TECHNISCHE UNIVERSITÄT WIEN

VIENNA UNIVERSITY OF TECHNOLOGY

DISSERTATION

Proof Transformations by Resolution

Computational Methods of Cut-Elimination

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften unter der Leitung von

Univ.Prof. Dr.phil. Alexander Leitsch
Institut für Computersprachen (E185)
Arbeitsgruppe Theoretische Informatik und Logik

eingereicht an der Technischen Universität Wien
Fakultät für Informatik

von

Clemens Richter
Mat.-Nr.: 9825117
Adlergasse 16/5/6
A-2512 Tribuswinkel

Tribuswinkel, am 22. Mai 2006 ....

(Clemens Richter)

Kurzfassung

In der Beweistheorie werden Beweise als formale Objekte repräsentiert. Schnittelimination ist eine auf Gentzen zurückzuführende Disziplin, welche eine bestimmte Regel — die Schnittregel — aus diesen formalen Beweisen entfernt. Diese Technik der Beweistransformation hat sich zu einer der bedeutendsten Disziplinen der Beweistheorie entwickelt. Der Effekt der Schnittelimination ist die Entfernung aller Anwendungen von Hilfsaussagen (Lemmata) innerhalb eines Beweises, was wiederum zu einem analytischen Beweis, im Sinne dass alle Aussagen innerhalb des Beweises Teilformeln des sich ergebenden Satzes sind, führt.

Die Schnitteliminationsmethode CERES (Schnittelimination mittels Resolution) analysiert zunächst die konkrete Beweisstruktur und bildet diese auf eine unerfüllbare Klauselmenge ab. Eine Resolutionswiderlegung dieser Klauselmenge dient dann als Skelett für eine analytische Variante des Beweises.

Die Gleichheit ist ein zentrales Paradigma in der Mathematik und spielt eine Schlüsselrolle in der automatischen Deduktion. Diese Bedeutung weckt daher die Notwendigkeit die Gleichheit in bestehende Schnitteliminationskonzepte zu integrieren. In dieser Arbeit wird eine Erweiterung von CERES zu CERESe, durch das Hinzufügen der Gleichheit in Form von Regeln zu den zugrunde liegenden Sequentialkalkülen, präsentiert wobei alle Vorzüge von CERES erhalten bleiben. Insbesondere bleibt CERESe den Schnittreduktionsystemen ähnlich zu Gentzen's Ansatz überlegen, sie ist flexibel bezüglich Resolution in Verbindung mit Paramodulation und all deren Verfeinerungen und erlaubt eine semantische Verwendung des Schnitts. Wir erweitern CERES auch durch das Konzept der Gleichheitstheorien was neben den bereits bestehenden Vorteilen der Methode zu einem System führt, das in erster Linie eine Vereinfachung der Beweisnotation mit sich bringt.

Weiters wird eine Implementierung von CERES in Form eines Computersystems zur Schnittelimination vorgestellt, welches die Transformation von konkreten, mathematisch relevanten Beweisen erlaubt. Das System unterstützt die beweistheoretische Analyse von Beweisen und ist auch in der Lage schnittfreie Beweisvarianten zu erzeugen, die sich hinsichtlich ihrer Argumentation deutlich unterscheiden. Dies ist die erste Implementierung eines solchen Systems und könnte der Anfang einer neuen Ära von computergestützter Beweistheorie sein. Einige Experimente mit konkreten, aussagekräftigeren Beweisen, die bereits einen gewissen Komplexitätsgrad aufweisen runden die Demonstration des Systems ab.

Proof Transformations by Resolution

Computational Methods of Cut-Elimination

by

Clemens Richter

DOCTORAL THESIS
AT THE
VIENNA UNIVERSITY OF TECHNOLOGY

Abstract

In proof theory mathematical proofs are represented as formal objects. Cut-elimination is a discipline due to Gentzen removing a certain rule — the cut rule — from these formal proofs. This proof transformation technique has advanced to one of the most important disciplines in proof theory. The effect of cut-elimination is the removal of all applications of intermediate statements (lemmas) within a proof resulting in a proof that is analytic in the sense, that all statements of the proof are subformulas of the resulting theorem.

The cut-elimination method CERES (cut-elimination by resolution) first analyzes and maps a proof structure to a clause term which evaluates to an unsatisfiable set of clauses. A resolution refutation of this clause set then serves as a skeleton for an analytic variant of the proof.

Equality is a central paradigm in mathematics and plays a key role in automated deduction. Therefore its importance awakes the necessity of integrating equality into existing cut-elimination concepts. In this thesis an extension of CERES to CERESe by adding equality in form of rules to the underlying sequent calculi is presented where all the benefits of CERES are preserved. In particular CERESe is superior to Gentzen like cut-reduction systems, it is flexible with respect to resolution in conjunction with paramodulation and all its refinements and admits a semantical use of cut. We also enrich CERES by the concept of equational theories yielding a system which adds mainly simplicity of proof notations to the existing advantages of the method.

Furthermore an implementation of CERES as a computational system for cut-elimination is presented allowing the transformation of concrete mathematically relevant proofs. The system supports the proof theoretical analysis of proofs and is also capable of generating variants of cut-free proofs which are clearly distinguishable by their argumentation. This is the first implementation of such a system and could be the beginning of a new era of computational proof theory. Some experiments with concrete proofs of some relevance which already have a certain complexity round off the demonstration of the system.

To Kathi

This kind of support must be love.

Contents

Preface

This thesis is the result of a lot of work as well as in theoretical as in computational perspectives. Work that couldn't have been done without the constant support of Alexander Leitsch.

Since my first lecture in mathematical logic in the winter term of 1998 I was regularly following his courses. He arouse my interest in mathematical logic, inspired me to a deeper insight into recursion theory which culminated in my master's thesis (Richter 2003) on the importance of the diagonalization method in recursion theory and logic under his advice. The preoccupation with the work of Kurt Gödel which was partly covered by my master's thesis (especially his incompleteness theorems (Gödel 1931)) and again his permanent motivation led me to the field of proof theory. During the years this led to a very friendly contact for what I am deeply grateful. It therefore gives me pleasure that he became again the advisor of a thesis of mine. After all this time I think this is the right place to express my gratitude to him for everything I learned from him personally and scientifically. Thank you, Alex.

Another person I am truly indebted to is my wife, Katharina. Not only for that she was willing to marry me during the work for this thesis. Also her constant support in everyday life enabled me to actually accomplish the work on this thesis. I feel also very much obligated to thank her and am proud of her especially because of the throwbacks and misery that life held ready for us during the last year even so she allowed me to concentrate on my work — as far as it was possible at certain periods of time. Thank you, Kathi.

CHAPTER 1

Introduction

In proof theory mathematical proofs are represented as formal objects. The main subjects are proof analysis and proof transformation. In order to be able to analyze proofs often proof transformations have to be applied a priori or sometimes the analysis of concepts and methods involve the transformation of proofs. Therefore proof transformations play a central role in modern proof theory.

The first field of application of proof theory were consistency proofs, i.e. showing the consistency of certain theories based on a formal specification by means of calculi. Most of these consistency proofs included several proof transformation steps. In his famous paper Gentzen (1934) used a specific technique to get rid of an auxiliary rule, namely the cut rule, to prove the consistency of natural deduction. In particular he proved that natural deduction can be simulated by a sequent calculus with cut; the cut-elimination theorem then yields the consistency result. Nowadays, cut-elimination has become one of the most prominent forms of proof transformation within proof theory.

A cut can be thought of as a meta-level version of the reasoning principle modus ponens. Hence the effect of cut-elimination is the removal of all applications of intermediate statements (lemmas) within a proof resulting in a proof that is analytic in the sense, that all statements of the proof are subformulas of the resulting theorem.

Robinson's resolution principle (Robinson 1965) is a revolutionary mathematical tool of the last century. Not only because of its impact to mathematics, also because of its importance in computational aspects, this sound and complete method for deciding the validity of a theorem by refutation still plays a central role in various fields. Especially automated deduction and automated theorem proving are largely based on this method. On the one hand because of its effectiveness and on the other hand because of its algorithmical simplicity. A lot of refinements such as hyper-resolution and extensions, e.g. a rule based approach of equality integration called paramodulation (e.g. in Nieuwenhuis and Rubio 2001), have been invented and investigated despite portings to numerous different calculi and other formal systems.

Gentzen also founded in his *Hauptsatz* an algorithmical or computational approach. In fact, this constructive proof allows also for extraction of a rule based cut-elimination method. Others followed, e.g. the similar methods of Schütte (1960) and Tait (1968), but none of them with a substantially different approach. Until Baaz and Leitsch (2000) came up with a cut-elimination method based on resolution (CERES for short). The core of the method is the following: A proof structure containing cuts is analyzed and mapped to a clause term which evaluates to an unsatisfiable clause set of which a resolution refutation serves as a skeleton for an analytic variant of the proof.

Motivated by investigations of proof theorists like Girard (1987) who were applying cut-elimination to relevant mathematical proofs, but of course only on an abstract and theoretical level, it becomes clear that CERES is a promising algorithm to serve as a basis for an implementation of a cut-elimination system. The main advantages over other computational methods lie in the speed-up gained by CERES and the existing and very sophisticated automated theorem provers which can be used.

Note that cut-elimination is non-unique, i.e. there is no single cut-free proof which represents the analytic version of a proof with lemmas. The application of cut-elimination by computational methods on concrete proofs allows the creation of interesting new proofs of the same statement but using an entirely different argumentation. This introduces a mechanism of extracting new proof theoretical methods and concepts out of existing proofs and could be the beginning of an era of computational proof theory.

The exceptional role of equality in mathematics in general hence also its central importance for the formalization of mathematical proofs causes the desire to integrate this concept into CERES. Equational reasoning is already a key component in automated deduction and verification systems therefore also an extension of the implementation benefits from many mature concepts. The advantages of proof formalization using equality are eclectic. While drastically simplifying proofs, the improvement of understanding of formalized proofs is invaluable. Allowing a more natural mathematical way of formalizing and interpreting proofs provides essential support to human users working with such a system.

The structure of this thesis is as follows. After giving some basic notions and definitions we present the traditional cut-elimination methods of Gentzen and Schütte-Tait in chapter 3. The method of cut-elimination by resolution is described in detail in the subsequent chapter including comparisons to the traditional approaches. Chapter 5 is dedicated to the extension of CERES by equality and followed by a chapter introducing equational theories within the logical inferences of the calculi presented in the previous chapters. The implementation of `CERES` is discussed in chapter 7, and interesting experiments performed with the system are the subject of the last chapter.

CHAPTER 2

Preliminaries

In the following $x, y, z, x_0, y_0, z_0, \ldots$ denote bound individual variables whereas $u, v, w, u_0, v_0, w_0, \ldots$ denote free individual variables.

Definition 2.1 (term, semi-term). Terms and semi-terms are defined inductively in the following way:

1. Individual constants are (semi-)terms.
2. Free individual variables are terms. Free and bound individual variables are semi-terms.
3. If f is a function symbol of arity n and $t_1, \ldots, t_n$ are (semi-)terms then $f(t_1, \ldots, t_n)$ is a (semi-)term.

Thus semi-terms are terms with bound variables, an analogous property holds for formulas.

Definition 2.2 (formula, semi-formula). Formulas and semi-formulas are inductively defined as follows:

1. If P is an n-ary predicate symbol and $t_1, \ldots, t_n$ are (semi-)terms, then $P(t_1, \ldots, t_n)$ is a (semi-)formula. It is called an atomic formula or an atom.
2. If A and B are (semi-)formulas, then $\neg A$, $A \wedge B$, $A \vee B$ and $A \supset B$ are (semi-)formulas.
3. If A is a (semi-)formula not containing the bound variable x, then $(\forall x)A(u/x)$ and $(\exists x)A(u/x)$ are (semi-)formulas.

A (semi)-formula not containing any quantifiers (only constructed by 1 and 2) is called a propositional formula.

A^n denotes a formula sequence $A, \ldots, A$ of n times A and $A \equiv B$ expresses that a formula A is structurally equal to B.

Obviously semi-formulas are formulas with free occurrences of bound variables. Note that our definition of terms and formulas is due to Takeuti (1987).

We will later be in the need to determine precisely the position of a term or formula within another term or formula; the following definition helps in this matter.

Definition 2.3 (position). Let t be a (semi-)term. We define positions within t inductively as follows:

1. Let t be an individual constant or a variable then 0 (the empty position) is the position in t representing the entire term t, i.e. $t.0 = t$.
2. Let t be of the form $t = f(t_1, \dots, t_n)$ then again 0 is the position in t representing the entire term t, i.e. $t.0 = t$. Let further $\xi_i : (k_l, \dots, k_1)$ be a position in a t_i, for $1 \leq i \leq n$ with $t_i.\xi_i = s$, then we define the nested position ξ in t such that $t.\xi = s$ as $\xi : (i, k_l, \dots, k_1)$.

Note that the empty position 0 is just an abbreviation for ().

Let $t.\xi = s$ then $t[r]_\xi$ denotes the (semi-)term t after replacement of the (semi-)term s on position ξ by r, in particular $t[r]_\xi = r$. Moreover if Ξ is a set of positions in t then $t[r]_\Xi$ is defined by replacing all sub-(semi-)terms $t.\xi$, for $\xi \in \Xi$, in t by r.

Positions in (semi-)formulas are defined analogously (simply consider all formulas as terms).

Definition 2.4 (substitution). A substitution σ is defined as a function from the set of variables to the set of terms, formally $\sigma = \{ u_1 \mapsto t_1, \dots, u_n \mapsto t_n \}$. The identity substitution is denoted by id.

The application $t\sigma$ of a substitution σ to a term t is defined inductively in the following way.

1. Let t be an individual constant or a variable then simply apply the substitution by $t\sigma$.
2. Let t be a term of the form $f(t_1, \dots, t_n)$ then $t\sigma = f(t_1\sigma, \dots, t_n\sigma)$.

The composition of two substitutions σ and θ, denoted by $\sigma\theta$, is defined by $t(\sigma\theta) = (t\sigma)\theta$.

We write $A(u)$ to indicate (potential) free occurrences of the variable u in A. Let t be an arbitrary term, then $A(u/t)$ stands for the replacement of all free occurrences of u in A by t, i.e. $A[t]_\Xi$ where $\Xi = \{ \xi \mid A.\xi = u \}$.

The logical complexity of a formula expresses the number of logical symbols of which the formula consists.

Definition 2.5 (logical complexity). The logical complexity of a formula A is defined inductively as follows:

1. $\text{comp}(A) = 0$ for A being an atom,
2. $\text{comp}(A) = \text{comp}(B) + 1$ for $A \equiv \neg B$,
3. $\text{comp}(A) = comp(B) + \text{comp}(C) + 1$, for $A \equiv B \wedge C$, $A \equiv B \vee C$ or $A \equiv B \supset C$,
4. $\text{comp}(A) = \text{comp}(B) + 1$ for $A \equiv (\forall x)B$ or $A \equiv (\exists x)B$.

In the following $\Gamma, \Delta, \Pi, \Lambda, \Gamma_0, \Delta_0, \Pi_0, \Lambda_0, \dots$ denote finite (possibly empty) sequences of formulas. The empty sequence of formulas is denoted by ϵ.

Definition 2.6 (sequent). A finite sequence of formulas, separated by the auxiliary syntactic symbol $\vdash$, is called a sequent, symbolically $S : \Gamma \vdash \Delta$, where Γ is called the antecedents respectively Δ the consequent of S. The empty sequent is denoted by $\vdash$. A sequent S is called atomic if Γ and Δ are sequences of atomic formulas.

For two sequents $S_1 : \Gamma \vdash \Delta$ and $S_2 : \Pi \vdash \Lambda$ the expression $S_1 \circ S_2$ denotes the composition of the sequents, i.e. $\Gamma, \Pi \vdash \Delta, \Lambda$. Because of the sequent representation as sequences of formulas, note that the composition operator is not commutative, symbolically $S_1 \circ S_2 \neq S_2 \circ S_1$.

Definition 2.7 (axiom set). A (possibly infinite) set $\mathcal{A}$ of atomic sequents is called an axiom set if it is closed under substitution, i.e. for all $S \in \mathcal{A}$ and for all substitutions σ we have $S\sigma \in \mathcal{A}$. S is called an axiom.

Definition 2.8 (LK). Let $\mathcal{A}$ be an axiom set. An axiom rule is of the form

$$\overline{S}$$

where the sequent S is contained in $\mathcal{A}$, i.e. $S \in \mathcal{A}$; S is called an axiom.

An inference rule is of the form

$$\frac{S'}{S}\ \rho_u \qquad \text{or} \qquad \frac{S_1 \quad S_2}{S}\ \rho_b$$

where ρ_u denotes a unary **LK** inference rule and ρ_b denotes a binary **LK** inference rule. The sequent S' is called premise (of ρ_u) respectively the sequents S_1 and S_2 are called left and right premise (of ρ_b). The sequent S is called the conclusion of the particular inference.

Unlike Gentzen's version of **LK** (see Gentzen 1934) we use the multiplicative version of **LK** similar to Girard (see Girard 1987). In the following definition the auxiliary formulas are put in bold face and the principal formulas are underlined, but usually these markings are avoided because the auxiliary and principal formulas are mostly uniquely identifiable by their outermost positioning (respectively the permutations are given explicitly where needed).

1. The structural rules of

   a) Weakening:

   $$\frac{\Gamma \vdash \Delta}{\underline{A_1}, \ldots, \underline{A_n}, \Gamma \vdash \Delta}\ \mathrm{w:l} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, \underline{A_1}, \ldots, \underline{A_n}}\ \mathrm{w:r}$$

   where $n > 0$.

   b) Contraction:

   $$\frac{\boldsymbol{A_1^{m_1}}, \ldots, \boldsymbol{A_n^{m_n}} \vdash \Delta}{\underline{A_1}, \ldots, \underline{A_n} \vdash \Delta}\ \mathrm{c}(m_1, \ldots, m_n):\mathrm{l} \qquad \frac{\Gamma \vdash \boldsymbol{A_1^{m_1}}, \ldots, \boldsymbol{A_n^{m_n}}}{\Gamma \vdash \underline{A_1}, \ldots, \underline{A_n}}\ \mathrm{c}(m_1, \ldots, m_n):\mathrm{r}$$

   where the number of occurrences of the i-th formula $m_i > 0$, $i \in \{1, \ldots, n\}$. The auxiliary formulas respectively the principal formulas are those A_i of the premises respectively those A_i of the conclusion where $m_i > 1$.

   c) Permutation:

   $$\frac{\boldsymbol{A_1}, \ldots, \boldsymbol{A_n} \vdash \Delta}{\underline{A_{\tau(1)}}, \ldots, \underline{A_{\tau(n)}} \vdash \Delta}\ \pi(\tau):\mathrm{l} \qquad \frac{\Gamma \vdash \boldsymbol{A_1}, \ldots, \boldsymbol{A_n}}{\Gamma \vdash \underline{A_{\tau(1)}}, \ldots, \underline{A_{\tau(n)}}}\ \pi(\tau):\mathrm{r}$$

   where τ is a permutation of $\{1, \ldots, n\}$. The auxiliary formulas respectively the principal formulas are those A_i of the premises respectively those $A_{\tau(i)}$ of the conclusions where $i \neq \tau(i)$, $i \in \{1, \ldots, n\}$, holds. The permutation τ is specified as a list of cycles. $-\tau$ denotes the inverse permutation of τ.

   d) Cut:

   $$\frac{\Gamma \vdash \Delta, \boldsymbol{A} \quad \boldsymbol{A}, \Pi \vdash \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda}\ \mathrm{cut}$$

   where A is called the cut-formula.

2. The logical rules for

   a) ¬-introduction:

$$\frac{\Gamma \vdash \Delta, \boldsymbol{A}}{\underline{\neg A}, \Gamma \vdash \Delta}\ \neg : \mathrm{l} \qquad \frac{\boldsymbol{A}, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \underline{\neg A}}\ \neg : \mathrm{r}$$

   b) ∧-introduction:

$$\frac{\boldsymbol{A}, \boldsymbol{B}, \Gamma \vdash \Delta}{\underline{A \wedge B}, \Gamma \vdash \Delta}\ \wedge : \mathrm{l} \qquad \frac{\Gamma \vdash \Delta, \boldsymbol{A} \quad \Pi \vdash \Lambda, \boldsymbol{B}}{\Gamma, \Pi \vdash \Delta, \Lambda, \underline{A \wedge B}}\ \wedge : \mathrm{r}$$

   c) ∨-introduction:

$$\frac{\boldsymbol{A}, \Gamma \vdash \Delta \quad \boldsymbol{B}, \Pi \vdash \Lambda}{\underline{A \vee B}, \Gamma, \Pi \vdash \Delta, \Lambda}\ \vee : \mathrm{l} \qquad \frac{\Gamma \vdash \Delta, \boldsymbol{A}, \boldsymbol{B}}{\Gamma \vdash \Delta, \underline{A \vee B}}\ \vee : \mathrm{r}$$

   d) ⊃-introduction:

$$\frac{\Gamma \vdash \Delta, \boldsymbol{A} \quad \boldsymbol{B}, \Pi \vdash \Lambda}{\underline{A \supset B}, \Gamma, \Pi \vdash \Delta, \Lambda}\ \supset : \mathrm{l} \qquad \frac{\boldsymbol{A}, \Gamma \vdash \Delta, \boldsymbol{B}}{\Gamma \vdash \Delta, \underline{A \supset B}}\ \supset : \mathrm{r}$$

   e) ∀-introduction:

$$\frac{\boldsymbol{A(x/t)}, \Gamma \vdash \Delta}{\underline{(\forall x)A(x)}, \Gamma \vdash \Delta}\ \forall : \mathrm{l} \qquad \frac{\Gamma \vdash \Delta, \boldsymbol{A(x/u)}}{\Gamma \vdash \Delta, \underline{(\forall x)A(x)}}\ \forall : \mathrm{r}$$

   where t is an arbitrary term and u does not occur in the conclusion. u is called an eigenvariable.

   f) ∃-introduction:

$$\frac{\boldsymbol{A(x/u)}, \Gamma \vdash \Delta}{\underline{(\exists x)A(x)}, \Gamma \vdash \Delta}\ \exists : \mathrm{l} \qquad \frac{\Gamma \vdash \Delta, \boldsymbol{A(x/t)}}{\Gamma \vdash \Delta, \underline{(\exists x)A(x)}}\ \exists : \mathrm{r}$$

   where x does not occur in the conclusion and t is an arbitrary term. u is called an Eigenvariable.

Definition 2.9 (standard axiom set). The standard axiom set $\mathcal{A}_{\mathbf{LK}}$ of **LK** is an axiom set defined by

$$\mathcal{A}_{\mathbf{LK}} = \{\, A \vdash A \mid A \text{ an atom} \,\}.$$

Definition 2.10 (proof, LK-proof). A proof φ of a sequent S from an axiom set $\mathcal{A}$ is a directed labelled tree, where the nodes represent occurrences of sequents and the edges are labelled according to the inference rule applications in the calculus **K**. The root is labelled by the occurrence of the end-sequent S and the leaves are labelled by occurrences of axioms, i.e. elements of $\mathcal{A}$.

Let ν be a proof node in φ, then $\varphi.\nu$ denotes the subproof of φ beginning at ν (with root ν). A proof node μ is called immediate successor of ν if μ is the occurrence of a premise of a rule ρ in φ with conclusion ν.

An **LK**-proof is a proof where $\mathcal{A}$ consists of atomic sequents including the standard axioms $\mathcal{A}_{\mathbf{LK}}$ of **LK** and the inference rules applied are those of **LK**.

In the following we assume that if the concrete axiom set $\mathcal{A}$ on which a specific **LK**-proof is based on is not specified explicitly then $\mathcal{A}$ is the standard axiom set of **LK**.

Definition 2.11 (ancestor). Let ρ, which is either of the form

$$\frac{S_1 : \Pi_1, \Gamma_1 \vdash \Delta_1, \Lambda_1}{S : \Pi, \Gamma_1 \vdash \Delta_1, \Lambda}\ \rho_u \qquad \text{or} \qquad \frac{S_1 : \Pi_1, \Gamma_1 \vdash \Delta_1, \Lambda_1 \quad S_2 : \Pi_2, \Gamma_2 \vdash \Delta_2, \Lambda_2}{S : \Pi, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, \Lambda}\ \rho_b$$

be an inference rule in an **LK**-proof φ, where Π_i and Λ_i respectively Π and Λ denote the (possibly empty) sequences of auxiliary formulas of the (one or two) premises respectively principal formulas of the conclusion; let further $\mathfrak{o}_k$ be the occurrence of the k-th principal formula in S and $\mathfrak{o}_{ij}$ be the occurrence of the j-th auxiliary formula in S_i, $i \in 1,2$ and $j,k \in \mathbb{N}$. Then all $\mathfrak{o}_{ij}$ are ancestors of all $\mathfrak{o}_k$.
Let $\mathfrak{o}_{ij}$ be the occurrence of the j-th non-principal formula in Γ_i respectively Δ_i of S, then $\mathfrak{o}_j$ being the occurrence of the j-th non-auxiliary formula in Γ_i respectively Δ_i of S_i is defined as the ancestor of $\mathfrak{o}_{ij}$.

The ancestor relation in φ is defined as the reflexive and transitive closure of the above relation.

If Ω is a set of formula occurrences in φ and let S be the sequent at the node ν of the **LK**-proof φ then by $\nu(\Omega)$ respectively $\overline{\nu}(\Omega)$ we denote the subsequent of S consisting of all formulas which are respectively are not ancestors of a formula occurrence in Ω.

Definition 2.12 (LKp). LKp is the calculus obtained from **LK** by adding the semantic cut rule (p-cut), also called pseudo-cut, to the existing rules of **LK**.

$$\frac{\Gamma \vdash \Delta, \boldsymbol{A} \quad \boldsymbol{B}, \Pi \vdash \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda}\ \text{p-cut}$$

if $A \supset B$ is valid.

By comparison of the semantic cut rule with an implication introduction rule

$$\frac{\Gamma \vdash \Delta, \boldsymbol{A} \quad \boldsymbol{B}, \Pi \vdash \Lambda}{\underline{A \supset B}, \Gamma, \Pi \vdash \Delta, \Lambda}\ \supset: \mathrm{l}$$

it is easy to see that $A \supset B$ can only be cut out if it is valid or in other words, that $\Gamma, \Pi \vdash \Delta, \Lambda$ is equivalent to $A \supset B, \Gamma, \Pi \vdash \Delta, \Lambda$ if $A \supset B$ is valid.

The introduction of the semantic cut rule is more ore less a replacement of the existing cut rule, which is just the specific case of the semantic cut rule where A is syntactically equivalent to B, i.e. $A \equiv B$.

Definition 2.13 (LKp-proof). An **LKp**-proof is a proof where $\mathcal{A}$ consists of atomic sequents including the standard axioms $\mathcal{A}_{\mathbf{LK}}$ of **LK** and the inference rules applied are those of **LKp**.

Definition 2.14 (grade, rank). Let φ be an **LK**-proof ending with a cut rule inference, i.e. φ is of the form

$$\frac{\begin{array}{c}(\varphi_l)\\ \Gamma \vdash \Delta, \boldsymbol{A}\end{array} \quad \begin{array}{c}(\varphi_r)\\ \boldsymbol{A}, \Pi \vdash \Lambda\end{array}}{\Gamma, \Pi \vdash \Delta, \Lambda}\ \text{cut}$$

Then the grade of φ is defined as the logical complexity of the cut-formula A, i.e.

$$\text{grade}(\varphi) = \text{comp}(A).$$

The rank of φ is defined as the sum of the ranks of φ_l (the left-rank) and φ_r (the right-rank), which are defined as the maximum ancestor-paths of the cut-formula A

$$\text{rank}_\text{l}(\varphi) = \max\{\, \text{length}(p) \mid p \in \text{ancpath}(A, \varphi_l) \,\}$$
$$\text{rank}_\text{l}(\varphi) = \max\{\, \text{length}(p) \mid p \in \text{ancpath}(A, \varphi_r) \,\}$$

hence

$$\text{rank}(\varphi) = \text{rank}_\text{l}(\varphi) + \text{rank}_\text{l}(\varphi).$$

Definition 2.15 (clause). A clause is an atomic sequent, i.e. a sequent of the form $\Gamma \vdash \Delta$, where Γ and Δ are sequences of atomic formulas. The empty clause is denoted by ϵ.

The definition of the resolution calculus **R** that we are giving below is a very common way of defining such a calculus nevertheless it is based on Robinson's resolution method (see Robinson 1965) and influenced by the general resolution concept in Leitsch (1997). Furthermore **R** has an explicit factoring rule similar to Loveland (1978).

Definition 2.16 (R). Let $\mathcal{C}$ be a clause set. An initial rule is of the form

$$\overline{C}$$

where the clause C is contained in $\mathcal{C}$, i.e. $C \in \mathcal{C}$; C is called an initial clause.

An **R** inference rule is of the form

$$\frac{C'}{C}\ \rho_u \qquad \text{or} \qquad \frac{C_l \quad C_r}{C}\ \rho_b$$

where ρ_u denotes a unary **R** inference rule and ρ_b denotes a binary **R** inference rule. The clause C' is called premise (of ρ_u) respectively the clauses C_l and C_r are called left and right premise (of ρ_b). The clause C is called the conclusion of the particular inference.

In the following definition the auxiliary formulas are put in bold face and the principal formulas are underlined, but usually these markings are avoided because the auxiliary and principal formulas are mostly uniquely identifiable by their outermost positioning (respectively the permutations are given explicitly where needed).

The rules of

1. Permutation:

$$\frac{A_1, \ldots, A_n \vdash \Delta}{A_{\tau(1)}, \ldots, A_{\tau(n)} \vdash \Delta}\ \pi(\tau) : \text{l} \qquad \frac{\Gamma \vdash A_1, \ldots, A_n}{\Gamma \vdash A_{\tau(1)}, \ldots, A_{\tau(n)}}\ \pi(\tau) : \text{r}$$

   where τ is a permutation of $\{\, 1, \ldots, n \,\}$. The auxiliary formulas respectively the principal formulas are those A_i of the premises respectively those $A_{\tau(i)}$ of the conclusions where $i \neq \tau(i)$, $i \in \{\, 1, \ldots, n \,\}$, holds.

2. Factoring:
$$\frac{\boldsymbol{A_1,\ldots,A_n},\Gamma\vdash\Delta}{(\underline{A_1},\Gamma\vdash\Delta)\sigma}\ \mathrm{f}(\sigma):\mathrm{l}\qquad\frac{\Gamma\vdash\Delta,\boldsymbol{A_1,\ldots,A_n}}{(\Gamma\vdash\Delta,\underline{A_1})\sigma}\ \mathrm{f}(\sigma):\mathrm{r}$$
where σ is an m.g.u. of the set $\{ A_i \}_{1\leq i\leq n}$. The conclusion of a factoring rule is called factor and additionally nontrivial in case $1 < n$.

3. Binary resolution:
$$\frac{\Gamma\vdash\Delta,\boldsymbol{A}\quad \boldsymbol{A'},\Pi\vdash\Lambda}{(\Gamma,\Pi\vdash\Delta,\Lambda)\sigma}\ \mathrm{r}(\sigma)$$
where σ is an m.g.u. of $\{ A, A' \}$ and the premises are variable disjoint clauses. The conclusion of a binary resolution inference is also called resolvent (a terminus going back to David Hilbert by the way) and the two formulas A and A' that are resolved upon are called (positive resp. negative) resolution-formulas.

The paramodulation rule for the resolution calculus, which we will be using, is defined as follows.

Definition 2.17 (paramodulation). Let the premises be variable disjoint clauses and σ be a most general unifier of $\{ s, s' \}$ then the paramodulation rules for some position set Ξ for the resolution calculus is defined as follows:
$$\frac{\Gamma\vdash\Delta,\boldsymbol{s=t}\quad \boldsymbol{A[s']_\Xi},\Pi\vdash\Lambda}{(\underline{A[t]_\Xi},\Gamma,\Pi\vdash\Delta,\Lambda)\sigma}\ \mathrm{p}(\sigma,\Xi):\mathrm{l}\qquad\frac{\Gamma\vdash\Delta,\boldsymbol{s=t}\quad \Pi\vdash\Lambda,\boldsymbol{A[s']_\Xi}}{(\Gamma,\Pi\vdash\Delta,\Lambda,\underline{A[t]_\Xi})\sigma}\ \mathrm{p}(\sigma,\Xi):\mathrm{r}$$
where s and t are arbitrary terms.

We also define versions of this paramodulation rules where the sides of the equation $s = t$ get flipped within the left premise.
$$\frac{\Gamma\vdash\Delta,\boldsymbol{t=s}\quad \boldsymbol{A[s']_\Xi},\Pi\vdash\Lambda}{(\underline{A[t]_\Xi},\Gamma,\Pi\vdash\Delta,\Lambda)\sigma}\ \mathrm{p}'(\sigma,\Xi):\mathrm{l}\qquad\frac{\Gamma\vdash\Delta,\boldsymbol{t=s}\quad \Pi\vdash\Lambda,\boldsymbol{A[s']_\Xi}}{(\Gamma,\Pi\vdash\Delta,\Lambda,\underline{A[t]_\Xi})\sigma}\ \mathrm{p}'(\sigma,\Xi):\mathrm{r}$$
where again s and t are arbitrary terms.

This just expresses an implicit application of the symmetry of equality. Therefore this rules are just shortcuts for
$$\frac{\dfrac{\Gamma\vdash\Delta,t=s\quad \overline{\vdash t=t}}{\Gamma\vdash\Delta,s=t}\ \mathrm{p}(\mathsf{id},(1)):\mathrm{r}\qquad A[s']_\Xi,\Pi\vdash\Lambda}{(A[t]_\Xi,\Gamma,\Pi\vdash\Delta,\Lambda)\sigma}\ \mathrm{p}(\sigma,\Xi):\mathrm{l}$$
respectively
$$\frac{\dfrac{\Gamma\vdash\Delta,t=s\quad \overline{\vdash t=t}}{\Gamma\vdash\Delta,s=t}\ \mathrm{p}(\mathsf{id},(1)):\mathrm{r}\qquad \Pi\vdash\Lambda,A[s']_\Xi}{(\Gamma,\Pi\vdash\Delta,\Lambda,A[t]_\Xi)\sigma}\ \mathrm{p}(\sigma,\Xi):\mathrm{r}$$

Note that the particular definition of the paramodulation rule does not matter, again all refinements of paramodulation such as ordered paramodulation or superposition might be used (see Chang and Lee (1973) or Degtyarev and Voronkov (2001) for more details on paramodulation and its refinements).

Definition 2.18 (R-deduction, PR-deduction). A deduction of a set of clauses using the rules of the resolution calculus **R** is called an **R**-deduction and **PR**-deduction if the rules of **R** are extended by paramodulation (the paramodulation rules specified in definition 2.17).

Definition 2.19 (R-proof, PR-proof). An **R**-deduction (**PR**-deduction) of a set of clauses $\mathcal{C}$ is called an **R**-proof (**PR**-proof) if it is a deduction of the empty clause. An **R**-proof (**PR**-proof) is also called an **R**-refutation (**PR**-refutation).

Definition 2.20 (regularity). Let φ be an **LK**-proof and ψ and ψ' are different subproofs of φ, i.e. the sets of proof nodes of ψ and ψ' are distinct. Then φ is called regular if every eigenvariable u that occurs in ψ does not occur in ψ'. φ is called strongly regular if every eigenvariable u that occurs in ψ does not occur in any proof node of φ that is not a proof node of ψ.

Let γ be an **R**-proof. Then γ is called regular if all initial sequents have distinct variables.

From now on, if we speak of regularity of an **LK**-proof we always mean strong regularity.

CHAPTER 3

Standard Cut-Elimination Methods

As far as standard or classical cut-elimination methods are concerned what definitely comes to ones mind is the first cut-elimination method ever (see Gentzen 1934). In this famous article Gerhard Gentzen's aim was to give a consistency proof of natural deduction and thus of intuitionistic logic. As tools he invented the sequent calculi **LK** and **LJ** (the intuitionistic counterpart of **LK**). To map natural deductions to sequent calculus derivations Gentzen had to add a syntactic rule - the cut rule. But to prove the consistency he had to show that all applications of this rule can be eliminated, since the empty sequent can only be the conclusion of a cut rule. He proved this conjecture with the *Hauptsatz* (or *cut-elimination theorem*), stating that every proof in **LK** can be transformed into a (not necessarily unique) normal form by means of cut-elimination.

