more than clausal tableaux and therefore their proofs can be shorter. From the point of view of automated reasoning, analytic tableaux are able to use unit clauses in a more effective way than clausal tableaux, fairly close to unit resolution. In that way they can produce intermediate clause sets in which the size of a number of clauses is shortened and therefore shorter proofs can be found.

As a consequence, a number of interesting questions are opened.

Question 1. *Can clausal tableaux with (atomic) cut polynomially simulate analytic tableaux with (atomic) cut?*

Question 2. *Can analytic tableaux polynomially simulate tree-resolution?*

The first question is particularly interesting in the light of the result by Arai [1], showing that admitting cut formulae of different size creates a proper hierarchy (wrt polynomial simulation) of tableau calculi, and the current implementation techniques of clausal tableaux which use atomic cut [7].

As for the second question, the author is convinced that the answer is negative, although we must be careful when looking for new counterexamples. The family of formulae proposed by D'Agostino and Mondadori [6] to compare the relative efficiency of tableaux and truth tables might be a good candidate. Yet, this may lead us to the same pitfall.

Indeed, a recent result by Arai [2] for resolution and tableau proofs in DAG form [2] shows that there is an upper bound of $O(n^{2+3\log n})$ for the translation of (DAG) resolution refutation of size n into (DAG) cut-free tableau proofs. It is worth noting that it is possible to recast our result on the proof size of Cook and Reckhow formulae Σ_n in the same way: there is a proof of size $O(|\Sigma_n|^{\log|\Sigma_n|})$. We can conjecture that the DAG result might be extended to the tree case:

Conjecture 1. *Analytic tableaux quasi-polynomially simulates tree resolution.*

We leave both questions open for future investigations.

Acknowledgements

I would like to thank L. Carlucci Aiello for her encouragement and support and N. Arai, R. Letz, and an anonymous reviewer for useful comments which improved

[1] This result can be strengthened by observing that model elimination cannot even polynomially simulate clausal tableaux with a weak form of cut such as factorization [7].

[2] The proof is represented as direct acyclic graph rather than a tree.

the quality of the paper. This work has been partly supported by MURST and CNR grants of the Dipartimento of Informatica and Sistemistica at the University of Roma "La Sapienza" and by a travel grant of ECCAI.

References

[1] N. Arai, A proper hierarchy of propositional sequent calculi, Theoret. Comput. Sci. 159 (2) (1996) 343–354.
[2] N. Arai, Cut-free LK quasi-polynomially simulates resolution, preprint FI-CXT1998-001, The Fields Institute for Research in Mathematical Sciences, University of Toronto, Canada, 1998.
[3] P. Baumgartner, U. Furbach, I. Niemelä, Hyper tableaux, in: J. Alferes, L. Pereira, E. Orlowska (Eds.), Proc. 6th European Workshop on Logics in Artificial Intelligence (JELIA-96), Lecture Notes in Artificial Intelligence, vol. 1126, Springer, Berlin, 1996,, pp. 1–17.
[4] S.A. Cook, R. Reckhow, On the lengths of proofs in the propositional calculus, Proc. 6th ACM Symp. on Theory of Computing (STOC-74), 1974, pp. 135–148. See corrections in SIGACT News, 6 (1974) 15–22. Extended version in [5].
[5] S.A. Cook, R. Reckhow, The relative efficiency of propositional proof systems, J. Symbolic Logic 44 (1979) 36–50.
[6] M. D'Agostino, M. Mondadori, The taming of the cut, J. Logic Comput. 4 (3) (1994) 285–319.
[7] R. Letz, K. Mayr, C. Goller, Controlled integration of the cut rule into connection tableau calculi, J. Automat. Reason. 13 (3) (1994) 297–337.
[8] D. Loveland, Mechanical theorem proving by model elimination, J. ACM 15 (2) (1968) 236–251.
[9] N. Murray, E. Rosenthal, On the computational intractability of analytic tableau methods, J. Interest Group in Pure Appl. Logic 2 (2) (1994) 205–228.
[10] N. Murray, E. Rosenthal, On the relative merits of path dissolution and analytic tableau, Theoret. Comput. Sci. 131 (1) (1994) 1–28.
[11] J. Robinson, A machine oriented logic based on the resolution principle, J. ACM 12 (1) (1965) 23–41.
[12] R. Smullyan, First Order Logic, Springer, Berlin, 1968. Republished by Dover, New York, 1995.
[13] T. Tammet, Gandalf, J. Automat. Reason. 18 (2) (1997) 199–204.
[14] A. Urquhart, The complexity of propositional proofs, Bull. Symbolic Logic 1 (4) (1995) 425–467.