Non-elementary Speed-ups in Proof Length by Different Variants of Classical Analytic Calculi*

Uwe Egly

Institut für Informationssysteme E184.3
Technische Universität Wien
Treitlstraße 3, A–1040 Wien, Austria
e-mail: uwe@kr.tuwien.ac.at

Abstract. In this paper, different variants of classical analytic calculi for first-order logic are compared with respect to the length of proofs possible in such calculi. A cut-free sequent calculus is used as a prototype for different other analytic calculi like analytic tableau or various connection calculi. With modified branching rules (β-rules), non-elementary shorter minimal proofs can be obtained for a class of formulae. Moreover, by a simple translation technique and a standard sequent calculus, analytic cuts, i.e., cuts where the cut formulae occur as subformulae in the input formula, can be polynomially simulated.

1 Introduction

Analytic first-order calculi like (free-variable) analytic tableaux [4, 15, 23] or various connection calculi [5] are well suited for implementing automated deduction on a computer. In order to search for a proof in such calculi, only subformulae of the input formula have to be considered. This property, often referred to as the subformula property, makes such calculi highly attractive for automated proof search in classical as well as in non-classical logics. Although the subformula property does not imply cut-free calculi, most of the aforementioned calculi are cut-free. The price that we have to pay for the simplicity introduced by these restrictions is the weakness of these calculi with respect to proof length. Short and structured proofs should be preferred because they are easier to understand and easier to check.

In this paper, we show that this weakness can be eliminated to an enormous extent without giving up these restrictions. In what follows, we will use different variants of the cut-free LK-calculus [16] because this calculus can be considered as a prototype for many other analytic calculi including the ones mentioned above (see [23] or [7] for details). We stress that our results also hold for free-variable tableaux which are often used in implementations. The idea on which the improvement is based is the introduction of a subformula in both polarities.

* The author would like to thank Hans Tompits and the referees for their useful comments on an earlier version of this paper.
although the subformula may occur in one polarity only in the input formula, i.e., the formula to be proven.

Our observation was that different translations to definitional form, namely Eder’s variant [12] and the variant of Plaisted and Greenbaum [19], yield different normal forms, where, for some classes of formulae, the lengths of shortest proofs are non-elementarily related [14]. Both translations introduce labels abbreviating subformulae but the former translation introduces equivalences, whereas the latter one introduces implications whenever possible. Roughly speaking, if equivalences are used then not only the subformula is available in the resulting formula but also the negation of the subformula. It is well known from the literature that resolution combined with the latter translation polynomially simulates the cut-free LK-calculus but resolution combined with the former translation yields a stronger calculus with respect to proof length (see [13] for the propositional case). Therefore, we introduce formulae in different polarities by two mechanisms, namely by using modified branching rules[2] and by a translation of the input formula. As an example, consider the following [K]-rule for \( \rightarrow_l \).

\[
\Gamma \vdash \Delta_1, A, \Delta_2 \quad \Gamma_1, B, \Gamma_2 \vdash \Pi
\]

\[
\Gamma, \Gamma_1, (A \rightarrow B), \Gamma_2 \vdash \Delta_1, \Delta_2, \Pi \rightarrow l
\]

This rule is replaced by the following two rules.

\[
\Gamma \vdash \Delta_1, A, B, \Delta_2 \quad \Gamma_1, B, \Gamma_2 \vdash \Pi
\]

\[
\Gamma, \Gamma_1, (A \rightarrow B), \Gamma_2 \vdash \Delta_1, \Delta_2, \Pi
\]

With these new rules, some restricted forms of analytic cuts can be simulated resulting in a non-elementary speed-up in minimal proof length for some classes of formulae.[3]

The second possibility to introduce formulae in both polarities is a translation of the input formula \( F \). Instead of proving \( F \), the equivalent formula \( (\bigwedge_{A \in \Sigma(F)} (A \rightarrow A)) \rightarrow F \) is proved, where \( \Sigma(F) \) denotes the set of all subformulae of \( F \). Observe that the length of the resulting formula[4] is at most quadratic in the length of \( F \). With a standard cut-free sequent calculus and this translation scheme, analytic cuts can be simulated with only a moderate increase of proof length. This is important for existing implementations because sequent- or tableau-based theorem provers can be extended by analytic cut without modifying them. Generating a slightly extended formula can be done in an inexpensive preprocessing step.

One may object that the introduction of additional formulae increases the branching degree of nodes in the search space. This is a correct observation but we reply with the following three arguments. First, it is common mathematical practise to use lemmata (and, therefore, cuts) in proofs. These lemmata usually improve the structure and readability of proofs. Second, weaker forms of

---

[2] I.e., the \( \beta \)-rule in tableau notation.
[3] In [2], non-elementary speed-ups are obtained by an optimized \( \delta \)-rule. Roughly speaking, antiprenexing is incorporated into the \( \delta \)-rule, such that the non-elementary speed-up described in [3] can be reproduced in the tableau format.
[4] I.e., the number of symbol occurrences.
asymmetric $\beta$-rules (e.g., the folding-down rule in SETHEO or KOMET) and lemma mechanisms have been proved useful in practical applications. It is immediately apparent that such extensions do not improve the search for a proof for all formulae, but some of them can only be proved with these extensions, even if they increase the branching degree. A third argument concerns tableaux with lean induction. In [1], we introduce induction principles into tableau calculi with equality in a simple and elegant way. Induction is implemented by modified closure conditions for branches. Such tableau calculi can be strengthened\(^5\) by allowing analytic cuts or modified branching rules. For instance, if the axiom $E$ is deleted from Example 2 (a weak form of arithmetic) in [1], then the resulting formula is not provable in our tableau calculus with lean induction, but it becomes provable if analytic cuts are allowed.