A corollary from the Hauptsatz for example is the subformula property, that under the absence of cut rule inferences all formulas occurring in the proof are instances of subformulas of the end-sequent. This opens up the perspective of the so called *midsequent* theorem, allowing a partition of an **LK**-proof such that all the quantifier inferences appear below all the propositional inference rule applications. Which gives a constructive method for obtaining the *Herbrand disjunction* in the sense of *Herbrand's theorem* (Herbrand 1930).

Most importantly, Gentzen's proof of the Hauptsatz is a constructive proof which allows the extraction of a method to actually *perform* cut-elimination on proofs in **LK**. Also there has never been the intention by Gentzen to actually perform cut-elimination with this algorithm, which probably explains its average performance inefficiencies. Another method for cut-elimination is the Schütte-Tait method (for details see Schütte 1960, Tait 1968).

The cut-elimination method of Gentzen eliminates an uppermost cut by a double induction on the complexity of the cut-formula and on its rank, where the rank of a formula is the sum of the maximal length of ancestor paths of the proofs of the premises. Whereas the Schütte-Tait procedure always eliminates a maximal cut with respect to the complexity of the cut-formula. Obviously these methods are similar, since they only differ in the selection strategy, which cut to be eliminated next. But what they have in common is the rule base they are working with.

3.1 Cut-Reduction

The basic idea behind the cut reduction rules is the following: move the cut rule upwards within the **LK**-proof φ (in direction to the axioms) using rank reduction until either the cut-formula is introduced in both premises immediately above the cut (i.e. $\text{rank}(\varphi) = 2$) and grade reduction has to be carried out or the cut can be dissolved against a standard axiom or a weakening introducing the cut-formula in one premise.

Without loss of generality we assume φ to be regular, since some of the reduction transformations contain shiftings of rules respectively duplications of proof parts which could violate eigenvariable conditions otherwise.

Remark 3.1. Our aim is not to eliminate all cuts. Merely we are interested in reducing φ to an **LK**-proof φ' which contains at most atomic cuts.

Definition 3.1 (cut-reduction). $\mathcal{R} = \langle R, \Rightarrow \rangle$ is a cut-reduction system such that R is the set of all **LK**-proofs and $\Rightarrow \subseteq R \times R$ is a binary relation. Let φ and ψ be **LK**-proofs then $\varphi \Rightarrow \psi$ iff φ transforms to ψ according to a cut-reduction rule specified in section 3.2 or 3.3.

Building up on this general definition of a cut-reduction system we formulate the specialized reductions of Gentzen and Schütte-Tait.

Definition 3.2 (Gentzen reduction). Let φ and ψ be **LK**-proofs. The Gentzen reduction relation $\Rightarrow_G$ is then defined $\varphi \Rightarrow_G \psi$ if $\varphi \Rightarrow \psi$ and the only cut rule occurring in φ is the last inference rule.

Definition 3.3 (Schütte-Tait reduction). Let φ and ψ be **LK**-proofs and let Ω be the set of all occurrences of cut formulas in φ. The Schütte-Tait reduction relation $\Rightarrow_{ST}$ is then defined $\varphi.\nu \Rightarrow_{ST} \psi$ if $\varphi.\nu \Rightarrow \psi$ and the last inference rule of $\varphi.\nu$ is the only maximal cut rule inference with cut-formula A, i.e. the logical complexity of A is greater or equal than the logical complexity of any formula of Ω, formally

$$\max\{\, \mathrm{comp}(B) \mid B \text{ a cut-formula of } \Omega \,\} \leq \mathrm{comp}(A),$$

and greater than the logical complexity of any other cut-formula occurrence in $\varphi.\nu$.

Now let $\varphi.\nu$ be of the form

$$\dfrac{\begin{matrix}(\varphi_l) \\ \Gamma \vdash \Delta, \boldsymbol{A}\end{matrix} \quad \begin{matrix}(\varphi_r) \\ \boldsymbol{A}, \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi \vdash \Delta, \Lambda}\ \mathrm{cut}$$

The cut reduction rules divide into two partitions. Into rules reducing the logical complexity of the cut-formula - called *grade reduction rules* - and into rules reducing the maximum rank of the cut rule within the proof - called *rank reduction rules.*

3.2 Grade Reduction

If the cut-formula A is immediately introduced (on both sides) above the cut then we distinguish the following cases depending on the form of the cut-formula A:

1. $A \equiv \neg A$ then let $\varphi.\nu =$

$$\dfrac{\dfrac{\overset{(\varphi_{l'})}{\boldsymbol{A}, \Gamma \vdash \Delta}}{\Gamma \vdash \Delta, \underline{\boldsymbol{\neg A}}}\;\neg:\mathrm{r} \qquad \dfrac{\overset{(\varphi_{r'})}{\Pi \vdash \Lambda, \boldsymbol{A}}}{\underline{\boldsymbol{\neg A}}, \Pi \vdash \Lambda}\;\neg:\mathrm{l}}{\Gamma, \Pi \vdash \Delta, \Lambda}\;\mathrm{cut}$$

which transforms to

$$\dfrac{\dfrac{\dfrac{\overset{(\varphi_{r'})}{\Pi \vdash \Lambda, \boldsymbol{A}} \quad \overset{(\varphi_{l'})}{\boldsymbol{A}, \Gamma \vdash \Delta}}{\boldsymbol{\Pi}, \boldsymbol{\Gamma} \vdash \Lambda, \Delta}\;\mathrm{cut}}{\underline{\Gamma}, \underline{\Pi} \vdash \boldsymbol{\Lambda}, \boldsymbol{\Delta}}\;\pi(\tau_l):\mathrm{l}}{\Gamma, \Pi \vdash \underline{\Delta}, \underline{\Lambda}}\;\pi(\tau_r):\mathrm{r}$$

Either of the permutation rule inferences may be omitted if they have no effect. Which is for the permutation on the left side the case if $\Gamma = \epsilon$, $\Pi = \epsilon$ or $\Gamma = \Pi$ and for the permutation on the right side if $\Delta = \epsilon$, $\Lambda = \epsilon$ or $\Delta = \Lambda$.

2. $A \equiv A \wedge B$ then let $\varphi.\nu =$

$$\dfrac{\dfrac{\overset{(\varphi_{l_l})}{\Gamma_l \vdash \Delta_l, \boldsymbol{A}} \quad \overset{(\varphi_{l_r})}{\Gamma_r \vdash \Delta_r, \boldsymbol{B}}}{\Gamma_l, \Gamma_r \vdash \Delta_l, \Delta_r, \underline{\boldsymbol{A} \wedge \boldsymbol{B}}}\;\wedge:\mathrm{r} \qquad \dfrac{\overset{(\varphi_{r'})}{\boldsymbol{A}, \boldsymbol{B}, \Pi \vdash \Lambda}}{\underline{\boldsymbol{A} \wedge \boldsymbol{B}}, \Pi \vdash \Lambda}\;\wedge:\mathrm{l}}{\Gamma_l, \Gamma_r, \Pi \vdash \Delta_l, \Delta_r, \Lambda}\;\mathrm{cut}$$

which transforms to

$$\dfrac{\overset{(\varphi_{l_l})}{\Gamma_l \vdash \Delta_l, \boldsymbol{A}} \qquad \dfrac{\dfrac{\overset{(\varphi_{l_r})}{\Gamma_r \vdash \Delta_r, \boldsymbol{B}} \quad \dfrac{\overset{(\varphi_{r'})}{\boldsymbol{A}, \boldsymbol{B}, \Pi \vdash \Lambda}}{\underline{\boldsymbol{B}}, \underline{A}, \Pi \vdash \Lambda}\;\pi((1\,2)):\mathrm{l}}{\boldsymbol{\Gamma_r}, \boldsymbol{A}, \Pi \vdash \Delta_r, \Lambda}\;\mathrm{cut}}{\underline{\boldsymbol{A}}, \underline{\Gamma_r}, \Pi \vdash \Delta_r, \Lambda}\;\pi(\tau):\mathrm{l}}{\Gamma_l, \Gamma_r, \Pi \vdash \Delta_l, \Delta_r, \Lambda}\;\mathrm{cut}$$

where the permutation rule applying τ can be omitted in case $\Gamma_r = \epsilon$ or $\{\, \Gamma_r \,\} = \{\, A \,\}$.

3. $A \equiv A \vee B$ then let $\varphi.\nu =$

$$\dfrac{\dfrac{\overset{(\varphi_{l'})}{\Gamma \vdash \Delta, \boldsymbol{A}, \boldsymbol{B}}}{\Gamma \vdash \Delta, \underline{\boldsymbol{A} \vee \boldsymbol{B}}}\;\vee:\mathrm{r} \qquad \dfrac{\overset{(\varphi_{r_l})}{\boldsymbol{A}, \Pi_l \vdash \Lambda_l} \quad \overset{(\varphi_{r_r})}{\boldsymbol{B}, \Pi_r \vdash \Lambda_r}}{\underline{\boldsymbol{A} \vee \boldsymbol{B}}, \Pi_l, \Pi_r \vdash \Lambda_l, \Lambda_r}\;\vee:\mathrm{l}}{\Gamma, \Pi_l, \Pi_r \vdash \Delta, \Lambda_l, \Lambda_r}\;\mathrm{cut}$$

which transforms to

$$\dfrac{\dfrac{\dfrac{\dfrac{\overset{(\varphi_{l'})}{\Gamma \vdash \Delta, \boldsymbol{A}, \boldsymbol{B}}}{\Gamma \vdash \Delta, \underline{B}, \underline{\boldsymbol{A}}}\;\pi(\tau_1):\mathrm{r} \quad \overset{(\varphi_{r_l})}{\boldsymbol{A}, \Pi_l \vdash \Lambda_l}}{\Gamma, \Pi_l \vdash \Delta, \boldsymbol{B}, \boldsymbol{\Lambda_l}}\;\mathrm{cut}}{\Gamma, \Pi_l \vdash \Delta, \underline{\Lambda_l}, \underline{\boldsymbol{B}}}\;\pi(\tau_2):\mathrm{r} \qquad \overset{(\varphi_{r_r})}{\boldsymbol{B}, \Pi_r \vdash \Lambda_r}}{\Gamma, \Pi_l, \Pi_r \vdash \Delta, \Lambda_l, \Lambda_r}\;\mathrm{cut}$$

where the permutation rule applying τ_2 can be omitted in case $\Lambda_l = \epsilon$ or $\{\, \Lambda_l \,\} = \{\, B \,\}$.

4. $A \equiv A \supset B$ then let $\varphi.\nu =$

$$\dfrac{\dfrac{\overset{(\varphi_{l'})}{\boldsymbol{A}, \Gamma \vdash \Delta, \boldsymbol{B}}}{\Gamma \vdash \Delta, \underline{\boldsymbol{A} \supset \boldsymbol{B}}}\ \vee:\mathrm{r} \qquad \dfrac{\overset{(\varphi_{r_l})}{\Pi_l \vdash \Lambda_l, \boldsymbol{A}} \quad \overset{(\varphi_{r_r})}{\boldsymbol{B}, \Pi_r \vdash \Lambda_r}}{\underline{\boldsymbol{A} \supset \boldsymbol{B}}, \Pi_l, \Pi_r \vdash \Lambda_l, \Lambda_r}\ \vee:\mathrm{l}}{\Gamma, \Pi_l, \Pi_r \vdash \Delta, \Lambda_l, \Lambda_r}\ \mathrm{cut}$$

which transforms to

$$\dfrac{\dfrac{\dfrac{\dfrac{\overset{(\varphi_{r_l})}{\Pi_l \vdash \Lambda_l, \boldsymbol{A}} \quad \overset{(\varphi_{l'})}{\boldsymbol{A}, \Gamma \vdash \Delta, B}}{\boldsymbol{\Pi_l}, \boldsymbol{\Gamma} \vdash \Lambda_l, \Delta, B}\ \mathrm{cut}}{\underline{\Gamma}, \underline{\Pi_l} \vdash \boldsymbol{\Lambda_l}, \boldsymbol{\Delta}, B}\ \pi(\tau_1):\mathrm{l}}{\Gamma, \Pi_l \vdash \underline{\Delta}, \underline{\Lambda_l}, \boldsymbol{B}}\ \pi(\tau_2):\mathrm{r} \qquad \overset{(\varphi_{r_r})}{\boldsymbol{B}, \Pi_r \vdash \Lambda_r}}{\Gamma, \Pi_l, \Pi_r \vdash \Delta, \Lambda_l, \Lambda_r}\ \mathrm{cut}$$

Either of the permutation rule inferences may be omitted if they have no effect. Which is for the permutation on the left side the case if $\Gamma = \epsilon$, $\Pi_l = \epsilon$ or $\Gamma = \Pi_l$ and for the permutation on the right side if $\Delta = \epsilon$, $\Lambda_l = \epsilon$ or $\Delta = \Lambda_l$.

5. $A \equiv (\forall x)A(x)$ then let $\varphi.\nu =$

$$\dfrac{\dfrac{\overset{(\varphi_{l'}(x/u))}{\Gamma \vdash \Delta, \boldsymbol{A(x/u)}}}{\Gamma \vdash \Delta, \underline{\boldsymbol{(\forall x)A(x)}}}\ \forall:\mathrm{r} \qquad \dfrac{\overset{(\varphi_{r'})}{\boldsymbol{A(x/t)}, \Pi \vdash \Lambda}}{\underline{\boldsymbol{(\forall x)A(x)}}, \Pi \vdash \Lambda}\ \forall:\mathrm{l}}{\Gamma, \Pi \vdash \Delta, \Lambda}\ \mathrm{cut}$$

which transforms to

$$\dfrac{\overset{(\varphi_{l'}(x/t))}{\Gamma \vdash \Delta, \boldsymbol{A(x/t)}} \quad \overset{(\varphi_{r'})}{\boldsymbol{A(x/t)}, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\ \mathrm{cut}$$

Remark: We assume regularity, hence u only occurs in $\varphi_{l'}$.

6. $A \equiv (\exists x)A(x)$ then let $\varphi.\nu =$

$$\dfrac{\dfrac{\overset{(\varphi_{l'})}{\Gamma \vdash \Delta, \boldsymbol{A(x/t)}}}{\Gamma \vdash \Delta, \underline{\boldsymbol{(\exists x)A(x)}}}\ \exists:\mathrm{r} \qquad \dfrac{\overset{(\varphi_{r'}(x/u))}{\boldsymbol{A(x/u)}, \Pi \vdash \Lambda}}{\underline{\boldsymbol{(\exists x)A(x)}}, \Pi \vdash \Lambda}\ \exists:\mathrm{l}}{\Gamma, \Pi \vdash \Delta, \Lambda}\ \mathrm{cut}$$

which transforms to

$$\dfrac{\overset{(\varphi_{l'})}{\Gamma \vdash \Delta, \boldsymbol{A(x/t)}} \quad \overset{(\varphi_{r'}(x/t))}{\boldsymbol{A(x/t)}, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\ \mathrm{cut}$$

Remark: We assume regularity, hence u only occurs in $\varphi_{l'}$.

3.3 Rank Reduction

1. the cut-formula A derives from a standard axiom rule introduction immediately above the cut.

a) end-sequent of φ_l is a standard axiom

$$\dfrac{\overline{A \vdash \boldsymbol{A}} \quad \begin{matrix}(\varphi_r)\\ \boldsymbol{A}, \Pi \vdash \Lambda\end{matrix}}{A, \Pi \vdash \Lambda}\ \text{cut}$$

which transforms to

$$\begin{matrix}(\varphi_r)\\ A, \Pi \vdash \Lambda\end{matrix}$$

b) end-sequent of φ_r is a standard axiom

$$\dfrac{\begin{matrix}(\varphi_l)\\ \Pi \vdash \Lambda, \boldsymbol{A}\end{matrix} \quad \overline{\boldsymbol{A} \vdash A}}{\Pi \vdash \Lambda, A}\ \text{cut}$$

which transforms to

$$\begin{matrix}(\varphi_l)\\ \Pi \vdash \Lambda, A\end{matrix}$$

2. the cut-formula A derives from a weakening immediately above the cut.

a) weakening on the right is the last rule of φ_l:

$$\dfrac{\dfrac{\begin{matrix}(\varphi_{l'})\\ \Gamma \vdash \Delta_1\end{matrix}}{\Gamma \vdash \Delta_1, \underline{\Delta_2}, \underline{\boldsymbol{A}}}\ \text{w : r} \quad \begin{matrix}(\varphi_r)\\ \boldsymbol{A}, \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi \vdash \Delta_1, \Delta_2, \Lambda}\ \text{cut}$$

which transforms to

$$\dfrac{\dfrac{\dfrac{\begin{matrix}(\varphi_{l'})\\ \Gamma \vdash \Delta_1\end{matrix}}{\underline{\boldsymbol{\Pi}}, \boldsymbol{\Gamma} \vdash \Delta_1}\ \text{w : l}}{\underline{\Gamma}, \underline{\Pi} \vdash \Delta_1}\ \pi(\tau) : \text{l}}{\Gamma, \Pi \vdash \Delta_1, \underline{\Delta_2}, \underline{\Lambda}}\ \text{w : r}$$

b) weakening on the left is the last rule of φ_r:

$$\dfrac{\begin{matrix}(\varphi_l)\\ \Gamma \vdash \Delta, \boldsymbol{A}\end{matrix} \quad \dfrac{\begin{matrix}(\varphi_{r'})\\ \Pi_2 \vdash \Lambda\end{matrix}}{\underline{\boldsymbol{A}}, \underline{\Pi_1}, \Pi_2 \vdash \Lambda}\ \text{w : l}}{\Gamma, \Pi_1, \Pi_2 \vdash \Delta, \Lambda}\ \text{cut}$$

which transforms to

$$\dfrac{\dfrac{\dfrac{\begin{matrix}(\varphi_{r'})\\ \Pi_2 \vdash \Lambda\end{matrix}}{\underline{\Gamma}, \underline{\Pi_1}, \Pi_2 \vdash \Lambda}\ \text{w : l}}{\Gamma, \Pi_1, \Pi_2 \vdash \boldsymbol{\Lambda}, \underline{\boldsymbol{\Delta}}}\ \text{w : r}}{\Gamma, \Pi_1, \Pi_2 \vdash \underline{\Delta}, \underline{\Lambda}}\ \pi(\tau) : \text{r}$$

3. the cut-formula A occurs

a) in the antecedent of the end-sequent of φ_l:

$$\dfrac{\begin{matrix}(\varphi_l)\\ \Gamma_1, A, \Gamma_2 \vdash \Delta, \boldsymbol{A}\end{matrix} \quad \begin{matrix}(\varphi_r)\\ \boldsymbol{A}, \Pi \vdash \Lambda\end{matrix}}{\Gamma_1, A, \Gamma_2, \Pi \vdash \Delta, \Lambda}\ \text{cut}$$

which transforms to

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{matrix}(\varphi_r)\\ A, \Pi \vdash \Lambda\end{matrix}}{\underline{A_1}, \ldots, \underline{A_{i-1}}, \underline{A_i}, \underline{\boldsymbol{A_{i+1}}}, \ldots, \underline{\boldsymbol{A_n}}, \boldsymbol{A_i}, \Pi \vdash \Lambda}\ \text{w : l}}{A_1, \ldots, A_{i-1}, \boldsymbol{A_i}, \underline{\boldsymbol{A_i}}, \underline{A_{i+1}}, \ldots, \underline{A_n}, \Pi \vdash \Lambda}\ \pi(\tau_1) : \text{l}}{A_1, \ldots, A_{i-1}, \underline{\boldsymbol{A_i}}, \boldsymbol{A_{i+1}}, \ldots, \boldsymbol{A_n}, \Pi \vdash \Lambda}\ \text{c}(m_1, \ldots, m_{i-1}, 2, m_{i+1}, \ldots, m_n) : \text{l}}{A_1, \ldots, A_{i-1}, \underline{A_{i+1}}, \ldots, \underline{A_n}, \underline{A_i}, \Pi \vdash \Lambda}\ \pi(\tau_2) : \text{l}}{\dfrac{\Gamma_1, A, \Gamma_2, \Pi \vdash \boldsymbol{\Lambda}, \underline{\boldsymbol{\Delta}}}{\Gamma_1, A, \Gamma_2, \Pi \vdash \underline{\Delta}, \underline{\Lambda}}\ \pi(\tau_3) : \text{r}}\ \text{w : r}$$

where $m_j = 1$, $j \in \{1, \ldots, i-1, i+1, \ldots, n\}$. Furthermore $A \equiv A_i$, $\Gamma_1 \equiv A_1, \ldots, A_{i-1}$ and $\Gamma_2 \equiv A_{i+1}, \ldots, A_n$.

Any of the permutation rule inferences may be omitted if they have no effect. Which is for the permutations on the left side the case if $\Gamma_2 = \epsilon$ or $\{\,\Gamma_2\,\} = \{\,A\,\}$ and for the permutation on the right side if $\Delta = \epsilon$, $\Lambda = \epsilon$ or $\Delta = \Lambda$.

b) in the consequent of the end-sequent of φ_r: symmetric to 3a.

4. the cut-formula A derives from a contraction immediately above the cut.

a) contraction on the right is the last rule of φ_l: .

$$\dfrac{\dfrac{\overset{(\varphi_{l'})}{\Gamma \vdash A_1^{m_1},\ldots,A_n^{m_n}}}{\Gamma \vdash A_1,\ldots,A_{n-1},\boldsymbol{A_n}}\;c(m_1,\ldots,m_n):\mathrm{r} \qquad \overset{(\varphi_r)}{\boldsymbol{A_n},\Pi\vdash\Lambda}}{\Gamma,\Pi\vdash A_1,\ldots,A_{n-1},\Lambda}\;\mathrm{cut}$$

which transforms to

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overset{(\varphi_{l_{m_n-1}})}{\Gamma,\Pi^{m_n-1}\vdash A_1^{m_1},\ldots,A_{n-1}^{m_{n-1}},\Lambda^{m_n-1},\boldsymbol{A_n}} \qquad \overset{(\varphi_r)}{\boldsymbol{A_n},\Pi\vdash\Lambda}}{\Gamma,\boldsymbol{B_1},\ldots,\boldsymbol{B_k},\ldots,\boldsymbol{B_1},\ldots,\boldsymbol{B_k}\vdash A_1^{m_1},\ldots,A_{n-1}^{m_{n-1}},\Lambda^{m_n}}\;\mathrm{cut}}{\Gamma,\underline{\boldsymbol{B_1^{m_n}}},\ldots,\underline{\boldsymbol{B_k^{m_n}}}\vdash A_1^{m_1},\ldots,A_{n-1}^{m_{n-1}},\Lambda^{m_n}}\;\pi(\tau_l):\mathrm{l}}{\Gamma,\underline{B_1},\ldots,\underline{B_k}\vdash A_1^{m_1},\ldots,A_{n-1}^{m_{n-1}},\boldsymbol{C_1},\ldots,\boldsymbol{C_l},\ldots,\boldsymbol{C_1},\ldots,\boldsymbol{C_l}}\;c(1,\ldots,1,m_n,\ldots,m_n):\mathrm{l}}{\Gamma,\Pi\vdash \boldsymbol{A_1^{m_1}},\ldots,\boldsymbol{A_{n-1}^{m_{n-1}}},\underline{\boldsymbol{C_1^{m_n}}},\ldots,\underline{\boldsymbol{C_l^{m_n}}}}\;\pi(\tau_r):\mathrm{r}}{\Gamma,\Pi\vdash\underline{A_1},\ldots,\underline{A_{n-1}},\underline{C_1},\ldots,\underline{C_l}}\;c(m_1,\ldots,m_{n-1},m_n,\ldots,m_n):\mathrm{r}$$

with the recursive proof part $\varphi_{l_{i+1}}$:

$$\dfrac{\dfrac{\overset{(\varphi_{l_i})}{\Gamma,\Pi^i\vdash A_1^{m_1},\ldots,A_{n-1}^{m_{n-1}},\Lambda^i,A_n^i,\boldsymbol{A_n}} \qquad \overset{(\varphi_{r_{i+1}})}{\boldsymbol{A_n},\Pi\vdash\Lambda}}{\Gamma,\Pi^{i+1}\vdash A_1^{m_1},\ldots,A_{n-1}^{m_{n-1}},\Lambda^i,\boldsymbol{A_n^i},\boldsymbol{\Lambda}}\;\mathrm{cut}}{\Gamma,\Pi^{i+1}\vdash A_1^{m_1},\ldots,A_{n-1}^{m_{n-1}},\Lambda^i,\underline{\Lambda},\underline{A_n^i}}\;\pi(\tau_i):\mathrm{r}$$

and $\Pi \equiv B_1,\ldots,B_k$, $\Lambda \equiv C_1,\ldots,C_l$.

The subproofs φ_{r_i} are syntactic copies of φ_r where the eigenvariables have been renamed to globally fresh variables. This is essential to keep up regularity of the transformed proof as a whole.

Any of the permutation rule inferences may be omitted if they have no effect. Which is for the permutation τ_l on the left side the case if $\Pi = \epsilon$, $\{\,\Pi\,\} = \{\,B_1\,\}$ or $m_n = 1$, for the permutation τ_r on the right side if $\Lambda = \epsilon$, $\{\,\Lambda\,\} = \{\,C_1\,\}$ or $m_n = 1$ and for the permutations τ_i on the right side if $\Lambda = \epsilon$ or $\{\,\Lambda\,\} = \{\,A_n\,\}$. Furthermore the contraction rule on the left side may be omitted if $\Pi = \epsilon$ or $m_n = 1$.

b) contraction on the left is the last rule of φ_r: symmetric to 4a.

5. the cut-formula A derives from a permutation immediately above the cut.

a) permutation on the right is the last rule of φ_l: .

$$\dfrac{\dfrac{\overset{(\varphi_{l'})}{\Gamma \vdash A_1, \ldots, A_{n-1}, A_n}}{\Gamma \vdash A_{\tau(1)}, \ldots, A_{\tau(n-1)}, \boldsymbol{A_{\tau(n)}}}\;\pi(\tau):\mathrm{r} \qquad \overset{(\varphi_r)}{\boldsymbol{A_{\tau(n)}}, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash A_{\tau(1)}, \ldots, A_{\tau(n-1)}, \Lambda}\;\mathrm{cut}$$

which transforms to

i. if $\tau(n) = n$:

$$\dfrac{\dfrac{\overset{(\varphi_{l'})}{\Gamma \vdash A_1, \ldots, A_{n-1}, \boldsymbol{A_n}} \qquad \overset{(\varphi_r)}{\boldsymbol{A_{\tau(n)}}, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash A_1, \ldots, A_{n-1}, \Lambda}\;\mathrm{cut}}{\Gamma, \Pi \vdash A_{\tau(1)}, \ldots, A_{\tau(n-1)}, \Lambda}\;\pi(\tau_1):\mathrm{r}$$

ii. otherwise let $\tau(n) = i$ for $i \neq n$, then we have to distinguish according to the rule above the permutation rule (the last rule of $\varphi_{l'}$):

A. weakening right (A_i is one of the principal formulas)

$$\dfrac{\overset{(\varphi_{l''})}{\Gamma \vdash \Delta_1}}{\Gamma \vdash \Delta_1, \underline{\Delta_2}, \underline{A_i}, \underline{\Delta_3}}\;\mathrm{w}:\mathrm{r}$$

which transforms to

$$\dfrac{\dfrac{\dfrac{\dfrac{\overset{(\varphi_{l''})}{\Gamma \vdash \Delta_1}}{\underline{\boldsymbol{\Pi}}, \boldsymbol{\Gamma} \vdash \Delta_1}\;\mathrm{w}:\mathrm{l}}{\underline{\Gamma}, \underline{\Pi} \vdash \Delta_1}\;\pi(\tau_2):\mathrm{l}}{\Gamma, \Pi \vdash \Delta_1, \underline{\Delta_2}, \underline{\Delta_3}, \underline{\Lambda}}\;\mathrm{w}:\mathrm{r}}{\Gamma, \Pi \vdash A_{\tau(1)}, \ldots, A_{\tau(n-1)}, \Lambda}\;\pi(\tau_3):\mathrm{r}$$

where $\Delta_1, \Delta_2 \equiv A_1, \ldots, A_{i-1}$ and $\Delta_3 \equiv A_{i+1}, \ldots, A_n$.

The permutation τ_3 need not be applied if $\Gamma = \epsilon$, $\Pi = \epsilon$ or $\Gamma = \Pi$.

B. permutation right

$$\dfrac{\overset{(\varphi_{l''})}{\Gamma \vdash \Gamma \vdash A_{-\tau'(1)}, \ldots, A_{-\tau'(n)}}}{\Gamma \vdash A_1, \ldots, A_n}\;\pi(\tau'):\mathrm{r}$$

where $-\tau'$ denotes the inverse permutation of τ'; which (combining the permutations) transforms to

$$\dfrac{\dfrac{\overset{(\varphi_{l''})}{\Gamma \vdash A_{-\tau'(1)}, \ldots, A_{-\tau'(n-1)}, A_{-\tau'(n)}}}{\Gamma \vdash A_{\tau(1)}, \ldots, A_{\tau(n-1)}, \boldsymbol{A_{\tau(n)}}}\;\pi(\tau_4):\mathrm{r} \qquad \overset{(\varphi_r)}{\boldsymbol{A_{\tau(n)}}, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash A_{\tau(1)}, \ldots, A_{\tau(n-1)}, \Lambda}\;\mathrm{cut}$$

C. contraction right

$$\dfrac{\begin{matrix}(\varphi_{l''})\\ \Gamma \vdash A_1^{m_1},\ldots,A_n^{m_n}\end{matrix}}{\Gamma \vdash A_1,\ldots,A_n}\ \mathrm{c}(m_1,\ldots,m_n):\mathrm{r}$$

which (shifting the permutation putting A_i in place above the contraction) transforms to

$$\dfrac{\dfrac{\dfrac{\begin{matrix}(\varphi_{l''})\\ \Gamma \vdash A_1^{m_1},\ldots,A_{i-1}^{m_{i-1}},\boldsymbol{A_i^{m_i}},\boldsymbol{A_{i+1}^{m_{i+1}}},\ldots,\boldsymbol{A_n^{m_n}}\end{matrix}}{\Gamma \vdash A_1^{m_1},\ldots,A_{i-1}^{m_{i-1}},\underline{A_{i+1}^{m_{i+1}}},\ldots,\underline{A_n^{m_n}},\underline{A_i^{m_i}}}\ \pi(\tau_5):\mathrm{r}}{\Gamma \vdash A_1,\ldots,A_{i-1},A_{i+1},\ldots,A_n,\boldsymbol{A_i}}\ \mathrm{c}(\overline{m}):\mathrm{r} \qquad \begin{matrix}(\varphi_r)\\ \boldsymbol{A_{\tau(n)}},\Pi \vdash \Lambda\end{matrix}}{\dfrac{\Gamma,\Pi \vdash A_1,\ldots,A_{i-1},A_{i+1},\ldots,A_n,\Lambda}{\Gamma,\Pi \vdash A_{\tau(1)},\ldots,A_{\tau(n-1)},\Lambda}\ \pi(\tau_6):\mathrm{r}}\ \mathrm{cut}$$

where $\overline{m} = m_1,\ldots,m_{i-1},m_{i+1},\ldots,m_n,m_i$.

D. the last rule of $\varphi_{l'}$ is an arbitrary unary rule (except the ones already covered by the cases 5(a)iiA to 5(a)iiC), i.e. $\varphi_{l'} =$

$$\dfrac{\begin{matrix}(\varphi_{l''})\\ \Gamma' \vdash \Delta_1, A_i, \Delta_2'\end{matrix}}{\Gamma \vdash \Delta_1, A_i, \Delta_2}\ \rho_u$$

which transforms to

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{matrix}(\varphi_{l''})\\ \Gamma' \vdash \Delta_1, A_i, \Delta_2'\end{matrix}}{\Gamma' \vdash \Delta_1, \Delta_2', \boldsymbol{A_i}}\ \pi(\tau_7):\mathrm{r} \qquad \begin{matrix}(\varphi_r)\\ \boldsymbol{A_i},\Pi \vdash \Lambda\end{matrix}}{\Gamma',\Pi \vdash \Delta_1, \boldsymbol{\Delta_2'}, \boldsymbol{\Lambda}}\ \mathrm{cut}}{\Gamma',\Pi \vdash \Delta_1, \underline{\Lambda}, \underline{\Delta_2'}}\ \pi(\tau_8):\mathrm{r}}{\dfrac{\Gamma,\Pi \vdash A_1,\ldots,A_{i-1},\Lambda,A_{i+1},\ldots,A_n}{\Gamma,\Pi \vdash A_{\tau(1)},\ldots,A_{\tau(n-1)},\Lambda}\ \pi(\tau_9):\mathrm{r}}\ \rho_u$$

where $\Delta_1 \equiv A_1,\ldots,A_{i-1}$ and $\Delta_2 \equiv A_{i+1},\ldots,A_n$.

The permutation τ_8 need not be applied if $\Delta_2' = \epsilon$, $\Lambda = \epsilon$ or $\Delta_2' = \Lambda$.

Remark: A_i is not a principal formula of ρ_u.

E. the last rule of $\varphi_{l'}$ is an arbitrary binary rule, hence either the cut-formula ancestor comes from the left premise of ρ_u, i.e. $\varphi_{l'} =$

$$\dfrac{\begin{matrix}(\varphi_{l'_l})\\ \Gamma_l' \vdash \Delta_{l_1}, A_i, \Delta_{l_2}'\end{matrix} \qquad \begin{matrix}(\varphi_{l'_r})\\ \Gamma_r' \vdash \Delta_r'\end{matrix}}{\Gamma_l, \Gamma_r \vdash \Delta_{l_1}, A_i, \Delta_{l_2}, \Delta_r}\ \rho_b$$

which transforms to

$$\dfrac{\dfrac{\dfrac{\overset{(\varphi_{l'_l})}{\Gamma'_l \vdash \Delta_{l_1}, \boldsymbol{A_i}, \boldsymbol{\Delta'_{l_2}}}}{\Gamma'_l \vdash \Delta_{l_1}, \underline{\Delta'_{l_2}}, \underline{\boldsymbol{A_i}}}\ \pi(\tau_{10}):\mathrm{r} \qquad \overset{(\varphi_r)}{\boldsymbol{A_i}, \Pi \vdash \Lambda}}{\dfrac{\Gamma'_l, \Pi \vdash \Delta_{l_1}, \boldsymbol{\Delta'_{l_2}}, \boldsymbol{\Lambda}}{\Gamma'_l, \Pi \vdash \Delta_{l_1}, \underline{\Lambda}, \underline{\Delta'_{l_2}}}\ \pi(\tau_{11}):\mathrm{r}}\ \mathrm{cut} \qquad \overset{(\varphi_{l'_r})}{\Gamma'_r \vdash \Delta'_r}}{\dfrac{\dfrac{\boldsymbol{\Gamma_l}, \boldsymbol{\Pi}, \Gamma_r \vdash \Delta_{l_1}, \Lambda, \Delta_{l_2}, \Delta_r}{\Gamma_l, \underline{\Gamma_r}, \underline{\Pi} \vdash A_1, \ldots, A_{i-1}, \Lambda, A_{i+1}, \ldots, A_n}\ \pi(\tau_{12}):\mathrm{l}}{\Gamma_l, \Gamma_r, \Pi \vdash A_{\tau(1)}, \ldots, A_{\tau(n-1)}, \Lambda}\ \pi(\tau_{13}):\mathrm{r}}\ \rho_b$$

where $\Gamma \equiv \Gamma_l, \Gamma_r$ and $\Delta_{l_1} \equiv A_1, \ldots, A_{i-1}$ and $\Delta_{l_2}, \Delta_r \equiv A_{i+1}, \ldots, A_n$.

The permutation τ_{11} need not be applied if $\Delta'_{l_2} = \epsilon$, $\Lambda = \epsilon$ or $\Delta'_{l_2} = \Lambda$.

