FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Structural Analysis of Cut-Elimination

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Computational Intelligence

eingereicht von

Simon Peter Wolfsteiner
Matrikelnummer 0705422

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Univ.Prof. Dr.phil. Alexander Leitsch

Wien, 17.04.2015

(Unterschrift Verfasser) (Unterschrift Betreuung)

Erklärung zur Verfassung der Arbeit

Simon Peter Wolfsteiner
Stift-Schlägler-Siedlung 6a, 4150 Rohrbach

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit—einschließlich Tabellen, Karten und Abbildungen—, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum) (Unterschrift Verfasser)

The characteristic excellence of mathematics is only to be found where the reasoning is rigidly logical: the rules of logic are to mathematics what those of structure are to architecture.

— Bertrand Russell

Acknowledgements

First and foremost, I would like to express my sincerest gratitude to my advisor Professor Alexander Leitsch for providing me with this interesting topic. Not only did he spend a vast amount of time on proofreading and discussing my work but he also has the ability to point one—with great patience—in the right direction if one gets stuck on a problem.

Furthermore, I am very grateful to Stefan Hetzl who made an important observation in Chapter 6 regarding term normal forms—without his observation the results of this thesis would not be as general as they are now. I am also very thankful for the opportunity to join his research group after completion of the master's program.

Special thanks go to my colleagues Gerald Berger, Patrick Hohenecker, Benjamin Kiesl, Anela Lolic, Janos Tapolczai and Manuel Wiesinger with whom I attended the majority of lectures during the course of my studies.

Although, during the course of this thesis, I did not have that much time for activities outside the scientific realm, I would like to thank my friends for their understanding and for helping me to take my mind off things from time to time.

Last but not least, I would like to thank my family Sonja Santana-Caraballo, Markus, Fabia, Monika and Leopold Wolfsteiner for continuously supporting me in every possible way and always believing in me.

I dedicate this thesis to my mother Monika and my late father Leopold Wolfsteiner, who sadly passed away right before I entered the Computational Intelligence program.

Abstract

Proof theory—a branch of mathematical logic—is concerned with analyzing properties of proofs mathematically by treating them as formal objects. Gerhard Gentzen proved one of the major results of proof theory—the cut-elimination theorem—which states that the so-called cut-rule can be eliminated from formal proof systems in the style of the original sequent calculus **LK**. An important consequence of the cut-elimination theorem is that a cut-free proof only uses subformulas of the formulas already present in the statement to be proved (i.e. they have the so-called subformula property). In the realm of concrete mathematical proofs, the elimination of cuts corresponds to the omission of lemmas.

Gentzen's cut-elimination method is reductive in the sense that it performs local proof rewriting steps on small parts of a proof. The method **CERES** (cut-elimination by resolution) constitutes an alternative cut-elimination approach that—as opposed to reductive methods—takes the global structure of a proof into account by analyzing all cuts simultaneously. Roughly speaking, **CERES** extracts an unsatisfiable set of clauses that encodes the structure of a proof containing cuts. A resolution refutation of this set of clauses then serves as a skeleton for a proof containing at most atomic cuts.

Due to a result by Baaz and Leitsch it is known that **CERES** simulates reductive cut-elimination methods up to the elimination of non-atomic cuts. In this thesis, we prove a new simulation result in order to positively answer the question whether the simulation also includes the elimination of atomic cuts. To this end, we define a specific indexed resolution method (similar to the method of atomic cut-linkage for swapped clause sets by Bruno Woltzenlogel Paleo) and prove its completeness—using a new resolution method for clause terms—with respect to special characteristic clause sets obtained by **CERES**.

The obtained result can play a crucial role in the completeness proof of **CERES** for intuitionistic logic and provide a partial answer to the conjecture posed by Giselle Reis whether **CERES** in conjunction with indexed resolution and the method of joining projections yields an intuitionistic proof.