The structure of the paper is as follows. In Section 2, definitions and notations are introduced. In Section 3, we present improved branching rules for LK. In Section 4, we show that these modified branching rules can enable non-elementary speed-ups of the length of shortest proofs. Section 5 is devoted to a simple translation of the input formula such that analytic cuts can be polynomially simulated by proving the result of the translation in LK without cut. Finally, we discuss related concepts relevant for theorem proving systems.

2 Definitions and Notations

We consider a usual first-order language with function symbols.

Definition 2.1

Given a formula $F$, we call $G$ an immediate $s$-subformula (structural subformula) of $F$ if either

1. $F$ is an atom and $G = F$, or
2. $F = (\neg G_1)$ and $G = G_1$, or
3. $F = (G_1 \circ G_2)$ ($\circ \in \{\land, \lor, \to\}$) and $G = G_1$ or $G = G_2$, or
4. $F = \forall x \, G_1(x)$ ($Q \in \{\forall, \exists\}$) and $G = G_1(x)$.

The relation “is $s$-subformula of” is the reflexive and transitive closure of the relation “is an immediate $s$-subformula of”. We call $G$ an immediate subformula of $F$ if either one of 1.–3. holds, or

4’. $F = \forall x \, G_1(x)$ ($Q \in \{\forall, \exists\}$) and $G = G_1(t)$ for an arbitrary term $t$.

The relation “is subformula of” is the reflexive and transitive closure of the relation “is an immediate subformula of”.

Definition 2.2

Let $F$ be a formula. $\Sigma(F)$ denotes the set of all $s$-subformulae of $F$ and $\forall F$ denotes the universal closure of $F$.

\(^5\) "Strengthening" here does not mean "proving with a shorter proof", but proving formulae which are unprovable in the weaker calculus.
Definition 2.3
Let \( A_1, \ldots, A_n \) and \( B_1, \ldots, B_m \) be first-order formulae. Then,
\[
A_1, \ldots, A_n \vdash B_1, \ldots, B_m
\]
is called a sequent. The informal meaning of the sequent is the same as the informal meaning of the formula \( (\bigwedge_{i=1}^n A_i) \rightarrow (\bigvee_{i=1}^m B_i) \).

Let \( S \) be either a formula or a sequent. An occurrence of a formula \( G \) occurs positively (negatively) in \( S \) if the number of explicit (\( \neg \)) or implicit (\( \rightarrow, \vdash \)) negation signs of \( G \) is even (odd). For instance, \( A \) and \( C \) occur positively in \( A \rightarrow B \vdash C \), whereas \( B \) occurs negatively in the same sequent.\(^6\)

We use the following “multiplicative” sequent calculus. In contrast to Gentzen’s original formulation in [16], rule applications are allowed at arbitrary places in the sequent. As a consequence, the exchange rule can be omitted. Let \( \Gamma, \Delta, \text{ and } \Lambda \) (possibly subscripted) denote sequences of formulae and let \( A, B, \) and \( F \) denote formulae.

The inference rules for LK are the logical rules, the quantifier rules, and the structural rules without cut. \( \text{LK}_{\text{cut}} \) is the calculus LK extended by the cut rule. For all sequent calculi in this paper, the initial sequents (or the axioms) are of the form \( \Gamma \vdash \Gamma \) for a formula \( \Gamma \).

**Logical Rules**

\[
\frac{\Delta_1, A, \Delta_2, B, \Delta_3 \vdash \Gamma}{\Delta_1, (A \land B), \Delta_2, \Delta_3 \vdash \Gamma} \quad \land l
\]

\[
\frac{\Gamma, \Delta_1, A, \Delta_2 \vdash \Pi_1, B, \Pi_2}{\Gamma, \Pi_1, (A \land B), \Pi_2 \vdash \Delta_1, \Delta_2} \quad \land r
\]

\[
\frac{\Gamma_1, A, \Gamma_2 \vdash \Delta_1, B, \Delta_2 \vdash \Gamma_3}{\Gamma_1, \Pi_1, (A \lor B), \Gamma_2, \Pi_2 \vdash \Delta_1, \Delta_2} \quad \lor l
\]

\[
\frac{\Gamma \vdash \Delta_1, A, \Delta_2, B, \Delta_3}{\Gamma \vdash \Delta_1, A, \Delta_2, B, \Delta_3} \quad \lor r
\]

\[
\frac{\Gamma \vdash \Delta_1, A, \Delta_2}{\Gamma \vdash \Delta_1, A, \Delta_2} \quad \rightarrow l
\]

\[
\frac{\Gamma, \Gamma_1, (A \rightarrow B), \Gamma_2 \vdash \Delta_1, \Delta_2, \Pi}{\Gamma, \Gamma_1, (A \rightarrow B), \Gamma_2 \vdash \Delta_1, \Delta_2, \Pi} \quad \rightarrow r
\]

\[
\frac{\Delta \vdash \Gamma_1, A, \Gamma_2}{\neg A, \Delta \vdash \Gamma_1, \Gamma_2} \quad l
\]

\[
\frac{\Gamma_1, A, \Gamma_2 \vdash \Delta}{\Gamma_1, \Gamma_2 \vdash \Delta, \neg A} \quad r
\]

For all logical rules, \( A \) and \( B \) are called auxiliary formulae and \( (A \land B) \), \( (A \lor B) \), \( (A \rightarrow B) \), and \( \neg A \) are called principal formulae of the corresponding \( \land, \lor, \rightarrow, \) and \( \neg \) rules. The inference rules \( \lor l, \rightarrow l \), and \( \land r \) are called branching rules.