Or the cut-formula ancestor comes from the right premise of ρ_b, i.e. $\varphi_{l'} =$

$$\dfrac{\overset{(\varphi_{l'_l})}{\Gamma'_l \vdash \Delta'_l} \quad \overset{(\varphi_{l'_r})}{\Gamma'_r \vdash \Delta_{r_1}, A_i, \Delta'_{r_2}}}{\Gamma_l, \Gamma_r \vdash \Delta_l, \Delta_{r_1}, A_i, \Delta_{r_2}}\ \rho_b$$

which is symmetric to the previous case.

Remark: A_i is not a principal formula of ρ_b.

b) permutation on the left is the last rule of φ_r: symmetric to 5a.

6. the last rule of

a) φ_l is an arbitrary unary rule on the left:

$$\dfrac{\dfrac{\overset{(\varphi_{l'})}{\Gamma' \vdash \Delta', A}}{\Gamma \vdash \Delta, \boldsymbol{A}}\ \rho_u : \mathrm{l} \qquad \overset{(\varphi_r)}{\boldsymbol{A}, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\ \mathrm{cut}$$

which transforms to

$$\dfrac{\dfrac{\overset{(\varphi_{l'})}{\Gamma' \vdash \Delta', \boldsymbol{A}} \quad \overset{(\varphi_r)}{\boldsymbol{A}, \Pi \vdash \Lambda}}{\Gamma', \Pi \vdash \Delta', \Lambda}\ \mathrm{cut}}{\Gamma, \Pi \vdash \Delta, \Lambda}\ \rho_u : \mathrm{l}$$

b) φ_r is an arbitrary unary rule on the right:

$$\dfrac{\overset{(\varphi_l)}{\Gamma \vdash \Delta, \boldsymbol{A}} \qquad \dfrac{\overset{(\varphi_{r'})}{A, \Pi' \vdash \Lambda'}}{\boldsymbol{A}, \Pi \vdash \Lambda}\ \rho_u : \mathrm{r}}{\Gamma, \Pi \vdash \Delta, \Lambda}\ \mathrm{cut}$$

which transforms to

$$\dfrac{\dfrac{\overset{(\varphi_l)}{\Gamma \vdash \Delta, \boldsymbol{A}} \quad \overset{(\varphi_{r'})}{\boldsymbol{A}, \Pi' \vdash \Lambda'}}{\Gamma, \Pi' \vdash \Delta, \Lambda'}\ \mathrm{cut}}{\Gamma, \Pi \vdash \Delta, \Lambda}\ \rho_u : \mathrm{r}$$

The parameters of ρ_u need to be adjusted in case of a contraction where for every formula of Π (for a contraction on the left) resp. Δ (for a contraction on the right) a 1 must be added to the appropriate side of the argument and in case of a permutation on the right where the permutation has to be shifted by the number of formulas of Δ. Note that the parameter of a permutation on the left need not be modified since in cyclic representation identities at the end are not expressed.

Remark: ρ_u does not operate on the cut-formula.

c) φ_l is an arbitrary binary rule on the left:

$$\dfrac{\dfrac{\overset{(\varphi_{l_l})}{\Gamma'_l \vdash \Delta'_l} \quad \overset{(\varphi_{l_r})}{\Gamma'_r \vdash \Delta'_r, A}}{\Gamma_l, \Gamma_r \vdash \Delta_l, \Delta_r, \boldsymbol{A}}\rho_b : \mathrm{l} \qquad \overset{(\varphi_r)}{\boldsymbol{A}, \Pi \vdash \Lambda}}{\Gamma_l, \Gamma_r, \Pi \vdash \Delta_l, \Delta_r, \Lambda}\mathrm{cut}$$

which transforms to

$$\dfrac{\dfrac{\overset{(\varphi_{l_l})}{\Gamma'_l \vdash \Delta'_l} \quad \dfrac{\dfrac{\overset{(\varphi_{l_r})}{\Gamma'_r \vdash \Delta'_r, \boldsymbol{A}} \quad \overset{(\varphi_r)}{\boldsymbol{A}, \Pi \vdash \Lambda}}{\Gamma'_r, \Pi \vdash \boldsymbol{\Delta'_r}, \boldsymbol{\Lambda}}\mathrm{cut}}{\Gamma'_r, \Pi \vdash \underline{\Lambda}, \underline{\underline{\Delta'_r}}}\pi(\tau_1) : \mathrm{r}}{\Gamma_l, \Gamma_r, \Pi \vdash \Delta_l, \boldsymbol{\Lambda}, \boldsymbol{\Delta_r}}\rho_b : \mathrm{l}}{\Gamma_l, \Gamma_r, \Pi \vdash \Delta_l, \underline{\underline{\Delta_r}}, \underline{\Lambda}}\pi(\tau_2) : \mathrm{r}$$

Both permutation rule inferences may be omitted if Δ'_r contains no auxiliary formula of ρ_b : l (the consequent of the right premise of ρ_b : l has to be thought adjusted then). Either of the permutation rule inferences may be omitted if they have no effect. Which is for the permutation τ_1 the case if $\Lambda = \epsilon$, $\Delta'_r = \epsilon$ or $\Lambda = \Delta'_r$ and for the permutation τ_2 if $\Lambda = \epsilon$, $\Delta_r = \epsilon$ or $\Lambda = \Delta_r$.

d) φ_r is an arbitrary binary rule on the right:

$$\dfrac{\overset{(\varphi_l)}{\Gamma \vdash \Delta, \boldsymbol{A}} \qquad \dfrac{\overset{(\varphi_{r_l})}{A, \Pi'_l \vdash \Lambda'_l} \quad \overset{(\varphi_{r_r})}{\Pi'_r \vdash \Lambda'_r}}{\boldsymbol{A}, \Pi_l, \Pi_r \vdash \Lambda_l, \Lambda_r}\rho_b : \mathrm{r}}{\Gamma, \Pi_l, \Pi_r \vdash \Delta, \Lambda_l, \Lambda_r}\mathrm{cut}$$

which transforms to

$$\dfrac{\dfrac{\dfrac{\dfrac{\overset{(\varphi_l)}{\Gamma \vdash \Delta, \boldsymbol{A}} \quad \overset{(\varphi_{r_l})}{\boldsymbol{A}, \Pi'_l \vdash \Lambda'_l}}{\boldsymbol{\Gamma}, \boldsymbol{\Pi'_l} \vdash \Delta, \Lambda'_l}\mathrm{cut}}{\underline{\underline{\Pi'_l}}, \underline{\Gamma} \vdash \Delta, \Lambda'_l}\pi(\tau_1) : \mathrm{l} \qquad \overset{(\varphi_{r_r})}{\Pi'_r \vdash \Lambda'_r}}{\boldsymbol{\Pi_l}, \boldsymbol{\Gamma}, \Pi_r \vdash \Delta, \Lambda_l, \Lambda_r}\rho_b : \mathrm{r}}{\underline{\Gamma}, \underline{\underline{\Pi_l}}, \Pi_r \vdash \Delta, \Lambda_l, \Lambda_r}\pi(\tau_2) : \mathrm{l}$$

Both permutation rule inferences may be omitted if Π'_l contains no auxiliary formula of ρ_b : r (the antecedent of the left premise of ρ_b : r has to be thought adjusted then). Either of the permutation rule inferences may be omitted if they have no effect. Which is for the permutation τ_1 the case if $\Gamma = \epsilon$, $\Pi'_l = \epsilon$ or $\Gamma = \Pi'_l$ and for the permutation τ_2 if $\Gamma = \epsilon$, $\Pi_l = \epsilon$ or $\Gamma = \Pi_l$.

Remark: ρ_b does not operate on the cut-formula.

CHAPTER 4

Cut-Elimination by Resolution

Cut-elimination by resolution (CERES) is a proof transformation method invented by Baaz and Leitsch (for the foundations see Baaz and Leitsch 2000, Baaz and Leitsch 2006). Which serves the purpose to transform arbitrary **LK**-proofs (possibly) containing cuts of arbitrary complexity into a cut-free **LK**-proof of the same theorem by means of the resolution principle. Since the method permits the use of atomic sequents as axioms it is in general only possible to obtain a proof still containing atomic cuts. But because of the negligible mathematical importance of these cuts this is a compromise worth taking.

CERES is more than just a transformation method; it also analyzes the proofs and detects certain redundancies; information that flows in at certain steps of the transformation process and has a big influence on the generated cut-free proof. In addition CERES should not be understood as kind of proof normalization method like intended by e.g. Gentzen's method. Quite contrary, CERES is not looking for unique or certain proofs. On the contrary, only the variety of possibilities offered during proof transformation allows some interesting experiments such as those performed by the system `CERES` in chapter 8. The ability to use all kinds of resolution refinements plays a central role here.

Because of the close connection between the calculi **LK** and **R** within CERES no one should be amazed that the definitions of these calculi (recall definitions 2.8 and 2.16) show many parallelisms. Especially the definition of the resolution calculus **R** has been designed to be very compatible with **LK**. Inference rules in **R** not making use of unification are already isomorphic to certain **LK**-rules. This design feature will become extra conspicuous in chapter 5.

Another big advantage of CERES is its robustness regarding the concrete definition of the **LK**-rules. A fact that will be pointed out many times within this thesis.

Within this chapter we will first introduce the cut-elimination method itself additionally to some concepts facilitating the proof transformation, which are of big use even outside this context. Followed by a demonstration of the method applied to some introductory examples. Some special cases will even point out some characteristics and properties of the method not obvious at first glance. Finally a comparison with the cut-elimination methods of Gentzen and Schütte-Tait will be performed.

4.1 The Method

In contrast to the traditional cut-elimination methods of chapter 3 which are based on local operations CERES transforms the entire **LK**-proof to its cut-free variant at once. Therefore no reduction rules exist in CERES which applied one by one perform the whole transformation.

The individual steps of CERES are subdivided as follows:

Starting point is an **LK**-proof ψ with a closed end-sequent S (i.e. S contains no free variables), then

1. skolemization of ψ gives $\varphi = \psi^{\mathcal{SK}}$ in case the end-sequent contains strong quantifiers otherwise $\varphi = \psi$,
2. extract the characteristic clause term $\Theta(\varphi)$ of φ,
3. compute the characteristic clause set $\mathrm{CL}(\varphi) = |\,\Theta(\varphi)\,|$ of φ,
4. refute $\mathrm{CL}(\varphi)$ by a resolution refutation γ (which has to be regular),
5. combine the most general unifiers of γ to a global m.g.u., self-application yields δ (of which all unifiers are id now),
6. build the proof projection schemes $\varphi(C)$ for any clause $C \in \mathrm{CL}(\varphi)$ of which an instance occurs as initial clause in δ,
7. generate the concrete instantiations of every proof projection scheme as required by δ,
8. transform δ concatenating the proof projection instances to a cut-free **LK**-proof $\varphi(\delta)$ (containing only atomic cuts).

Skolemization of φ is needed in case the end-sequent contains strong quantifiers since:

- resolution does not distinguish between variables bound by strong and weak quantifiers in φ (clauses can always be thought of as universal lemmas) and
- eigenvariable conditions could be violated by skipping rules of φ during generation of proof projection schemes.

Some remarks on proof skolemization: skolemization of a proof means replacing all variables bound by occurrences of strong quantifiers within the end-sequent by Skolem functions after dropping these quantifiers and propagating the changes upwards in the proof (see e.g. Andrews 1971). Details regarding skolemization and de-skolemization can also be found here (Baaz and Leitsch 1994).

Important for understanding is that cut-formulas cannot be skolemized, e.g. think of the cut

$$\frac{\Gamma \vdash \Delta, \boldsymbol{(\forall x)(\exists y)A(x,y)} \quad \boldsymbol{(\forall x)(\exists y)A(x,y)}, \Pi \vdash \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda}\ \mathrm{cut}$$

which would yield two different skolemizations. The cut-formula occurrence in the left premise becomes $(\exists y)A(c,y)$ and the cut-formula occurrence in the right premise becomes $(\forall x)A(x,f(x))$. Which does not coincide with the semantic of the cut rule since

$$\vdash (\forall x)A(x,f(x)) \supset (\exists y)A(c,y)$$

but more importantly

$$\nvdash (\exists y)A(c,y) \supset (\forall x)A(x,f(x))$$

First, we need to define our main tool. As mentioned earlier the following structure is not only an auxiliary means to an end, moreover it is a symbolic and abstract representation of the clauses and the underlying proof as an extensional algebraic object. As we will see these clause terms remember the ontogenesis of a set of clauses.

Definition 4.1 (clause term). Clause terms are defined inductively in the following way:

1. (Finite) sets of clauses are clause terms.
2. If X and Y are clause terms, then $X \oplus Y$ and $X \otimes Y$ are clause terms.

We will use this structure to build up an abstract binary tree representation of the underlying **LK**-proof. Taking those subsequents of the original initial sequents which contain all ancestors of cut-formula occurrences as leaf nodes, which then get connected in the same way as the binary rule inferences (unary rule inferences are neglected) connect them in the **LK**-proof. More formally this is done by the following algorithm.

Definition 4.2 (characteristic clause term). Let φ be an **LK**-proof and let Ω be the set of all occurrences of cut-formulas in φ. The characteristic clause term $\Theta(\varphi)$ is defined inductively as follows.

For every proof node ν in φ either:

1. ν is the occurrence of an axiom in φ, then the characteristic clause term of φ at the proof node ν corresponds to its subsequent consisting of all formulas which are ancestors of an occurrence in Ω, i.e.
$$\Theta(\varphi.\nu) = \{\, \nu(\Omega)\, \}.$$
2. ν is the immediate successor of the proof node μ at a unary inference rule ρ_u application within φ, i.e. a unary rule ρ_u applied to μ gives ν. Then we simply define
$$\Theta(\varphi.\nu) = \Theta(\varphi.\mu).$$
3. ν is the immediate successor of the proof nodes μ_l and μ_r at a binary inference rule application ρ_b within φ, i.e. a binary rule ρ_b applied to μ_l and μ_r gives ν. Then we distinguish between:
   a) All of the auxiliary and principal formulas of ρ_b are ancestors of Ω, i.e. the auxiliary formulas occur in $\mu_l(\Omega)$ respectively $\mu_r(\Omega)$ and the principal formulas occur in $\nu(\Omega)$. Then
   $$\Theta(\varphi.\nu) = \Theta(\varphi.\mu_l) \oplus \Theta(\varphi.\mu_r).$$
   b) None of the auxiliary and principal formulas of ρ_b is an ancestor of Ω, i.e. the auxiliary formulas occur in $\overline{\mu_l}(\Omega)$ respectively $\overline{\mu_r}(\Omega)$ and the principal formulas occur in $\overline{\nu}(\Omega)$. Then
   $$\Theta(\varphi.\nu) = \Theta(\varphi.\mu_l) \otimes \Theta(\varphi.\mu_r).$$

Finally, the characteristic clause term $\Theta(\varphi)$ of φ is defined as $\Theta(\varphi.\nu)$ where ν is the root node of φ.

Note that in the **LK**-calculus used (see definition 2.8) either all auxiliary and principal formulas of binary rule inferences are ancestors of cut-formulas or none of them. Easy to see since all binary rules are single conclusion rules.

Definition 4.3. Let X, Y be clause terms. We define a mapping $|\ |$ from clause terms to sets of clauses as follows.

$$\begin{aligned} |\,\mathcal{C}\,| &= \mathcal{C} \text{ for sets of clauses } \mathcal{C}, \\ |\,X \oplus Y\,| &= |\,X\,| \cup |\,Y\,|, \\ |\,X \otimes Y\,| &= |\,X\,| \times |\,Y\,|, \end{aligned}$$

where

$$\{\,\Gamma_1 \vdash \Delta_1, \ldots, \Gamma_n \vdash \Delta_n\,\} \times \{\,\Pi_1 \vdash \Lambda_1, \ldots, \Pi_m \vdash \Lambda_m\,\} = \{\,\Gamma_i, \Pi_j \vdash \Delta_i, \Lambda_j \mid 1 \leq i \leq n, 1 \leq j \leq m\,\}.$$

Two clause terms X and Y are equivalent iff the corresponding sets of clauses are equal, i.e.

$$X \sim Y \quad \Leftrightarrow \quad |\,X\,| = |\,Y\,|.$$

This mapping allows the dissolution of the algebraic structure to a pure clause set which enables further processing with well known tools and making it also easier to define a simple notion for the equality of clause terms.

Definition 4.4 (characteristic clause set). Let φ be an **LK**-proof and $\Theta(\varphi)$ be the characteristic clause term of φ. Then the characteristic clause set $\mathrm{CL}(\varphi)$ of φ is given by

$$\mathrm{CL}(\varphi) = |\,\Theta(\varphi)\,|.$$

Remark 4.1. If φ is a cut-free **LK**-proof then there are no occurrences of cut-formulas in φ hence $\mathrm{CL}(\varphi) = \{\vdash\}$.

The following important property of characteristic clause sets is motivated by the thought that starting with exactly the material of an **LK**-proof which forms up the cut-formulas of the proof and reassembling it structurally in the same way would yield a derivation of the empty sequent in **LK**.

Theorem 4.1. Let φ be an **LK**-proof. Then $\mathrm{CL}(\varphi)$ is unsatisfiable, i.e. there exists a resolution refutation of $\mathrm{CL}(\varphi)$.

Proof. In Baaz and Leitsch (2000) or Baaz and Leitsch (2006). □

Since the characteristic clause sets of **LK**-proofs have this nice property it will be possible to use the concept of resolution to break down this atomic structures.

By starting at the initial sequents relative to a clause C of the characteristic clause set $\mathrm{CL}(\varphi)$ of an **LK**-proof φ and skipping all inference rule applications in an **LK**-proof φ operating on ancestors of cut-formula occurrences of φ we get an **LK**-proof $\varphi(C)$ of the form,

$$\begin{array}{c} (\varphi(C)) \\ \Gamma, \Gamma_C \vdash \Delta_C, \Delta \end{array}$$

where $\Gamma \vdash \Delta$ is a subsequent of the end-sequent of φ and $C = \Gamma_C \vdash \Delta_C$.

Or equally speaking just applying those rules not operating on ancestors of cut-formulas yields a proof projection scheme of φ with respect to C.

Definition 4.5 (proof projection scheme). Let φ be an **LK**-proof and let Ω be the set of all occurrences of cut-formulas in φ. The proof projection scheme $\varphi(C)$ of φ with respect to the clause $C \in \mathrm{CL}(\varphi)$ is constructed inductively as follows.

For every proof node ν in φ and for some clause C' either:

1. ν is the occurrence of an axiom S in φ, more detailed

$$\varphi.\nu : \overline{S : \Gamma \vdash \Delta}$$

   then the proof projection scheme of φ at the proof node ν corresponds to the axiom itself (and perhaps some rearrangements afterwards), i.e. $\varphi.\nu(C') =$

$$\cfrac{\cfrac{\overline{\mathbf{\Gamma} \vdash \Delta}}{\underline{\Gamma'}, \underline{\Gamma_{C'}} \vdash \mathbf{\Delta}}\ \pi(\tau_l) : \mathrm{l}}{\Gamma', \Gamma_{C'} \vdash \underline{\Delta_{C'}}, \underline{\Delta'}}\ \pi(\tau_r) : \mathrm{r}$$

   where $\nu(\Omega) : \Gamma_{C'} \vdash \Delta_{C'}$ and $C' = \Gamma_{C'} \vdash \Delta_{C'}$ holds. Furthermore Γ' results from Γ respectively Δ' from Δ by just resecting the formulas occurring within $\nu(\Omega)$. The permutations on the respective sides are only necessary if the sequences of formulas Γ and Δ are not already in the target sort sequence, which is the case if $\Gamma \neq \Gamma', \Gamma_{C'}$ or $\Delta \neq \Delta_{C'}, \Delta'$, and can be omitted otherwise.

   Moving the ancestors of cut-formulas right inside of S ensures that the permutation expense necessary during the construction of the remaining proof projection scheme can be minimized.

2. ν is the immediate successor of the proof node μ at a unary inference rule application ρ_u within φ, i.e.

$$\cfrac{\begin{matrix}(\psi)\\ \mu\ :\ S'\end{matrix}}{\nu\ :\ S}\ \rho_u$$

   where ψ denotes the subproof $\varphi.\mu$ of φ.

   Then we distinguish between:

   a) All of the auxiliary and principal formulas of ρ_u are ancestors of Ω, i.e. the auxiliary formulas occur in $\mu(\Omega)$ and the principal formulas occur in $\nu(\Omega)$. Then we leave it out in the projection scheme by simply not applying it, hence

$$\varphi.\nu(C') = \psi(C').$$

   b) Some of the auxiliary or principal formulas of ρ_u are not ancestors of Ω. Then

$$\cfrac{\begin{matrix}\psi(C')\\ \mathbf{\Pi'}, \Gamma, \Gamma_{C'} \vdash \Delta_{C'}, \Delta, \mathbf{\Lambda'}\end{matrix}}{\underline{\Pi}, \Gamma, \Gamma_{C'} \vdash \Delta_{C'}, \Delta, \underline{\Lambda}}\ \rho'_u$$

where $C' = \Gamma_{C'} \vdash \Delta_{C'}$. Furthermore Π results from Π' respectively Λ from Λ' by application of ρ'_u.

For rules with a single principal formula all of the auxiliary formulas and the principal formula are not ancestors of Ω. Therefore for a logical unary rule ρ_u (as defined in definition 2.8) ρ'_u does not differ from ρ_u.

For rules with multiple principal formulas (in the calculus used the unary structural rules are concerned) the auxiliary and principal formulas are not solely non-ancestors of Ω. Thus ρ'_u has to be a reduction of ρ_u to those parts of the auxiliary and principal formulas that are actually not ancestors of Ω. So all auxiliary formulas of ρ'_u occur in $\overline{\mu}(\Omega)$ and all principal formulas of ρ'_u occur in $\overline{\nu}(\Omega)$. Furthermore the rule parameters of ρ'_u have to be adapted appropriately due to the changed number of auxiliary or principal formulas and eventually also to the changed number of formulas in the context.

An exception could arise in case of ρ'_u being a permutation rule inference of which the applied permutation corresponds to the identity after reducing the auxiliary and principal formulas in the way just mentioned. If this is the case the rule application of ρ'_u can be omitted and $\varphi.\nu(C') = \psi(C')$.

3. ν is the immediate successor of the proof nodes μ_l and μ_r at a binary inference rule application ρ_b within φ, i.e.

$$\dfrac{\begin{matrix}(\psi_l) & (\psi_r)\\ \mu_l : S_l & \mu_r : S_r\end{matrix}}{\nu : S}\ \rho_b$$

where ψ_l and ψ_r denote the subproofs of $\varphi.\mu_l$ respectively $\varphi.\mu_r$.

Then we distinguish between:

a) All of the auxiliary and principal formulas of ρ_b are ancestors of Ω, i.e. the auxiliary formulas occur in $\mu_l(\Omega)$ respectively $\mu_r(\Omega)$ and the principal formulas occur in $\nu(\Omega)$. Then we distinguish the following cases:

i. $C' \in \mathrm{CL}(\psi_l)$ and $C' \notin \mathrm{CL}(\psi_r)$: Then the proof projection scheme $\varphi.\nu(C')$ is obtained by

$$\dfrac{\dfrac{\dfrac{\begin{matrix}\psi_l(C')\\ \Gamma, \Gamma_{C'} \vdash \Delta_{C'}, \Delta\end{matrix}}{\underline{\mathbf{\Pi}}, \mathbf{\Gamma}, \Gamma_{C'} \vdash \Delta_{C'}, \Delta}\ \mathrm{w}:\mathrm{l}}{\underline{\Gamma}, \underline{\Pi}, \Gamma_{C'} \vdash \Delta_{C'}, \Delta}\ \pi(\tau):\mathrm{l}}{\Gamma, \Pi, \Gamma_{C'} \vdash \Delta_{C'}, \Delta, \underline{\Lambda}}\ \mathrm{w}:\mathrm{r}$$

where $\overline{S_r}(\Omega) : \Pi \vdash \Lambda$. $\Gamma_{C'}$ and $\Delta_{C'}$, such that $C' = \Gamma_{C'} \vdash \Delta_{C'}$, are those parts of the end-sequent of the proof projection $\psi_l(C')$ which were ancestors of Ω within the original proof φ.

The above weakening rules can be omitted if the corresponding added sequences of formulas are empty. The permutation rule doesn't need to be introduced if it has no effect, which is the case if either $\Pi = \epsilon$, $\Gamma = \epsilon$ or $\Pi = \Gamma$.

In this case the clause C' is entirely descended from the left subproof φ_l.

ii. $C' \notin \mathrm{CL}(\psi_l)$ and $C' \in \mathrm{CL}(\psi_r)$: Then the proof projection scheme $\varphi.\nu(C')$ is obtained by

$$\dfrac{\dfrac{\dfrac{\dfrac{\psi_r(C')}{\Pi, \Pi_{C'} \vdash \Lambda_{C'}, \Lambda}}{\underline{\Gamma}, \Pi, \Pi_{C'} \vdash \Lambda_{C'}, \Lambda}\ \mathrm{w:l}}{\Gamma, \Pi, \Pi_{C'} \vdash \Lambda_{C'}, \boldsymbol{\Lambda}, \underline{\boldsymbol{\Delta}}}\ \mathrm{w:r}}{\Gamma, \Pi, \Pi_{C'} \vdash \Lambda_{C'}, \underline{\Delta}, \underline{\Lambda}}\ \pi(\tau):\mathrm{r}$$

where $\overline{S_l}(\Omega) : \Gamma \vdash \Delta$. $\Pi_{C'}$ and $\Lambda_{C'}$, such that $C' = \Pi_{C'} \vdash \Lambda_{C'}$, are those parts of the end-sequent of the proof projection $\psi_r(C')$ which were ancestors of Ω within the original proof φ.

The above weakening rules can be omitted if the corresponding added sequences of formulas are empty. The permutation rule doesn't need to be introduced if it has no effect, which is the case if either $\Delta = \epsilon$, $\Lambda = \epsilon$ or $\Delta = \Lambda$.

In this case the clause C' is entirely descended from the right subproof φ_r.

iii. $C' \in \mathrm{CL}(\psi_l)$ and $C' \in \mathrm{CL}(\psi_r)$: In this case we have the possibility to make a decision between proceeding in either the left or the right branch of the proof, since the clause in question descends from both. Choosing the left branch needs handling as in 3(a)i and choosing the right branch needs handling as in 3(a)ii.

b) None of the auxiliary and principal formulas of ρ_b is an ancestor of Ω, i.e. the auxiliary formulas occur in $\overline{\varphi.\mu_l}(\Omega)$ respectively $\overline{\varphi.\mu_r}(\Omega)$ and the principal formulas occur in $\overline{\varphi.\nu}(\Omega)$. Then potentially all possible clause pairs gained from the premises whose composition yields C' would be candidates to proceed further. Hence let

$$\mathcal{C}_p = \{\, (C_1, C_2) \mid C_1 \circ C_2 = C' \wedge C_1 \in \mathrm{CL}(\psi_l) \wedge C_2 \in \mathrm{CL}(\psi_r) \,\}$$

be this set.

After choosing the clause tuple (C_l, C_r), by some strategy, the proof projection scheme $\varphi.\nu(C')$ gets constructed as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\psi_l(C_l)}{\boldsymbol{\Pi_l}, \Gamma_l, \Gamma_{C_l} \vdash \Delta_{C_l}, \Delta_l, \boldsymbol{\Lambda_l}} \quad \dfrac{\psi_r(C_r)}{\boldsymbol{\Pi_r}, \Gamma_r, \Gamma_{C_r} \vdash \Delta_{C_r}, \Delta_r, \boldsymbol{\Lambda_r}}}{\underline{\Pi}, \Gamma_l, \boldsymbol{\Gamma_{C_l}}, \boldsymbol{\Gamma_r}, \Gamma_{C_r} \vdash \Delta_{C_l}, \Delta_l, \Delta_{C_r}, \Delta_r, \underline{\Lambda}}\ \rho_b}{\Pi, \Gamma_l, \underline{\Gamma_r}, \underline{\Gamma_{C_l}}, \Gamma_{C_r} \vdash \Delta_{C_l}, \boldsymbol{\Delta_l}, \boldsymbol{\Delta_{C_r}}, \Delta_r, \Lambda}\ \pi(\tau_l):\mathrm{l}}{\Pi, \Gamma_l, \Gamma_r, \Gamma_{C_l}, \Gamma_{C_r} \vdash \Delta_{C_l}, \underline{\Delta_{C_r}}, \underline{\Delta_l}, \Delta_r, \Lambda}\ \pi(\tau_r):\mathrm{r}$$

where $C' = \Gamma_{C_l}, \Gamma_{C_r} \vdash \Delta_{C_l}, \Delta_{C_r}$. Furthermore Π results from Π_l and Π_r respectively Λ from Λ_l and Λ_r by application of ρ_b.

Either of the permutation rule inferences may be omitted if they have no effect. Which is for the permutation on the left side the case if $\Gamma_{C_l} = \epsilon$, $\Gamma_r = \epsilon$ or $\Gamma_{C_l} = \Gamma_r$ and for the permutation on the right side if $\Delta_{C_r} = \epsilon$, $\Delta_l = \epsilon$ or $\Delta_{C_r} = \Delta_l$.

Note that in the calculus used (see definition 2.8) either all auxiliary and principal formulas of binary rule inferences are ancestors of cut-formulas or none of them.

Finally, the proof projections scheme $\varphi(C)$ of φ with respect to the clause C is defined as $\varphi.\nu(C)$ where ν is the uppermost node of φ of which the root node ν' of its associated proof projection scheme $\varphi.\nu(C)$ fulfills the following two properties:

- $\nu'(\Omega) = C$ and
- $\overline{\nu'}(\Omega)$ is a subsequent of the end-sequent of φ.

The necessary but additional weakening and permutation rule inferences are due to the multiplicative version of the underlying **LK**. An additive version of the calculus would have been advantageous from this perspective but would at the same time entail other major drawbacks which will among other things be discussed in chapter 7.

Remark 4.2. During formation of the proof projection schemes no eigenvariable conditions may be violated because on the one hand either there occur no strong quantifiers in the end-sequent at all or the proof has been skolemized and on the other hand the proof projection schemes are cut-free hence the subformula property guarantees that no new strong quantifiers (with this new eigenvariables) could have been introduced.

Definition 4.6 (proof projection instance). Let φ be an **LK**-proof and let $\varphi(C)$ be a proof projection scheme of φ with respect to the clause $C \in \mathrm{CL}(\varphi)$. Then a proof projection instance $\varphi(C)\sigma$ is a concrete instantiation of the proof projection scheme $\varphi(C)$ by the substitution σ of which the domain is the set of all free variables of $\varphi(C)$ and the range is the set of terms.

The Idea behind this definition is that those formulas of the end-sequent of a proof projection $\varphi(C)$ of an **LK**-proof φ which are going into the end-sequent of φ do not contain any free variables since it consists only of closed formulas. Therefore an arbitrary instantiation of the variables of C, which can of course only occur freely in $\varphi(C)$, will be *absorbed* by the weak quantifier introductions within the projection scheme — which have to occur if $\varphi(C)$ contains free variables at all.

Now what remains to be done is to refute the characteristic clause set $\mathrm{CL}(\varphi)$ of the **LK**-proof φ by a regular resolution refutation γ. Combine all most general unifiers to a global most general unifier of the entire refutation γ and apply it to γ to obtain δ a refutation of $\mathrm{CL}(\varphi)$ where all unifiers are id.

This refutation δ serves as a skeleton of the cut-free variant of φ (containing only atomic cuts) by replacing every leaf node ν in δ which is labelled by a clause C' — an instance of a clause $C \in \mathrm{CL}(\varphi)$ by the substitution σ — with the appropriate proof projection instance $\varphi(C)\sigma$ and all rule inferences of δ are mapped to equivalent rule inferences in **LK**.

Definition 4.7. Let φ be an **LK**-proof of S and $\mathrm{PROJ}(\varphi)$ be the set of all possible proof projection schemes of φ, i.e.

$$\mathrm{PROJ}(\varphi) = \{\, \varphi(C) \mid C \in \mathrm{CL}(\varphi) \,\}.$$

Additionally let δ be a resolution refutation of the characteristic clause set $\mathrm{CL}(\varphi)$ of φ where all unifiers are the identity; obtained from a resolution refutation γ by application of a global most general unifier θ, i.e. $\delta = \gamma\theta$. Then the concatenating transformation $\varphi(\delta)$ is constructed inductively by induction on the structure of δ as follows.

For every proof node ν in δ either:

1. ν is the occurrence of an initial clause C' in δ, more detailed

$$\delta.\nu : \overline{C' : \Gamma_{C'} \vdash \Delta_{C'}}$$

which transforms to

$$\cfrac{\cfrac{\begin{array}{c}(\varphi(C)(\sigma\theta))\\ \mathbf{\Gamma}, \mathbf{\Gamma_{C'}} \vdash \Delta_{C'}, \Delta\end{array}}{\underline{\Gamma_{C'}}, \underline{\Gamma} \vdash \mathbf{\Delta_{C'}}, \mathbf{\Delta}}\ \pi(\tau_l) : \mathrm{l}}{\Gamma_{C'}, \Gamma \vdash \underline{\Delta}, \underline{\Delta_{C'}}}\ \pi(\tau_r) : \mathrm{r}$$

where $\varphi(C)\sigma$ is a proof projection instance of a $\varphi(C) \in \mathrm{PROJ}(\varphi)$ instantiated by the (matching) substitution σ such that $C' = C\sigma$ and $C \in \mathrm{CL}(\varphi)$.Should be $\Gamma = \epsilon$, $\Gamma_{C'} = \epsilon$ or $\Gamma = \Gamma_{C'}$ respectively $\Delta = \epsilon$, $\Delta_{C'} = \epsilon$ or $\Delta = \Delta_{C'}$ then the permutation rule applying τ_l resp. τ_l can be skipped.

2. ν is the immediate successor of the proof node μ at a unary inference rule application ρ_u within δ, then $\delta.\nu$ transforms to

$$\cfrac{\begin{array}{c}(\varphi(\delta.\mu))\\ \mathbf{\Pi'}, \Gamma \vdash \Delta, \mathbf{\Lambda'}\end{array}}{\underline{\Pi}, \Gamma \vdash \Delta, \underline{\Lambda}}\ \rho'_u$$

where Π results from Π' respectively Λ from Λ' by application of ρ'_u.

In case ρ_u is a permutation rule (on the left/right side) in **R** then ρ'_u becomes a permutation rule (on the left/right side) in **LK** where the permutations applied have to be extended by identity permutations for the additional innermost formulas. Otherwise ρ_u is a factoring rule (on the left/right side) in **R** then ρ'_u becomes a contraction rule (on the left/right side) in **LK** where the argument is a vector of 1s except for the outermost atom formula where the multiplicity of this atom has to be considered.

3. ν is the immediate successor of the proof nodes μ_l and μ_r at a binary inference rule application ρ_b within δ, then $\delta.\nu$ transforms to

$$\cfrac{\begin{array}{c}(\varphi(\delta.\mu_l)) \qquad (\varphi(\delta.\mu_r))\\ \mathbf{\Pi_l}, \Gamma \vdash \Delta, \mathbf{\Lambda_l}\mathbf{\Pi_r}, \Gamma \vdash \Delta, \mathbf{\Lambda_r}\end{array}}{\underline{\Pi}, \Gamma \vdash \Delta, \underline{\Lambda}}\ \rho'_b$$

where Π results from Π_l and Π_r respectively Λ from Λ_l and Λ_r by application of ρ'_b.

ρ'_u becomes a cut rule inference of **LK** for ρ_b being a resolution rule of **R** where the cut-formula is the formula resolved upon (which is unique since the m.g.u. of ρ_b is id).

Note, certainly ρ_b can only be a resolution rule for **R** but to keep this transformation algorithm as general as possible (intending future extensions) it is defined this way.

Finally, the transformation $\varphi(\delta)$ is defined as taking the transformation after ν, ν being the root of δ, and applying the following last sequence of **LK**-inferences (if necessary):

- a permutation on each side to achieve the same order of the formulas as in S,
- a contraction on each side to achieve the same multiplicity of the formulas as in S and
- a weakening on each side to add missing formulas of S.

This last sequence of inferences ensures the syntactical equivalence of the end-sequents of φ and $\varphi(\delta)$ — skip them if this is not desired.