Kurzfassung

Die Beweistheorie – ein Teilgebiet der mathematischen Logik – betrachtet Beweise als formale Objekte und untersucht deren Eigenschaften mit Hilfe mathematischer Methoden. Der Schnitteliminationssatz – einer der wichtigsten Sätze der Beweistheorie – welcher von Gerhard Gentzen bewiesen wurde, besagt, dass die sogenannte Schnittregel ohne Weiteres aus einem formalen Beweissystem in der Art des Sequentialkalküls **LK** entfernt werden kann. Eine wichtige Eigenschaft schnittfreier Beweise ist die Tatsache, dass solche Beweise nur Teilformeln jener Formeln enthalten, welche bereits im zu beweisenden Satz enthalten sind (d. h. sie besitzen die sogenannte Teilformel-Eigenschaft). Betrachtet man konkrete mathematische Beweise, so entspricht die Schnittelimination dem Entfernen von Hilfssätzen (Lemmata) aus solchen Beweisen.

Bei der Gentzenschen Schnitteliminationsmethode handelt es sich um eine reduktive Methode, da sie lokale Beweistransformationen an einem kleinen Teil des gesamten Beweises durchführt. Die **CERES**-Methode (cut-elimination by resolution) stellt hingegen einen alternativen Ansatz dar, indem sie – im Gegensatz zu reduktiven Methoden – durch die gleichzeitige Analyse aller Schnitte die globale Struktur eines Beweises berücksichtigt. Grob gesagt extrahiert **CERES** eine widerlegbare Klauselmenge, welche die Struktur eines Beweises mit Schnitten repräsentiert. Eine Resolutionswiderlegung ebendieser Klauselmenge dient in weiterer Folge als Skelett für einen Beweis, welcher höchstens atomare Schnitte enthält.

Baaz und Leitsch konnten zeigen, dass die **CERES**-Methode die reduktiven Ansätze bis zu jenem Punkt simulieren kann, an dem nur noch atomare Schnitte im Beweis vorhanden sind. In der vorliegenden Arbeit wird ein neues Simulationsresultat bewiesen, welches die bisherige Simulation auf die Elimination atomarer Schnitte ausweitet. Zu diesem Zweck wird eine spezielle indizierte Resolutionsmethode definiert (ähnlich der von Bruno Woltzenlogel Paleo eingeführten “atomic cut-linkage”-Methode für sogenannte “swapped clause sets”) und ihre Vollständigkeit – unter Zuhilfenahme einer neuen Klauselterm-Resolutionsmethode – für eine gewisse Klasse von charakteristischen Klauselmengen (welche durch Anwendung der **CERES**-Methode erhalten wurden) bewiesen.

Das erzielte Hauptresultat könnte eine wichtige Rolle im Beweis der Vollständigkeit der **CERES**-Methode für intuitionistische Logik einnehmen. Weiters kann damit eine Teilantwort auf die von Giselle Reis ausgesprochene Vermutung gegeben werden, ob **CERES** in Verbindung mit indizierter Resolution und einer weiteren Methode namens “joining projections” einen intuitionistischen Beweis liefert.

Contents

1 Introduction **1**
1.1 Structure of the Thesis . . . 3

2 Preliminaries **5**
2.1 First-order Logic . . . 5
2.2 Sequent Calculus . . . 12
2.3 Resolution Calculus . . . 21

3 The Problem of Cut-Elimination **27**
3.1 Motivation . . . 27
3.2 Cut-Elimination Theorem & Consequences . . . 28
3.3 Reductive Cut-Elimination . . . 30

4 Cut-Elimination by Resolution **45**
4.1 Motivation & Overview . . . 45
4.2 Clause Terms . . . 47
4.3 The Method CERES . . . 54

5 Complexity Analysis of CERES **63**
5.1 Canonic Resolution Refutations . . . 63
5.2 Characteristic Terms and Cut-Reduction . . . 66
5.3 Speed-up Results . . . 67

6 A More General Analysis of Characteristic Clause Sets and Cut-Elimination **73**
6.1 Term Resolution . . . 73
6.2 Indexed Resolution . . . 83
6.3 Completeness and All That . . . 105

7 Conclusion **121**
7.1 Applications . . . 123
7.2 Possible Future Work . . . 123

Bibliography **125**

CHAPTER 1

Introduction

In mathematics, it is of utmost importance to demonstrate that mathematical statements are valid by means of *proof*, i.e. a chain of reasoning starting from assumed statements or a priori truths following strict logical rules that eventually lead to a particular conclusion [33].