**Quantifier Rules**

\[
\frac{\Delta_1, A(t), \Delta_2 \vdash \Gamma}{\Delta_1, \forall x A(x), \Delta_2 \vdash \Gamma} \quad \forall l
\]

\[
\frac{\Gamma \vdash \Delta_1, A(y), \Delta_2}{\Gamma \vdash \Delta_1, \forall x A(x), \Delta_2} \quad \forall r
\]

\(^6\) Positive resp. negative occurrences coincide with positive resp. negative parts in [22].
∀r and ∃l must fulfill the eigenvariable condition, i.e., the (free) variable y does not occur in Γ, Δ₁, Δ₂, or A(x). The term t is any term not containing a bound variable. A(t) and A(y) are auxiliary formulae and ∀xA(x) and ∃xA(x) are principal formulae.

**Structural Rules**

**Weakening**

\[
\frac{\Gamma \vdash \Delta, \Delta_1, \Delta_2 \vdash \Gamma}{\Gamma \vdash \Delta, \Delta_1, \Delta_2 \vdash \Gamma} \text{ \(wl\)}
\]

\[
\frac{\Delta \vdash \Gamma_1, \Gamma_2}{\Delta \vdash \Gamma_1, \Gamma_2} \text{ \(wr\)}
\]

A is called the **weakening formula**.

**Contraction**

\[
\frac{\Gamma_1, A, \Gamma_2, A, \Gamma_3 \vdash \Delta}{\Gamma_1, \Gamma_2, \Gamma_3 \vdash \Delta} \text{ \(cl\)}
\]

\[
\frac{\Delta \vdash \Gamma_1, A, \Gamma_2, A, \Gamma_3}{\Delta \vdash \Gamma_1, A, \Gamma_2, A, \Gamma_3} \text{ \(cr\)}
\]

**Cut**

\[
\frac{\Gamma \vdash \Delta_1, A, \Delta_2 \quad \Delta_1, A, \Delta_2 \vdash \Pi}{\Gamma, \Delta_1, A, \Delta_2 \vdash \Delta_1, \Delta_2, \Pi} \text{ \(cut\)}
\]

A is called the **cut formula**.

We say that inference I **introduces** F if F is the principal or weakening formula of I.

Let LK be a meta-variable for sequent calculi. A **derivation** (in tree form) of a sequent S in LK is defined as usual.

**Definition 2.4**

An application of the cut rule in an LKcut-derivation of \( \vdash F \) is called **analytic** if the cut formula is a subformula of F. LKacut is the calculus LKcut, where all cuts are analytic.

Analytic cuts preserve the subformula property because the cut formula occurs (as a subformula) in the formula of the end sequent.

**Definition 2.5**

A sequence of sequents in an LK-derivation \( \alpha \) of S is called a **branch** of \( \alpha \) if the following conditions are satisfied.

1. The sequence begins with an initial sequent and ends with S.
2. Every sequent in the sequence except S is an upper sequent of an inference I, and is immediately followed by the lower sequent of I.

A path is a partial branch beginning in a sequent (not necessarily an initial sequent) and ending with S.
Definition 2.6
The length of a formula \( F \), denoted by \(|F|\), is the number of symbol occurrences in the string representation of \( F \). If \( \Delta = F_1, \ldots, F_n \) or \( \Delta = \{F_1, \ldots, F_n\} \), then \(|\Delta| = \sum_{i=1}^{n} |F_i|\). For a sequent \( S \) of the form \( \Gamma \vdash \Delta, |S| = |\Gamma| + |\Delta| \). The length of an \( \mathcal{LK} \)-derivation \( \alpha \), denoted by \(|\alpha|\), is \( \sum_{S \in T} |S| \), where \( T \) is the multiset of sequences occurring in \( \alpha \). The number of sequents occurring in \( \alpha \) is denoted by \( \#\text{seq}(\alpha) \). The height \( h(\alpha) \) of \( \alpha \) is the number of sequents occurring on the longest branch.

Let \( s \) be the non-elementary function with \( s(0) = 1 \) and \( s(n+1) = 2s(n) \) for all \( n \in \mathbb{N} \). The following definition of a polynomial simulation (resp. an elementary simulation) is adapted from \[12\] and restricted to the case that the connectives in both calculi are identical.

Definition 2.7
A calculus \( P_1 \) can polynomially simulate (elementarily simulate) a calculus \( P_2 \) if there is a polynomial \( p \) (an elementary function \( e \)) such that the following holds. For every proof of a formula \( F \) in \( P_2 \) of length \( n \), there is a proof of \( F \) in \( P_1 \), whose length is not greater than \( p(n) \) (\( e(n) \)).

3 Alternative Branching Rules

In this section, we define \( \mathcal{LK}^a \), an asymmetric variant of \( \mathcal{LK} \) for first-order logic by replacing each of the branching rules \( \forall l \), \( \forall r \), and \( \rightarrow l \) by two variants. Moreover, we show that \( \mathcal{LK}^a \) is sound and complete. In the next section, we demonstrate that these modified branching rules are strongly related with the analytic cut rule.