"Mission accomplished." Starting with an **LK**-proof φ of a skolemized and closed end-sequent S by refuting $\mathrm{CL}(\varphi)$ by γ and applying the combined global most general unifier to γ we obtain δ and finally $\varphi(\delta)$ yields an **LK**-proof ψ of S containing at most atomic cut rule inferences, i.e. the cut-formulas have logical complexity 0.

Proposition 4.1. CERES also eliminates semantic cuts, i.e. is a cut-elimination method for **LKp**.

Proof. In Baaz and Leitsch (2000). □

Note that the standard methods of cut-elimination (e.g. Gentzen's method) are not capable of eliminating semantic cuts.

4.2 Examples

To illustrate the method we will demonstrate some features and specialties of CERES with the following examples. The first example is kept simple on purpose to facilitate understanding of the work flow of CERES. The algorithms and definitions are roughly repeated and explained when used.

Example 4.1. Now, let φ be the proof

$$\dfrac{\varphi_l \qquad \varphi_r}{(\forall x)(\forall y)(P(x,y) \supset Q(x,y)) \vdash (\exists x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y))}\ \text{cut}$$

where φ_l is

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\boldsymbol{P(u,a)}^+ \vdash P(u,a)}}{\vdash \boldsymbol{P(u,a)}, \underline{\boldsymbol{\neg P(u,a)}}^+}\ \neg:\mathrm{r}}{\vdash \underline{\neg P(u,a)}^+, \underline{\boldsymbol{P(u,a)}}}\ \pi((1\,2)):\mathrm{r} \qquad \dfrac{}{\boldsymbol{Q(u,a)} \vdash Q(u,a)^+}}{\underline{P(u,a) \supset Q(u,a)} \vdash \boldsymbol{\neg P(u,a)}^+, \boldsymbol{Q(u,a)}^+}\ \supset:\mathrm{l}}{\boldsymbol{P(u,a) \supset Q(u,a)} \vdash \underline{\neg P(u,a) \vee Q(u,a)}^+}\ \vee:\mathrm{r}}{\underline{\boldsymbol{(\forall y)(P(u,y) \supset Q(u,y))}} \vdash \neg P(u,a) \vee Q(u,a)^+}\ \forall:\mathrm{l}}{\underline{(\forall x)(\forall y)(P(x,y) \supset Q(x,y))} \vdash \boldsymbol{\neg P(u,a) \vee Q(u,a)}^+}\ \forall:\mathrm{l}}{(\forall x)(\forall y)(P(x,y) \supset Q(x,y)) \vdash \underline{\boldsymbol{(\exists y)(\neg P(u,y) \vee Q(u,y))}}^+}\ \exists:\mathrm{r}}{(\forall x)(\forall y)(P(x,y) \supset Q(x,y)) \vdash \underline{\boldsymbol{(\forall x)(\exists y)(\neg P(x,y) \vee Q(x,y))}}^+}\ \forall:\mathrm{r}$$

and φ_r is

$$
\cfrac{
\cfrac{
\cfrac{
\cfrac{
\cfrac{}{P(b,v) \vdash P(b,v)^+}
}{\neg P(b,v)^+, P(b,v) \vdash}\ \neg : \mathrm{l}
}{P(b,v), \neg P(b,v)^+ \vdash}\ \pi((1\,2)) : \mathrm{l}
}{\neg P(b,v)^+ \vdash \neg P(b,v)}\ \neg : \mathrm{r}
\qquad
\cfrac{
\cfrac{
\cfrac{}{Q(b,v)^+ \vdash Q(b,v)}
}{\neg Q(b,v), Q(b,v)^+ \vdash}\ \neg : \mathrm{l}
}{Q(b,v)^+, \neg Q(b,v) \vdash}\ \pi((1\,2)) : \mathrm{l}
}{\neg P(b,v) \vee Q(b,v)^+, \neg Q(b,v) \vdash \neg P(b,v)}\ \vee : \mathrm{l}
$$

$$
\cfrac{
\cfrac{
\cfrac{
\cfrac{
\cfrac{
\cfrac{\neg P(b,v) \vee Q(b,v)^+, \neg Q(b,v) \vdash \neg P(b,v)}{\neg Q(b,v), \neg P(b,v) \vee Q(b,v)^+ \vdash \neg P(b,v)}\ \pi((1\,2)) : \mathrm{l}
}{\neg P(b,v) \vee Q(b,v)^+ \vdash \neg Q(b,v) \supset \neg P(b,v)}\ \supset : \mathrm{r}
}{\neg P(b,v) \vee Q(b,v)^+ \vdash (\exists y)(\neg Q(b,y) \supset \neg P(b,y))}\ \exists : \mathrm{r}
}{\neg P(b,v) \vee Q(b,v)^+ \vdash (\exists x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y))}\ \exists : \mathrm{r}
}{(\exists y)(\neg P(b,y) \vee Q(b,y))^+ \vdash (\exists x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y))}\ \exists : \mathrm{l}
}{(\forall x)(\exists y)(\neg P(x,y) \vee Q(x,y))^+ \vdash (\exists x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y))}\ \forall : \mathrm{l}
$$

The extraction of the characteristic clause term happens top down starting with those parts of the initial sequents that are marked as ancestors of cut-formulas (by crosses) which are now interpreted as sets. At every occurrence of a binary rule the two clause terms resulting from the premises are connected by a binary operator. Depending whether the auxiliary formulas of the inference were ancestors of cut-formulas or not the operator will either be $\oplus$ or $\otimes$. All unary inference rules have no influence on the clause term and hence it remains unchanged.

For the example this yields the following characteristic clause term

$$\Theta(\varphi) = ((\{\, P(u,a) \vdash \,\} \otimes \{\, \vdash Q(u,a) \,\}) \oplus (\{\, \vdash P(b,v) \,\} \oplus \{\, Q(b,v) \vdash \,\}))$$

which characterizes those parts of the axiom sequents which have been used to derive the cut-formula (on both sides).

The operator $\oplus$ of the clause term is interpreted as union and the operator $\otimes$ as merge, i.e. the antecedents and consequent parts of different sequents are exchanged such that only one part is exchanged at once.

Hence by evaluation of $\Theta(\varphi)$ for the characteristic clause set $|\,\Theta(\varphi)\,|$ of φ we obtain

$$
\begin{aligned}
\mathrm{CL}(\varphi) = \{ \vdash P(b,v), & \qquad (C_1)\\
P(u,a) \vdash Q(u,a), & \qquad (C_2)\\
Q(b,v) \vdash \}. & \qquad (C_3)
\end{aligned}
$$

The characteristic clause set of an **LK**-proof is always unsatisfiable. Therefore one can always find a resolution refutation of the characteristic clause set.

In particular, we define a resolution refutation γ of $\mathrm{CL}(\varphi)$:

$$
\cfrac{
\cfrac{\cfrac{}{\vdash P(b,v)} \quad \cfrac{}{P(u,a) \vdash Q(u,a)}}{\vdash Q(b,a)}\ \mathrm{r}(\sigma)
\qquad \cfrac{}{Q(b,v) \vdash}
}{\vdash}\ \mathrm{r}(\sigma)
$$

and a corresponding (ground) refutation δ of γ, i.e. $\delta = \gamma\sigma$:

$$\dfrac{\dfrac{\dfrac{}{\vdash \boldsymbol{P(b,a)}} \quad \dfrac{}{\boldsymbol{P(b,a)} \vdash Q(b,a)}}{\vdash \boldsymbol{Q(b,a)}}\ \mathsf{r(id)} \quad \dfrac{}{\boldsymbol{Q(b,a)} \vdash}}{\vdash}\ \mathsf{r(id)}$$

with the (ground) substitution $\sigma = \{\, u \mapsto b, v \mapsto a \,\}$.

Now we have to reduce φ to proof projection schemes of the clauses used as initial clauses in the resolution refutation of $\mathrm{CL}(\varphi)$. A proof projection scheme of φ w.r.t. a clause $C \in \mathrm{CL}(\varphi)$ is defined by skipping all inferences going into cuts, which leads to a cut-free proof of (a subsequent of) the end-sequent extended by C.

Again, we start at the initial sequents (without those parts marked as ancestors of cut-formulas and not necessary for the creation of the clause in question) and apply all inference rules not operating on ancestors of cut-formulas until all such binary rules have been applied and a subsequent of the end-sequent has been composed.

The proof projection scheme $\varphi(C_1)$ of φ corresponding to the clause C_1 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\boldsymbol{P(b,v)} \vdash P(b,v)}}{\vdash P(b,v), \underline{\neg P(b,v)}}\ \neg:\mathrm{r}}{\underline{\boldsymbol{\neg Q(b,v)}} \vdash P(b,v), \boldsymbol{\neg P(b,v)}}\ \mathrm{w}:\mathrm{l}}{\vdash P(b,v), \underline{\boldsymbol{\neg Q(b,v) \supset \neg P(b,v)}}}\ \supset:\mathrm{r}}{\vdash P(b,v), \underline{\boldsymbol{(\exists y)(\neg Q(b,y) \supset \neg P(b,y))}}}\ \exists:\mathrm{r}}{\vdash P(b,v), \underline{(\exists x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y))}}\ \exists:\mathrm{r}$$

and let $\chi_1 = \varphi(C_1)\sigma$ be a proof projection instance.

The proof projection scheme $\varphi(C_2)$ of φ corresponding to the clause C_2 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{P(u,a) \vdash \boldsymbol{P(u,a)}} \quad \dfrac{}{\boldsymbol{Q(u,a)} \vdash Q(u,a)}}{\underline{\boldsymbol{P(u,a) \supset Q(u,a)}}, P(u,a) \vdash Q(u,a)}\ \supset:\mathrm{l}}{\underline{\boldsymbol{(\forall y)(P(u,y) \supset Q(u,y))}}, P(u,a) \vdash Q(u,a)}\ \forall:\mathrm{l}}{\underline{(\forall x)(\forall y)(P(x,y) \supset Q(x,y))}, P(u,a) \vdash Q(u,a)}\ \forall:\mathrm{l}$$

and let $\chi_2 = \varphi(C_2)\sigma$ be a proof projection instance.

The proof projection scheme $\varphi(C_3)$ of φ corresponding to the clause C_3 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Q(b,v) \vdash \boldsymbol{Q(b,v)}}}{\underline{\neg Q(b,v)}, Q(b,v) \vdash}\ \neg:\mathrm{l}}{\boldsymbol{\neg Q(b,v)}, Q(b,v) \vdash \underline{\boldsymbol{\neg P(b,v)}}}\ \mathrm{w}:\mathrm{r}}{Q(b,v) \vdash \underline{\boldsymbol{\neg Q(b,v) \supset \neg P(b,v)}}}\ \supset:\mathrm{r}}{Q(b,v) \vdash \underline{\boldsymbol{(\exists y)(\neg Q(b,y) \supset \neg P(b,y))}}}\ \exists:\mathrm{r}}{Q(b,v) \vdash \underline{(\exists x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y))}}\ \exists:\mathrm{r}$$

and let $\chi_3 = \varphi(C_3)\sigma$ be a proof projection instance.

Finally the proof projection instances can be composed to a cut-free proof of φ, i.e. a proof of φ containing only atomic cuts, using its resolution refutation δ as a skeleton.

$$
\dfrac{\dfrac{\dfrac{\dfrac{(\chi_1)}{\vdash \boldsymbol{P(b,a)}, \boldsymbol{Y}}}{\vdash \underline{Y}, \underline{\boldsymbol{P(b,a)}}}\;\pi((1\,2)):\mathrm{r} \qquad \dfrac{\dfrac{(\chi_2)}{\boldsymbol{X}, \boldsymbol{P(b,a)} \vdash Q(b,a)}}{\underline{\boldsymbol{P(b,a)}}, \underline{X} \vdash Q(b,a)}\;\pi((1\,2)):\mathrm{l}}{X \vdash Y, \boldsymbol{Q(b,a)}}\;\mathrm{cut} \qquad \dfrac{(\chi_3)}{\boldsymbol{Q(b,a)} \vdash Y}}{\dfrac{X \vdash \boldsymbol{Y}, \boldsymbol{Y}}{X \vdash \underline{Y}}\;\mathrm{c}(2):\mathrm{r}}\;\mathrm{cut}
$$

where $X = (\forall x)(\forall y)(P(x,y) \supset Q(x,y))$ and $Y = (\exists x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y))$.

The following example demonstrates the necessary changes caused by a slightly modified end-sequent requiring skolemization.

Example 4.2. Let ψ be the proof

$$
\dfrac{\psi_l \quad \psi_r}{(\forall x)((\forall y)P(x,y) \supset (\exists y)Q(x,y)) \vdash (\forall x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y))}\;\mathrm{cut}
$$

with the proof parts ψ_l:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\boldsymbol{P(u,v)} \vdash P(u,v)}}{\vdash P(u,v), \underline{\neg P(u,v)}}\;\neg:\mathrm{r}}{\vdash P(u,v), \boldsymbol{\neg P(u,v)}, \underline{\boldsymbol{Q(u,v)}}}\;\mathrm{w}:\mathrm{r}}{\vdash P(u,v), \underline{\boldsymbol{\neg P(u,v) \lor Q(u,v)}}}\;\lor:\mathrm{r}}{\vdash \boldsymbol{P(u,v)}, \underline{\boldsymbol{(\exists y)(\neg P(u,y) \lor Q(u,y))}}}\;\exists:\mathrm{r}}{\vdash \underline{(\exists y)(\neg P(u,y) \lor Q(u,y))}, \underline{\boldsymbol{P(u,v)}}}\;\pi((1\,2)):\mathrm{r}}{\vdash (\exists y)(\neg P(u,y) \lor Q(u,y)), \underline{\boldsymbol{(\forall y)P(u,y)}}}\;\forall:\mathrm{r} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Q(u,v) \vdash Q(u,v)}}{Q(u,v) \vdash \boldsymbol{Q(u,v)}, \underline{\boldsymbol{\neg P(u,v)}}}\;\mathrm{w}:\mathrm{r}}{Q(u,v) \vdash \underline{\boldsymbol{\neg P(u,v)}}, \underline{\boldsymbol{Q(u,v)}}}\;\pi((1\,2)):\mathrm{r}}{Q(u,v) \vdash \underline{\boldsymbol{\neg P(u,v) \lor Q(u,v)}}}\;\lor:\mathrm{r}}{\boldsymbol{Q(u,v)} \vdash \underline{(\exists y)(\neg P(u,y) \lor Q(u,y))}}\;\exists:\mathrm{r}}{\underline{\boldsymbol{(\exists y)Q(u,y)}} \vdash (\exists y)(\neg P(u,y) \lor Q(u,y))}\;\exists:\mathrm{l}}{\dfrac{\underline{(\forall y)P(u,y) \supset (\exists y)Q(u,y)} \vdash \boldsymbol{(\exists y)(\neg P(u,y) \lor Q(u,y))}, \boldsymbol{(\exists y)(\neg P(u,y) \lor Q(u,y))}}{\dfrac{\boldsymbol{(\forall y)P(u,y) \supset (\exists y)Q(u,y)} \vdash \underline{(\exists y)(\neg P(u,y) \lor Q(u,y))}}{\dfrac{\underline{(\forall x)((\forall y)P(x,y) \supset (\exists y)Q(x,y))} \vdash \boldsymbol{(\exists y)(\neg P(u,y) \lor Q(u,y))}}{(\forall x)((\forall y)P(x,y) \supset (\exists y)Q(x,y)) \vdash \underline{\boldsymbol{(\forall x)(\exists y)(\neg P(x,y) \lor Q(x,y))}}}\;\forall:\mathrm{r}}\;\forall:\mathrm{l}}\;\mathrm{c}(2):\mathrm{r}}\;\supset:\mathrm{l}
$$

and ψ_r:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P(u,v) \vdash P(u,v)}}{\neg P(u,v), P(u,v) \vdash}\;\neg:\mathrm{l}}{P(u,v), \neg P(u,v) \vdash}\;\pi((1\,2)):\mathrm{l}}{\neg P(u,v) \vdash \neg P(u,v)}\;\neg:\mathrm{r} \qquad \dfrac{\dfrac{\dfrac{}{Q(u,v) \vdash Q(u,v)}}{\neg Q(u,v), Q(u,v) \vdash}\;\neg:\mathrm{l}}{Q(u,v), \neg Q(u,v) \vdash}\;\pi((1\,2)):\mathrm{l}}{\neg P(u,v) \lor Q(u,v), \neg Q(u,v) \vdash \neg P(u,v)}\;\lor:\mathrm{l}}{\neg Q(u,v), \neg P(u,v) \lor Q(u,v) \vdash \neg P(u,v)}\;\pi((1\,2)):\mathrm{l}}{\neg P(u,v) \lor Q(u,v) \vdash \neg Q(u,v) \supset \neg P(u,v)}\;\supset:\mathrm{r}}{\neg P(u,v) \lor Q(u,v) \vdash (\exists y)(\neg Q(u,y) \supset \neg P(u,y))}\;\exists:\mathrm{r}}{(\exists y)(\neg P(u,y) \lor Q(u,y)) \vdash (\exists y)(\neg Q(u,y) \supset \neg P(u,y))}\;\exists:\mathrm{l}}{(\forall x)(\exists y)(\neg P(x,y) \lor Q(x,y)) \vdash (\exists y)(\neg Q(u,y) \supset \neg P(u,y))}\;\forall:\mathrm{l}}{(\forall x)(\exists y)(\neg P(x,y) \lor Q(x,y)) \vdash (\forall x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y))}\;\forall:\mathrm{r}
$$

Three strong quantifiers occurring in the end-sequent, obviously force this particular proof to be skolemized. The skolemized variant of ψ is called $\psi^{\mathcal{SK}}$.

Now, $\psi^{\mathcal{SK}}$ results to be the proof

$$
\dfrac{\psi_l^{\mathcal{SK}} \qquad \psi_r^{\mathcal{SK}}}{(\forall x)(P(x,f(x)) \supset Q(x,g(x))) \vdash (\exists y)(\neg Q(c,y) \supset \neg P(c,y))}\;\mathrm{cut}
$$

with the skolemized proof parts $\psi_l^{\mathcal{SK}}$:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P(u,f(u)) \vdash P(u,f(u))}}{\vdash P(u,f(u)), \neg P(u,f(u))}\;\neg:\mathrm{r}}{\vdash P(u,f(u)), \neg P(u,f(u)), Q(u,f(u))}\;\mathrm{w}:\mathrm{r}}{\vdash P(u,f(u)), \neg P(u,f(u)) \lor Q(u,f(u))}\;\lor:\mathrm{r}}{\vdash P(u,f(u)), (\exists y)(\neg P(u,y) \lor Q(u,y))}\;\exists:\mathrm{r}}{\vdash (\exists y)(\neg P(u,y) \lor Q(u,y)), P(u,f(u))}\;\pi((1\,2)):\mathrm{r} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Q(u,g(u)) \vdash Q(u,g(u))}}{Q(u,g(u)) \vdash Q(u,g(u)), \neg P(u,g(u))}\;\mathrm{w}:\mathrm{r}}{Q(u,g(u)) \vdash \neg P(u,g(u)), Q(u,g(u))}\;\pi((1\,2)):\mathrm{r}}{Q(u,g(u)) \vdash \neg P(u,g(u)) \lor Q(u,g(u))}\;\lor:\mathrm{r}}{Q(u,g(u)) \vdash (\exists y)(\neg P(u,y) \lor Q(u,y))}\;\exists:\mathrm{r}}{P(u,f(u)) \supset Q(u,g(u)) \vdash (\exists y)(\neg P(u,y) \lor Q(u,y)), (\exists y)(\neg P(u,y) \lor Q(u,y))}\;\supset:\mathrm{l}}{P(u,f(u)) \supset Q(u,g(u)) \vdash (\exists y)(\neg P(u,y) \lor Q(u,y))}\;\mathrm{c}(2):\mathrm{r}}{(\forall x)(P(x,f(x)) \supset Q(x,g(x))) \vdash (\exists y)(\neg P(u,y) \lor Q(u,y))}\;\forall:\mathrm{l}}{(\forall x)(P(x,f(x)) \supset Q(x,g(x))) \vdash (\forall x)(\exists y)(\neg P(x,y) \lor Q(x,y))}\;\forall:\mathrm{r}
$$

and $\psi_r^{\mathcal{SK}}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P(c,v) \vdash \boldsymbol{P(c,v)}}}{\boldsymbol{\neg P(c,v)}, \boldsymbol{P(c,v)} \vdash}\ \neg:\mathrm{l}}{\boldsymbol{P(c,v)}, \neg P(c,v) \vdash}\ \pi((1\,2)):\mathrm{l}}{\boldsymbol{\neg P(c,v)} \vdash \neg P(c,v)}\ \neg:\mathrm{r} \qquad \dfrac{\dfrac{\dfrac{}{Q(c,v) \vdash \boldsymbol{Q(c,v)}}}{\boldsymbol{\neg Q(c,v)}, \boldsymbol{Q(c,v)} \vdash}\ \neg:\mathrm{l}}{\boldsymbol{Q(c,v)}, \neg Q(c,v) \vdash}\ \pi((1\,2)):\mathrm{l}}{\boldsymbol{\neg P(c,v) \vee Q(c,v)}, \boldsymbol{\neg Q(c,v)} \vdash \neg P(c,v)}\ \vee:\mathrm{l}}{\boldsymbol{\neg Q(c,v)}, \neg P(c,v) \vee Q(c,v) \vdash \boldsymbol{\neg P(c,v)}}\ \pi((1\,2)):\mathrm{l}}{\neg P(c,v) \vee Q(c,v) \vdash \boldsymbol{\neg Q(c,v) \supset \neg P(c,v)}}\ \supset:\mathrm{r}}{\boldsymbol{\neg P(c,v) \vee Q(c,v)} \vdash (\exists y)(\neg Q(c,y) \supset \neg P(c,y))}\ \exists:\mathrm{r}}{\boldsymbol{(\exists y)(\neg P(c,y) \vee Q(c,y))} \vdash (\exists y)(\neg Q(c,y) \supset \neg P(c,y))}\ \exists:\mathrm{l}}{\boldsymbol{(\forall x)(\exists y)(\neg P(x,y) \vee Q(x,y))} \vdash (\exists y)(\neg Q(c,y) \supset \neg P(c,y))}\ \forall:\mathrm{l}$$

Construction of the characteristic clause term of $\psi^{\mathcal{SK}}$,

$$\Theta(\psi^{\mathcal{SK}}) = ((\{\, P(u,f(u)) \vdash \} \otimes \{ \vdash Q(u,g(u)) \}) \oplus (\{ \vdash P(c,v) \} \oplus \{\, Q(c,v) \vdash \}))$$

and computation of the associated characteristic clause set

$$\begin{aligned} \mathrm{CL}(\psi^{\mathcal{SK}}) = \{ & \vdash P(c,v), && (C_1)\\ & P(u,f(u)) \vdash Q(u,g(u)), && (C_2)\\ & Q(c,v) \vdash \}. && (C_3) \end{aligned}$$

yields, afterwards, a resolution refutation γ of $\mathrm{CL}(\psi^{\mathcal{SK}})$:

$$\dfrac{\dfrac{\dfrac{}{\vdash P(c,v)} \quad \dfrac{}{P(u,f(u)) \vdash Q(u,g(u))}}{\vdash Q(c,g(c))}\ \mathrm{r}(\theta) \qquad \dfrac{}{Q(c,v') \vdash}}{\vdash}\ \mathrm{r}(\theta)$$

and a corresponding (ground) refutation δ of γ, i.e. $\delta = \gamma\theta$:

$$\dfrac{\dfrac{\dfrac{}{\vdash P(c,f(c))} \quad \dfrac{}{P(c,f(c)) \vdash Q(c,g(c))}}{\vdash Q(c,g(c))}\ \mathrm{r}(\mathsf{id}) \qquad \dfrac{}{Q(c,g(c)) \vdash}}{\vdash}\ \mathrm{r}(\mathsf{id})$$

with the (ground) substitution $\theta = \{\, u \mapsto c, v \mapsto f(c), v' \mapsto g(c) \,\}$.

The proof projection scheme $\psi^{\mathcal{SK}}(C_1)$ of $\psi^{\mathcal{SK}}$ corresponding to the clause C_1 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\boldsymbol{P(c,v)} \vdash P(c,v)}}{\vdash P(c,v), \neg P(c,v)}\ \neg:\mathrm{r}}{\boldsymbol{\neg Q(c,v)} \vdash P(c,v), \boldsymbol{\neg P(c,v)}}\ \mathrm{w}:\mathrm{l}}{\vdash P(c,v), \boldsymbol{\neg Q(c,v) \supset \neg P(c,v)}}\ \supset:\mathrm{r}}{\vdash P(c,v), (\exists y)(\neg Q(c,y) \supset \neg P(c,y))}\ \exists:\mathrm{r}$$

and let $\chi_1 = \psi^{\mathcal{SK}}(C_1)\theta$ be a proof projection instance.

The proof projection scheme $\psi^{\mathcal{SK}}(C_2)$ of $\psi^{\mathcal{SK}}$ corresponding to the clause C_2 is:

$$
\dfrac{\dfrac{\dfrac{}{P(u,f(u)) \vdash \boldsymbol{P(u,f(u))}} \quad \dfrac{}{\boldsymbol{Q(u,g(u))} \vdash Q(u,g(u))}}{\boldsymbol{P(u,f(u)) \supset Q(u,g(u))}, P(u,f(u)) \vdash Q(u,g(u))}\ \supset: \mathrm{l}}{(\forall x)(P(x,f(x)) \supset Q(x,g(x))), P(u,f(u)) \vdash Q(u,g(u))}\ \forall: \mathrm{l}
$$

and let $\chi_2 = \psi^{\mathcal{SK}}(C_2)\theta$ be a proof projection instance.

The proof projection scheme $\psi^{\mathcal{SK}}(C_3)$ of $\psi^{\mathcal{SK}}$ corresponding to the clause C_3 is:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Q(c,v) \vdash \boldsymbol{Q(c,v)}}}{\neg Q(c,v), Q(c,v) \vdash}\ \neg: \mathrm{l}}{\boldsymbol{\neg Q(c,v)}, Q(c,v) \vdash \boldsymbol{\neg P(c,v)}}\ \mathrm{w}: \mathrm{r}}{Q(c,v) \vdash \boldsymbol{\neg Q(c,v) \supset \neg P(c,v)}}\ \supset: \mathrm{r}}{Q(c,v) \vdash (\exists y)(\neg Q(c,y) \supset \neg P(c,y))}\ \exists: \mathrm{r}
$$

and let $\chi_3 = \psi^{\mathcal{SK}}(C_3)(\{\, v \mapsto v' \,\}\theta)$ be a proof projection instance.

Finally the proof projection instances can be composed to a cut-free proof of $\psi^{\mathcal{SK}}$, i.e. a proof of $\psi^{\mathcal{SK}}$ containing only atomic cuts, using its resolution refutation δ as a skeleton.

$$
\dfrac{\dfrac{\dfrac{\dfrac{(\chi_1)}{\vdash P(c,f(c)), Y}}{\vdash Y, P(c,f(c))}\ \pi((1\,2)): \mathrm{r} \quad \dfrac{\dfrac{(\chi_2)}{X, P(c,f(c)) \vdash Q(c,g(c))}}{P(c,f(c)), X \vdash Q(c,g(c))}\ \pi((1\,2)): \mathrm{l}}{X \vdash Y, Q(c,g(c))}\ \mathrm{cut} \quad \dfrac{(\chi_3)}{Q(c,g(c)) \vdash Y}}{\dfrac{X \vdash Y, Y}{X \vdash Y}\ \mathrm{c}(2): \mathrm{r}}\ \mathrm{cut}
$$

where $X = (\forall x)(P(x,f(x)) \supset Q(x,g(x)))$ and $Y = (\exists y)(\neg Q(c,y) \supset \neg P(c,y))$.

Last but not least, an interesting though simple example showing the behavior of the method CERES in case of an extended axiom set. Let us assume that the next example uses the standard axiom set $\mathcal{A}_{\mathbf{LK}}$ of **LK** allowing arbitrary context in each axiom. This extended axiom set in conjunction with skolemization leads to an increased effort or overhead of some parts of the cut-elimination transformation.

Example 4.3. Let ψ' be the proof

$$
\dfrac{\psi'_l \quad \psi'_r}{(\forall x)((\forall y)P(x,y) \supset (\exists y)Q(x,y)) \vdash (\forall x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y))}\ \mathrm{cut}
$$

with the proof parts ψ_l':

$$
\begin{prooftree}
\AxiomC{$\boldsymbol{P(u,v)} \vdash P(u,v), Q(u,v)$}
\RightLabel{$\neg:\mathrm{r}$}
\UnaryInfC{$\vdash P(u,v), \boldsymbol{Q(u,v)}, \boldsymbol{\neg P(u,v)}$}
\RightLabel{$\pi((2\,3)):\mathrm{r}$}
\UnaryInfC{$\vdash P(u,v), \boldsymbol{\neg P(u,v)}, \boldsymbol{Q(u,v)}$}
\RightLabel{$\vee:\mathrm{r}$}
\UnaryInfC{$\vdash P(u,v), \boldsymbol{\neg P(u,v) \vee Q(u,v)}$}
\RightLabel{$\exists:\mathrm{r}$}
\UnaryInfC{$\vdash \boldsymbol{P(u,v)}, \boldsymbol{(\exists y)(\neg P(u,y) \vee Q(u,y))}$}
\RightLabel{$\pi((1\,2)):\mathrm{r}$}
\UnaryInfC{$\vdash (\exists y)(\neg P(u,y) \vee Q(u,y)), \boldsymbol{P(u,v)}$}
\RightLabel{$\forall:\mathrm{r}$}
\UnaryInfC{$\vdash (\exists y)(\neg P(u,y) \vee Q(u,y)), \boldsymbol{(\forall y)P(u,y)}$}
\AxiomC{$\boldsymbol{P(u,v)}, Q(u,v) \vdash Q(u,v)$}
\RightLabel{$\neg:\mathrm{r}$}
\UnaryInfC{$Q(u,v) \vdash \boldsymbol{Q(u,v)}, \boldsymbol{\neg P(u,v)}$}
\RightLabel{$\pi((1\,2)):\mathrm{r}$}
\UnaryInfC{$Q(u,v) \vdash \boldsymbol{\neg P(u,v)}, \boldsymbol{Q(u,v)}$}
\RightLabel{$\vee:\mathrm{r}$}
\UnaryInfC{$Q(u,v) \vdash \boldsymbol{\neg P(u,v) \vee Q(u,v)}$}
\RightLabel{$\exists:\mathrm{r}$}
\UnaryInfC{$\boldsymbol{Q(u,v)} \vdash (\exists y)(\neg P(u,y) \vee Q(u,y))$}
\RightLabel{$\exists:\mathrm{l}$}
\UnaryInfC{$\boldsymbol{(\exists y)Q(u,y)} \vdash (\exists y)(\neg P(u,y) \vee Q(u,y))$}
\RightLabel{$\supset:\mathrm{l}$}
\BinaryInfC{$(\forall y)P(u,y) \supset (\exists y)Q(u,y) \vdash \boldsymbol{(\exists y)(\neg P(u,y) \vee Q(u,y))}, \boldsymbol{(\exists y)(\neg P(u,y) \vee Q(u,y))}$}
\RightLabel{$c(2):\mathrm{r}$}
\UnaryInfC{$\boldsymbol{(\forall y)P(u,y) \supset (\exists y)Q(u,y)} \vdash (\exists y)(\neg P(u,y) \vee Q(u,y))$}
\RightLabel{$\forall:\mathrm{l}$}
\UnaryInfC{$(\forall x)((\forall y)P(x,y) \supset (\exists y)Q(x,y)) \vdash \boldsymbol{(\exists y)(\neg P(u,y) \vee Q(u,y))}$}
\RightLabel{$\forall:\mathrm{r}$}
\UnaryInfC{$(\forall x)((\forall y)P(x,y) \supset (\exists y)Q(x,y)) \vdash \boldsymbol{(\forall x)(\exists y)(\neg P(x,y) \vee Q(x,y))}$}
\end{prooftree}
$$

and ψ_r':

$$
\begin{prooftree}
\AxiomC{$P(u,v) \vdash \boldsymbol{P(u,v)}$}
\RightLabel{$\neg:\mathrm{l}$}
\UnaryInfC{$\boldsymbol{\neg P(u,v)}, \boldsymbol{P(u,v)} \vdash$}
\RightLabel{$\pi((1\,2)):\mathrm{l}$}
\UnaryInfC{$\boldsymbol{P(u,v)}, \neg P(u,v) \vdash$}
\RightLabel{$\neg:\mathrm{r}$}
\UnaryInfC{$\boldsymbol{\neg P(u,v)} \vdash \neg P(u,v)$}
\AxiomC{$Q(u,v) \vdash \boldsymbol{Q(u,v)}$}
\RightLabel{$\neg:\mathrm{l}$}
\UnaryInfC{$\boldsymbol{\neg Q(u,v)}, \boldsymbol{Q(u,v)} \vdash$}
\RightLabel{$\pi((1\,2)):\mathrm{l}$}
\UnaryInfC{$\boldsymbol{Q(u,v)}, \neg Q(u,v) \vdash$}
\RightLabel{$\vee:\mathrm{l}$}
\BinaryInfC{$\boldsymbol{\neg P(u,v) \vee Q(u,v)}, \boldsymbol{\neg Q(u,v)} \vdash \neg P(u,v)$}
\RightLabel{$\pi((1\,2)):\mathrm{l}$}
\UnaryInfC{$\boldsymbol{\neg Q(u,v)}, \neg P(u,v) \vee Q(u,v) \vdash \boldsymbol{\neg P(u,v)}$}
\RightLabel{$\supset:\mathrm{r}$}
\UnaryInfC{$\neg P(u,v) \vee Q(u,v) \vdash \boldsymbol{\neg Q(u,v) \supset \neg P(u,v)}$}
\RightLabel{$\exists:\mathrm{r}$}
\UnaryInfC{$\boldsymbol{\neg P(u,v) \vee Q(u,v)} \vdash (\exists y)(\neg Q(u,y) \supset \neg P(u,y))$}
\RightLabel{$\exists:\mathrm{l}$}
\UnaryInfC{$\boldsymbol{(\exists y)(\neg P(u,y) \vee Q(u,y))} \vdash (\exists y)(\neg Q(u,y) \supset \neg P(u,y))$}
\RightLabel{$\forall:\mathrm{l}$}
\UnaryInfC{$(\forall x)(\exists y)(\neg P(x,y) \vee Q(x,y)) \vdash \boldsymbol{(\exists y)(\neg Q(u,y) \supset \neg P(u,y))}$}
\RightLabel{$\forall:\mathrm{r}$}
\UnaryInfC{$\boldsymbol{(\forall x)(\exists y)(\neg P(x,y) \vee Q(x,y))} \vdash (\forall x)(\exists y)(\neg Q(x,y) \supset \neg P(x,y))$}
\end{prooftree}
$$