Proof theory—a branch of mathematical logic—treats proofs as formal, mathematical objects and investigates their properties and structure by mathematical means [47]. The advent of proof theory can be traced back to Hilbert's attempt—known as *Hilbert's Program*—to formalize mathematics in axiomatic systems and to prove their consistency by finitary means [47, 52].

In 1931, Hilbert's original endeavour was, however, shattered by Gödel's celebrated *incompleteness theorems* [25]. Roughly speaking, Gödel's *first incompleteness theorem* states that in any sufficiently strong formal system of arithmetic (that is also consistent), there are true propositions, which are not provable within this system [13, 25, 28]. Moreover, the *second incompleteness theorem* shows that the proposition stating the consistency of a sufficiently strong formal system of arithmetic is not provable within this system, provided that the system is indeed consistent [25, 28, 47].

In his seminal papers *Untersuchungen über das logische Schließen I+II*, Gentzen introduced the *sequent calculi* **LK** and **LJ** for classical and intuitionistic logic, respectively [21]. Amongst others, sequent calculi contain the so-called *cut-rule*, which allows the use of lemmas (i.e. intermediary statements) in proofs. The main result of the paper was the "Hauptsatz" (or *cut-elimination theorem*), which basically states that any theorem of first-order logic can be proved without detours, i.e. without the use of instances of the cut-rule [47]. Cut-free proofs have the so-called *subformula property*, i.e. all formulas used in the proof are (instances of) subformulas of the statement to be proved [9, 35]. The subformula property is of great importance, as it implies the consistency of both **LK** and **LJ**. Indeed, if there would be a proof of the empty sequent (roughly speaking, a false statement without subformulas), it would be provable without using the cut-rule at all—this, however, is impossible by the subformula property [47].

The method of cut-elimination is by no means restricted to proof theory in an abstract sense (e.g. consistency proofs), but it can actually be applied to concrete mathematical proofs. A famous example of such an application is Girard's analysis (see [24]) of Fürstenberg and Weiss' topological proof [19] of van der Waerden's theorem [49] on partitions. It turned out that van der Waerden's original elementary proof was the result of applying cut-elimination to the proof of Fürstenberg and Weiss [35].

Inspired by the idea of fully automating cut-elimination on concrete mathematical proofs in order to obtain new interesting elementary proofs, Baaz and Leitsch introduced a new cut-elimination method based on resolution called **CERES**[1] (*cut-elimination by resolution*) [7, 35]. The method **CERES** takes the global structure of an **LK**-proof into account, whereas reductive methods (e.g. Gentzen's method) only operate on small parts of an **LK**-proof. **CERES** roughly works as follows: extract the *characteristic clause set*, which is an unsatisfiable set of clauses encoding the structure of a proof that contains cuts. With the help of a first-order theorem prover, obtain a *resolution refutation* γ of the characteristic clause set, which will then serve as a skeleton for a proof ψ containing at most atomic cuts. The resolution refutation γ is then transformed into ψ by replacing its leaves by so-called *projections* (i.e. cut-free parts of the original proof) [10, 35].

In [9], it was shown that the characteristic clause set of an **ACNF** (atomic cut normal form, i.e. a proof containing at most atomic cuts) under **CERES** always subsumes the characteristic clause set of an **ACNF** under reductive methods. This means the characteristic clause sets obtained from proofs after reductive cut-elimination are redundant w.r.t. the one obtained from the original proof. In other words, **CERES** simulates reductive methods up to the elimination of non-atomic cuts.

A natural question that arises in this context is whether **CERES** still simulates reductive methods if we eliminate the atomic cuts from the **ACNF** as well. This thesis serves as a recipe towards answering the question positively. The main ingredients are the resolution refinement of *indexed resolution* (cf. the method of *atomic cut-linkage* for *swapped clause sets* by Bruno Woltzenlogel Paleo [51]) and a new method called *term resolution*.