We replace \( \forall l \) by \( \forall l_1 \) and \( \forall l_2 \), \( \forall r \) by \( \forall r_1 \) and \( \forall r_2 \), and \( \rightarrow l \) by \( \rightarrow l_1 \) and \( \rightarrow l_2 \) and obtain the calculus \( \mathcal{LK}^a \) from \( \mathcal{LK} \). The new branching rules are as follows.

\[
\frac{\Gamma, A, \Gamma_2 \vdash \Delta_1, B}{\Gamma, \Pi_1, (A \lor B), \Gamma_2, \Pi_2 \vdash \Delta_1, \Delta_2} \quad \forall l_1
\]

\[
\frac{\Gamma, A, \Gamma_2 \vdash \Delta_1, B \quad \Pi_1, B, \Pi_2 \vdash \Delta_2, A}{\Gamma, \Pi_1, (A \lor B), \Gamma_2, \Pi_2 \vdash \Delta_1, \Delta_2} \quad \forall l_2
\]

\[
\frac{\Gamma, B \vdash \Delta_1, A, \Delta_2 \quad A \vdash \Pi_1, B, \Pi_2}{\Gamma, A \vdash \Delta_1, \Pi_1, (A \land B), \Delta_2, \Pi_2} \quad \land r_1
\]

\[
\frac{\Gamma \vdash \Delta_1, A, \Delta_2 \quad A \vdash \Pi_1, B, \Pi_2}{\Gamma, \Pi_1, (A \land B), \Delta_2, \Pi_2} \quad \land r_2
\]

\[
\frac{\Gamma \vdash \Delta_1, A, B, \Delta_2 \quad \Gamma_1, B, \Gamma_2 \vdash \Pi}{\Gamma, \Gamma_1, (A \rightarrow B), \Gamma_2 \vdash \Delta_1, \Delta_2, \Pi} \quad \rightarrow l_1
\]

\[
\frac{\Gamma \vdash \Delta_1, A, B \quad \Gamma_1, B, A, \Gamma_2 \vdash \Pi}{\Gamma, \Gamma_1, (A \rightarrow B), \Gamma_2 \vdash \Delta_1, \Delta_2, \Pi} \quad \rightarrow l_2
\]

Obviously, \( \mathcal{LK}^a \) is an analytic calculus and the subformula property is retained by these replacements of inference rules. What is new is the possibility to get a subformula occurring in one polarity only in the lower sequent in different polarities in the upper sequents of the new branching rules. It is immediately apparent by the translations below that, for any \( \mathcal{LK} \)-derivation \( \alpha \) of \( \vdash F \), there
exists an $\text{LK}^a$-derivation $\alpha^a$ of the same end sequent and $|\alpha^a| \leq 2 \cdot |\alpha|$. Hence, $\text{LK}^a$ polynomially simulates $\text{LK}$ but the reverse simulation is not elementary as we will demonstrate in the next section.

We consider a transformation of $\text{LK}$-derivations into $\text{LK}^a$-derivations of the same end sequent. An application of $\forall l$ of the form

$$
\frac{\Gamma_1, A, \Gamma_2 \vdash \Delta_1 \quad \Pi_1, B, \Pi_2 \vdash \Delta_2}{\Gamma_1, \Pi_1, (A \lor B), \Gamma_2, \Pi_2 \vdash \Delta_1, \Delta_2} \, \forall l
$$

is replaced by

$$
\frac{\Gamma_1, A, \Gamma_2 \vdash \Delta_1, B \quad \Pi_1, B, \Pi_2 \vdash \Delta_2}{\Gamma_1, \Pi_1, (A \lor B), \Gamma_2, \Pi_2 \vdash \Delta_1, \Delta_2} \, \forall l_1
$$

The replacements for the rules $\rightarrow l$ and $\land r$ are similar to the one presented above, i.e., the necessary formula is introduced by weakening above the new branching rule. We get the following theorem.

**Theorem 3.1**

$\text{LK}^a$ polynomially simulates $\text{LK}$.

We consider a transformation of $\text{LK}^a$-derivations into $\text{LK}^a_{\text{cut}}$-derivations of the same end sequent. An application of $\forall l_1$ of the form

$$
\frac{\Gamma_1, A, \Gamma_2 \vdash \Delta_1, B \quad \Pi_1, B, \Pi_2 \vdash \Delta_2}{\Gamma_1, \Pi_1, (A \lor B), \Gamma_2, \Pi_2 \vdash \Delta_1, \Delta_2} \, \forall l_1
$$

is replaced by

$$
\frac{\Gamma_1, (A \lor B), \Gamma_2 \vdash \Delta_1, B \quad \Pi_1, B, \Pi_2 \vdash \Delta_2}{\Gamma_1, \Pi_1, (A \lor B), \Gamma_2, \Pi_2 \vdash \Delta_1, \Delta_2} \, \forall l_2
$$

$$
\frac{\Gamma_1, (A \lor B), \Gamma_2 \vdash \Delta_1, B \quad \Pi_1, B, \Pi_2 \vdash \Delta_2}{\Gamma_1, \Pi_1, (A \lor B), \Gamma_2, \Pi_2 \vdash \Delta_1, \Delta_2} \, \forall r_1
$$

$$
\frac{\Gamma_1, (A \lor B), \Gamma_2 \vdash \Delta_1, B \quad \Pi_1, B, \Pi_2 \vdash \Delta_2}{\Gamma_1, \Pi_1, (A \lor B), \Gamma_2, \Pi_2 \vdash \Delta_1, \Delta_2} \, \forall r_2
$$

The replacements for the new rules $\forall l_2$, $\rightarrow l_1$, $\rightarrow l_2$, $\land r_1$, and $\land r_2$ are similar to the one presented above. We get the following theorem.