Again, the proof ψ' has to be skolemized, yielding the proof $\psi'^{\mathcal{SK}}$:

$$
\frac{\psi'^{\mathcal{SK}}_l \qquad \psi'^{\mathcal{SK}}_r}{(\forall x)(P(x,f(x)) \supset Q(x,g(x))) \vdash (\exists y)(\neg Q(c,y) \supset \neg P(c,y))}\ \mathrm{cut}
$$

with the proof parts $\psi'^{\mathcal{SK}}_l$:

$$
\begin{array}{ll}
P(u,f(u)) \vdash P(u,f(u)), Q(u,f(u)) & \\
\vdash P(u,f(u)), Q(u,f(u)), \neg P(u,f(u)) & \neg : \mathrm{r} \\
\vdash P(u,f(u)), \neg P(u,f(u)), Q(u,f(u)) & \pi((2\,3)) : \mathrm{r} \\
\vdash P(u,f(u)), \neg P(u,f(u)) \vee Q(u,f(u)) & \vee : \mathrm{r} \\
\vdash P(u,f(u)), (\exists y)(\neg P(u,y) \vee Q(u,y)) & \exists : \mathrm{r} \\
\vdash (\exists y)(\neg P(u,y) \vee Q(u,y)), P(u,f(u)) & \pi((1\,2)) : \mathrm{r}
\end{array}
\qquad
\begin{array}{ll}
P(u,g(u)), Q(u,g(u)) \vdash Q(u,g(u)) & \\
Q(u,g(u)) \vdash Q(u,g(u)), \neg P(u,g(u)) & \neg : \mathrm{r} \\
Q(u,g(u)) \vdash \neg P(u,g(u)), Q(u,g(u)) & \pi((1\,2)) : \mathrm{r} \\
Q(u,g(u)) \vdash \neg P(u,g(u)) \vee Q(u,g(u)) & \vee : \mathrm{r} \\
Q(u,g(u)) \vdash (\exists y)(\neg P(u,y) \vee Q(u,y)) & \exists : \mathrm{r}
\end{array}
$$

$$
\begin{array}{ll}
P(u,f(u)) \supset Q(u,g(u)) \vdash (\exists y)(\neg P(u,y) \vee Q(u,y)), (\exists y)(\neg P(u,y) \vee Q(u,y)) & \supset : \mathrm{l} \\
P(u,f(u)) \supset Q(u,g(u)) \vdash (\exists y)(\neg P(u,y) \vee Q(u,y)) & \mathrm{c}(2) : \mathrm{r} \\
(\forall x)(P(x,f(x)) \supset Q(x,g(x))) \vdash (\exists y)(\neg P(u,y) \vee Q(u,y)) & \forall : \mathrm{l} \\
(\forall x)(P(x,f(x)) \supset Q(x,g(x))) \vdash (\forall x)(\exists y)(\neg P(x,y) \vee Q(x,y)) & \forall : \mathrm{r}
\end{array}
$$

and ${\psi'}^{\mathcal{SK}}_r$:

$$
\begin{array}{ll}
P(c,v) \vdash P(c,v) & \\
\neg P(c,v), P(c,v) \vdash & \neg : \mathrm{l} \\
P(c,v), \neg P(c,v) \vdash & \pi((1\,2)) : \mathrm{l} \\
\neg P(c,v) \vdash \neg P(c,v) & \neg : \mathrm{r}
\end{array}
\qquad
\begin{array}{ll}
Q(c,v) \vdash Q(c,v) & \\
\neg Q(c,v), Q(c,v) \vdash & \neg : \mathrm{l} \\
Q(c,v), \neg Q(c,v) \vdash & \pi((1\,2)) : \mathrm{l}
\end{array}
$$

$$
\begin{array}{ll}
\neg P(c,v) \vee Q(c,v), \neg Q(c,v) \vdash \neg P(c,v) & \vee : \mathrm{l} \\
\neg Q(c,v), \neg P(c,v) \vee Q(c,v) \vdash \neg P(c,v) & \pi((1\,2)) : \mathrm{l} \\
\neg P(c,v) \vee Q(c,v) \vdash \neg Q(c,v) \supset \neg P(c,v) & \supset : \mathrm{r} \\
\neg P(c,v) \vee Q(c,v) \vdash (\exists y)(\neg Q(c,y) \supset \neg P(c,y)) & \exists : \mathrm{r} \\
(\exists y)(\neg P(c,y) \vee Q(c,y)) \vdash (\exists y)(\neg Q(c,y) \supset \neg P(c,y)) & \exists : \mathrm{l} \\
(\forall x)(\exists y)(\neg P(x,y) \vee Q(x,y)) \vdash (\exists y)(\neg Q(c,y) \supset \neg P(c,y)) & \forall : \mathrm{l}
\end{array}
$$

Construction of the characteristic clause term of ${\psi'}^{\mathcal{SK}}$ now yields

$$
\begin{aligned}
\Theta({\psi'}^{\mathcal{SK}}) = ((\{\, P(u,f(u)) \vdash Q(u,f(u)) \,\} \otimes \{\, P(u,g(u)) \vdash Q(u,g(u)) \,\}) \oplus \\
(\{\, \vdash P(c,v) \,\} \oplus \{\, Q(c,v) \vdash \,\}))
\end{aligned}
$$

and computation of the associated characteristic clause set results in clauses containing more literals than in the previous example, which are additionally — thanks to skolemization — not redundant.

$$
\begin{aligned}
\mathrm{CL}({\psi'}^{\mathcal{SK}}) = \{\, & \vdash P(c,v), && (C_1) \\
& P(u,f(u)), P(u,g(u)) \vdash Q(u,f(u)), Q(u,g(u)), && (C_2) \\
& Q(c,v) \vdash \}. && (C_3)
\end{aligned}
$$

Of course, we also obtain a more complex resolution refutation γ of $\mathrm{CL}(\psi'^{\mathcal{SK}})$:

$$\dfrac{\dfrac{\overset{(\gamma')}{\vdash Q(c,f(c)),Q(c,g(c))} \qquad \dfrac{}{Q(c,v')\vdash}}{\vdash Q(c,f(c))}\;\mathrm{r}(\theta) \qquad \dfrac{}{Q(c,v)\vdash}}{\vdash}\;\mathrm{r}(\theta)$$

with $\gamma' =$

$$\dfrac{\dfrac{}{\vdash P(c,v')} \qquad \dfrac{\dfrac{}{\vdash P(c,v)} \qquad \dfrac{}{P(u,f(u)),P(u,g(u))\vdash Q(u,f(u)),Q(u,g(u))}}{P(u,g(u))\vdash Q(c,f(c)),Q(c,g(c))}\;\mathrm{r}(\theta)}{\vdash Q(c,f(c)),Q(c,g(c))}\;\mathrm{r}(\theta)$$

and a corresponding (ground) refutation δ of γ, i.e. $\delta = \gamma\theta$:

$$\dfrac{\dfrac{\overset{(\delta')}{\vdash Q(c,f(c)),Q(c,g(c))} \qquad \dfrac{}{Q(c,g(c))\vdash}}{\vdash Q(c,f(c))}\;\mathrm{r}(\mathsf{id}) \qquad \dfrac{}{Q(c,f(c))\vdash}}{\vdash}\;\mathrm{r}(\mathsf{id})$$

with $\delta' =$

$$\dfrac{\dfrac{}{\vdash P(c,g(c))} \qquad \dfrac{\dfrac{}{\vdash P(c,f(c))} \qquad \dfrac{}{P(c,f(c)),P(c,g(c))\vdash Q(c,f(c)),Q(c,g(c))}}{P(c,g(c))\vdash Q(c,f(c)),Q(c,g(c))}\;\mathrm{r}(\mathsf{id})}{\vdash Q(c,f(c)),Q(c,g(c))}\;\mathrm{r}(\mathsf{id})$$

and with the (ground) substitution $\theta = \{\, u \mapsto c, v \mapsto f(c), v' \mapsto g(c)\,\}$.

The proof projection scheme $\psi'^{\mathcal{SK}}(C_1)$ of $\psi'^{\mathcal{SK}}$ corresponding to the clause C_1 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\boldsymbol{P(c,v)}\vdash P(c,v)}}{\vdash P(c,v),\underline{\neg P(c,v)}}\;\neg:\mathrm{r}}{\underline{\neg\boldsymbol{Q(c,v)}}\vdash P(c,v),\neg\boldsymbol{P(c,v)}}\;\mathrm{w}:\mathrm{l}}{\vdash P(c,v),\underline{\neg\boldsymbol{Q(c,v)}\supset\neg\boldsymbol{P(c,v)}}}\;\supset:\mathrm{r}}{\vdash P(c,v),\underline{(\exists y)(\neg Q(c,y)\supset\neg P(c,y))}}\;\exists:\mathrm{r}$$

and let $\chi_1 = \psi'^{\mathcal{SK}}(C_1)\theta$ and $\chi_1' = \psi'^{\mathcal{SK}}(C_1)(\{\, v \mapsto v'\,\}\theta)$ be proof projection instances.

The proof projection scheme $\psi'^{\mathcal{SK}}(C_2)$ of $\psi'^{\mathcal{SK}}$ corresponding to the clause C_2 is:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{P(u,f(u))\vdash \boldsymbol{P(u,f(u))},\boldsymbol{Q(u,f(u))}}}{P(u,f(u))\vdash \underline{Q(u,f(u))},\underline{\boldsymbol{P(u,f(u))}}}\;\pi((1\,2)):\mathrm{r} \qquad \dfrac{\dfrac{}{\boldsymbol{P(u,g(u))},\boldsymbol{Q(u,g(u))}\vdash Q(u,g(u))}}{\underline{\boldsymbol{Q(u,g(u))}},\underline{P(u,g(u))}\vdash Q(u,g(u))}\;\pi((1\,2)):\mathrm{l}}{\underline{\boldsymbol{P(u,f(u))\supset Q(u,g(u))}},P(u,f(u)),P(u,g(u))\vdash Q(u,f(u)),Q(u,g(u))}\;\supset:\mathrm{l}}{\underline{(\forall x)(P(x,f(x))\supset Q(x,g(x)))},P(u,f(u)),P(u,g(u))\vdash Q(u,f(u)),Q(u,g(u))}\;\forall:\mathrm{l}$$

and let $\chi_2 = \psi'^{\mathcal{SK}}(C_2)\theta$ be a proof projection instance.

The proof projection scheme $\psi'^{\mathcal{SK}}(C_3)$ of $\psi'^{\mathcal{SK}}$ corresponding to the clause C_3 is:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\overline{Q(c,v) \vdash \boldsymbol{Q(c,v)}}}{\neg Q(c,v), Q(c,v) \vdash}\ \neg : \mathrm{l}}{\boldsymbol{\neg Q(c,v)}, Q(c,v) \vdash \boldsymbol{\neg P(c,v)}}\ \mathrm{w} : \mathrm{r}}{Q(c,v) \vdash \boldsymbol{\neg Q(c,v) \supset \neg P(c,v)}}\ \supset : \mathrm{r}}{Q(c,v) \vdash (\exists y)(\neg Q(c,y) \supset \neg P(c,y))}\ \exists : \mathrm{r}
$$

and let $\chi_3 = \psi'^{\mathcal{SK}}(C_3)\theta$ and $\chi'_3 = \psi'^{\mathcal{SK}}(C_3)(\{\, v \mapsto v' \,\}\theta)$ be proof projection instances.

Finally the proof projection instances can be composed to a cut-free proof of $\psi'^{\mathcal{SK}}$, i.e. a proof of $\psi'^{\mathcal{SK}}$ containing only atomic cuts, using the resolution refutation δ as a skeleton.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overset{(\chi'_1)}{\vdash P(c,g(c)), Y}}{\vdash Y, P(c,g(c))}\ \pi((1\,2)) : \mathrm{r} \qquad \chi}{X \vdash Y, Y, Q(c,f(c)), Q(c,g(c))}\ \mathrm{cut} \qquad \overset{(\chi'_3)}{Q(c,g(c)) \vdash Y}}{X \vdash Y, Y, Q(c,f(c)), Y}\ \mathrm{cut}}{X \vdash Y, Y, Y, Q(c,f(c))}\ \pi((3\,4)) : \mathrm{r} \qquad \overset{(\chi_3)}{Q(c,f(c)) \vdash Y}}{X \vdash Y, Y, Y, Y}\ \mathrm{cut}}{X \vdash Y}\ \mathrm{c}(4) : \mathrm{r}
$$

with $\chi =$

$$
\dfrac{\dfrac{\overset{(\chi_1)}{\vdash P(c,f(c)), Y}}{\vdash Y, P(c,f(c))}\ \pi((1\,2)) : \mathrm{r} \qquad \dfrac{\overset{(\chi_2)}{X, P(c,f(c)), P(c,g(c)) \vdash Q(c,f(c)), Q(c,g(c))}}{P(c,f(c)), P(c,g(c)), X \vdash Q(c,f(c)), Q(c,g(c))}\ \pi((1\,3\,2)) : \mathrm{l}}{P(c,g(c)), X \vdash Y, Q(c,f(c)), Q(c,g(c))}\ \mathrm{cut}
$$

where $X = (\forall x)(P(x,f(x)) \supset Q(x,g(x)))$ and $Y = (\exists y)(\neg Q(c,y) \supset \neg P(c,y))$.

Concluding, one could say: "Keep it simple!" Indeed, simpler axioms yield a simpler characteristic clause set hence also a simpler resolution refutation and even simpler proof projection schemes and instances. Therefore *optimal* as far as CERES is concerned would be a reduction of the number of formulas occurring in every axiom S by means of additional weakenings such that S is minimal w.r.t. the number of ancestors of cut-formulas occurring within S (and of course still $S \in \mathcal{A}$ holds).

4.3 Comparison of Cut-Elimination Methods

Even though the cut-elimination methods presented in chapter 3 and CERES operate differently some convincing comparisons can be made.

- *Extensibility*: Without doubt extending CERES to support additional rules or calculi turns out to be very simple and is feasible in most cases without increasing the complexity and corrupt the clarity of the method. An example of an extension of CERES by equality can be found in chapter 5 of this thesis, even an extension to many valued logics is possible (see Baaz and Leitsch 2004). Especially unary rule applications are more or less insignificant for the method, hence even powerful unary rules like negation normal form transformations could be supported. On the contrary the rules of the cut-reduction systems are not very flexible. Slight modifications of the calculus may demand substantial changes.

- *Intermediate Results*: After each single step of a reduction rule application in Gentzen's or Schütte-Tait's method we obtain an intact and valid **LK**-proof of the same theorem. Since CERES eliminates all cuts at once this is not possible, only at the very end, after back-transformation of the refutation into **LK** and concatenation with instances of the proof projection schemes a proof of the original theorem is constructed. A fact that makes it more complicated to eliminate just a certain cut rule inference from an **LK**-proof containing several non-atomic cuts with CERES. Whereas for the cut-reduction systems presented each cut is algorithmically (more or less) equally hard to eliminate.
- *Speed-up Results*: CERES has been shown (e.g. in Baaz and Leitsch 2006) to yield non-elementary speed-ups (the complexity of cut-elimination itself) with respect to other cut-elimination methods based on reduction systems (as in definition 3.1) like Gentzen's or Schütte-Tait's cut-elimination methods. On the other hand these cut-reduction systems do not have non-elementary speed-ups with respect to CERES, more precisely the expense of CERES is just exponentially bounded by them. An important point especially as far as implementing a cut-elimination system is concerned — be referred to chapter 7 for more information on this topic.

Remark 4.3. Regarding regularity. The rule based cut-reduction systems need the initial **LK**-proof to be regular (more accurate a stronger form of regularity requiring all eigenvariables to be unique within the proof — fresh variables). This is due to the shifting of rules above quantifier introduction rules and multiplication of entire proof parts (e.g. at contraction rules).

One could expect that regularity of the proof is not necessary in case of CERES. But this turns out to be a fallacy since skolemization supposes exactly the same kind of regularity as for the cut-reduction systems. Hence in general whenever skolemization has to be performed also strong regularity has to be assured.

Our last example shows a regular proof which cannot be skolemized e.g. with Andrews' method (see Andrews 1971).

Example 4.4.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{P(u) \vdash P(u)}}{P(u) \vdash P(u), Q(v,u)}\ \mathrm{w:r}}{P(u) \vdash P(u), (\forall z)Q(z,u)}\ \forall:\mathrm{r}}{\vdash P(u), P(u) \supset (\forall z)Q(z,u)}\ \supset:\mathrm{r}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{P(u) \vdash P(u)}}{Q(v,u), P(u) \vdash P(u)}\ \mathrm{w:l}}{Q(v,u) \land P(u) \vdash P(u)}\ \land:\mathrm{l} \qquad \dfrac{\dfrac{}{Q(v,u) \vdash Q(v,u)}}{(\forall z)Q(z,u) \vdash Q(v,u)}\ \forall:\mathrm{l}}{P(u) \supset (\forall z)Q(z,u), Q(v,u) \land P(u) \vdash Q(v,u)}\ \supset:\mathrm{l}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdash P(u), P(u) \supset (\forall z)Q(z,u) \qquad P(u) \supset (\forall z)Q(z,u), Q(v,u) \land P(u) \vdash Q(v,u)}{Q(v,u) \land P(u) \vdash P(u), Q(v,u)}\ \mathrm{cut}}{Q(v,u) \land P(u) \vdash P(u) \lor Q(v,u)}\ \lor:\mathrm{r}}{(\forall y)(Q(y,u) \land P(u)) \vdash P(u) \lor Q(v,u)}\ \forall:\mathrm{l}}{(\forall x)(\forall y)(Q(y,x) \land P(x)) \vdash P(u) \lor Q(v,u)}\ \forall:\mathrm{l}}{(\forall x)(\forall y)(Q(y,x) \land P(x)) \vdash (\forall x)(P(x) \lor Q(v,x))}\ \forall:\mathrm{r}}{(\forall x)(\forall y)(Q(y,x) \land P(x)) \vdash (\exists y)(\forall x)(P(x) \lor Q(y,x))}\ \exists:\mathrm{r}$$

During skolemization it would be necessary to apply the substitution $\{ u \mapsto f(v) \}$ along the component path of the eigenvariable A, for a Skolem function f. But this would prevent the introduction of the universal quantifier on the right by violating the eigenvariable condition.

Note that this proof is regular, since all eigenvariables, i.e. u and v, are distinct — even different variables for free and bound variable occurrences are used.

Hence regularity is not sufficient. Only the use of fresh (new) variables for each eigenvariable solves the dilemma.

Further note that this **LK**-proof does not use any "special" properties of the multiplicative calculus, thus the problem remains even in a pure additive calculus.

The root of the problem seems to be the cut rule inference where an "eigenvariable beholder" (strong quantifier) disappears and allows the variable to occur below the cut.

CHAPTER 5

Extension of CERES to Equality

There exist various substantially different approaches how to integrate equality into **LK**. Some of them are solely based on axiomatization, i.e. adding equality axioms to the existing axioms, without any extension of the rules (e.g. see Takeuti (1987) for more details). This kind of equality integration would of course be possible in CERES without any changes (note that we allow arbitrary atomic sequents as axioms), but has at least two major drawbacks. On the one hand it is mathematically a very unnatural way of using equality within a proof, concerning formalization and interpretation of proofs. On the other hand the computational expense is much higher, e.g. by propagating the used axioms throughout the entire proof to the antecedent of the end-sequent. In addition it would not be possible to use the existing paradigm of paramodulation in combination with CERES which is especially designed to handle equality reasoning within resolution, which is a special design goal (also see Leitsch and Richter (2005)).

This disadvantages are overcome by introducing the theory of equality to **LK** by means of rules. Some might argue that this is a trade-off against the loss of the subformula property and depending on the specific rules the introduction of implicit cuts. Loosing the subformula property is not avoidable if you intend to use equality in a mathematically natural and intuitive way. The argument of implicit cuts is immediately dismantled in CERES as we only intend to eliminate non-atomic cuts (since we are not using axioms of the form $A \vdash A$ — which of course also applies to the approach by axiomatization).

5.1 The Extension

Again there are many different variants how to extend **LK** by equality with help of rules (e.g. see Schwichtenberg and Troelstra (2000)). We will now define the best suitable version for our needs (similar to Degtyarev and Voronkov (2001)).

Definition 5.1 (LKe). **LKe** is the calculus obtained from **LK** by adding the following equality introduction rules to the existing rules of **LK** (see definition 2.8):

$$\frac{\Gamma \vdash \Delta, \boldsymbol{s = t} \quad \boldsymbol{A[s]_\Xi}, \Pi \vdash \Lambda}{\underline{A[t]_\Xi}, \Gamma, \Pi \vdash \Delta, \Lambda} = (\Xi) : \mathrm{l} \qquad \frac{\Gamma \vdash \Delta, \boldsymbol{s = t} \quad \Pi \vdash \Lambda, \boldsymbol{A[s]_\Xi}}{\Gamma, \Pi \vdash \Delta, \Lambda, \underline{A[t]_\Xi}} = (\Xi) : \mathrm{r}$$

where s and t are arbitrary terms and Ξ is a set of positions.

For practical reasons we will additionally use the following rules in **LKe**:

$$\frac{\Gamma \vdash \Delta, \boldsymbol{t = s} \quad \boldsymbol{A[s]_\Xi}, \Pi \vdash \Lambda}{\underline{A[t]_\Xi}, \Gamma, \Pi \vdash \Delta, \Lambda} =' (\Xi) : \mathrm{l} \qquad \frac{\Gamma \vdash \Delta, \boldsymbol{t = s} \quad \Pi \vdash \Lambda, \boldsymbol{A[s]_\Xi}}{\Gamma, \Pi \vdash \Delta, \Lambda, \underline{A[t]_\Xi}} =' (\Xi) : \mathrm{r}$$

Note that this rules could also be derived from the ones above using an additional equality rule inference. Hence, they are just shortcuts for

$$\frac{\dfrac{\Gamma \vdash \Delta, \boldsymbol{t = s} \quad \overline{\vdash \boldsymbol{t = t}}}{\Gamma \vdash \Delta, \underline{\boldsymbol{s = t}}} = ((1)) : \mathrm{r} \quad \boldsymbol{A[s]_\Xi}, \Pi \vdash \Lambda}{\underline{A[t]_\Xi}, \Gamma, \Pi \vdash \Delta, \Lambda} = (\Xi) : \mathrm{l}$$

respectively

$$\frac{\dfrac{\Gamma \vdash \Delta, \boldsymbol{t = s} \quad \overline{\vdash \boldsymbol{t = t}}}{\Gamma \vdash \Delta, \underline{\boldsymbol{s = t}}} = ((1)) : \mathrm{r} \quad \Pi \vdash \Lambda, \boldsymbol{A[s]_\Xi}}{\Gamma, \Pi \vdash \Delta, \Lambda, \underline{A[t]_\Xi}} = (\Xi) : \mathrm{r}$$

As you will notice the definitions of the additional equality inference rules within **LKe** look very familiar. They exactly correspond to the paramodulation rules defined for the resolution calculus (recall definition 2.17) which enables CERESe to benefit from the extended abilities of **PR**-resolution in one of its crucial steps - the generation of the resolution refutation skeleton.

Definition 5.2 (LKe-proof). An **LKe**-proof is a proof where $\mathcal{A}$ consists of atomic sequents including the standard axioms $\mathcal{A}_{\mathbf{LK}}$ of **LK** and the axiom set of reflexivity, i.e.

$$\{\vdash t = t \mid t \text{ a term}\},$$

and the inference rules applied are those of **LKe**.

The extension by equality is also possible for **LKp**.

Definition 5.3 (LKep). **LKep** is the calculus obtained from **LKe** by again adding the semantic cut (see definition 2.12) rule to the existing rules of **LKe**.

Definition 5.4 (LKep-proof). An **LKep**-proof is a proof where $\mathcal{A}$ consists of atomic sequents including the standard axioms $\mathcal{A}_{\mathbf{LK}}$ of **LK** and the reflexivity axioms and the inference rules applied are those of **LKep**.

For the extension of CERES to CERESe neither a redefinition of the characteristic clause term nor of the characteristic clause set and its computation is necessary; the equality introduction rules are treated as "ordinary" binary rules. The proof projection schemes are built in exactly the same manner, hence also the necessity of skolemization a priori and (eventually) de-skolemization a posteriori remains.

Definition 5.5 (E-unsatisfiable). A set of clauses $\mathcal{C}$ is E-unsatisfiable if $\mathcal{C}$ has no equational model (i.e. a model where $=$ is interpreted as equality).

Therefore the only thing that remains to be shown is the following theorem.

Theorem 5.1. Let φ be an **LKe**-proof and Ω the set of all cut-formula occurrences in φ. Then $\mathrm{CL}(\varphi)$ is E-unsatisfiable, i.e. there exists a **PR**-refutation with resolution and paramodulation of $\mathrm{CL}(\varphi)$.

Proof. As in Baaz and Leitsch (2000) we show that, from the set $\mathrm{CL}(\varphi.\nu)$ for any proof node ν in φ we can derive $\nu(\Omega)$ (the subsequent at ν consisting just of the ancestors of a cut). As there is no ancestor of a cut in the end-sequent, we obtain an **LKe** derivation of $\vdash$ from $\mathrm{CL}(\varphi)$. The equality introduction rules on the left and on the right behave like any other binary rule in **LK**, and the construction goes through like for **LK**. As **LKe** is sound on equational interpretations and we have derived $\vdash$, $\mathrm{CL}(\varphi)$ must be E-unsatisfiable. □

Corollary 5.1. Let φ be an **LKep**-proof and Ω the set of all cut-formula occurrences in φ. Then $\mathrm{CL}(\varphi)$ is E-unsatisfiable, i.e. there exists a **PR**-refutation with resolution and paramodulation of $\mathrm{CL}(\varphi)$.

Proof. The proof of theorem 5.1 also holds for **LKep**. □

Proposition 5.1. CERESe is a cut-elimination method for **LKe**.

Proof. By theorem 5.1 $\mathrm{CL}(\varphi)$ is E-unsatisfiable. As **PR**-deduction is complete there exists a **PR**-refutation γ of $\mathrm{CL}(\varphi)$. Let γ' be a version of γ after applying the combined most general unifier of γ to itself (γ has to be regular). Then γ' is an **LKe** derivation of $\vdash$ from the axiom set defined by $\mathrm{CL}(\varphi)$. γ' contains only atomic cuts. By inserting the proof projection instances on every leaf of γ' we obtain a proof of the original sequent with only atomic cuts. □

Corollary 5.2. CERESe is a cut-elimination method for **LKep**.

Proof. The proof of proposition 5.1 also holds for **LKep**. □

5.2 Example

Now we will demonstrate the strength of this method on a well-known example from group theory. The proof φ below verifies that a 2-nilpotent group is commutative using the cancellation principle as a lemma. Therefore we need to extend the set of axioms by all instances of the necessary group theoretic axioms:

$$\vdash (u \circ v) \circ w = u \circ (v \circ w), \tag{A}$$

$$\vdash e \circ u = u \qquad \vdash u \circ e = u, \qquad (E_l),\ (E_r)$$

$$\vdash u^{-1} \circ u = e \qquad \vdash u \circ u^{-1} = e, \qquad (I_l),\ (I_r)$$

$$\vdash u \circ u = e, \tag{N_2}$$

where u^{-1} denotes the inverse element of u.

Since the original proof of $\vdash (\forall x)(\forall y)x \circ y = y \circ x$ contains strong quantifiers it has to be skolemized in advance and the resulting cut-free proof de-skolemized afterwards.

We only give the skolemized proof (of the sequent $\vdash a \circ b = b \circ a$ for two individual constant symbols a and b). The proof of $\vdash (\forall x)(\forall y) x \circ y = y \circ x$ can be directly obtained by generalizing a to u and b to v and by afterwards applying $\forall : \mathrm{r}$ twice on $\vdash u \circ v = v \circ u$.

Within this section the following formula abbreviations are used:

$$P\colon \quad (a \circ b) \circ (a \circ b) = (b \circ a) \circ (a \circ b),$$

$$C\colon \quad a \circ b = b \circ a,$$

$$S\colon \quad u \circ w = v \circ w.$$

Then, let the main proof φ be defined as follows.

$\varphi =$

$$\dfrac{\overline{\vdash \boldsymbol{e = e}} \quad \dfrac{\overline{\vdash \boldsymbol{b \circ b = e}} \quad \dfrac{\overline{\vdash \boldsymbol{e \circ b = b}} \quad \dfrac{\overline{\vdash \boldsymbol{a \circ a = e}} \quad \overset{(\varphi')}{\boldsymbol{e = b \circ ((a \circ a) \circ b)} \vdash C}}{\underline{\boldsymbol{e = b \circ (e \circ b)}} \vdash C}\, {=}\,((2,2,1))\ :\mathrm{l}}{\underline{\boldsymbol{e = b \circ b}} \vdash C}\, {=}\,((2,2))\ :\mathrm{l}}{\underline{\boldsymbol{e = e}} \vdash C}\, {=}\,((2))\ :\mathrm{l}}{\vdash a \circ b = b \circ a}\,\mathrm{cut}$$

$\varphi' =$

$$\dfrac{\overline{\vdash \boldsymbol{(a \circ a) \circ b = a \circ (a \circ b)}} \quad \dfrac{\overline{\vdash \boldsymbol{(b \circ a) \circ (a \circ b) = b \circ (a \circ (a \circ b))}} \quad \varphi''}{\underline{\boldsymbol{e = b \circ (a \circ (a \circ b))}} \vdash C}\, {=}\,(0)\ :\mathrm{l}}{\underline{e = b \circ ((a \circ a) \circ b)} \vdash C}\, {=}'\,((2,2))\ :\mathrm{l}$$

$\varphi'' =$

$$\dfrac{\overline{\vdash \boldsymbol{(a \circ b) \circ (a \circ b) = e}} \quad \dfrac{\varphi_c \quad \varphi_p}{\underline{\boldsymbol{(a \circ b) \circ (a \circ b) = (b \circ a) \circ (a \circ b)}} \vdash C}\,\text{p-cut}}{\underline{e = (b \circ a) \circ (a \circ b)} \vdash C}\, {=}\,((1))\ :\mathrm{l}$$

This is the subproof of the cancellation lemma used in φ, $\varphi_c =$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{\vdash \boldsymbol{v \circ e = v}} \quad \dfrac{\overline{\vdash \boldsymbol{u \circ e = u}} \quad \dfrac{\overline{\vdash \boldsymbol{w \circ w^{-1} = e}} \quad \dfrac{\overline{\vdash \boldsymbol{(v \circ w) \circ w^{-1} = v \circ (w \circ w^{-1})}} \quad \varphi'_c}{S \vdash \underline{\boldsymbol{u \circ (w \circ w^{-1}) = v \circ (w \circ w^{-1})}}}\, {=}\,((2))\ :\mathrm{r}}{S \vdash \underline{\boldsymbol{u \circ e = v \circ e}}}\, {=}\,((1,2),(2,2))\ :\mathrm{r}}{S \vdash \underline{\boldsymbol{u = v \circ e}}}\, {=}\,((1))\ :\mathrm{r}}{\boldsymbol{u \circ w = v \circ w} \vdash \underline{\boldsymbol{u = v}}}\, {=}\,((2))\ :\mathrm{r}}{\vdash \underline{\boldsymbol{u \circ w = v \circ w \supset u = v}}}\,\supset :\mathrm{r}}{\vdash \underline{\boldsymbol{(\forall z)(u \circ z = v \circ z \supset u = v)}}}\,\forall :\mathrm{r}}{\dfrac{\vdash \underline{\boldsymbol{(\forall y)(\forall z)(u \circ z = y \circ z \supset u = y)}}}{\vdash \underline{(\forall x)(\forall y)(\forall z)(x \circ z = y \circ z \supset x = y)}}\,\forall :\mathrm{r}}\,\forall :\mathrm{r}$$

$\varphi'_c =$

$$\dfrac{\vdash \boldsymbol{(u \circ w) \circ w^{-1} = u \circ (w \circ w^{-1})} \qquad \dfrac{S \vdash \boldsymbol{u \circ w = v \circ w} \qquad \vdash \boldsymbol{(u \circ w) \circ w^{-1} = (u \circ w) \circ w^{-1}}}{S \vdash \boldsymbol{(u \circ w) \circ w^{-1} = (v \circ w) \circ w^{-1}}} = ((2,1)) : \mathrm{r}}{S \vdash u \circ (w \circ w^{-1}) = (v \circ w) \circ w^{-1}} = ((1)) : \mathrm{r}$$

$\varphi_p =$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P \vdash \boldsymbol{(a \circ b) \circ (a \circ b) = (b \circ a) \circ (a \circ b)} \qquad \boldsymbol{a \circ b = b \circ a} \vdash C}{\boldsymbol{(a \circ b) \circ (a \circ b) = (b \circ a) \circ (a \circ b) \supset a \circ b = b \circ a}, P \vdash C} \supset : \mathrm{l}}{\boldsymbol{(\forall z_1)((a \circ b) \circ (a \circ z_1) = (b \circ a) \circ (a \circ z_1) \supset a \circ b = b \circ a)}, P \vdash C} \forall : \mathrm{l}}{\boldsymbol{(\forall z_0)(\forall z_1)((a \circ b) \circ (z_0 \circ z_1) = (b \circ a) \circ (z_0 \circ z_1) \supset a \circ b = b \circ a)}, P \vdash C} \forall : \mathrm{l}}{\boldsymbol{(\forall y)(\forall z_0)(\forall z_1)((a \circ b) \circ (z_0 \circ z_1) = y \circ (z_0 \circ z_1) \supset a \circ b = y)}, P \vdash C} \forall : \mathrm{l}$$

$$\dfrac{\boldsymbol{(\forall y)(\forall z_0)(\forall z_1)((a \circ b) \circ (z_0 \circ z_1) = y \circ (z_0 \circ z_1) \supset a \circ b = y)}, P \vdash C}{(\forall x)(\forall y)(\forall z_0)(\forall z_1)(x \circ (z_0 \circ z_1) = y \circ (z_0 \circ z_1) \supset x = y), P \vdash C} \forall : \mathrm{l}$$

Hence for the characteristic clause set $\mathrm{CL}(\varphi)$ of φ we obtain

$$\begin{aligned}
\mathrm{CL}(\varphi) = \{\, & a \circ b = b \circ a \vdash, && (C_1)\\
& (a \circ b) \circ (a \circ b) = (b \circ a) \circ (a \circ b) \vdash (a \circ b) \circ (a \circ b) = (b \circ a) \circ (a \circ b), && (C_2)\\
& \vdash (u \circ w) \circ w^{-1} = (u \circ w) \circ w^{-1}, && (C_3)\\
& u \circ w = v \circ w \vdash u \circ w = v \circ w, && (C_4)\\
& \vdash (u \circ w) \circ w^{-1} = u \circ (w \circ w^{-1}), && (C_5)\\
& \vdash (v \circ w) \circ w^{-1} = v \circ (w \circ w^{-1}), && (C_6)\\
& \vdash w \circ w^{-1} = e, && (C_7)\\
& \vdash u \circ e = u, && (C_8)\\
& \vdash v \circ e = v, && (C_9)\\
& \vdash (a \circ b) \circ (a \circ b) = e, && (C_{10})\\
& \vdash (b \circ a) \circ (a \circ b) = b \circ (a \circ (a \circ b)), && (C_{11})\\
& \vdash (a \circ a) \circ b = a \circ (a \circ b), && (C_{12})\\
& \vdash a \circ a = e, && (C_{13})\\
& \vdash e \circ b = b, && (C_{14})\\
& \vdash b \circ b = e, && (C_{15})\\
& \vdash e = e \,\}. && (C_{16})
\end{aligned}$$

Since resolution admits subsumption and deletion of tautologies we obtain a reduced characteristic clause set by omitting the clauses C_2, C_4, C_6, C_9 and C_{16}:

$$\begin{aligned}
\mathrm{CL}(\varphi)' = \{\, & a \circ b = b \circ a \vdash, && (C_1)\\
& \vdash (u \circ w) \circ w^{-1} = (u \circ w) \circ w^{-1}, && (C_3)\\
& \vdash (u \circ w) \circ w^{-1} = u \circ (w \circ w^{-1}), && (C_5)\\
& \vdash w \circ w^{-1} = e, && (C_7)\\
& \vdash u \circ e = u, && (C_8)\\
& \vdash (a \circ b) \circ (a \circ b) = e, && (C_{10})\\
& \vdash (b \circ a) \circ (a \circ b) = b \circ (a \circ (a \circ b)), && (C_{11})\\
& \vdash (a \circ a) \circ b = a \circ (a \circ b), && (C_{12})\\
& \vdash a \circ a = e, && (C_{13})\\
& \vdash e \circ b = b, && (C_{14})\\
& \vdash b \circ b = e \,\}. && (C_{15})
\end{aligned}$$

A **PR**-refutation of $\mathrm{CL}(\varphi)'$ is given by the following derivations.

Derivation of C_{17}:

$$\dfrac{\dfrac{\overset{(C_7)}{\vdash w \circ w^{-1} = e} \quad \overset{(C_8)}{\vdash u \circ e = u}}{\vdash u \circ (w \circ w^{-1}) = u}\ \mathrm{p}'(\mathsf{id}, (1,2)) : \mathrm{r} \qquad \overset{(C_5\sigma_1)}{\vdash (u' \circ w') \circ w'^{-1} = u' \circ (w' \circ w'^{-1})}}{\vdash (u \circ w) \circ w^{-1} = u}\ \mathrm{p}(\sigma_2, (2)) : \mathrm{r} \qquad (C_{17})$$

where $\sigma_1 = \{u \mapsto u', w \mapsto w'\}$ and $\sigma_2 = \{u' \mapsto u, w' \mapsto w\}$.