Starting from a proof with atom indexing (i.e. each atomic subformula of each cut-formula has a different index), we consider a corresponding **ACNF** under reductive methods in which all atomic cuts have been shifted to the top (= $\mathbf{ACNF}^{\mathbf{top}}$). As a consequence, the resulting characteristic clause set contains indexed clauses of a specific form. Then we show—with term resolution as an auxiliary means—that indexed resolution is complete w.r.t. clause sets in this specific form. Furthermore, we show that eliminating atomic cuts by reductive methods from a proof in $\mathbf{ACNF}^{\mathbf{top}}$ amounts to a specific form of indexed resolution on the corresponding characteristic clause sets. Finally, it will turn out that the characteristic clause set of the original proof still subsumes the one of the proof obtained after reductive elimination of atomic cuts.

Subsequently, our main result can play a crucial role in the completeness proof of the **CERES** method for intuitionistic logic. In particular, the result provides a partial answer to the conjecture posed by Giselle Reis (see [40]) whether **CERES** in conjunction with

[1] An implementation of **CERES** is available at `http://www.logic.at/ceres/`.

indexed resolution and the method of joining projections indeed yields an intuitionistic proof.

1.1 Structure of the Thesis

After this introductory chapter—in an attempt to make this thesis as self-contained as possible and to fix notation and terminology—we present the basic notions and definitions in Chapter 2.

In Chapter 3, we give a general overview on the problem of cut-elimination and its most important consequences. The chapter is then concluded with the definition of a proof rewriting system for cut-elimination based on Gentzen's constructive proof of the cut-elimination theorem.

Chapter 4 serves the purpose to introduce the cut-elimination method **CERES** (cut-elimination by resolution) and to prove some of its most important properties.

We will take a closer look at the computational complexity of the method **CERES** in Chapter 5. By comparing **CERES** and reductive methods, it will turn out that the former can simulate the latter. As a consequence, **CERES** has a nonelementary speed-up over reductive methods.

Chapter 6 is devoted to the proof of our main result, namely that **CERES** still simulates reductive methods if we include the elimination of atomic cuts. This is done by defining a new method that performs some sort of resolution on the syntax of clause terms. With the help of this method, we will show that the resolution refinement of indexed resolution is complete w.r.t. a certain class of clause sets and that each atomic cut-elimination step can be simulated by indexed resolution on the corresponding characteristic clause sets.

This thesis is then concluded in Chapter 7 by summarizing the main results of the thesis, presenting its most important applications and giving an outlook on possible future work.

CHAPTER 2

Preliminaries

We assume that the reader is familiar with the basic concepts of classical first-order logic. Nevertheless—to fix notation and terminology—the following chapter will give a short overview on the logical notions that will be used throughout this thesis. In Section 2.1, we will introduce the syntax and semantics of classical first-order logic, followed by our formulation of the sequent calculus **LK** in Section 2.2. Finally, in Section 2.3, we will introduce the resolution calculus, which will be needed in order to define the method **CERES**.

2.1 First-order Logic

Syntax

Definition 2.1.1 is based on [34].

Definition 2.1.1 (Language). The language $\mathcal{L}$ of classical first-order logic consists of the following elements:

- a countably infinite set of individual variables V,
- a countably infinite set of constant symbols CS,
- a countably infinite set of function symbols $FS = \bigcup_{i=1}^{\infty} FS_i$, where all sets FS_i, for $i \geq 1$, are countably infinite (FS_i is the set of i-ary function symbols),
- a countably infinite set of predicate symbols $PS = \bigcup_{i=1}^{\infty} PS_i$, where all sets PS_i, for $i \geq 1$, are countably infinite (PS_i is the set of i-ary predicate symbols),
- the logical connectives $\wedge$, $\vee$, $\neg$ and $\rightarrow$,
- the quantifiers $\forall, \exists$,
- $\top$ (verum) and $\bot$ (falsum).