**Theorem 3.2**

$\text{LK}^a_{\text{cut}}$ polynomially simulates $\text{LK}^a$.

Theorems 3.1 and 3.2 imply the following one.

**Theorem 3.3**

$\text{LK}^a$ is sound and complete.

We will demonstrate in the next section that the new branching rules indeed have the power of cut for some class of formulae. Observe that the cuts, introduced by the translation, are analytic cuts.
4 A Non-Elementary Speed-Up Result

In this section, we compare $\text{LK}$ with $\text{LK}^a$. Our discussion is based on a sequence $H_1, H_2, \ldots$ of formulae which are modified and extended variants of formulae $F_1, F_2, \ldots$ presented in [18].

Definition 4.1

Let $F_k$ occur in the infinite sequence of formulae $(F_k)_{k \in \mathbb{N}}$ where

$$F_k = \forall b \left( (\forall w_0 \exists v_0 \, p(w_0, b, v_0) \wedge \forall w_0 \exists v_0 \, p(w_0, b, v_0) \wedge \forall z \left( p(v, y, z) \wedge p(z, y, w) \right) \rightarrow p(v, u, w) \right) \wedge \ldots \wedge \exists v_0 \, p(b, v_1, v_0) \ldots .$$

Using this sequence $(F_k)_{k \in \mathbb{N}}$, Orevkov showed that cut elimination can tremendously affect proof length. More precisely, he proved that there exists an $\text{LK}_{\text{cut}}$-derivation of $\vdash F_{Fa}$ with a single occurrence of the cut rule and the number of sequents in this derivation is linear in $k$, but any cut-free $\text{LK}$-derivation of $\vdash F_{Fa}$ has height $\geq 2 \cdot s(k) + 1$. We first sketch a slightly modified version of Orevkov's $\text{LK}_{\text{cut}}$-derivation with one application of the cut rule. Let $\psi_k$ be this $\text{LK}_{\text{cut}}$-derivation. The cut is then changed to an analytic cut by extending $F_k$ by $(q \lor \neg q) \lor A$, where $A$ is the cut formula and $q$ is an atom neither occurring in $F_k$ nor in $A$. The remaining derivation is adjusted accordingly. Instead of $F_k$, we get an extended formula $H_k$. Then we show that any $\text{LK}$-derivation of $\vdash H_k$ has length $> (2 \cdot s(k))^{1/2}$ and that we can simulate the cut in the derivation of $\vdash H_k$ by the modified branching rules. Hence, we obtain a short cut-free derivation of $\vdash H_k$ in $\text{LK}^a$.

Abbreviations shown in Figure 1 are used in the following in order to simplify the notation. We do not present the $\text{LK}_{\text{cut}}$-derivation of $\vdash F_k$ in all details but sketch the proof stressing the relevant details. Slightly differing $\text{LK}_{\text{cut}}$-derivations of $F_k$ can be found in [18] or in [14].

There are two kinds of $\text{LK}$-derivations, namely $\beta_k$ and $\delta_k(b_0)$, which are relevant in the following for $k > 0$. For $k = 0$, only $\delta_0(b_0)$ is relevant, which is an $\text{LK}$-derivation of the sequent $A_0(b_0), C \vdash B_0(b_0)$. Then, the $\text{LK}$-derivation of $\vdash F_0$, denoted by $\psi_0$, is as follows.

$$\frac{\delta_0(b_0)}{\vdash \forall b \left( (A_0(b) \land C) \rightarrow B_0(b) \right) \land, \rightarrow r, \forall r}$$

For $k > 0$, $\beta_k$ is an $\text{LK}$-derivation of $A_0(b_0), C \vdash A_k(b_0)$ and $\delta_k(b_0)$ is an $\text{LK}$-derivation of $A_0(b_0), C, A_k(b_0) \vdash B_k(b_0)$. Then, $\psi_k$ is as follows.

$$\frac{\beta_k \quad \delta_k(b_0)}{\vdash \forall b \left( (A_0(b) \land C) \rightarrow B(k)(b) \right) \land, \rightarrow r, \forall r \quad \text{cut, cl, cl}}$$

The derivation $\psi_k$ of $\vdash F_k$ in $\text{LK}_{\text{cut}}$ discussed so far has one application of the cut rule where the cut formula has a free variable. We transform $\psi_k$ into $\phi_k$ and make the application of the cut rule analytic.
\[ C_1(\alpha, \beta, \gamma) = \exists z (p(\alpha, \beta, z) \land p(z, \beta, \gamma)) \]
\[ C_2(\alpha, \beta, \gamma) = \exists y (p(y, b_0, \alpha) \land C_1(\beta, y, \gamma)) \]
\[ C = \forall u \forall v w (C_2(u, v, w) \rightarrow p(v, u, w)) \]
\[ B_0(\alpha) = \exists v_0 p(b_0, \alpha, v_0) \]
\[ B_{i+1}(\alpha) = \exists v_{i+1} (p(b_0, \alpha, v_{i+1}) \land B_i(v_{i+1})) \]
\[ A_0(\alpha) = \forall w_0 \exists v_0 p(w_0, \alpha, v_0) \]
\[ A_{i+1}(\alpha) = \forall w_{i+1} (A_i(w_{i+1}) \rightarrow A_{i+1}(\alpha)) \]
\[ \overline{A}_{i+1}(\alpha, \delta) = \exists v_{i+1} (A_i(v_{i+1}) \land p(\alpha, \delta, v_{i+1})) \]

Fig. 1. Abbreviations used in the following.