Derivation of C_{18}:

$$\dfrac{\dfrac{\overset{(C_{13})}{\vdash a \circ a = e} \quad \overset{(C_{14})}{\vdash e \circ b = b}}{\vdash (a \circ a) \circ b = b}\ \mathrm{p}'(\mathsf{id}, (1,1)) : \mathrm{r} \qquad \overset{(C_{12})}{\vdash (a \circ a) \circ b = a \circ (a \circ b)}}{\vdash b = a \circ (a \circ b)}\ \mathrm{p}(\mathsf{id}, (1)) : \mathrm{r} \qquad (C_{18})$$

Derivation of C_{19}:

$$\dfrac{\dfrac{\overset{(C_{18})}{\vdash b = a \circ (a \circ b)} \quad \overset{(C_{15})}{\vdash b \circ b = e}}{\vdash b \circ (a \circ (a \circ b)) = e}\ \mathrm{p}(\mathsf{id}, (1,2)) : \mathrm{r} \qquad \overset{(C_{11})}{\vdash (b \circ a) \circ (a \circ b) = b \circ (a \circ (a \circ b))}}{\vdash (b \circ a) \circ (a \circ b) = e}\ \mathrm{p}(\mathsf{id}, (2)) : \mathrm{r} \qquad (C_{19})$$

Derivation of C_{20}:

$$\dfrac{\overset{(C_{17})}{\vdash (u \circ w) \circ w^{-1} = u} \quad \overset{(C_{17}\sigma_3)}{\vdash (u' \circ w') \circ w'^{-1} = u'}}{\vdash u \circ (w^{-1})^{-1} = u \circ w}\ \mathrm{p}(\sigma_4, (1,1)) : \mathrm{r} \qquad (C_{20})$$

where $\sigma_3 = \{u \mapsto u', w \mapsto w'\}$ and $\sigma_4 = \{u' \mapsto u \circ w, w' \mapsto w^{-1}\}$.

Derivation of C_{21}:

$$\dfrac{\overset{(C_{10})}{\vdash (a \circ b) \circ (a \circ b) = e} \quad \overset{(C_{17})}{\vdash (u \circ w) \circ w^{-1} = u}}{\vdash e \circ (a \circ b)^{-1} = a \circ b}\ \mathrm{p}(\sigma_5, (1,1)) : \mathrm{r} \qquad (C_{21})$$

where $\sigma_5 = \{u \mapsto a \circ b, w \mapsto a \circ b\}$.

Derivation of C_{22}:

$$\dfrac{\overset{(C_{22})}{\vdash e \circ w = w} \quad \overset{(C_{21})}{\vdash e \circ (a \circ b)^{-1} = a \circ b}}{\vdash (a \circ b)^{-1} = a \circ b}\ \mathrm{p}(\sigma_6, (1)) : \mathrm{r} \qquad (C_{22})$$

where $\sigma_6 = \{w \mapsto (a \circ b)^{-1}\}$.

Derivation of C_{23}:

$$\dfrac{\dfrac{}{\vdash e \circ w = w} \quad \dfrac{\overset{(C_{22})}{\vdash (a \circ b)^{-1} = a \circ b} \quad \dfrac{\overset{(C_{19})}{\vdash (b \circ a) \circ (a \circ b) = e} \quad \overset{(C_{17})}{\vdash (u \circ w) \circ w^{-1} = u}}{\vdash e \circ (a \circ b)^{-1} = b \circ a}\ \mathrm{p}(\sigma_7, (1,1)) : \mathrm{r}}{\vdash e \circ (a \circ b) = b \circ a}\ \mathrm{p}(\mathsf{id}, (1,2)) : \mathrm{r}}{\vdash a \circ b = b \circ a}\ \mathrm{p}(\sigma_8, (1)) : \mathrm{r} \qquad (C_{23})$$

where $\sigma_7 = \{u \mapsto b \circ a, w \mapsto a \circ b\}$ and $\sigma_8 = \{w \mapsto a \circ b\}$.

And finally we have a refutation:

$$\dfrac{\overset{(C_{23})}{\vdash a \circ b = b \circ a} \quad \overset{(C_1)}{a \circ b = b \circ a \vdash}}{\vdash}\ \mathrm{r}(\mathsf{id})$$

Let γ be the **PR**-refutation defined above (in form of a tree). By applying a global m.g.u. σ we get δ of γ, i.e. $\delta = \gamma\sigma$, we obtain a derivation of $\vdash$ in **LKe** (and hence also in **LKep**) from instances of $\mathrm{CL}(\varphi)$. There is only one non-trivial proof projection required, namely this to the clause C_1. All other proof projections to the clauses used from $\mathrm{CL}(\varphi)$ are trivial, i.e. the instantiated clauses itself are already the proof projection instances to themselves, simply because there are no inferences in φ operating on non-ancestors of cut-formulas and most of the axioms are of such a simple shape. The proof of $\vdash a \circ b = b \circ a$ with only atomic cuts φ^* is therefore:

$$\dfrac{\overset{(\delta')}{\vdash a \circ b = b \circ a} \quad a \circ b = b \circ a \vdash a \circ b = b \circ a}{\vdash a \circ b = b \circ a}\ \mathrm{cut}$$

where δ' represents the subproof of δ originally yielding the clause C_{23}.

By de-skolemizing φ^* we obtain a proof $\hat{\varphi}$ of $\vdash (\forall x)(\forall y) x \circ y = y \circ x$ with only atomic cuts (and clearly without making use of the cancellation principle).

CHAPTER 6

Equational Theories

Making use of existing well-known or even arbitrary equational theories within each inference step has the following advantages:

- gain of more expressiveness,
- proofs get shorter, what makes them more understandable and allows concentrating on the gist, especially for substantial complex proofs and
- proofs can be performed and realized in a much more natural and mathematical way.

First we present variants of the calculi defined so far using equational theories (as initiated in Leitsch and Richter 2005) whereas an example is given afterwards demonstrating how more complex mathematical proofs may be processed with this concept in combination with CERES (or rather with the corresponding extension CERESe).

6.1 The Concept

Definition 6.1 (equational axiom set). A (possibly infinite) set $\mathcal{E}$ of term equations, i.e.

$$\mathcal{E} = \{\, s_1 = t_1, s_2 = t_2, s_3 = t_3 \ldots \},$$

is called an equational axiom set if it is closed under substitution, i.e. for all $E \in \mathcal{E}$ and for all substitutions σ we have $E\sigma \in \mathcal{E}$.

Definition 6.2 (equational theory). Let $\mathcal{E}$ be an equational axiom set. An equational theory is defined as a congruence relation on $\mathcal{E}$ in the following way:

$$s =_{\mathcal{E}} t \quad \Leftrightarrow \quad \mathcal{E} \models s = t.$$

This definition is extended to formulas by the reflexive and transitive closure of the relation

$$A[s]_{\Xi} =_{\mathcal{E}} A[t]_{\Xi} \quad \text{if} \quad s =_{\mathcal{E}} t.$$

Based on equational theories the presented calculi can be redefined to use the equational theory at every inference. Effected are the logical rules which can now be applied not only to syntactic equivalent formulas but to all auxiliary formulas which are equal modulo the underlying equational theory $\mathcal{E}$.

Definition 6.3 ($\mathbf{LK}^{\mathcal{E}}$, $\mathbf{LKp}^{\mathcal{E}}$). $\mathbf{LK}^{\mathcal{E}}$ respectively $\mathbf{LKp}^{\mathcal{E}}$ are the calculi obtained by extending the definitions of **LK** (see definition 2.8) respectively **LKp** (see definition 2.12) in the sense that we replace their logical rules by ones making use of an equational axiom set $\mathcal{E}$ at each inference step, such that the following inference rules may be applied if the equalities $A =_{\mathcal{E}} A^*$ and $B =_{\mathcal{E}} B^*$ hold.

The logical rules for

1. $\neg$-introduction:

$$\frac{\Gamma \vdash \Delta, \boldsymbol{A}^*}{\underline{\neg A}, \Gamma \vdash \Delta} \neg : \mathrm{l} \qquad \frac{\boldsymbol{A}^*, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \underline{\neg A}} \neg : \mathrm{r}$$

2. $\wedge$-introduction:

$$\frac{\boldsymbol{A}, \boldsymbol{B}^*, \Gamma \vdash \Delta}{\underline{A \wedge B}, \Gamma \vdash \Delta} \wedge : \mathrm{l} \qquad \frac{\Gamma \vdash \Delta, \boldsymbol{A}^* \quad \Pi \vdash \Lambda, \boldsymbol{B}^*}{\Gamma, \Pi \vdash \Delta, \Lambda, \underline{A \wedge B}} \wedge : \mathrm{r}$$

3. $\vee$-introduction:

$$\frac{\boldsymbol{A}^*, \Gamma \vdash \Delta \quad \boldsymbol{B}^*, \Pi \vdash \Lambda}{\underline{A \vee B}, \Gamma, \Pi \vdash \Delta, \Lambda} \vee : \mathrm{l} \qquad \frac{\Gamma \vdash \Delta, \boldsymbol{A}^*, \boldsymbol{B}}{\Gamma \vdash \Delta, \underline{A \vee B}} \vee : \mathrm{r}$$

4. $\supset$-introduction:

$$\frac{\Gamma \vdash \Delta, \boldsymbol{A}^* \quad \boldsymbol{B}^*, \Pi \vdash \Lambda}{\underline{A \supset B}, \Gamma, \Pi \vdash \Delta, \Lambda} \supset : \mathrm{l} \qquad \frac{\boldsymbol{A}^*, \Gamma \vdash \Delta, \boldsymbol{B}^*}{\Gamma \vdash \Delta, \underline{A \supset B}} \supset : \mathrm{r}$$

5. $\forall$-introduction:

$$\frac{\boldsymbol{A}^*(\boldsymbol{x}/\boldsymbol{t}), \Gamma \vdash \Delta}{\underline{(\forall x)A(x)}, \Gamma \vdash \Delta} \forall : \mathrm{l} \qquad \frac{\Gamma \vdash \Delta, \boldsymbol{A}^*(\boldsymbol{x}/\boldsymbol{u})}{\Gamma \vdash \Delta, \underline{(\forall x)A(x)}} \forall : \mathrm{r}$$

   where t is an arbitrary term and u does not occur in the conclusion.

6. $\exists$-introduction:

$$\frac{\boldsymbol{A}^*(\boldsymbol{x}/\boldsymbol{u}), \Gamma \vdash \Delta}{\underline{(\exists x)A(x)}, \Gamma \vdash \Delta} \exists : \mathrm{l} \qquad \frac{\Gamma \vdash \Delta, \boldsymbol{A}^*(\boldsymbol{x}/\boldsymbol{t})}{\Gamma \vdash \Delta, \underline{(\exists x)A(x)}} \exists : \mathrm{r}$$

   where u does not occur in the conclusion and t is an arbitrary term.

Definition 6.4 ($\mathbf{LK}^{\mathcal{E}}$-proof, $\mathbf{LKp}^{\mathcal{E}}$-proof). An $\mathbf{LK}^{\mathcal{E}}$-proof respectively $\mathbf{LKp}^{\mathcal{E}}$-proof is a proof where $\mathcal{A}$ consists of atomic sequents including the standard axioms $\mathcal{A}_{\mathbf{LK}}$ of **LK** and the inference rules applied are those of $\mathbf{LK}^{\mathcal{E}}$ respectively $\mathbf{LKp}^{\mathcal{E}}$.

Definition 6.5 ($\mathbf{LKe}^{\mathcal{E}}$). $\mathbf{LKe}^{\mathcal{E}}$ is the calculus obtained from $\mathbf{LK}^{\mathcal{E}}$ by adding the following equality introduction rules to the existing rules of $\mathbf{LK}^{\mathcal{E}}$:

$$\frac{\Gamma \vdash \Delta, \boldsymbol{s} = \boldsymbol{t} \quad \boldsymbol{A}^*[\boldsymbol{s}]_{\boldsymbol{\Xi}^*}, \Pi \vdash \Lambda}{\underline{A[t]_\Xi}, \Gamma, \Pi \vdash \Delta, \Lambda} = (\Xi)\ : \mathrm{l} \qquad \frac{\Gamma \vdash \Delta, \boldsymbol{s} = \boldsymbol{t} \quad \Pi \vdash \Lambda, \boldsymbol{A}^*[\boldsymbol{s}]_{\boldsymbol{\Xi}^*}}{\Gamma, \Pi \vdash \Delta, \Lambda, \underline{A[t]_\Xi}} = (\Xi)\ : \mathrm{r}$$

respectively

$$\frac{\Gamma \vdash \Delta, \boldsymbol{t} = \boldsymbol{s} \quad \boldsymbol{A}^*[\boldsymbol{s}]_{\boldsymbol{\Xi}^*}, \Pi \vdash \Lambda}{\underline{A[t]_\Xi}, \Gamma, \Pi \vdash \Delta, \Lambda} =' (\Xi)\ : \mathrm{l} \qquad \frac{\Gamma \vdash \Delta, \boldsymbol{t} = \boldsymbol{s} \quad \Pi \vdash \Lambda, \boldsymbol{A}^*[\boldsymbol{s}]_{\boldsymbol{\Xi}^*}}{\Gamma, \Pi \vdash \Delta, \Lambda, \underline{A[t]_\Xi}} =' (\Xi)\ : \mathrm{r}$$

where s and t are arbitrary terms, Ξ and Ξ^* are sets of positions and $A =_{\mathcal{E}} A^*$.

Definition 6.6 ($\mathbf{LKe}^{\mathcal{E}}$-proof). An $\mathbf{LKe}^{\mathcal{E}}$-proof is a proof where $\mathcal{A}$ consists of atomic sequents including the standard axioms $\mathcal{A}_{\mathbf{LK}}$ of **LK** and the axiom set of reflexivity, i.e.

$$\{\vdash t = t \mid t \text{ a term}\},$$

and the inference rules applied are those of $\mathbf{LKe}^{\mathcal{E}}$.

Definition 6.7 ($\mathbf{LKep}^{\mathcal{E}}$). $\mathbf{LKep}^{\mathcal{E}}$ is the calculus obtained from $\mathbf{LKe}^{\mathcal{E}}$ by again adding the semantic cut rule (see definition 2.12) to the existing rules of $\mathbf{LKe}^{\mathcal{E}}$.

Definition 6.8 ($\mathbf{LKep}^{\mathcal{E}}$-proof). An $\mathbf{LKep}^{\mathcal{E}}$-proof is a proof where $\mathcal{A}$ consists of atomic sequents including the standard axioms $\mathcal{A}_{\mathbf{LK}}$ of **LK** and the axiom set of reflexivity and the inference rules applied are those of $\mathbf{LKep}^{\mathcal{E}}$.

6.2 Example

The following example demonstrates the usage of the concept of equational theories within $\mathbf{LKep}^{\mathcal{E}}$. In this example we will use $\mathbf{LKep}^{\mathcal{E}}$-inferences modulo groups, i.e. the underlying equational theory is the theory of groups G (the binary connective of G is $\circ$, the neutral element is e).

We define the following axioms:

$$\begin{array}{ll} A_1: & (\forall x)(x \neq s(x) \wedge x \neq s(s(x))), \\ A_2: & (\forall x)(\forall y)(x = s(y) \vee y = s(x) \vee x = y), \\ A_3: & (\forall x)(\forall y)(s(x) = s(y) \supset x = y) \end{array}$$

and the conclusion:

$$C: \quad (\forall x)(\forall y) x \circ y = y \circ x$$

Furthermore we are using the following abbreviation $\Gamma_A \equiv A_1, A_2, A_3$ for the sequence of all three axioms.

We consider the following proof with G-pseudo-cuts:

$$\frac{\overset{(\phi)}{\Gamma_A \vdash (\forall y)(y = e \vee y = s(e) \vee y = s(s(e)))} \quad \overset{(\psi)}{(\exists z)(\exists z')(\exists z'')(\forall y)(y = z \circ z' \vee y = z \circ z'' \vee y = z) \vdash C}}{\Gamma_A \vdash C} \text{ p-cut}$$

Informal proof ϕ: From A_2 by setting x to e we get

$$(\forall y)(e = s(y) \vee y = s(e) \vee e = y). \qquad (*)$$

and setting y to s(e)) we get

$$e = s(s(s(e))) \vee s(s(e)) = s(e) \vee e = s(s(e)).$$

From A_1 we infer $e = s(s(s(e)))$. By $e = s(s(s(e)))$ and A_3 we get

$$(\forall y)(e = s(y) \leftrightarrow y = s(s(e))). \tag{**}$$

Therefore, from $(*)$ and the equivalence $(**)$ we obtain the left cut-formula

$$(\forall y)(y = e \vee y = s(e) \vee y = s(s(e))). \tag{I}$$

Note that no group theoretic inferences are required in ϕ.

Informal proof ψ: From the left cut-formula

$$(\exists z)(\exists z')(\exists z'')(\forall y)(y = z \circ z' \vee y = z \circ z'' \vee y = z) \tag{II}$$

it follows that the structure consists of three elements only. As the underlying structure is G we have a group with 3 elements. But there is only one such group and this is commutative.

Therefore (II) and G imply C.

Moreover we have a pseudo-cut w.r.t. G. In fact (I) $\supset$ (II) is valid under G: just choose $z = e$, $z' = s(e)$, and $z'' = s(s(e))$. This cut can be eliminated with CERESe under use of G.

Note that, within the example, the subproof ψ, i.e. the proof of the group containing only three elements, demonstrates also the expressive power of the combined method since it would not be possible (for practical reasons) to prove the theorem without having equality also as a rule.

The example above should be treated as an appetizer for what is possible by means of the CERESe method using equational theories.

CHAPTER 7

The Cut-Elimination System CERES

Implementing a cut-elimination system is motivated by being actually willing to perform cut-elimination on proofs of mathematical significance. These proofs easily grow that big (especially formalized in a calculus like **LK**) that no human being (even with a lot of patience) is able to do it by hand. Now why should one be interested in making implicit lemma applications explicit gaining a much bigger proof (at least in general) of the same theorem and what is the reward.

No doubt cut-elimination is one of the most important techniques of proof analyzation and transformation in modern proof theory. Besides usage as a tool showing the consistency of certain calculi and logics mainly three arguments are essential:

- Cut-elimination can be used to construct elementary proofs from non-elementary ones, as e.g. done in Girard's analysis of Van der Waerden's theorem (Girard 1987). Reducing a topological argumentation to a useful and purely combinatorial proof by means of cut-elimination. Clearly it does not make sense to make all proofs elementary but some.

- Cut-elimination is crucial for constructing structures or extracting information from proofs of which the techniques used to obtain these can only operate on cut-free proofs, e.g. the extraction of Herbrand disjunctions.

- Cut-elimination could serve the higher purpose of extracting constructive algorithms or proof methods and schemes directly from proofs which are already contained in the argumentation but still "hidden" in the proofs.

Agreed, cut-elimination of very simple toy-proofs can easily be done and the results are only boring. Whereas concrete proofs of more mathematical significance are far to complex to apply cut-elimination without a system and without an aim.

Based on the comparisons of the cut-elimination methods we have seen so far CERES brings the best qualifications with it. An additional advantage speaking for an implementation of CERES in contrast to others like Gentzen's is — besides the speedup results — the use of the resolution concept. Allowing to chose from a broad variety of tools, especially automated theorem provers, of which both the methods and algorithms as the implementations have been fine tuned to "almost" perfection.

7.1 The System

The system `CERES`[1] implements exactly the cut-elimination method CERES including the equality extensions to CERESe (described in the chapters 4 and 5). So from now on all properties associated with the system derive from CERESe.

Before going into detail we will give some technical key data.

The cut-elimination system `CERES` is written in ANSI-C++[2] using intensively the framework of the Standard Template Library (STL[3]). Since the structural data representation language XML is used for input and output the system also builds on the Libxml2[4] library. Currently the system is available only on Linux based operating systems (including Darwin — Mac OS X).

The main tasks are:

- Computation of an unsatisfiable set of clauses characterizing the cut-formulas. This is done by automatically extracting the characteristic clause term from an input proof φ formalized in **LKe** and computing the resulting *characteristic clause set* $\mathrm{CL}(\varphi)$.
- Generating a resolution refutation of $\mathrm{CL}(\varphi)$ (aided by an external automated theorem prover[5]) and the proof projection schemes of φ with respect to $\mathrm{CL}(\varphi)$ including the necessary proof projection instances (the ones actually used in the refutation). The properly instantiated projection schemes are then concatenated, using the refutation obtained by the theorem prover as a skeleton of a proof with at most atomic cuts.
- Internal tasks which in the mean time take up a non-negligible expense. Among others are some proof transformations (translating different calculi), unification and matching algorithms, regularization and skolemization algorithms, proof validation and a bunch of elementary proof operations.

The program is called via the command line specifying the file names of the input and output XML files (for the concrete format see section 7.2). On execution the system runs without any interaction with the user (in case of successful execution, otherwise errors are reported).

The work flow of the program during run-time is the following:

1. Process the input file, validate the XML file against a specification in form of a DTD and build up the internal representation of the input **LKe**-proof φ as a data structure.
2. Verify the correctness of φ according to the definition of **LKe**.
3. Skolemization of the proof on demand (after regularization).
4. Extract the characteristic clause term $\Theta(\varphi)$ of φ.
5. Compute the characteristic clause set $\mathrm{CL}(\varphi)$ of φ.

[1] The documentation of the method and the current version of the system `CERES` are available at `http://www.logic.at/ceres/`.

[2] The C++ Programming Language following the International Standard 14882:1998 approved as an American National Standard (see `http://www.ansi.org`).

[3] See `http://www.sgi.com/tech/stl/` for more details about the STL.

[4] Available from `http://www.xmlsoft.org`.

[5] The current version of `CERES` uses the automated theorem prover Otter (see `http://www-unix.mcs.anl.gov/AR/otter/`), but any refutational theorem prover based on resolution and paramodulation may be used.

6. Build the proof projection schemes $\varphi(C)$ for every $C \in \mathrm{CL}(\varphi)$.

7. Generate an input file for the automated theorem prover Otter containing $\mathrm{CL}(\varphi)$ and the axiom set and execute Otter.

8. Process the output file of Otter (see section 7.3 for details) and transform the refutation into a regular **PR**-proof γ.

9. Compute a combined global m.g.u. of γ and apply it to γ yielding δ.

10. Transform δ into an **LKe**-proof ψ.

11. Generate the required proof projection instances and concatenate them with ψ propagating those parts of the proof projections which are parts of the end-sequent of φ to the root.

12. Write the XML output file containing the proofs φ, γ, ψ and the proof projection schemes and $\Theta(\varphi)$, $\mathrm{CL}(\varphi)$ and the used axiom set.

Since the restriction to skolemized proofs is crucial to the CERES method, the system also performs skolemization (according to Andrew's method (Andrews 1971)) on the input proof if required.

To increase the performance and avoid redundancy, the data structures of formulas and terms are internally represented as directed acyclic graphs. This representation turns out to be very handy, also for the internal unification algorithms (see Baader and Snyder (2001) for performance investigations). The calculi on which `CERES` is operating are defined on sequents respectively clauses (i.e. atomic sequents) which are represented as sequential lists of formulas, in the sense of data structures. This is a very important point for the implementation of a system since e.g. a definition as multisets of formulas would require special treatment by the system (arbitrary permutations might occur at inferences) and of course the ancestor relation could not be uniquely defined.

The formal analysis of mathematical proofs (especially by a mathematician as a pre- and post-"processor") relies on a suitable representation format for the input and output of proofs, and on an appropriate aid in dealing with them. We developed an intermediary proof language[6] connecting the language of mathematical proofs with **LKe**. Furthermore we implemented a proof tool[7] acting as viewer and editor with a graphical user interface, allowing a convenient input and analysis of the output of `CERES`. Thereby the integration of the equality rules into the underlying calculus plays an essential role in overlooking, understanding and analyzing complex mathematical proofs by humans (see also Baaz, Hetzl, Leitsch, Richter and Spohr 2006).

`CERES` already proofed to be efficient and reliable in performing cut-elimination on input **LKe**-proofs with hundreds of proof nodes. A first impression of the capabilities demonstrated on some examples of `CERES` can be found in chapter 8 and also on the official website[8].

[6] See `http://www.logic.at/hlk/` for more information.
[7] Details can be found at `http://www.logic.at/prooftool/`.
[8] The website of `CERES` is `http://www.logic.at/ceres/`.

7.2 Input and Output

The system `CERES` expects a proof of a theorem formalized as an **LKe**-proof φ and the used axiom set as input and outputs several proofs (input proof, proof projection schemes $\varphi(C)$, resolution refutation and cut-free output proof), the characteristic clause term $\Theta(\varphi)$, the characteristic clause set $\mathrm{CL}(\varphi)$ and again the used axiom set.

Input and output are formatted using the well known data representation language XML. This allows the use of arbitrary and well known utilities for editing, transformation and presentation as well as standardized programming libraries.

To get a better impression how proofs are structured in the input and output XML format we give a commented excerpt from the DTD specifying proofs:

```
<!ELEMENT proof                ((rule|sequent),mark*)>
<!ATTLIST proof
          symbol               CDATA         #IMPLIED
          calculus             CDATA         #IMPLIED
          >

<!ELEMENT rule                 (sequent,(rule|sequent|prooflink)+,mark*)>
<!ATTLIST rule
          symbol               CDATA         #IMPLIED
          type                 CDATA         #IMPLIED
          param                CDATA         #IMPLIED
          >

<!ELEMENT prooflink            EMPTY>
<!ATTLIST prooflink
          symbol               CDATA         #IMPLIED
          >

<!ELEMENT sequent              (formulalist,formulalist,mark*)>
```

Note that we also support markings of important parts of the structure for special purposes, e.g. forthcoming user interactions or cut-formula ancestor relations.

```
<!ELEMENT mark                 (#PCDATA)>
<!ATTLIST mark
          type                 CDATA         #IMPLIED
          >
```

The type of the calculus can be `LK` in case of an **LKe**-proof or `PR` in case of a **PR**-proof.

The rule types of the calculus **LK** follow the mapping:

LKe-rule	attribute `type`
axiom	`ax`
w : l	`weakl`
w : r	`weakr`
c : l	`contrl`

LKe-rule	attribute `type`
c : r	`contrr`
π : l	`perml`
π : r	`permr`
cut	`cut`

LKe-rule	attribute `type`
$\neg$: l	`negl`
$\neg$: r	`negr`
$\wedge$: l	`andl`
$\wedge$: r	`andr`

LKe-rule	attribute type
$\vee : \mathrm{l}$	orl
$\vee : \mathrm{r}$	orr
$\supset : \mathrm{l}$	impll
$\supset : \mathrm{r}$	implr

LKe-rule	attribute type
$\forall : \mathrm{l}$	foralll
$\forall : \mathrm{r}$	forallr
$\exists : \mathrm{l}$	existsl
$\exists : \mathrm{r}$	existsr

LKe-rule	attribute type
$= : \mathrm{l}$	eql1
$= : \mathrm{r}$	eqr1
$=' : \mathrm{l}$	eql2
$=' : \mathrm{r}$	eqr2

and the rule types of the resolution calculus **R** are defined as follows:

PR-rule	attribute type
initial clause	init
$\pi : \mathrm{l}$	perml
$\pi : \mathrm{r}$	permr
$\mathrm{f} : \mathrm{l}$	factl
$\mathrm{f} : \mathrm{r}$	factr

PR-rule	attribute type
r	res
$\mathrm{p} : \mathrm{l}$	paral1
$\mathrm{p} : \mathrm{r}$	parar1
$\mathrm{p}' : \mathrm{l}$	paral2
$\mathrm{p}' : \mathrm{r}$	parar2

Formulas are represented by

```
<!ENTITY % formula
        '(formulavariable|conjunctiveformula|quantifiedformula|atomformula)'>

<!ELEMENT formulavariable     EMPTY>
<!ATTLIST formulavariable
          symbol              CDATA         #IMPLIED
          >

<!ENTITY % formulalistelement '(%formula;|formulalistvariable)'>
<!ELEMENT formulalist         (%formulalistelement;)*>

<!ELEMENT formulalistvariable EMPTY>
<!ATTLIST formulalistvariable
          symbol              CDATA         #IMPLIED
          >

<!ELEMENT conjunctiveformula  ((%formula;)+,mark*)>
<!ATTLIST conjunctiveformula
          type                 CDATA         #IMPLIED
          >

<!ELEMENT quantifiedformula   (variable,%formula;,mark*)>
<!ATTLIST quantifiedformula
          type                CDATA         #IMPLIED
          >
```

and terms by

```
<!ENTITY % term
        '(function|variable|constant)'>

<!ELEMENT atomformula        ((%term;)*,mark*)>
<!ATTLIST atomformula
          symbol             CDATA         #IMPLIED
          >

<!ELEMENT function           ((%term;)+,mark*)>
<!ATTLIST function
          symbol             CDATA         #IMPLIED
          >

<!ELEMENT variable           (mark*)>
<!ATTLIST variable
          symbol             CDATA         #IMPLIED
          >

<!ELEMENT constant           (mark*)>
<!ATTLIST constant
          symbol             CDATA         #IMPLIED
          >
```

7.3 The Theorem Prover: Otter

As already mentioned, CERES is not choosy when it comes to the used automated theorem prover. Therefore a large variety of theorem provers can be considered.

Since performance is not the issue flexibility is a more interesting property of a theorem prover making experiments (as in chapter 8) easier.

We decided to give the automated theorem prover Otter a try since at the time we started developing the system it was the only theorem prover, we were aware of, generating a formalized output proof intended for post processing, the so called *proof objects*.

CERES uses this proof objects of Otter, which are resolution derivations according to the calculus definition given below, as an input and transforms them into a resolution refutation of **PR**.

Otter clauses are disjunctions of literals.

Definition 7.1 (Otter clause). We define Otter clauses inductively as follows:

1. The empty Otter clause $\Box$ is an Otter clause.
2. Literals, i.e. A and $\neg A$ for an atom A, are Otter clauses.
3. If C and D are Otter clauses, then $C \vee D$ is an Otter clause.

The empty Otter clause is denoted by $\Box$. Furthermore we define the identities:

$$C \vee \Box \vee D = C \vee D$$
$$\Box \vee \Box = \Box.$$

Definition of the Otter resolution calculus **ORES**.

Definition 7.2 (ORES). The derivation rules are:

1. Input:
$$\frac{}{L_1 \vee \ldots \vee L_n}\ \mathrm{in}$$

2. Instantiation:
$$\frac{L_1 \vee \ldots \vee L_n}{(L_1 \vee \ldots \vee L_n)\sigma}\ \mathrm{i}(\sigma)$$
for a substitution σ.

3. Propositional:
$$\frac{L_1 \vee \ldots \vee L_i \vee \ldots \vee L_n}{L_1 \vee \ldots \vee L_{i-1} \vee L_{i+1} \vee \ldots \vee L_n}\ \mathrm{prop}$$
where $L_i = L_j$ for some $j \neq i$, $1 \leq i, j \leq n$.

4. (Binary) resolvent:
$$\frac{C \vee \neg A \vee D \quad C' \vee A \vee D'}{C \vee D \vee C' \vee D'}\ \mathrm{res}$$
where C, C', D and D' denote Otter clauses.

5. Paramodulation:
$$\frac{C \vee s = t \vee D \quad C' \vee A[s] \vee D'}{C \vee D \vee C' \vee A[t] \vee D'}\ \mathrm{para}$$

6. Flipping:
$$\frac{C \vee s = t \vee D}{C \vee t = s \vee D}\ \mathrm{f}_{pos} \qquad \frac{C \vee \neg s = t \vee D}{C \vee \neg t = s \vee D}\ \mathrm{f}_{neg}$$
Otter knows only a single flipping rule, which we splitted into a positive and a negative variant.

CHAPTER **8**

Experiments with CERES

This chapter is dedicated to some interesting experiments with CERES. On the one hand demonstrating some of the abilities of the program and on the other hand to do some fire tests on more or less known mathematical proofs. These considerable and more extensive examples will show the ease of doing real work with CERES and which different schemes of things arise from it.

8.1 Effects of Resolution Refinements

The use of the resolution refutation of the characteristic clause set as a skeleton for the cut-free proof makes it possible to change the mathematical character of the resulting proof via different resolution refutations, e.g. using different resolution refinements. Within these refutations *universal lemmas*, i.e. clauses containing variables representing universal formulas, appear which do neither occur in the original proof nor in the cut-eliminated proofs, where they are already instantiated. Now we are doing exactly such an experiment using an input proof already analyzed and defined as an **LK**-derivation in Urban (2000) with the program CERES. This experiment has already been demonstrated at the LPAR 2004[1] (see also Baaz, Hetzl, Leitsch, Richter and Spohr 2004).

The proof deals with the following situation: We are given an infinite tape where each cell is either labelled '0' or '1'. We prove that on this tape there are two cells labelled with the same number. The contents of a cell of the tape is denoted by the unary function f taking the cell index as an argument (e.g. $f(u) = 0$ means that the cell with index u is labelled '0'), s is the successor function and $m^{u,v}$ is the maximum of u and v.

[1]The LPAR 2004 (11th International Conference on Logic for Programming, Artificial Intelligence and Reasoning) was held on March 14-18th, 2005 in Montevideo, Uruguay.

Within this section the following formula abbreviations are used:

$$\begin{aligned}
M_1 &= (\forall y)(\forall x)x \le m^{x,y} \\
M_2 &= (\forall y)(\forall x)y \le m^{x,y} \\
S &= (\forall x)(\forall y)(s(x) \le y \supset x < y) \\
T &= (\forall i)(\forall x)(\forall y)((f(x) = i \wedge f(y) = i) \supset f(x) = f(y)) \\
A &= (\forall x)(f(x) = 0 \vee f(x) = 1) \\
P &= (\exists p)(\exists q)(p < q \wedge f(p) = f(q)) \\
\infty_0 &= (\forall n)(\exists k)(n \le k \wedge f(k) = 0) \\
\infty_1 &= (\forall m)(\exists l)(m \le l \wedge f(l) = 1)
\end{aligned}$$

moreover 1 is an abbreviation for $s(0)$.

Then, let the proof φ be defined as follows.

$\varphi =$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overset{(\tau)}{M_1, M_2, A \vdash \infty_0, \infty_1} \quad \overset{(\epsilon_1)}{\infty_1, S, T \vdash P}}{M_1, M_2, A, S, T \vdash, \infty_0, P}\ \text{cut}}{M_1, M_2, A, S, T \vdash, P, \infty_0}\ \pi((1\,2)):\mathrm{r} \quad \overset{(\epsilon_0)}{\infty_0, S, T \vdash P}}{M_1, M_2, A, S, T, S, T \vdash P, P}\ \text{cut}}{M_1, M_2, S, S, T, T, A \vdash P, P}\ \pi((3\,6\,7)):\mathrm{l}}{M_1, M_2, S, T, A \vdash P, P}\ \mathrm{c}((1,1,2,2,1)):\mathrm{l}}{M_1, M_2, S, T, A \vdash P}\ \mathrm{c}((2)):\mathrm{r}$$

The mathematical proof argument of φ says from the facts that there are infinitely many indices labelled '0' and infinitely many indices labelled '1' we can conclude that there are two different indices which have the same content.

The used subproofs τ, ϵ_0 and ϵ_1 are given below (for more details be referred to Urban 2000).