$\triangle$

In the following (unless stated otherwise), we will use the following notational conventions [34]:

- Variables: $x, y, z, u, v, w, x_1, y_1, \dots$
- Constant symbols: $a, b, c, d, e, a_1, b_1, \dots$
- Function symbols: $f, g, h, f_1, g_1, \dots$
- Predicate symbols: $P, Q, R, P_1, Q_1, \dots$

The following definition of terms is taken from [34, Definition 2.1.1].

Definition 2.1.2 (Term). The set of terms T is inductively defined as follows:

(i) $V \subseteq \mathsf{T}$ (variables are terms),

(ii) $CS \subseteq \mathsf{T}$ (constant symbols are terms),

(iii) If $t_1, \dots, t_n \in \mathsf{T}$ and $f \in FS_n$ with $n \geq 1$, then $f(t_1, \dots, t_n) \in \mathsf{T}$.

(iv) No other objects are terms.

$\triangle$

Statements like in point (iv) "No other objects are ..." will henceforth be omitted and will be considered as included in the concept "definition" [34].

The following definitions are based on [34]:

If $t = f(t_1, \dots, t_n)$, for an $f \in FS_n$ and terms $t_1, \dots, t_n$, then t is called a *functional term*; the terms t_i are called the *arguments* of t. Variables and constant symbols have no arguments.

The occurrence of terms can be defined inductively: A term s *occurs* in a term t if either $s = t$ or s occurs in an argument of t.

The set of all variables occurring in a term t is denoted by $V(t)$. A term t with $V(t) = \emptyset$ is called a *ground term*.

Definition 2.1.3 is based on [34, Definition 2.1.3].

Definition 2.1.3 (Formula). The set of first-order logic formulas PL is inductively defined as follows:

(i) If $P \in PS_n$, where $n \geq 1$ and $t_1, \dots, t_n \in \mathsf{T}$, then $P(t_1, \dots, t_n) \in \mathsf{PL}$,

(ii) $\top \in \mathsf{PL}$ and $\bot \in \mathsf{PL}$,

(iii) If $A \in \mathsf{PL}$, then $\neg A \in \mathsf{PL}$,

(iv) If $A, B \in \mathsf{PL}$, then $A \wedge B \in \mathsf{PL}$,

(v) If $A, B \in \mathsf{PL}$, then $A \vee B \in \mathsf{PL}$,

(vi) If $A, B \in \mathsf{PL}$, then $A \to B \in \mathsf{PL}$,

(vii) If $A \in \mathsf{PL}$ and $x \in V$, then $(\forall x)A \in \mathsf{PL}$,

(viii) If $A \in \mathsf{PL}$ and $x \in V$, then $(\exists x)A \in \mathsf{PL}$.

$\triangle$

We follow [34] by defining:

Formulas obtained by (i) are called *atomic formulas* or *atoms*, and the $t_1, \ldots, t_n$ are called the *arguments* of $P(t_1, \ldots, t_n)$. Let A be a formula such that $A = A_1 \odot A_2$, $A = \neg B$, or $A = (Qx)B$, for $\odot \in \{\wedge, \vee, \to\}, x \in V$ and $Q \in \{\forall, \exists\}$. Then A_1, A_2 and B are called *immediate subformulas* of A.

A formula A *occurs* in a formula B, if either $A = B$ or A occurs in an immediate subformula of B. A is called a *subformula* of B if A occurs in B. In (vii) and (viii) A and all terms occurring in A as well as all its subformulas are said to be *in the scope of* $(\forall x)$ and $(\exists x)$, respectively.

Let s be a term and A be an atomic formula. Then s occurs in A, if s occurs in an argument of A. If A is an arbitrary formula, then s occurs in A if it occurs in some subformula of A.

Example 2.1.4 (cf. [34], Example 2.1.1)**.** Let $f \in FS_1$ and $x, y \in V$, then $f(x), f(y) \in \mathsf{T}$. If $P \in PS_2$, then $P(x, y)$ and $P(f(x), y)$ are atomic formulas. Thus, $(P(x, y) \vee P(f(x), y)) \in \mathsf{PL}$ and $((\forall x)(\forall y)(P(x, y) \vee P(f(x), y))) \in \mathsf{PL}$. $\triangle$

Definition 2.1.5 is taken from [34, Definition 2.1.4].