**Definition 4.2**

Let \( H_k \) \((k \in \mathbb{N})\) be a formula of the form
\[
\forall b (((q \lor \neg q) \lor A_k(b)) \land (A_0(b) \land C)) \rightarrow B_k(b)
\]
where \( q \) is a predicate with a predicate symbol not occurring elsewhere in \( A_i(b), B_i(b) \) \((0 \leq i \leq k)\), and \( C \).

An \( \text{LK}_{\text{cut}} \)-derivation of \( \vdash H_k \) is obtained from the derivation of \( \vdash F_k \) in \( \text{LK}_{\text{cut}} \) presented above by simply adding a \( \text{wl} \)-inference with weakening formula \((q \lor \neg q) \lor A_k(b)\) directly below \( \beta_k (k \geq 1) \) or directly below \( \delta_0(b_0) \) \((k = 0)\). Then \( \phi_0 \) and \( \phi_k \) \((k \geq 1)\) are as follows.
\[
\delta_0(b_0)
\]
\[
(\neg \neg q) \lor A_0(b_0), A_0(b_0), C \vdash B_0(b_0) \quad \text{wl}
\]
\[
((\neg \neg q) \lor A_0(b_0)) \land (A_0(b_0) \land C) \vdash B_0(b_0) \quad \text{l, l}
\]
\[
\vdash \forall b (((\neg \neg q) \lor A_0(b)) \land (A_0(b) \land C)) \rightarrow B_0(b))
\]

\[
\beta_k
\]
\[
(\neg \neg q) \lor A_k(b_0), A_0(b_0), C \vdash A_k(b_0) \quad \text{wl}
\]
\[
(\neg \neg q) \lor A_k(b_0), A_0(b_0), C \vdash B_k(b_0) \quad \text{cut, l, l, cl}
\]
\[
((\neg \neg q) \lor A_k(b_0)) \land (A_0(b_0) \land C) \vdash B_k(b_0) \quad \text{l, l}
\]
\[
\vdash \forall b (((\neg \neg q) \lor A_k(b)) \land (A_0(b) \land C)) \rightarrow B_k(b))
\]

The \( \text{LK}_{\text{cut}} \)-derivation \( \phi_k \) has only one application of the analytic cut rule. Hence, \( \phi_k \) is an \( \text{LK}_{\text{acut}} \)-derivation of \( \vdash H_k \).

**Lemma 4.1**

Let \( \phi_k \) be the \( \text{LK}_{\text{acut}} \)-derivation of \( \vdash H_k \) with one application of an analytic cut described above. Then, the length of \( \phi_k \) is \( \leq c \cdot 2^{d \cdot k} \) for constants \( c, d \).
Proof. By Theorem 2 in [18], the number of sequents in the LK_{cut}-derivation of $\vdash F_k$ is less than $e \cdot k$, where $e$ is a constant. Since $|(q \lor \neg q) \lor A_k| \leq a \cdot 2^{b \cdot k}$ for constants $a, b$, there exist constants $c, d$ such that the length of $\phi_k$ is $\leq c \cdot 2^{d \cdot k}$. □

Next, we transform the LK_{cut}-derivation $\phi_k$ with one occurrence of an analytic cut into a cut-free LK^a-derivation of the same end sequent.

**Lemma 4.2**
There exists an LK^a-derivation $\rho_k$ of $\vdash H_k$ such that the length of $\rho_k$ is $\leq c \cdot 2^{d \cdot k}$ for constants $c, d$.

Proof. The derivations $\rho_0$ and $\rho_k$ ($k \geq 1$) are as follows.

\[
\begin{align*}
& A_0(b_0) \vdash A_0(b_0) \\
& \frac{q \lor \neg q, A_0(b_0) \vdash A_0(b_0)}{wl} \quad \delta_0(b_0) \quad \forall l_1 \\
& \frac{(q \lor \neg q) \lor A_0(b_0), A_0(b_0), C \vdash B_0(b_0)}{\land l, \land} \\
& \vdash \forall b (((q \lor \neg q) \lor A(b)) \land (A_0(b) \land C)) \rightarrow B_0(b) \\
& A_k(b_0) \vdash A_k(b_0) \\
& \frac{q \lor \neg q, A_k(b_0) \vdash A_k(b_0)}{wl} \quad \delta_k(b_0) \quad \forall l_1, cl, cl \\
& \frac{(q \lor \neg q) \lor A_k(b_0), A_0(b_0), C \vdash B_k(b_0)}{\land l, \land} \\
& \vdash \forall b (((q \lor \neg q) \lor A_k(b)) \land (A_0(b) \land C)) \rightarrow B_k(b)
\end{align*}
\]

Hence, we have an LK^a-derivation $\rho_k$ of $\vdash H_k$ and the length of $\rho_k$ is $\leq c \cdot 2^{d \cdot k}$ for constants $c, d$. □

In contrast to the short LK_{cut}-derivation of $\vdash F_k$, any derivation of the same end sequent in LK has length non-elementary in $k$. The following lemma is a corollary of Theorem 1 in [18].

**Lemma 4.3**
Let $C_k$ be an LK-derivation of $\vdash F_k$. Then, $h(\psi_k) \geq 2 \cdot s(k) + 1$.

As a consequence, any derivation of $\vdash F_k$ in LK has length $\geq 2 \cdot s(k) + 1$. In the remainder of this section, we prove a non-elementary lower bound on the length of any LK-derivation of $\vdash H_k$.

**Lemma 4.4**
Let $\phi$ be an LK-derivation of $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$. Then, $\#\text{seq}(\phi) \geq n + m - 1$.