$\tau =$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{\boldsymbol{v \le m^{u,v}} \vdash v \le m^{u,v}}}{\underline{\boldsymbol{(\forall x)v \le m^{x,v}}} \vdash v \le m^{u,v}}\ \forall:\mathrm{l}}{\underline{(\forall y)(\forall x)y \le m^{x,y}} \vdash \boldsymbol{v \le m^{u,v}}}\ \forall:\mathrm{l} \quad \dfrac{\dfrac{\dfrac{\dfrac{\overline{\boldsymbol{u \le m^{u,v}} \vdash u \le m^{u,v}}}{\underline{\boldsymbol{(\forall x)x \le m^{x,v}}} \vdash u \le m^{u,v}}\ \forall:\mathrm{l}}{\underline{(\forall y)(\forall x)x \le m^{x,y}} \vdash \boldsymbol{u \le m^{u,v}}}\ \forall:\mathrm{l} \quad (\tau')}{M_1, A \vdash \boldsymbol{f(m^{u,v}) = 1}, \underline{\boldsymbol{u \le m^{u,v} \wedge f(m^{u,v}) = 0}}}\ \wedge:\mathrm{r}}{M_1, A \vdash \underline{u \le m^{u,v} \wedge f(m^{u,v}) = 0}, \underline{\boldsymbol{f(m^{u,v}) = 1}}}\ \pi((1\,2)):\mathrm{r}}{M_1, M_2, A \vdash u \le m^{u,v} \wedge f(m^{u,v}) = 0, \underline{\boldsymbol{v \le m^{u,v} \wedge f(m^{u,v}) = 1}}}\ \wedge:\mathrm{r}}{M_1, M_2, A \vdash \boldsymbol{u \le m^{u,v} \wedge f(m^{u,v}) = 0}, \underline{\boldsymbol{(\exists l)(v \le l \wedge f(l) = 1)}}}\ \exists:\mathrm{r}}{M_1, M_2, A \vdash \underline{(\exists l)(v \le l \wedge f(l) = 1)}, \underline{\boldsymbol{u \le m^{u,v} \wedge f(m^{u,v}) = 0}}}\ \pi((1\,2)):\mathrm{r}}{M_1, M_2, A \vdash (\exists l)(v \le l \wedge f(l) = 1), \underline{\boldsymbol{(\exists k)(u \le k \wedge f(k) = 0)}}}\ \exists:\mathrm{r}}{M_1, M_2, A \vdash \boldsymbol{(\exists l)(v \le l \wedge f(l) = 1)}, \underline{\boldsymbol{(\forall n)(\exists k)(n \le k \wedge f(k) = 0)}}}\ \forall:\mathrm{r}}{M_1, M_2, A \vdash \underline{(\forall n)(\exists k)(n \le k \wedge f(k) = 0)}, \underline{\boldsymbol{(\exists l)(v \le l \wedge f(l) = 1)}}}\ \pi((1\,2)):\mathrm{r}}{M_1, M_2, A \vdash \infty_0, \underline{(\forall m)(\exists l)(m \le l \wedge f(l) = 1)}}\ \forall:\mathrm{r}$$

$\tau' =$

$$
\begin{array}{cl}
f(m^{u,v}) = 0 \vdash f(m^{u,v}) = 0 \qquad f(m^{u,v}) = 1 \vdash f(m^{u,v}) = 1 & \\ \hline
f(m^{u,v}) = 0 \lor f(m^{u,v}) = 1 \vdash f(m^{u,v}) = 0, f(m^{u,v}) = 1 & \lor : \mathrm{l} \\ \hline
(\forall x)(f(x) = 0 \lor f(x) = 1) \vdash f(m^{u,v}) = 0, f(m^{u,v}) = 1 & \forall : \mathrm{l} \\ \hline
(\forall x)(f(x) = 0 \lor f(x) = 1) \vdash f(m^{u,v}) = 1, f(m^{u,v}) = 0 & \pi((1\,2)) : \mathrm{r}
\end{array}
$$

$\epsilon_0 =$

$$
\begin{array}{cl}
(\epsilon_0') & \\
s(u) \leq v \land f(v) = 0, 0 \leq u \land f(u) = 0, S, T \vdash P & \\ \hline
(\exists k)(s(u) \leq k \land f(k) = 0), 0 \leq u \land f(u) = 0, S, T \vdash P & \exists : \mathrm{l} \\ \hline
(\forall n)(\exists k)(n \leq k \land f(k) = 0), 0 \leq u \land f(u) = 0, S, T \vdash P & \forall : \mathrm{l} \\ \hline
0 \leq u \land f(u) = 0, (\forall n)(\exists k)(n \leq k \land f(k) = 0), S, T \vdash P & \pi((1\,2)) : \mathrm{l} \\ \hline
(\exists k)(0 \leq k \land f(k) = 0), (\forall n)(\exists k)(n \leq k \land f(k) = 0), S, T \vdash P & \exists : \mathrm{l} \\ \hline
(\forall n)(\exists k)(n \leq k \land f(k) = 0), (\forall n)(\exists k)(n \leq k \land f(k) = 0), S, T \vdash P & \forall : \mathrm{l} \\ \hline
(\forall n)(\exists k)(n \leq k \land f(k) = 0), S, T \vdash P & \mathrm{c}(2,1,1) : \mathrm{l}
\end{array}
$$

$\epsilon_1 =$

$$
\begin{array}{cl}
(\epsilon_1') & \\
s(u) \leq v \land f(v) = 1, 1 \leq u \land f(u) = 1, S, T \vdash P & \\ \hline
(\exists l)(s(u) \leq l \land f(l) = 1), 1 \leq u \land f(u) = 1, S, T \vdash P & \exists : \mathrm{l} \\ \hline
(\forall m)(\exists l)(m \leq l \land f(l) = 1), 1 \leq u \land f(u) = 1, S, T \vdash P & \forall : \mathrm{l} \\ \hline
1 \leq u \land f(u) = 1, (\forall m)(\exists l)(m \leq l \land f(l) = 1), S, T \vdash P & \pi((1\,2)) : \mathrm{l} \\ \hline
(\exists l)(1 \leq l \land f(l) = 1), (\forall m)(\exists l)(m \leq l \land f(l) = 1), S, T \vdash P & \exists : \mathrm{l} \\ \hline
(\forall m)(\exists l)(m \leq l \land f(l) = 1), (\forall m)(\exists l)(m \leq l \land f(l) = 1), S, T \vdash P & \forall : \mathrm{l} \\ \hline
(\forall m)(\exists l)(m \leq l \land f(l) = 1), S, T \vdash P & \mathrm{c}(2,1,1) : \mathrm{l}
\end{array}
$$

$\epsilon_j' =$

$$
\begin{array}{cl}
s(u) \leq v \vdash s(u) \leq v \qquad u < v \vdash u < v & \\ \hline
s(u) \leq v \supset u < v, s(u) \leq v \vdash u < v & \supset : \mathrm{l} \\ \hline
(\forall y)(s(u) \leq y \supset u < y), s(u) \leq v \vdash u < v & \forall : \mathrm{l} \\ \hline
(\forall x)(\forall y)(s(x) \leq y \supset x < y), s(u) \leq v \vdash u < v \qquad (\epsilon_j'') & \forall : \mathrm{l} \\ \hline
S, s(u) \leq v, T, f(u) = j, f(v) = j \vdash u < v \land f(u) = f(v) & \land : \mathrm{r} \\ \hline
s(u) \leq v, f(v) = j, S, T, f(u) = j \vdash u < v \land f(u) = f(v) & \pi((1\,2\,5\,4\,3)) : \mathrm{l} \\ \hline
s(u) \leq v \land f(v) = j, f(u) = j, S, T \vdash u < v \land f(u) = f(v) & \land : \mathrm{l} \\ \hline
f(u) = j, s(u) \leq v \land f(v) = j, S, T \vdash u < v \land f(u) = f(v) & \pi((1\,2)) : \mathrm{l} \\ \hline
j \leq u, f(u) = j, s(u) \leq v \land f(v) = j, S, T \vdash u < v \land f(u) = f(v) & \mathrm{w} : \mathrm{l} \\ \hline
j \leq u \land f(u) = j, s(u) \leq v \land f(v) = j, S, T \vdash u < v \land f(u) = f(v) & \land : \mathrm{l} \\ \hline
j \leq u \land f(u) = j, s(u) \leq v \land f(v) = j, S, T \vdash (\exists q)(u < q \land f(u) = f(q)) & \exists : \mathrm{r} \\ \hline
j \leq u \land f(u) = j, s(u) \leq v \land f(v) = j, S, T \vdash (\exists p)(\exists q)(p < q \land f(p) = f(q)) & \exists : \mathrm{r} \\ \hline
s(u) \leq v \land f(v) = j, j \leq u \land f(u) = j, S, T \vdash P & \pi((1\,2)) : \mathrm{l}
\end{array}
$$

$\epsilon_j'' =$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{f(u)=j \vdash \boldsymbol{f(u)=j}} \quad \overline{f(v)=j \vdash \boldsymbol{f(v)=j}}}{f(u)=j, f(v)=j \vdash \boldsymbol{f(u)=j \wedge f(v)=j}}\ \wedge:\mathrm{r} \qquad \overline{\boldsymbol{f(u)=f(v)} \vdash f(u)=f(v)}}{\boldsymbol{((f(u)=j \wedge f(v)=j) \supset f(u)=f(v))}, f(u)=j, f(v)=j \vdash f(u)=f(v)}\ \supset:\mathrm{l}}{\boldsymbol{(\forall y)((f(u)=j \wedge f(y)=j) \supset f(u)=f(y))}, f(u)=j, f(v)=j \vdash f(u)=f(v)}\ \forall:\mathrm{l}}{\boldsymbol{(\forall x)(\forall y)((f(x)=j \wedge f(y)=j) \supset f(x)=f(y))}, f(u)=j, f(v)=j \vdash f(u)=f(v)}\ \forall:\mathrm{l}}{(\forall i)(\forall x)(\forall y)((f(x)=i \wedge f(y)=i) \supset f(x)=f(y)), f(u)=j, f(v)=j \vdash \boldsymbol{f(u)=f(v)}}\ \forall:\mathrm{l}}{}
$$

The characteristic clause term $\Theta(\varphi)$ extracted from φ is

$$
\begin{aligned}
\Theta(\varphi) =& (((\{\vdash v \leq m^{u,v}\} \oplus (\{\vdash u \leq m^{u,v}\} \oplus (\{\vdash f(m^{u,v})=0\} \otimes \{\vdash f(m^{u,v})=1\}))) \\
& \oplus ((\{s(u) \leq v \vdash\} \otimes \{\vdash\}) \otimes ((\{f(u)=1 \vdash\} \otimes \{f(v)=1 \vdash\}) \otimes \{\vdash\}))) \\
& \oplus ((\{s(u) \leq v \vdash\} \otimes \{\vdash\}) \otimes ((\{f(u)=0 \vdash\} \otimes \{f(v)=0 \vdash\}) \otimes \{\vdash\})))
\end{aligned}
$$

and the corresponding characteristic clause set $\mathrm{CL}(\varphi)$ obtained from $\Theta(\varphi)$ results to

$$
\begin{aligned}
\mathrm{CL}(\varphi) = \{ & \vdash v \leq m^{u,v}, && (C_1)\\
& \vdash u \leq m^{u,v}, && (C_2)\\
& \vdash f(m^{u,v})=0, f(m^{u,v})=1, && (C_3)\\
& s(u) \leq v, f(u)=1, f(v)=1 \vdash, && (C_4)\\
& s(u) \leq v, f(u)=0, f(v)=0 \vdash \} && (C_5)
\end{aligned}
$$

The projection schemes obtained from φ for the five clauses above are the following:

$\varphi(C_1) =$

$$
\dfrac{\dfrac{\overline{v \leq m^{u,v} \vdash v \leq m^{u,v}}}{(\forall x) v \leq m^{x,v} \vdash v \leq m^{u,v}}\ \forall:\mathrm{l}}{(\forall y)(\forall x) y \leq m^{x,y} \vdash v \leq m^{u,v}}\ \forall:\mathrm{l}
$$

$\varphi(C_2) =$

$$
\dfrac{\dfrac{\overline{u \leq m^{u,v} \vdash u \leq m^{u,v}}}{(\forall x) x \leq m^{x,v} \vdash u \leq m^{u,v}}\ \forall:\mathrm{l}}{(\forall y)(\forall x) x \leq m^{x,y} \vdash u \leq m^{u,v}}\ \forall:\mathrm{l}
$$

$\varphi(C_3) =$

$$
\dfrac{\dfrac{\overline{f(m^{u,v})=0 \vdash f(m^{u,v})=0} \quad \overline{f(m^{u,v})=1 \vdash f(m^{u,v})=1}}{f(m^{u,v})=0 \vee f(m^{u,v})=1 \vdash f(m^{u,v})=0, f(m^{u,v})=1}\ \vee:\mathrm{l}}{(\forall x)(f(x)=0 \vee f(x)=1) \vdash f(m^{u,v})=0, f(m^{u,v})=1}\ \forall:\mathrm{l}
$$

$\varphi(C_4) = \psi_1$

$\varphi(C_5) = \psi_0$

where ψ_j is defined:

$\psi_j =$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{s(u) \leq v \vdash s(u) \leq v} \quad \overline{u < v \vdash u < v}}{s(u) \leq v \supset u < v, s(u) \leq v \vdash u < v}\ \supset : \mathrm{l}}{(\forall y)(s(u) \leq y \supset u < y), s(u) \leq v \vdash u < v}\ \forall : \mathrm{l}}{(\forall x)(\forall y)(s(x) \leq y \supset x < y), s(u) \leq v \vdash u < v}\ \forall : \mathrm{l} \qquad \psi_j'}{S, s(u) \leq v, T, f(u) = j, f(v) = j \vdash u < v \wedge f(u) = f(v)}\ \wedge : \mathrm{r}}{S, s(u) \leq v, T, f(u) = j, f(v) = j \vdash (\exists q)(u < q \wedge f(u) = f(q))}\ \exists : \mathrm{r}}{S, s(u) \leq v, T, f(u) = j, f(v) = j \vdash (\exists p)(\exists q)(p < q \wedge f(p) = f(q))}\ \exists : \mathrm{r}
$$

and $\psi_j' =$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{f(u) = j \vdash f(u) = j} \quad \overline{f(v) = j \vdash f(v) = j}}{f(u) = j, f(v) = j \vdash f(u) = j \wedge f(v) = j}\ \wedge : \mathrm{r} \qquad \overline{f(u) = f(v) \vdash f(u) = f(v)}}{(f(u) = j \wedge f(v) = j) \supset f(u) = f(v), f(u) = j, f(v) = j \vdash f(u) = f(v)}\ \supset : \mathrm{l}}{(\forall y)((f(u) = j \wedge f(y) = j) \supset f(u) = f(y)), f(u) = j, f(v) = j \vdash f(u) = f(v)}\ \forall : \mathrm{l}}{(\forall x)(\forall y)((f(x) = j \wedge f(y) = j) \supset f(x) = f(y)), f(u) = j, f(v) = j \vdash f(u) = f(v)}\ \forall : \mathrm{l}}{(\forall i)(\forall x)(\forall y)((f(x) = i \wedge f(y) = i) \supset f(x) = f(y)), f(u) = j, f(v) = j \vdash f(u) = f(v)}\ \forall : \mathrm{l}
$$

The resolution refutations yielding two mathematically different proofs of φ are demonstrated in the following two subsections. The resulting cut-free proofs have been omitted because of their sizes.

8.1.1 Positive Hyperresolution

Derivation of C_6:

$$
\dfrac{\overset{(C_3)}{\vdash f(m^{u,v}) = 0, f(m^{u,v}) = 1} \qquad \dfrac{\overset{(C_2\sigma_2)}{\vdash u \leq m^{u,w}} \quad \overset{(C_4\sigma_1)}{s(u') \leq v', f(u') = 1, f(v') = 1 \vdash}}{f(u') = 1, f(m^{s(u'),w}) = 1 \vdash}\ \mathrm{r}(\sigma_3)}{\underbrace{f(m^{s(m^{u,v}),w}) = 1 \vdash f(m^{u,v}) = 0}_{C_X}}\ \mathrm{r}(\sigma_4)
$$

$$
\dfrac{\overset{(C_3\sigma_5)}{\vdash f(m^{u',v'}) = 0, f(m^{u',v'}) = 1} \quad C_X}{\vdash f(m^{u,v}) = 0, f(m^{s(m^{u,v}),w}) = 0}\ \mathrm{r}(\sigma_6) \qquad (C_6)
$$

where $\sigma_1 = \{u \mapsto u', v \mapsto v'\}$, $\sigma_2 = \{v \mapsto w\}$, $\sigma_3 = \{u \mapsto s(u'), v' \mapsto m^{s(u'),w}\}$, $\sigma_4 = \{u' \mapsto m^{u,v}\}$, $\sigma_5 = \{u \mapsto u', v \mapsto v'\}$ and $\sigma_6 = \{u' \mapsto s(m^{u,v}), v' \mapsto w\}$.

For arbitrary u, v and w either the cell with index $i = m^{u,v}$ is labelled '0' or the cell with index $m^{i+1,w}$.

Derivation of C_7:

$$\dfrac{\overset{(C_6)}{\vdash f(m^{u,v}) = 0, f(m^{s(m^{u,v}),w}) = 0} \qquad \dfrac{\overset{(C_1\sigma_8)}{\vdash v \leq m^{u'',v}} \quad \overset{(C_5\sigma_7)}{s(u') \leq v', f(u') = 0, f(v') = 0 \vdash}}{f(u') = 0, f(m^{u'',s(u')}) = 0 \vdash}\ \mathrm{r}(\sigma_9)}{\underbrace{f(m^{u'',s(m^{s(m^{u,v}),w})}) = 0 \vdash f(m^{u,v}) = 0}_{C_Y}}\ \mathrm{r}(\sigma_{10})$$

$$\dfrac{\dfrac{\overset{(C_6\sigma_{11})}{\vdash f(m^{u',v'}) = 0, f(m^{s(m^{u',v'}),w'}) = 0} \quad C_Y}{\vdash f(m^{u',v'}) = 0, f(m^{u,v}) = 0}\ \mathrm{r}(\sigma_{12})}{\vdash f(m^{u,v}) = 0}\ \mathrm{f}(\sigma_{13}) : \mathrm{r} \qquad (C_7)$$

where $\sigma_7 = \{u \mapsto u', v \mapsto v'\}$, $\sigma_8 = \{u \mapsto u''\}$, $\sigma_9 = \{v \mapsto s(u'), v' \mapsto m^{u'',s(u')}\}$, $\sigma_{10} = \{u' \mapsto m^{s(m^{u,v}),w}\}$, $\sigma_{11} = \{u \mapsto u', v \mapsto v', w \mapsto w'\}$, $\sigma_{12} = \{u'' \mapsto s(m^{u',v'}), w' \mapsto s(m^{s(m^{u,v}),w})\}$ and $\sigma_{13} = \{u' \mapsto u, v' \mapsto v\}$.

For arbitrary u and v the cell with index $i = m^{u,v}$ is labelled '0'.

$$\dfrac{\overset{(C_7\sigma_{18})}{\vdash f(m^{u,v''}) = 0} \qquad \dfrac{\overset{(C_7\sigma_{16})}{\vdash f(m^{u',v}) = 0} \qquad \dfrac{\overset{(C_2\sigma_{14})}{\vdash u' \leq m^{u',v'}} \quad \overset{(C_5)}{s(u) \leq v, f(u) = 0, f(v) = 0 \vdash}}{f(u) = 0, f(m^{s(u),v'}) = 0 \vdash}\ \mathrm{r}(\sigma_{15})}{f(m^{s(m^{u',v}),v'}) = 0 \vdash}\ \mathrm{r}(\sigma_{17})}{\vdash}\ \mathrm{r}(\sigma_{19})$$

where $\sigma_{14} = \{u \mapsto u', v \mapsto v'\}$, $\sigma_{15} = \{u' \mapsto s(u), v \mapsto m^{s(u),v'}\}$, $\sigma_{16} = \{u \mapsto u'\}$, $\sigma_{17} = \{u \mapsto m^{u',v}\}$, $\sigma_{18} = \{v \mapsto v''\}$ and $\sigma_{19} = \{u \mapsto s(m^{u',v}), v'' \mapsto v'\}$.

For arbitrary u and v where $u < v$ at least one of the cells with index u or v should be labelled '1' but again for arbitrary u' and v' the cell with index $i = m^{u',v'}$ is labelled '0'. Hence choosing one time u as u' and one time v as v' leads to a contradiction.

8.1.2 Negative Hyperresolution

Derivation of C_6':

$$\dfrac{\overset{(C_1\sigma_1)}{\vdash v' \leq m^{u,v'}} \quad \overset{(C_4\sigma_2)}{s(v) \leq u', f(v) = 1, f(u') = 1 \vdash}}{f(v) = 1, f(m^{u,s(v)}) = 1 \vdash}\ \mathrm{r}(\sigma_3) \qquad (C_6')$$

where $\sigma_1 = \{v \mapsto v'\}$, $\sigma_2 = \{u \mapsto v, v \mapsto u'\}$ and $\sigma_3 = \{u' \mapsto m^{u,s(v)}, v' \mapsto s(v)\}$.

If a cell with index v is labelled '1' then no cell with an index bigger than v is labelled '1'.

Derivation of C'_7:

$$\dfrac{\overset{(C_2\sigma_4)}{\vdash u' \leq m^{u',v}} \quad \overset{(C_5\sigma_5)}{s(u) \leq v', f(u) = 0, f(v') = 0 \vdash}}{f(u) = 0, f(m^{s(u),v}) = 0 \vdash}\ \mathrm{r}(\sigma_6) \qquad (C'_7)$$

where $\sigma_4 = \{u \mapsto u'\}$, $\sigma_5 = \{v \mapsto v'\}$ and $\sigma_6 = \{u' \mapsto s(u), v' \mapsto m^{s(u),v}\}$.

If a cell with index u is labelled '0' then no cell with an index bigger than u is labelled '0'.

Derivation of C'_8:

$$\dfrac{\dfrac{\overset{(C_3\sigma_7)}{\vdash f(m^{u',v'}) = 1, f(m^{u',v'}) = 0} \quad \overset{(C'_7)}{f(m^{s(u),v}) = 0, f(u) = 0 \vdash}}{f(u) = 0 \vdash f(m^{s(u),v'}) = 1}\ \mathrm{r}(\sigma_8) \quad \overset{(C'_6\sigma_9)}{f(m^{u',s(v)}) = 1, f(v) = 1 \vdash}}{f(u) = 0, f(v) = 1 \vdash}\ \mathrm{r}(\sigma_{10}) \qquad (C'_8)$$

where $\sigma_7 = \{u \mapsto u', v \mapsto v'\}$, $\sigma_8 = \{u' \mapsto s(u), v \mapsto v'\}$, $\sigma_9 = \{u \mapsto u'\}$ and $\sigma_{10} = \{u' \mapsto s(u), v' \mapsto s(v)\}$.

If a cell with index v is labelled '1' then there is no cell with index u labelled '0', i.e. all cells are either only labelled '0' or only labelled '1'.

Derivation of C'_9:

$$\dfrac{\dfrac{\dfrac{\overset{(C_3\sigma_{11})}{\vdash f(m^{u',v'}) = 1, f(m^{u',v'}) = 0} \quad \overset{(C'_7)}{f(m^{s(u),v}) = 0, f(u) = 0 \vdash}}{f(u) = 0 \vdash f(m^{s(u),v}) = 1}\ \mathrm{r}(\sigma_{12}) \quad \overset{(C'_8\sigma_{13})}{f(v') = 1, f(u') = 0 \vdash}}{f(u) = 0, f(u') = 0 \vdash}\ \mathrm{r}(\sigma_{14})}{f(u) = 0 \vdash}\ \mathrm{f}(\sigma_{15}) : 1 \qquad (C'_9)$$

where $\sigma_{11} = \{u \mapsto u', v \mapsto v'\}$, $\sigma_{12} = \{v' \mapsto v, u' \mapsto s(u)\}$, $\sigma_{13} = \{u \mapsto u', v \mapsto v'\}$, $\sigma_{14} = \{v' \mapsto m^{s(u),v}\}$ and $\sigma_{15} = \{u' \mapsto u\}$.

No cell is labelled '0'.

Derivation of C'_{10}:

$$\dfrac{\dfrac{\dfrac{\overset{(C_3\sigma_{16})}{\vdash f(m^{u',v'}) = 1, f(m^{u',v'}) = 0} \quad \overset{(C'_8)}{f(u) = 0, f(v) = 1 \vdash}}{f(v) = 1 \vdash f(m^{u',v'}) = 1}\ \mathrm{r}(\sigma_{17}) \quad \overset{(C'_6\sigma_{18})}{f(m^{u,s(v'')}) = 1, f(v'') = 1 \vdash}}{f(v) = 1, f(v'') = 1 \vdash}\ \mathrm{r}(\sigma_{19})}{f(v) = 1 \vdash}\ \mathrm{f}(\sigma_{20}) : 1 \qquad (C'_{10})$$

where $\sigma_{16} = \{u \mapsto u', v \mapsto v'\}$, $\sigma_{17} = \{u \mapsto m^{u',v'}\}$, $\sigma_{18} = \{v \mapsto v''\}$, $\sigma_{19} = \{u' \mapsto u, v' \mapsto s(v'')\}$ and $\sigma_{20} = \{v'' \mapsto v\}$.

No cell is labelled '1'.

$$\dfrac{\dfrac{\overset{(C_3)}{\vdash f(m^{u,v}) = 1, f(m^{u,v}) = 0} \quad \overset{(C_9'\sigma_{21})}{f(u') = 0 \vdash}}{\vdash f(m^{u,v}) = 1}\,\mathrm{r}(\sigma_{22}) \quad \overset{(C_{10}'\sigma_{23})}{f(v') = 1 \vdash}}{\vdash}\,\mathrm{r}(\sigma_{24})$$

where $\sigma_{21} = \{u \mapsto u'\}$, $\sigma_{22} = \{u' \mapsto m^{u,v}\}$, $\sigma_{23} = \{v \mapsto v'\}$ and $\sigma_{24} = \{v' \mapsto m^{u,v}\}$.

The contradiction follows from the axiom that for arbitrary u and v the cell with the index $m^{u,v}$ is either labelled with '0' or with '1' in combination with the facts that no cell is labelled '0' and no cell is labelled '1'.

8.2 Orevkov's Proof

Independently, Orevkov (1982) and Statman (1979) showed a worst-case scenario of cut-elimination demonstrating that the complexity of cut-elimination in general is non-elementary (taking the number of proof nodes as the measure of complexity). The difference between the two proofs is more technical whereas Statman uses functions for formalization Orevkov uses only variables as terms.

Theorem 8.1. There exists a sequence $\{ D_k \}_{k \in \mathbb{N}}$ of sequents such that

1. there exist **LK**-proofs (with cuts) of D_k for every k which have a linear proof complexity in k,
2. while all cut-free **LK**-proofs of $D_{k>0}$ have hyper-exponential complexity.

Proof. In Orevkov (1982). □

This proof implies that there is no elementary bound on the complexity of cut-elimination.

The hyper-exponential function hyp is inductively defined by

1. $\mathrm{hyp}(0) = 1$ and
2. $\mathrm{hyp}(n+1) = 2^{\mathrm{hyp}(n)}$.

The nomenclature we are using within this proof is due to Orevkov. To facilitate understanding we can interpret them by

$$
\begin{aligned}
0 \in \mathcal{A}_0 &\leftrightarrow \mathcal{A}_0 \equiv (\forall x)(\exists y) x + 2^0 = y \\
k \in \mathcal{A}_n &\leftrightarrow (\forall z)(z \in \mathcal{A}_{n-1} \supset z + 2^k \in \mathcal{A}_{n-1})
\end{aligned}
$$

Since Orevkov's proof contains two strong quantifier occurrences in the end-sequent (independently of k) we have to introduce a constant 0 interpreted as zero and a function symbol s interpreted as the successor function. Note that we will only give the skolemized proof.

Within this section the following formula abbreviations are used (due to Orevkov's original proof):

$$
\begin{aligned}
\mathcal{A}_0(t) &= (\forall x_0)\bar{\mathcal{A}}_0(x_0, t) \\
\mathcal{A}_{i+1}(t) &= (\forall x_{i+1})(\mathcal{A}_i(x_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(x_{i+1}, t))
\end{aligned}
$$

$$
\begin{aligned}
\bar{\mathcal{A}}_0(t_1, t_2) &= (\exists y_0) P(t_1, t_2, y_0) \\
\bar{\mathcal{A}}_{i+1}(t_1, t_2) &= (\exists y_{i+1})(\mathcal{A}_i(y_{i+1}) \wedge P(t_1, t_2, y_{i+1}))
\end{aligned}
$$

$$
\mathcal{A}_0^s = (\forall x_0) P(x_0, 0, s(x_0))
$$

$$
\begin{aligned}
\mathcal{C} &= (\forall z)(\forall y)(\forall x)(\mathcal{C}_2(z, y, x) \supset P(y, z, x)) \\
\mathcal{C}_2(t_1, t_2, t_3) &= (\exists z_1)(P(z_1, 0, t_1) \wedge \mathcal{C}_1(t_2, z_1, t_3)) \\
\mathcal{C}_1(t_1, t_2, t_3) &= (\exists z_2)(P(t_1, t_2, z_2) \wedge P(z_2, t_2, t_3))
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{B}_0(t) &= (\exists y_0) P(0, t, y_0) \\
\mathcal{B}_{i+1}(t) &= (\exists y_{i+1})(P(0, t, y_{i+1}) \wedge \mathcal{B}_i(y_{i+1}))
\end{aligned}
$$

The proof of D_k is given below (starting with a proof scheme of the same name).

$D_k =$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(\bar{D}_k)}{\mathcal{A}_0^s, \mathcal{C} \vdash \boldsymbol{\mathcal{A}_k}} \quad \dfrac{\dfrac{(\tilde{D}_k)}{\boldsymbol{\mathcal{A}_0^s}, \boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{A}_k} \vdash \mathcal{B}_k}}{\underline{\boldsymbol{\mathcal{A}_k}}, \underline{\mathcal{A}_0^s}, \underline{\mathcal{C}} \vdash \mathcal{B}_k}\ \pi((132)) : \mathrm{l}}{\mathcal{A}_0^s, \boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{A}_0^s}, \mathcal{C} \vdash \mathcal{B}_k}\ \mathrm{cut}}{\boldsymbol{\mathcal{A}_0^s}, \underline{\boldsymbol{\mathcal{A}_0^s}}, \underline{\boldsymbol{\mathcal{C}}}, \mathcal{C} \vdash \mathcal{B}_k}\ \pi((23)) : \mathrm{l}}{\underline{\boldsymbol{\mathcal{A}_0^s}}, \underline{\boldsymbol{\mathcal{C}}} \vdash \mathcal{B}_k}\ \mathrm{c}(2,2) : \mathrm{l}}{\underline{\boldsymbol{\mathcal{A}_0^s} \wedge \boldsymbol{\mathcal{C}}} \vdash \boldsymbol{\mathcal{B}_k}}\ \wedge : \mathrm{l}}{\vdash \underline{(\mathcal{A}_0^s \wedge \mathcal{C}) \supset \mathcal{B}_k}}\ \supset : \mathrm{r}
$$

$\tilde{D}_0(t) =$

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{P(0,t,v_0) \vdash \boldsymbol{P(0,t,v_0)}}}{\boldsymbol{P(0,t,v_0)} \vdash \underline{(\exists y_0)P(0,t,y_0)}}\ \exists:\mathrm{r}}{\underline{(\exists \boldsymbol{y_0})\boldsymbol{P(0,t,y_0)}} \vdash \mathcal{B}_0(t)}\ \exists:\mathrm{l}}{\underline{(\forall x_0)\bar{\mathcal{A}}_0(x_0,t)} \vdash \mathcal{B}_0(t)}\ \forall:\mathrm{l}}{\underline{\mathcal{A}_0^s},\underline{\mathcal{C}},\mathcal{A}_0(t) \vdash \mathcal{B}_0(t)}\ \mathrm{w}:\mathrm{l}$$

$\tilde{D}_{i+1}(t) =$

$$\dfrac{\dfrac{\dfrac{\overset{(\bar{D}_i)}{\mathcal{A}_0^s,\mathcal{C} \vdash \boldsymbol{\mathcal{A}_i}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{P(0,t,v_{i+1}) \vdash \boldsymbol{P(0,t,v_{i+1})}} \quad \overset{(\tilde{D}_i(v_{i+1}))}{\mathcal{A}_0^s,\mathcal{C},\mathcal{A}_i(v_{i+1}) \vdash \boldsymbol{\mathcal{B}_i(v_{i+1})}}}{\boldsymbol{P(0,t,v_{i+1})},\boldsymbol{\mathcal{A}_0^s},\boldsymbol{\mathcal{C}},\boldsymbol{\mathcal{A}_i(v_{i+1})} \vdash \underline{P(0,t,v_{i+1}) \wedge \mathcal{B}_i(v_{i+1})}}\ \wedge:\mathrm{r}}{\underline{\mathcal{A}_i(v_{i+1})},\underline{P(0,t,v_{i+1})},\underline{\mathcal{A}_0^s},\underline{\mathcal{C}} \vdash \boldsymbol{P(0,t,v_{i+1}) \wedge \mathcal{B}_i(v_{i+1})}}\ \pi((1234)):\mathrm{l}}{\boldsymbol{\mathcal{A}_i(v_{i+1})},\boldsymbol{P(0,t,v_{i+1})},\mathcal{A}_0^s,\mathcal{C} \vdash \underline{(\exists y_{i+1})(P(0,t,y_{i+1}) \wedge \mathcal{B}_i(y_{i+1}))}}\ \exists:\mathrm{r}}{\underline{\boldsymbol{\mathcal{A}_i(v_{i+1}) \wedge P(0,t,v_{i+1})}},\mathcal{A}_0^s,\mathcal{C} \vdash \mathcal{B}_{i+1}(t)}\ \wedge:\mathrm{l}}{\underline{\boldsymbol{(\exists y_{i+1})(\mathcal{A}_i(y_{i+1}) \wedge P(0,t,y_{i+1}))}},\mathcal{A}_0^s,\mathcal{C} \vdash \mathcal{B}_{i+1}(t)}\ \exists:\mathrm{l}}{\underline{\boldsymbol{\mathcal{A}_i \supset \bar{\mathcal{A}}_{i+1}(0,t)}},\mathcal{A}_0^s,\mathcal{C},\mathcal{A}_0^s,\mathcal{C} \vdash \mathcal{B}_{i+1}(t)}\ \supset:\mathrm{l}}{\underline{\boldsymbol{(\forall x_{i+1})(\mathcal{A}_i(x_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(x_{i+1},t))}},\boldsymbol{\mathcal{A}_0^s},\mathcal{C},\boldsymbol{\mathcal{A}_0^s},\boldsymbol{\mathcal{C}} \vdash \mathcal{B}_{i+1}(t)}\ \forall:\mathrm{l}}{\dfrac{\underline{\boldsymbol{\mathcal{A}_0^s}},\underline{\boldsymbol{\mathcal{A}_0^s}},\boldsymbol{\mathcal{C}},\underline{\mathcal{C}},\underline{\mathcal{A}_{i+1}(t)} \vdash \mathcal{B}_{i+1}(t)}{\underline{\mathcal{A}_0^s},\underline{\mathcal{C}},\mathcal{A}_{i+1}(t) \vdash \mathcal{B}_{i+1}(t)}\ \mathrm{c}(2,2,1):\mathrm{l}}\ \pi((1245)):\mathrm{l}$$

$\varphi_0(t) =$

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{P(w_0,t,v_0) \vdash \boldsymbol{P(w_0,t,v_0)}}}{\boldsymbol{P(w_0,t,v_0)} \vdash \underline{(\exists y_0)P(w_0,t,y_0)}}\ \exists:\mathrm{r}}{\underline{(\exists \boldsymbol{y_0})\boldsymbol{P(w_0,t,y_0)}} \vdash \bar{\mathcal{A}}_0(w_0,t)}\ \exists:\mathrm{l}}{\underline{(\forall x_0)\bar{\mathcal{A}}_0(x_0,t)} \vdash \boldsymbol{\bar{\mathcal{A}}_0(w_0,t)}}\ \forall:\mathrm{l}}{\mathcal{A}_0(t) \vdash \underline{(\forall x_0)\bar{\mathcal{A}}_0(x_0,t)}}\ \forall:\mathrm{r}$$