Definition 2.1.5 (Free and Bounded Occurrences of Variables). Let A be an atomic formula and x be a variable occurring in A, then x occurs *free* in A. If x occurs free in A and B is of the form $A \odot C$, $C \odot A$, $\neg A$, or $(Qy)A$ (for $\odot \in \{\wedge, \vee, \to\}, y \neq x, Q \in \{\forall, \exists\}$), then x occurs free in B.

x occurs *bounded* in A if there exists a subformula of A of the form $(Qx)B$ such that x occurs in B. $\triangle$

A formula without free variables is called *closed* or a *sentence*. If a formula does not contain bounded variables, it is called *open* [34].

Example 2.1.6 (cf. [34], Example 2.1.2)**.** Let $A = P(x) \to (\exists x)(\forall y)R(x, y)$ be a formula. Then x occurs both free and bounded in A; y only occurs bounded in A. x occurs free in the subformula $(\forall y)R(x, y)$. The subformula $(\exists x)(\forall y)R(x, y)$ is a sentence whereas the subformula $R(x, y)$ is open. $\triangle$

The following definition is taken from [34]:

Definition 2.1.7 (Universal Closure). If A is an open formula containing the free variables $x_1, \ldots, x_n$, then $(\forall x_1), \ldots, (\forall x_n)A$ is called the *universal closure* of A. $\triangle$

The universal closure is not unique, as the order of the variables is not fixed. However, all closures are semantically equivalent [34].

Definition 2.1.8 is based on [9, Definition 2.1].

Definition 2.1.8 (Position). We inductively define positions within terms as follows:

(i) If $t \in V$ or $t \in CS$, then 0 is a position in t and $t.0 = t$.

(ii) Let $t = f(t_1, \ldots, t_n)$, where $f \in FS_n$ and $t_1, \ldots, t_n \in \mathsf{T}$. Then 0 is a position in t and $t.0 = t$. Let $\mu : (0, k_1, \ldots, k_l)$ be a position in a t_j (for $1 \leq j \leq n$) and $t_j.\mu = s$, then $\nu : (0, j, k_1, \ldots, k_l)$ is a position in t and $t.\nu = s$.

$\triangle$

The following is taken from [9]:

Positions serve the purpose to locate subterms in a term and to perform replacements on subterms. A subterm s of t is just a term with $t.\nu = s$, for some position ν in t. Let $t.\nu = s$, then $t[r]_\nu$ is the term t after replacement of s on position ν by r; in particular $t[r]_\nu.\nu = r$. Let P be a set of positions in t, then $t[r]_P$ is defined from t by replacing all $t.\nu$ with $\nu \in P$ by r.

Positions within formulas can be defined in the same way (e.g. consider all formulas as terms).

Remark. Considering the syntax tree of a term, positions can be viewed as paths in the tree to the occurrence of the respective subterms.

Example 2.1.9 (cf. [10], Example 3.1.2)**.** Let $t = f(g(x), h(x, y), c)$ be a term. Then

$$\begin{array}{rcl}
t.0 & = & t, \\
t.(0,1) & = & g(x), \\
t.(0,2) & = & h(x,y), \\
t.(0,3) & = & c, \\
t.(0,1,1) & = & x, \\
t.(0,2,1) & = & x, \\
t.(0,2,2) & = & y, \\
t[g_1(d)]_{(0,2,1)} & = & f(g(x), h(g_1(d), y), c).
\end{array}$$

$\triangle$

Definition 2.1.10 is based on [34, Definition 2.1.10] and [9].

Definition 2.1.10 (Substitution). A *substitution* is a mapping σ of type $V \rightarrow \mathsf{T}$ such that $\sigma(v) \neq v$ for only finitely many $v \in V$.

If σ is a substitution, then the set $\{v \mid v \in V, \sigma(v) \neq v\}$ is called the *domain* of σ (notation: $dom(\sigma)$).