Proof. By induction on the structure of $\phi$. □

**Lemma 4.5**
Let $\phi_k$ be an LK-derivation of $\vdash H_k$. Then, $|\phi_k| > (2 \cdot s(k))^{1/2}$. 
Proof. There are two possible inferences by which the formula $Q = (q \lor \neg q) \lor A_k(b_0)$ can be introduced into $\phi_k$, namely $\forall l$ and $\forall l$. We first eliminate all occurrences of $Q$ introduced by $\forall l$. Then, all occurrences of $Q$ introduced by $\forall l$ are eliminated. The resulting derivation $\phi_k'$ is an LK-derivation of $\vdash F_k$ and $|\phi_k'| > 2 \cdot s(k)$. A simple calculation yields the desired result.

Step 1. $Q$ is introduced by $\forall l$. Select the first $\forall l$-inference (with respect to some tree ordering) such that $\alpha_1$ and $\alpha_2$ do not have an $\forall l$-inference with principal formula $Q$. If there is no such inference, then goto Step 2. Otherwise, this first inference has the following form ($I_1$ and $I_2$ are LK-inferences.):

\[
\frac{\alpha_1}{\Gamma_1, q \lor \neg q, \Gamma_2 \vdash \Delta_1} I_1 \quad \frac{\alpha_2}{\Pi_1, A_k(b_0), \Pi_2 \vdash \Delta_2} I_2 \quad \forall l
\]

\[
\beta
\]

Construct an LK-derivation of $\vdash H_k$ of the form

\[
\frac{\alpha'_1}{\Gamma_1, \Gamma'_2 \vdash \Delta'_1} I'_1 \quad \frac{\alpha_2}{\Pi_1, \Pi_2, \Gamma_2 \vdash \Delta_1, \Delta_2} I_2 \quad \forall l
\]

\[
\beta
\]

\[
\vdots \quad \text{wl, wr} \quad (*)
\]

\[
\vdots \quad \text{wl, wr} \quad (**)\]

where $\alpha'_1$ is obtained from $\alpha_1$ by omitting all weakenings introducing formulae $q$, $\neg q$, or $q \lor \neg q$ and by omitting contractions upon such formulae. Moreover, all $\forall l$-inferences with principal formula $q \lor \neg q$ are omitted. $I'_1$ is either $I_1$ or an inference occurring in $\alpha_1$. The number of sequents occurring in the resulting derivation is not greater than the number of sequents in $\phi_k$ because $q$, $\neg q$, and $q \lor \neg q$ occurring in $\Gamma_1$, $\Gamma_2$, or $\Delta_1$ are re-introduced at $(*)$, and $\#seq(\alpha_2)$ is not less than the number of $\text{wl}$ and $\text{wr}$ in $(**)$). Replace all such $\forall l$-inferences without increasing the number of sequents resulting in an LK-derivation where all occurrences of $Q$ are introduced by $\forall l$.

Step 2. Omit all $\forall l$ introducing $q$, $\neg q$, $q \lor \neg q$, or $Q$ and adjust the derivation by omitting all contractions upon these formulae and all inferences with auxiliary formulae $q$, $\neg q$, $q \lor \neg q$ or $Q$. Since all occurrences of $Q$ are introduced by $\forall l$, $\vdash F_k$ is derived.

Now, we have an LK-derivation $\phi'_k$ of $\vdash F_k$, and $\#seq(\phi'_k) \leq \#seq(\phi_k)$. Since any sequent occurring in $\phi'_k$ has length less than $|\phi_k|$, the increase of length is at most quadratic. Now, $2 \cdot s(k) < |\phi'_k| \leq |\phi_k|^2$ and $|\phi_k|$ is greater than $(2 \cdot s(k))^{1/2}$.

Combining Lemma 4.2 and Lemma 4.5 yields the following theorem.
Theorem 4.1
LK cannot elementary simulate \( \text{LK}^a \).

Observe that similar results also hold for closely related calculi including free-variable tableaux. Moreover, the search space decreases non-elementarily (for some classes of formulae) if \( \text{LK}^a \) is applied instead of LK.

5 A Simple Transformation of the Input Formula

In this section, we introduce a simple translation scheme based on the introduction of closed formulae \( \forall (A \rightarrow A) \) for any s-subformula \( A \) of the given formula \( F \). We show that \( \text{LK}_{\text{acut}} \) can be polynomially simulated by this transformation in combination with LK. Moreover, the reverse simulation is also polynomial. We start with the definition of the implicational form of a formula.

Definition 5.1
Let \( \Sigma(F) \) denote the set of all s-subformulae of a given formula \( F \). Then, the implicational form of \( F \), denoted by \( \im(F) \), is defined as follows.

\[
\im(F) = \bigwedge_{A \in \Sigma(F)} \forall (A \rightarrow A)
\]

Instead of proving \( \vdash F \), the sequent \( \vdash \im(F) \rightarrow F \) is considered. This is possible because \( F \) is equivalent to \( \im(F) \rightarrow F \). The length of \( \im(F) \) is estimated as follows. Since the number of s-subformulae of \( F \) is \( \leq |F| \),

\[
|\im(F)| \leq |F| \cdot |F \rightarrow F| \leq |F| \cdot (2 \cdot |F| + 1).
\]

Definition 5.2
A sequent \( \vdash F \) is derivable in \( \text{LK}_{\rightarrow} \) if \( \vdash \im(F) \rightarrow F \) is derivable in LK.

Theorem 5.1
\( \text{LK}_{\rightarrow} \) polynomially simulates \( \text{LK}_{\text{acut}} \).