$\varphi_{i+1}(t) =$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overset{(\varphi_i(w_{i+1}))}{\mathcal{A}_i(w_{i+1}) \vdash \boldsymbol{\mathcal{A}_i(w_{i+1})}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\overset{(\varphi_i(v_{i+1}))}{\mathcal{A}_i(v_{i+1}) \vdash \boldsymbol{\mathcal{A}_i(v_{i+1})}} \quad \overline{P(w_{i+1},t,v_{i+1}) \vdash \boldsymbol{P(w_{i+1},t,v_{i+1})}}}{\boldsymbol{\mathcal{A}_i(v_{i+1})},\boldsymbol{P(w_{i+1},t,v_{i+1})} \vdash \underline{\mathcal{A}_i(v_{i+1}) \wedge P(w_{i+1},t,v_{i+1})}}\ \wedge:\mathrm{r}}{\underline{\mathcal{A}_i(v_{i+1}) \wedge P(w_{i+1},t,v_{i+1})} \vdash \boldsymbol{\mathcal{A}_i(v_{i+1}) \wedge P(w_{i+1},t,v_{i+1})}}\ \wedge:\mathrm{l}}{\boldsymbol{\mathcal{A}_i(v_{i+1}) \wedge P(w_{i+1},t,v_{i+1})} \vdash \underline{(\exists y_{i+1})(\mathcal{A}_i(y_{i+1}) \wedge P(w_{i+1},t,y_{i+1}))}}\ \exists:\mathrm{r}}{\underline{\boldsymbol{(\exists y_{i+1})(\mathcal{A}_i(y_{i+1}) \wedge P(w_{i+1},t,y_{i+1}))}} \vdash \bar{\mathcal{A}}_{i+1}(w_{i+1},t)}\ \exists:\mathrm{l}}{\underline{\boldsymbol{\mathcal{A}_i(w_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(w_{i+1},t)}},\boldsymbol{\mathcal{A}_i(w_{i+1})} \vdash \bar{\mathcal{A}}_{i+1}(w_{i+1},t)}\ \supset:\mathrm{l}}{\underline{\boldsymbol{\mathcal{A}_i(w_{i+1})}},\underline{\mathcal{A}_i(w_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(w_{i+1},t)} \vdash \boldsymbol{\bar{\mathcal{A}}_{i+1}(w_{i+1},t)}}\ \pi((12)):\mathrm{l}}{\boldsymbol{\mathcal{A}_i(w_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(w_{i+1},t)} \vdash \underline{\mathcal{A}_i(w_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(w_{i+1},t)}}\ \supset:\mathrm{r}}{\dfrac{\underline{(\forall x_{i+1})(\mathcal{A}_i(x_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(x_{i+1},t))} \vdash \boldsymbol{\mathcal{A}_i(w_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(w_{i+1},t)}}{\mathcal{A}_{i+1}(t) \vdash \underline{(\forall x_{i+1})(\mathcal{A}_i(x_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(x_{i+1},t))}}\ \forall:\mathrm{r}}\ \forall:\mathrm{l}$$

$\bar{D}_0 =$

$$
\begin{array}{cl}
\hline
P(w_0,0,s(w_0)) \vdash P(w_0,0,s(w_0)) & \\
\hline
P(w_0,0,s(w_0)) \vdash (\exists y_0)P(w_0,0,y_0) & \exists:\mathrm{r} \\
\hline
(\forall x_0)P(x_0,0,s(x_0)) \vdash \bar{\mathcal{A}}_0(w_0,0) & \forall:\mathrm{l} \\
\hline
\mathcal{C}, \mathcal{A}_0^s \vdash \bar{\mathcal{A}}_0(w_0,0) & \mathrm{w}:\mathrm{l} \\
\hline
\mathcal{A}_0^s, \mathcal{C} \vdash \bar{\mathcal{A}}_0(w_0,0) & \pi((1,2)):\mathrm{l} \\
\hline
\mathcal{A}_0^s, \mathcal{C} \vdash (\forall x_0)\bar{\mathcal{A}}_0(x_0,0) & \forall:\mathrm{r}
\end{array}
$$

$\bar{D}_{i+1} =$

$$
\begin{array}{cl}
\overset{(\bar{D}'_{i+1})}{\mathcal{A}_i(w_{i+1}), \mathcal{C}, P(w_{i+1},0,s(w_{i+1})) \vdash \mathcal{A}_i(s(w_{i+1}))} \qquad \overline{P(w_{i+1},0,s(w_{i+1})) \vdash P(w_{i+1},0,s(w_{i+1}))} & \\
\hline
\mathcal{A}_i(w_{i+1}), \mathcal{C}, P(w_{i+1},0,s(w_{i+1})), P(w_{i+1},0,s(w_{i+1})) \vdash \mathcal{A}_i(s(w_{i+1})) \wedge P(w_{i+1},0,s(w_{i+1})) & \wedge:\mathrm{r} \\
\hline
\mathcal{A}_i(w_{i+1}), \mathcal{C}, P(w_{i+1},0,s(w_{i+1})) \vdash \mathcal{A}_i(s(w_{i+1})) \wedge P(w_{i+1},0,s(w_{i+1})) & \mathrm{c}(1,1,2):\mathrm{l} \\
\hline
\mathcal{A}_i(w_{i+1}), \mathcal{C}, P(w_{i+1},0,s(w_{i+1})) \vdash (\exists y_{i+1})(\mathcal{A}_i(y_{i+1}) \wedge P(w_{i+1},0,y_{i+1})) & \exists:\mathrm{r} \\
\hline
\mathcal{C}, P(w_{i+1},0,s(w_{i+1})) \vdash \mathcal{A}_i(w_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(w_{i+1},0) & \supset:\mathrm{r} \\
\hline
P(w_{i+1},0,s(w_{i+1})), \mathcal{C} \vdash \mathcal{A}_i(w_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(w_{i+1},0) & \pi((12)):\mathrm{l} \\
\hline
(\forall x_0)P(x_0,0,s(x_0)), \mathcal{C} \vdash \mathcal{A}_i(w_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(w_{i+1},0) & \forall:\mathrm{l} \\
\hline
\mathcal{A}_0^s, \mathcal{C} \vdash (\forall x_{i+1})(\mathcal{A}_i(x_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(x_{i+1},0)) & \forall:\mathrm{r}
\end{array}
$$

$\bar{D}'_1 =$

$$
\begin{array}{cl}
\overline{X_2 \vdash P(w_0,w_1,v_1)} \qquad \overline{X_3 \vdash P(v_1,w_1,v_0)} & \\
\hline
X_2, X_3 \vdash P(w_0,w_1,v_1) \wedge P(v_1,w_1,v_0) & \wedge:\mathrm{r} \\
\hline
\overline{X_1 \vdash X_1} \qquad X_2, X_3 \vdash (\exists z_2)(P(w_0,w_1,z_2) \wedge P(z_2,w_1,v_0)) & \exists:\mathrm{r} \\
\hline
X_1, X_2, X_3 \vdash P(w_1,0,s(w_1)) \wedge \mathcal{C}_1(w_0,w_1,v_0) & \wedge:\mathrm{r} \\
\hline
X_1, X_2, X_3 \vdash (\exists z_1)(P(z_1,0,s(w_1)) \wedge \mathcal{C}_1(w_0,z_1,v_0)) & \exists:\mathrm{r}
\end{array}
\qquad
\begin{array}{cl}
\overline{X_4 \vdash P(w_0,s(w_1),v_0)} & \\
\hline
X_4 \vdash (\exists y_0)P(w_0,s(w_1),y_0) & \exists:\mathrm{r}
\end{array}
$$

$$
\begin{array}{cl}
\hline
\mathcal{C}_2(s(w_1),w_0,v_0) \supset P(w_0,s(w_1),v_0), X_1, X_2, X_3 \vdash \bar{\mathcal{A}}_0(w_0,s(w_1)) & \supset:\mathrm{l} \\
\hline
(\forall x)(\mathcal{C}_2(s(w_1),w_0,x) \supset P(w_0,s(w_1),x)), X_1, X_2, X_3 \vdash \bar{\mathcal{A}}_0(w_0,s(w_1)) & \forall:\mathrm{l} \\
\hline
(\forall y)(\forall x)(\mathcal{C}_2(s(w_1),y,x) \supset P(y,s(w_1),x)), X_1, X_2, X_3 \vdash \bar{\mathcal{A}}_0(w_0,s(w_1)) & \forall:\mathrm{l} \\
\hline
(\forall z)(\forall y)(\forall x)(\mathcal{C}_2(z,y,x) \supset P(y,z,x)), X_1, X_2, X_3 \vdash \bar{\mathcal{A}}_0(w_0,s(w_1)) & \forall:\mathrm{l} \\
\hline
P(v_1,w_1,v_0), P(w_0,w_1,v_1), \mathcal{C}, P(w_1,0,s(w_1)) \vdash \bar{\mathcal{A}}_0(w_0,s(w_1)) & \pi((1\,4\,2\,3)):\mathrm{l} \\
\hline
(\exists y_0)P(v_1,w_1,y_0), X_2, \mathcal{C}, X_1 \vdash \bar{\mathcal{A}}_0(w_0,s(w_1)) & \exists:\mathrm{l} \\
\hline
(\forall x_0)\bar{\mathcal{A}}_0(x_0,w_1), P(w_0,w_1,v_1), \mathcal{C}, X_1 \vdash \bar{\mathcal{A}}_0(w_0,s(w_1)) & \forall:\mathrm{l} \\
\hline
P(w_0,w_1,v_1), \mathcal{A}_0(w_1), \mathcal{C}, X_1 \vdash \bar{\mathcal{A}}_0(w_0,s(w_1)) & \pi((1\,2)):\mathrm{l} \\
\hline
(\exists y_0)P(w_0,w_1,y_0), \mathcal{A}_0(w_1), \mathcal{C}, X_1 \vdash \bar{\mathcal{A}}_0(w_0,s(w_1)) & \exists:\mathrm{l} \\
\hline
(\forall x_0)\bar{\mathcal{A}}_0(x_0,w_1), \mathcal{A}_0(w_1), \mathcal{C}, X_1 \vdash \bar{\mathcal{A}}_0(w_0,s(w_1)) & \forall:\mathrm{l} \\
\hline
\mathcal{A}_0(w_1), \mathcal{A}_0(w_1), \mathcal{C}, X_1 \vdash (\forall x_0)\bar{\mathcal{A}}_0(x_0,s(w_1)) & \forall:\mathrm{r} \\
\hline
\mathcal{A}_0(w_1), \mathcal{C}, X_1 \vdash \mathcal{A}_0(s(w_1)) & \mathrm{c}(2,1,1):\mathrm{l}
\end{array}
$$

where $X_1 = P(w_1, 0, s(w_1))$, $X_2 = P(w_0, w_1, v_1)$, $X_3 = P(v_1, w_1, v_0)$ and $X_4 = P(w_0, s(w_1), v_0)$.

$\bar{D}'_{i+2} =$

$$Y_2 \vdash P(w_{i+1}, w_{i+2}, v_{i+2}) \qquad Y_3 \vdash P(v_{i+2}, w_{i+2}, v_{i+1})$$
$$\wedge : r$$
$$Y_2, Y_3 \vdash P(w_{i+1}, w_{i+2}, v_{i+2}) \wedge P(v_{i+2}, w_{i+2}, v_{i+1})$$
$$\exists : r$$
$$Y_1 \vdash Y_1 \qquad Y_2, Y_3 \vdash (\exists z_2)(P(w_{i+1}, w_{i+2}, z_2) \wedge P(z_2, w_{i+2}, v_{i+1}))$$
$$\wedge : r$$
$$Y_1, Y_2, Y_3 \vdash P(w_{i+2}, 0, s(w_{i+2})) \wedge \mathcal{C}_1(w_{i+1}, w_{i+2}, v_{i+1})$$
$$\exists : r$$
$$Y_1, Y_2, Y_3 \vdash (\exists z_1)(P(z_1, 0, s(w_{i+2})) \wedge \mathcal{C}_1(w_{i+1}, z_1, v_{i+1})) \qquad \theta_1$$
$$\supset : l$$
$$\mathcal{C}_2(s(w_{i+2}), w_{i+1}, v_{i+1}) \supset P(w_{i+1}, s(w_{i+2}), v_{i+1}), Y_1, Y_2, Y_3, \mathcal{A}_i(v_{i+1}) \vdash Y_4$$
$$\forall : l$$
$$(\forall x)(\mathcal{C}_2(s(w_{i+2}), w_{i+1}, x) \supset P(w_{i+1}, s(w_{i+2}), x)), Y_1, Y_2, Y_3, \mathcal{A}_i(v_{i+1}) \vdash Y_4$$
$$\forall : l$$
$$(\forall y)(\forall x)(\mathcal{C}_2(s(w_{i+2}), y, x) \supset P(y, s(w_{i+2}), x)), Y_1, Y_2, Y_3, \mathcal{A}_i(v_{i+1}) \vdash Y_4$$
$$\forall : l$$
$$(\forall z)(\forall y)(\forall x)(\mathcal{C}_2(z, y, x) \supset P(y, z, x)), Y_1, Y_2, Y_3, \mathcal{A}_i(v_{i+1}) \vdash Y_4$$
$$\pi((153)(24)) : l$$
$$\mathcal{A}_i(v_{i+1}), Y_3, \mathcal{C}, Y_1, Y_2 \vdash Y_4$$
$$\wedge : l$$
$$\mathcal{A}_i(v_{i+1}) \wedge P(v_{i+2}, w_{i+2}, v_{i+1}), \mathcal{C}, Y_1, Y_2 \vdash Y_4$$
$$\exists : l$$
$$\theta_2 \qquad (\exists y_{i+1})(\mathcal{A}_i(y_{i+1}) \wedge P(v_{i+2}, w_{i+2}, y_{i+1})), \mathcal{C}, Y_1, Y_2 \vdash Y_4$$
$$\supset : l$$
$$\mathcal{A}_i(v_{i+2}) \supset \bar{\mathcal{A}}_{i+1}(v_{i+2}, w_{i+2}), \mathcal{A}_i(v_{i+2}), \mathcal{C}, Y_1, Y_2 \vdash Y_4$$
$$\forall : l$$
$$(\forall x_{i+1})(\mathcal{A}_i(x_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(x_{i+1}, w_{i+2})), \mathcal{A}_i(v_{i+2}), \mathcal{C}, Y_1, Y_2 \vdash Y_4$$
$$\pi((12543)) : l$$
$$\mathcal{A}_i(v_{i+2}), Y_2, \mathcal{A}_{i+1}(w_{i+2}), \mathcal{C}, Y_1 \vdash Y_4$$
$$\wedge : l$$
$$\mathcal{A}_i(v_{i+2}) \wedge P(w_{i+1}, w_{i+2}, v_{i+2}), \mathcal{A}_{i+1}(w_{i+2}), \mathcal{C}, Y_1 \vdash Y_4$$
$$\exists : l$$
$$\theta_3 \qquad (\exists y_{i+1})(\mathcal{A}_i(y_{i+1}) \wedge P(w_{i+1}, w_{i+2}, y_{i+1})), \mathcal{A}_{i+1}(w_{i+2}), \mathcal{C}, Y_1 \vdash Y_4$$
$$\supset : l$$
$$\mathcal{A}_i(w_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(w_{i+1}, w_{i+2}), \mathcal{A}_i(w_{i+1}), \mathcal{A}_{i+1}(w_{i+2}), \mathcal{C}, Y_1 \vdash Y_4$$
$$\forall : l$$
$$(\forall x_{i+1})(\mathcal{A}_i(x_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(x_{i+1}, w_{i+2})), \mathcal{A}_i(w_{i+1}), \mathcal{A}_{i+1}(w_{i+2}), \mathcal{C}, Y_1 \vdash Y_4$$
$$\pi((12)) : l$$
$$\mathcal{A}_i(w_{i+1}), \mathcal{A}_{i+1}(w_{i+2}), \mathcal{A}_{i+1}(w_{i+2}), \mathcal{C}, Y_1 \vdash Y_4$$
$$c(1,2,1,1) : l$$
$$\mathcal{A}_i(w_{i+1}), \mathcal{A}_{i+1}(w_{i+2}), \mathcal{C}, Y_1 \vdash \bar{\mathcal{A}}_{i+1}(w_{i+1}, s(w_{i+2}))$$
$$\supset : r$$
$$\mathcal{A}_{i+1}(w_{i+2}), \mathcal{C}, Y_1 \vdash \mathcal{A}_i(w_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(w_{i+1}, s(w_{i+2}))$$
$$\forall : r$$
$$\mathcal{A}_{i+1}(w_{i+2}), \mathcal{C}, Y_1 \vdash (\forall x_{i+1})(\mathcal{A}_i(x_{i+1}) \supset \bar{\mathcal{A}}_{i+1}(x_{i+1}, s(w_{i+2})))$$

where $Y_1 = P(w_{i+2}, 0, s(w_{i+2}))$, $Y_2 = P(w_{i+1}, w_{i+2}, v_{i+2})$, $Y_3 = P(v_{i+2}, w_{i+2}, v_{i+1})$ and $Y_4 = \bar{\mathcal{A}}_{i+1}(w_{i+1}, s(w_{i+2}))$ and the proof parts θ_i are:

$\theta_1 =$

$$\begin{array}{c}(\varphi_i(v_{i+1}))\\ \mathcal{A}_i(v_{i+1}) \vdash \mathcal{A}_i(v_{i+1})\end{array} \qquad P(w_{i+1}, s(w_{i+2}), v_{i+1}) \vdash P(w_{i+1}, s(w_{i+2}), v_{i+1})$$
$$\wedge : r$$
$$\mathcal{A}_i(v_{i+1}), P(w_{i+1}, s(w_{i+2}), v_{i+1}) \vdash \mathcal{A}_i(v_{i+1}) \wedge P(w_{i+1}, s(w_{i+2}), v_{i+1})$$
$$\pi((12)) : l$$
$$P(w_{i+1}, s(w_{i+2}), v_{i+1}), \mathcal{A}_i(v_{i+1}) \vdash \mathcal{A}_i(v_{i+1}) \wedge P(w_{i+1}, s(w_{i+2}), v_{i+1})$$
$$\exists : r$$
$$P(w_{i+1}, s(w_{i+2}), v_{i+1}), \mathcal{A}_i(v_{i+1}) \vdash (\exists y_{i+1})(\mathcal{A}_i(y_{i+1}) \wedge P(w_{i+1}, s(w_{i+2}), y_{i+1}))$$

$\theta_2 =$

$$\begin{array}{c}(\varphi_i(v_{i+2}))\\ \mathcal{A}_i(v_{i+2}) \vdash \mathcal{A}_i(v_{i+2})\end{array}$$

and $\theta_3 =$

$$\begin{array}{c}(\varphi_i(w_{i+1}))\\ \mathcal{A}_i(w_{i+1}) \vdash \boldsymbol{\mathcal{A}_i(w_{i+1})}\end{array}$$

The characteristic clause set $\mathrm{CL}(D_k)$ obtained from $\Theta(D_k)$, after post-processing by means of tautology elimination and subsumption, is

$$\begin{aligned}|\Theta(D_0)| = \{ &\vdash P(w_0, 0, s(w_0)), && (C_1)\\ &P(0, 0, v_0) \vdash \} && (C_3)\end{aligned}$$

$$\begin{aligned}|\Theta(D_{k>0})| = \{ &\vdash P(w_k, 0, s(w_k)), && (C_1)\\ &P(w_{k-1}, w_k, v_k), P(v_k, w_k, v_{k-1}) \vdash P(w_{k-1}, s(w_k), v_{k-1}), && (C_2)\\ &P(0, 0, v_k), P(0, v_k, v_{k-1}), P(0, v_{k-1}, v_{k-2}), \ldots, P(0, v_1, v_0) \vdash \} && (C_3)\end{aligned}$$

Interestingly we have two clauses (C_1 and C_2) of constant size and only one additional clause with $k+1$ negative literals.

By interpretation of $P(u, v, w)$ by $u + 2^v = w$ we can informally interpret the clauses by

$$\begin{gathered}\vdash u + 2^0 = s(u)\\ u + 2^v = w', w' + 2^v = w \vdash u + 2^{s(v)} = w\\ 0 + 2^0 = u_k, 0 + 2^{u_k} = u_{k-1}, 0 + 2^{u_{k-1}} = u_{k-2}, \ldots, 0 + 2^{u_1} = u_0 \vdash\end{gathered}$$

which calculates exactly hyp.

Therefore C_3 "computes" the value of $\mathrm{hyp}(k)$ (stored in v_0). C_1 defines the successor function of addition, whereas the clause C_2 defines the successor function on exponents, which perfectly explains its absence in the case $k = 0$.

The proof projection schemes are:

$D_k(C_1) =$

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}(D_k(C_1)')\\ \mathcal{A}_0^s(0), \mathcal{C} \vdash P(w_k, 0, s(w_k))\end{array}}{\boldsymbol{\mathcal{A}_0^s(0), \mathcal{C}} \vdash P(w_k, 0, s(w_k)), \underline{\mathcal{B}_k(0)}}\ \mathrm{w:r}}{\underline{\boldsymbol{\mathcal{A}_0^s(0)}} \wedge \boldsymbol{\mathcal{C}} \vdash P(w_k, 0, s(w_k)), \boldsymbol{\mathcal{B}_k(0)}}\ \wedge:\mathrm{l}}{\vdash \boldsymbol{P(w_k, 0, s(w_k))}, \underline{\boldsymbol{(\mathcal{A}_0^s(0) \wedge \mathcal{C}) \supset \mathcal{B}_k(0)}}}\ \supset:\mathrm{r}}{\vdash \underline{(\mathcal{A}_0^s(0) \wedge \mathcal{C}) \supset \mathcal{B}_k(0)}, \underline{P(w_k, 0, s(w_k))}}\ \pi((12)):\mathrm{r}$$

$D_0(C_1)' =$

$$\dfrac{\dfrac{\dfrac{\overline{\boldsymbol{P(w_0, 0, s(w_0))} \vdash P(w_0, 0, s(w_0))}}{\underline{(\forall x_0)P(x_0, 0, s(x_0))} \vdash P(w_0, 0, s(w_0))}\ \forall:\mathrm{l}}{\underline{\mathcal{C}}, \boldsymbol{\mathcal{A}_0^s(0)} \vdash P(w_0, 0, s(w_0))}\ \mathrm{w:l}}{\underline{\mathcal{A}_0^s(0)}, \underline{\mathcal{C}} \vdash P(w_0, 0, s(w_0))}\ \pi((12)):\mathrm{l}$$

$D_{k>0}(C_1)' =$

$$
\begin{array}{ll}
P(w_k,0,s(w_k)) \vdash P(w_k,0,s(w_k)) & \\
\mathcal{C}, P(w_k,0,s(w_k)) \vdash P(w_k,0,s(w_k)) & \mathrm{w:l} \\
P(w_k,0,s(w_k)), \mathcal{C} \vdash P(w_k,0,s(w_k)) & \pi((12)):\mathrm{l} \\
(\forall x_0)P(x_0,0,s(x_0)), \mathcal{C} \vdash P(w_k,0,s(w_k)) & \forall:\mathrm{l}
\end{array}
$$

$D_k(C_2) =$

$$
\begin{array}{ll}
C_2^1 \vdash P(w_{k-1},w_k,v_k) \quad C_2^2 \vdash P(v_k,w_k,v_{k-1}) & \\
C_2^1, C_2^2 \vdash P(w_{k-1},w_k,v_k) \wedge P(v_k,w_k,v_{k-1}) & \wedge:\mathrm{r} \\
Z_1 \vdash Z_1 \quad C_2^1, C_2^2 \vdash (\exists z'')(P(w_{k-1},w_k,z'') \wedge P(z'',w_k,v_{k-1})) & \exists:\mathrm{r} \\
Z_1, C_2^1, C_2^2 \vdash P(w_k,0,s(w_k)) \wedge \mathcal{C}_1(w_{k-1},w_k,v_{k-1}) & \wedge:\mathrm{r} \\
Z_1, C_2^1, C_2^2 \vdash (\exists z')(P(z',0,s(w_k)) \wedge \mathcal{C}_1(w_{k-1},z',v_{k-1})) \quad C_2^3 \vdash C_2^3 & \exists:\mathrm{r} \\
\mathcal{C}_2(s(w_k),w_{k-1},v_{k-1}) \supset P(w_{k-1},s(w_k),v_{k-1}), Z_1, C_2^1, C_2^2 \vdash C_2^3 & \supset:\mathrm{l} \\
(\forall x)(\mathcal{C}_2(s(w_k),w_{k-1},x) \supset P(w_{k-1},s(w_k),x)), Z_1, C_2^1, C_2^2 \vdash C_2^3 & \forall:\mathrm{l} \\
(\forall y)(\forall x)(\mathcal{C}_2(s(w_k),y,x) \supset P(y,s(w_k),x)), Z_1, C_2^1, C_2^2 \vdash C_2^3 & \forall:\mathrm{l} \\
(\forall z)(\forall y)(\forall x)(\mathcal{C}_2(z,y,x) \supset P(y,z,x)), P(w_k,0,s(w_k)), C_2^1, C_2^2 \vdash C_2^3 & \forall:\mathrm{l} \\
P(w_k,0,s(w_k)), \mathcal{C}, C_2^1, C_2^2 \vdash C_2^3 & \pi((12)):\mathrm{l} \\
(\forall x_0)P(x_0,0,s(x_0)), \mathcal{C}, C_2^1, C_2^2 \vdash C_2^3 & \forall:\mathrm{l} \\
\mathcal{A}_0^s(0), \mathcal{C}, C_2^1, C_2^2 \vdash C_2^3, \mathcal{B}_k(0) & \mathrm{w:r} \\
\mathcal{A}_0^s(0) \wedge \mathcal{C}, C_2^1, C_2^2 \vdash C_2^3, \mathcal{B}_k(0) & \wedge:\mathrm{l} \\
C_2^1, C_2^2 \vdash C_2^3, (\mathcal{A}_0^s(0) \wedge \mathcal{C}) \supset \mathcal{B}_k(0) & \supset:\mathrm{r} \\
C_2^1, C_2^2 \vdash (\mathcal{A}_0^s(0) \wedge \mathcal{C}) \supset \mathcal{B}_k(0), C_2^3 & \pi((12)):\mathrm{r}
\end{array}
$$

where $C_2^1 = P(w_{k-1},w_k,v_k)$, $C_2^2 = P(v_k,w_k,v_{k-1})$, $C_2^3 = P(w_{k-1},s(w_k),v_{k-1})$ and $Z_1 = P(w_k,0,s(w_k))$.

$D_k(C_3) =$

$$
\begin{array}{ll}
(\psi_k(0)) & \\
\mathcal{A}_0^s(0), \mathcal{C}, P(0,0,v_k), P(0,v_k,v_{k-1}), P(0,v_{k-1},v_{k-2}), \ldots, P(0,v_1,v_0) \vdash \mathcal{B}_k(0) & \\
\mathcal{A}_0^s(0) \wedge \mathcal{C}, P(0,0,v_k), P(0,v_k,v_{k-1}), P(0,v_{k-1},v_{k-2}), \ldots, P(0,v_1,v_0) \vdash \mathcal{B}_k(0) & \wedge:\mathrm{l} \\
P(0,0,v_k), P(0,v_k,v_{k-1}), P(0,v_{k-1},v_{k-2}), \ldots, P(0,v_1,v_0) \vdash (\mathcal{A}_0^s(0) \wedge \mathcal{C}) \supset \mathcal{B}_k(0) & \supset:\mathrm{r}
\end{array}
$$

$\psi_0(t) =$

$$
\begin{array}{ll}
P(0,t,v_0) \vdash P(0,t,v_0) & \\
P(0,t,v_0) \vdash (\exists y_0)P(0,t,y_0) & \exists:\mathrm{r} \\
\mathcal{A}_0^s(0), \mathcal{C}, P(0,t,v_0) \vdash \mathcal{B}_0(t) & \mathrm{w:l}
\end{array}
$$

$\psi_k(t) =$

$$
\dfrac{\dfrac{\dfrac{\overline{P(0,t,v_k) \vdash \boldsymbol{P(0,t,v_k)}} \quad \overset{(\psi_{k-1}(v_k))}{\mathcal{A}_0^s(0), \mathcal{C}, P(0,v_k,v_{k-1}),\ldots,P(0,v_1,v_0) \vdash \boldsymbol{\mathcal{B}_{k-1}(v_k)}}}{\boldsymbol{P(0,t,v_k)}, \boldsymbol{\mathcal{A}_0^s(0)}, \boldsymbol{\mathcal{C}}, P(0,v_k,v_{k-1}),\ldots,P(0,v_1,v_0) \vdash \underline{P(0,t,v_k) \wedge \mathcal{B}_{k-1}(v_k)}}\ \wedge:\mathrm{r}}{\underline{\mathcal{A}_0^s(0)}, \underline{\mathcal{C}}, \underline{P(0,t,v_k)}, P(0,v_k,v_{k-1}),\ldots,P(0,v_1,v_0) \vdash \boldsymbol{P(0,t,v_k) \wedge \mathcal{B}_{k-1}(v_k)}}\ \pi((123)):\mathrm{l}}{\mathcal{A}_0^s(0), \mathcal{C}, P(0,t,v_k), P(0,v_k,v_{k-1}),\ldots,P(0,v_1,v_0) \vdash \underline{(\exists y_k)(P(0,t,y_k) \wedge \mathcal{B}_{k-1}(y_k))}}\ \exists:\mathrm{r}
$$

Bibliography

Andrews, P. B.

[1971] Resolution In Type Theory, *Journal of Symbolic Logic* **36**(3), 1971, 414–432.

Baader, F. and Snyder, W.

[2001] Unification Theory, *in* Robinson and Voronkov (2001), 445–532.

Baader, F. and Voronkov, A. (eds)

[2005] *Logic for Programming, Artificial Intelligence, and Reasoning, 11th International Conference, LPAR 2004, Montevideo, Uruguay, March 14-18, 2005, Proceedings*, vol. 3452 of *Lecture Notes in Computer Science*, Springer, 2005.

Baaz, M. and Leitsch, A.

[1994] On Skolemization and Proof Complexity, *Fundamenta Informaticae* **20**, 1994, 353–379.

[2000] Cut-Elimination and Redundancy-Elimination by Resolution, *Journal of Symbolic Computation* **29**(4), 2000, 149–176.

[2004] CERES in Many-Valued Logics, *in* Baader and Voronkov (2005), 1–20.

[2006] Towards a Clausal Analysis of Cut-Elimination, *Journal of Symbolic Computation* **41**(3-4), 2006, 381–410.

Baaz, M., Hetzl, S., Leitsch, A., Richter, C. and Spohr, H.

[2004] Cut-Elimination: Experiments with CERES, *in* Baader and Voronkov (2005), 481–495.

[2006] Proof Transformation by CERES, *in* J. Borwein and W. Farmer (eds), *MKM 2006*, Springer, 2006, to appear.

Chang, C.-L. and Lee, R. C.-T.

[1973] *Symbolic Logic and Mechanical Theorem Proving*, Computer Science and Applied Mathematics Series, Academic Press, New York, 1973.

Degtyarev, A. and Voronkov, A.

[2001] Equality Reasoning in Sequent-Based Calculi, *in* Robinson and Voronkov (2001), 611–706.

Gentzen, G.

[1934] Untersuchungen über das logische Schließen, *Mathematische Zeitschrift* **39**(1), 1934, 405–431.

Girard, J.-Y.

[1987] *Proof Theory and Logical Complexity*, Studies in Proof Theory, Bibliopolis, 1987.

Gödel, K.

[1931] Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, *Monatshefte für Mathematik und Physik* **38**, 1931, 173–198.

Herbrand, J.

[1930] *Recherches sur la théorie de la démonstration*, PhD thesis, University of Paris, 1930.

Leitsch, A.

[1997] *The Resolution Calculus*, EATCS Texts in Theoretical Computer Science, Springer Verlag, 1997.

Leitsch, A. and Richter, C.

[2005] Equational Theories in CERES, 2005, unpublished — available at `http://www.logic.at/ceres/`.

Loveland, D. W.

[1978] *Automated Theorem Proving: A Logical Basis*, EATCS Texts in Theoretical Computer Science, North-Holland, Amsterdam, 1978.

Nieuwenhuis, R. and Rubio, A.

[2001] Paramodulation-Based Theorem Proving, *in* Robinson and Voronkov (2001), 371–443.

Orevkov, V. P.

[1982] Lower Bounds for Increasing Complexity of Derivations After Cut-Elimination, *Journal of Soviet Mathematics* **20**(4), 1982, 2337–2350.

Richter, C.

[2003] *Diagonalization and Self-Application — Applications in Logic and Computer Science*, Master's thesis, Vienna University of Technology, Austria, 2003.

Robinson, J. A.

[1965] A Machine-Oriented Logic Based on the Resolution Principle, *Journal of the ACM* **12**(1), 1965, 23–41.

Robinson, J. A. and Voronkov, A. (eds)

[2001] *Handbook of Automated Reasoning (in 2 volumes)*, Elsevier and MIT Press, 2001.

Schütte, K.

[1960] *Beweistheorie*, Springer-Verlag, Berlin, 1960.

Schwichtenberg, H. and Troelstra, A. S.

[2000] *Basic Proof Theory*, 2nd ed., Cambridge University Press, 2000.

Statman, R.

[1979] Lower Bounds on Herbrand's Theorem, *Proceedings of the American Mathematical Society* **75**(1), 1979, 104–107.

Tait, W. W.

[1968] Normal Derivability in Classical Logic, *in* J. Barwise (ed.), *The Syntax and Semantics of Infinitary Languages*, vol. 72 of *Lecture Notes in Mathematics*, Springer-Verlag, Berlin, 1968, 204–236.

Takeuti, G.

[1987] *Proof Theory*, no. 81 in *Studies in Logic and the Foundations of Mathematics*, 2nd ed., North-Holland, Amsterdam, 1987.

Urban, C.

[2000] *Classical Logic and Computation*, PhD thesis, University of Cambridge Computer Laboratory, 2000.

Index

Symbols

A

C

D

E

F

G

I

L

O

P

R

S

T

Curriculum Vitae

Born in Mödling as son of Karl and Brigitte Richter on the 17th of November 1976.

Full name:	Clemens Carl Richter
Citizenship:	Austria
Marital status:	Married

Education

since 2003	Ph.D. student of Computer Sciences at Vienna University of Technology under Alexander Leitsch.
May 2003	Master degree (Diplom Ingenieur) in Computer Sciences with a concentration in artificial intelligence and theory of computer science, thesis on methods in recursion theory, advisor Alexander Leitsch; passed cum laude.
since 1998	Student of Computer Sciences at Vienna University of Technology.
June 1997	School-leaving exam ("Matura") passed with excellence (Bundeshandelsakademie Baden).
since 1987	Secondary school.
1983-1987	Elementary school.

Participation in Scientific Projects

Participation in the following projects of the Austrian Science Fund (FWF):

P17995-N12	Automated Analysis of Mathematical Proofs
P17503-N12	Skolem Functions
P16264-N05	Proof Transformation by Resolution

Related Activities

- Organization of Collegium Logicum 2005: Cut-Elimination.

- since 2004 Vice Publicity Chair of the Kurt Gödel Society

Publications

- Proof Transformation by CERES, *in* J. Borwein and W. Farmer (eds), *MKM 2006*, Springer, 2006, to appear (with M. Baaz, S. Hetzl, A. Leitsch and H. Spohr).
- Equational Theories in CERES, 2005, unpublished — available at `http://www.logic.at/ceres/` (with A. Leitsch).
- Cut-Elimination: Experiments with CERES, *in* F. Baader and A. Voronkov (eds), *LPAR*, vol. 3452 of *Lecture Notes in Computer Science*, Springer, 2004, 481–495 (with M. Baaz, S. Hetzl, A. Leitsch and H. Spohr).
- *Diagonalization and Self-Application — Applications in Logic and Computer Science*, Master's thesis, Vienna University of Technology, Austria, 2003.

Conference and Workshop Presentations

August 2005	*System Demonstration of CERES.* CERES system demonstration within the advanced course *Computational Analysis of Proofs* at the 17th European Summer School in Logic, Language and Information (ESSLLI 2005), Heriot-Watt University, Edinburgh, Scotland.
July 2005	*Cut-Elimination: Experiments with CERES.* Talk at the Collegium Logicum 2005: Cut-Elimination, Vienna University of Technology, Vienna, Austria.
December 2004	*The Cut-Elimination Program CERES.* Talk at the Paris-Vienna Workshop on Proofs, University Paris VII, Paris, France.
May 2004	*The Cut-Elimination Program CERES.* Talk at the third Moscow-Vienna Workshop on Logic and Computation, Steklov Mathematical Institute, Russian Academy of Sciences, Moscow, Russia.