The set $\{\sigma(v) \mid v \in dom(\sigma)\}$ is called the *range* of σ (notation: $rg(\sigma)$).

If σ is a substitution with $\sigma(x_i) = t_i$, for $x_i \neq t_i$ $(1 \leq i \leq n)$ and $\sigma(v) = v$, for $v \notin \{x_1, \ldots, x_n\}$, then we denote σ by $\{x_1 \leftarrow t_1, \ldots, x_n \leftarrow t_n\}$. Substitutions are written in postfix notation, i.e. we write $F\sigma$ instead of $\sigma(F)$. $\triangle$

Substitutions can be extended to terms, atoms and formulas in a homomorphic way [10].

Definition 2.1.11 ([10], Definition 3.1.4). A substitution σ is called *more general* than a substitution ϑ (denoted by $\sigma \leq_s \vartheta$) if there exists a substitution μ such that $\vartheta = \sigma\mu$. $\triangle$

Example 2.1.12 ([10], Example 3.1.3)**.** Let $\vartheta = \{x \leftarrow a, y \leftarrow a\}$ and $\sigma = \{x \leftarrow y\}$. Then $\sigma\mu = \vartheta$, for $\mu = \{y \leftarrow a\}$, and thus $\sigma \leq_s \vartheta$. Note that for the identical substitution we get $\emptyset \leq_s \tau$, for all substitutions τ. $\triangle$

Let $F \in \mathsf{T}$ or $F \in \mathsf{PL}$. Then we write $F(x)$ to indicate (potential) free occurrences of the variable x in F. Let t be an arbitrary term, then $F(x/t)$ stands for $F[t]_P$, where $P = \{\nu \mid F.\nu = x\}$ [9].

Definition 2.1.13 is taken from [9, Definition 2.2].

Definition 2.1.13 (Complexity of Formulas). If $F \in \mathsf{PL}$, then the complexity $comp(F)$ is the number of logical symbols occurring in F. Formally we define:

- $comp(F) = 0$ if F is an atomic formula,
- $comp(F) = 1 + comp(A)$ if F is of the form $\neg A$ or $(Qx)A$, for $Q \in \{\forall, \exists\}$, $x \in V$,
- $comp(F) = 1 + comp(A) + comp(B)$ if F is of the form $A \odot B$, for $\odot \in \{\wedge, \vee, \rightarrow\}$.

$\triangle$

Example 2.1.14. Let $A = P(x) \rightarrow (\exists x)(\forall y)R(x,y)$. Then $comp(A) = 1 + comp(P(x)) + comp((\exists x)(\forall y)R(x,y)) = 1 + 0 + (1 + comp((\forall y)R(x,y))) = 2 + (1 + comp(R(x,y))) = 3 + 0 = 3$. $\triangle$

Semantics

Having laid down the syntax of first-order logic, we are now able to define its semantics. The semantical key concept is that of an interpretation [34, Definition 2.1.6].

Definition 2.1.15 (Interpretation). An *interpretation* of a formula $F \in \mathsf{PL}$ is a triple $\mathcal{M} = (D, \Phi, I)$ having the following properties:

(i) D is a nonempty set, called the *domain* of $\mathcal{M}$.

(ii) Φ is a mapping defined on $CS(F) \cup FS(F) \cup PS(F)$ such that

   (a) $\Phi(c) \in D$, for $c \in CS(F)$.

   (b) $\Phi(f)\colon D^n \rightarrow D$, for $f \in FS_n(F)$.

   (c) $\Phi(P) \subseteq D^n$, for $P \in PS_n(F)$ (i.e. $\Phi(P)$ is an n-ary predicate over D).

(iii) $I\colon V \rightarrow D$; I is called the *environment* or *variable assignment*.

$\triangle$

Interpretations $\mathcal{M}$ are the basis for the interpretation functions $u_{\mathcal{M}}$ for terms and $v_{\mathcal{M}}$ for formulas [34].

The following definition is based on [34].