Proof Idea. Let \( \phi \) be a derivation of \( \vdash F \) in \( \text{LK}_{\text{acut}} \) and let \( A \) be an s-subformula of \( F \). We show that replacing an application of cut with cut formula \( A \) by an application of \( \rightarrow l \) and \( \forall l \)-inferences to close the resulting formula \( A \rightarrow A \) yield an \( \text{LK}_{\text{acut}} \)-derivation with only a moderate increase of derivation length. After the replacements of all analytic cuts, we eventually get an \( \text{LK}_{\rightarrow} \)-derivation of \( \vdash F \) after some additional inferences.

Theorem 5.2
\( \text{LK}_{\text{acut}} \) polynomially simulates \( \text{LK}_{\rightarrow} \).
Proof Idea. Let $\phi$ be a derivation of $\vdash F$ in $\text{LK}_\to$, i.e., an $\text{LK}$-derivation of $\vdash \iota(F) \rightarrow F$. The idea is to replace some $\rightarrow$-inferences with principal formula $A \rightarrow A$ by an analytic cut and $\text{wl}$.

Consider an application of $\rightarrow l$ with lower sequent $S$ in $\phi$ (shown on the left below) such that the length of the path $\beta$ is minimal ($I_1$ and $I_2$ are $\text{LK}$-inferences).

$$
\frac{\alpha_1}{\Gamma \vdash \Delta_1, A, \Delta_2} \quad \frac{\alpha_2}{I_1} \quad I_2 \\
\frac{(A \rightarrow A), \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, \Pi}{\beta} \\
\frac{\Gamma \vdash \Delta_1, A, \Delta_2}{\beta'} \\
\frac{I_1, A, \Gamma_2 \vdash \Pi}{\text{cut}}
$$

Replace the left inference figure by the right inference figure, where sequents in $\beta'$ are adjusted accordingly. Iterating this elimination of $\rightarrow l$, deleting some weakenings and contractions yields an $\text{LK}_{\text{acut}}$-derivation $\psi$ of $\vdash F$, because only such formula occurrences are deleted which occur in $\iota(F)$. The length of $\psi$ is $\leq |\phi|$.

Theorem 5.1 and Theorem 3.2 imply

Theorem 5.3

$\text{LK}_\to$ and $\text{LK}_{\text{acut}}$ polynomially simulate $\text{LK}^a$.

This theorem, together with Lemma 4.2 imply the following

Corollary 5.1

There exists an $\text{LK}_\to$-derivation of $\vdash H_k$ with length exponential in $k$.

6 Conclusion and Discussion

We have shown that a slight modification of the branching rules of an $\text{LK}$-calculus yields a non-elementary decrease of proof length for a sequence of formulae. The reason for such a tremendous "speed-up" by the new rules is the utilization of both polarities of a subformula of $F$, which occurs only in one polarity in $\vdash F$.

The idea to use asymmetric branching rules in classical propositional sequent calculi is not new. Prawitz [21] (see also [5, 6]) used a similar idea in propositional logic. He introduced a reduction which can be simulated by (a sequence of) asymmetric branching rules and contractions.

More recently, D'Agostino showed in [11] that a class $(T_n)_{n \in \mathbb{N}}$ of propositional formulae, which was used by Cook and Reckhow [10] to indicate that analytic tableaux are not super, possess short proofs (in the length of the input formula) in a tableau system with modified branching rules, but any proof of a formula from this class in a standard tableau system has exponential length. He observed that, for $(T_n)_{n \in \mathbb{N}}$, the truth table method is computationally superior to analytic semantic tableaux. Semantically, a $\beta$-rule represents a case analysis, e.g., $A \lor B$ is true if either $A$ is true or $B$ is true. The observation resulting in asymmetric $\beta$-rules is the possibility to assume the negation of one case in
the other without violating correctness. Hence, we can additionally assume \( \neg A \)
in case \( B \), or \( \neg B \) in case \( A \). The formulae \((T_n)_{n \in \mathbb{N}}\) have short resolution proofs (even in tree resolution) and short tableau proofs if a linear (sequence) form is allowed, i.e., the tableau derivation has dag\(^7\) form instead of tree form. Bibel [6] showed that \( T_n \), which he calls "complete matrix in \( n \) variables", possess short proofs in connection calculi allowing a kind of factorization.

In contrast to the propositional case and \((T_n)_{n \in \mathbb{N}}\), where resolution yields short proofs, it can be shown that any resolution proof of \( H_k \) must have length non-elementary in the length of the input formula if the formula is translated to clause form using the standard method (without introducing additional definitions for subformulae). Hence, we get a non-elementary speed-up of \( \text{LK}^\alpha \) over resolution in the first-order case. Observe that not only proof length decreases non-elementarily, but also the size of the search space if, for instance, breadth-first search is assumed. The reason is that the increase of the size of the search space by the additional formulae introduced by the modified rules is only elementary, which is more than compensated by the much shorter proof length. Similar results for the search space hold for \( \text{LK}^{\rightarrow} \) and \( \text{LK}_{\text{acut}} \).

Another phenomenon has been observed in [20]. Shannon graphs (or Binary Decision Diagrams) use essentially the idea of the modified \( \beta \)-rules. In some sense, BDDs mimic the behavior of tableau systems with modified \( \beta \)-rules.

Finally, the folding-down operation in clausal connection tableaux implements the modified \( \beta \)-rules restricted to clauses. It is shown in [17] that folding-down can be viewed as an implementation technique for factorization in these tableaux.

References


\(^7\) directed acyclic graph