Definition 2.1.16. Let $F \in \mathsf{PL}$ and $\mathcal{M}$ be an interpretation of F, then we define the interpretation function $u_{\mathcal{M}} \colon T(F) \to D$ by

$$\begin{array}{rcll} u_{\mathcal{M}}(x) & = & I(x) & \text{for } x \in V, \\ u_{\mathcal{M}}(c) & = & \Phi(c) & \text{for } c \in CS(F) \text{ and} \\ u_{\mathcal{M}}(f(t_1, \ldots, t_n)) & = & \Phi(f)(u_{\mathcal{M}}(t_1), \ldots, u_{\mathcal{M}}(t_n)) & \text{for } f(t_1, \ldots, t_n) \in T(F), \end{array}$$

where $T(F)$ denotes the set of terms occurring in F. △

In order to define an interpretation function for quantified formulas, we require the concept of variable-equivalence of interpretations [34].

Definition 2.1.17 is based on [34, Definition 2.1.7]

Definition 2.1.17 (Equivalence of Interpretations). Two interpretations $\mathcal{M}$ and $\mathcal{M}'$ of a formula F are called *equivalent modulo* $x_1, \ldots, x_k$ if there are D, Φ, I, J such that $\mathcal{M} = (D, \Phi, I)$, $\mathcal{M}' = (D, \Phi, J)$ and $I(v) = J(v)$ for $v \in V \setminus \{x_1, \ldots, x_k\}$ (i.e. I and J differ at most on some of the x_i). If $\mathcal{M}$ is equivalent to $\mathcal{M}$' modulo x, we write $\mathcal{M} \sim_x \mathcal{M}'$. △

Equivalent interpretations have the same domain, and they interpret constant, function and predicate symbols in the same way, but they may differ on a finite set of variables [34].

Now, we are ready to define the evaluation of formulas in $\mathsf{PL}(F)$ via an interpretation $\mathcal{M}$, where $\mathsf{PL}(F)$ denotes the set of formulas over the language of F [34].

Definition 2.1.18 (cf. [13, 34]). Let $F \in \mathsf{PL}$ and $\mathcal{M} = (D, \Phi, I)$ be an interpretation of F.

$v_{\mathcal{M}} \colon \mathsf{PL}(F) \to \{\textbf{true}, \textbf{false}\}$ is defined inductively over the structure of formulas in $\mathsf{PL}(F)$:

(i) If A is an atomic formula in $\mathsf{PL}(F)$ and $A = P(t_1, \ldots, t_n)$, then $v_{\mathcal{M}}(A) = \textbf{true}$ if and only if $(u_{\mathcal{M}}(t_1), \ldots, u_{\mathcal{M}}(t_n)) \in \Phi(P)$.

(ii) $v_{\mathcal{M}}(\top) = \textbf{true}$ and $v_{\mathcal{M}}(\bot) = \textbf{false}$.

(iii) $v_{\mathcal{M}}(\neg A) = \textbf{true}$ iff $v_{\mathcal{M}}(A) = \textbf{false}$

(iv) $v_{\mathcal{M}}(A \land B) = \textbf{true}$ iff $v_{\mathcal{M}}(A) = \textbf{true}$ and $v_{\mathcal{M}}(B) = \textbf{true}$.

(v) $v_{\mathcal{M}}(A \lor B) = \textbf{true}$ iff $v_{\mathcal{M}}(A) = \textbf{true}$ or $v_{\mathcal{M}}(B) = \textbf{true}$.

(vi) $v_{\mathcal{M}}(A \to B) = \textbf{true}$ iff $v_{\mathcal{M}}(A) = \textbf{false}$ or $v_{\mathcal{M}}(B) = \textbf{true}$.

(vii) $v_{\mathcal{M}}((\forall x)A) = \textbf{true}$ iff for all $\mathcal{M}'$ such that $\mathcal{M} \sim_x \mathcal{M}'$ we have $v_{\mathcal{M}'}(A) = \textbf{true}$.

(viii) $v_{\mathcal{M}}((\exists x)A) = \textbf{true}$ iff for some $\mathcal{M}'$ such that $\mathcal{M} \sim_x \mathcal{M}'$ we have $v_{\mathcal{M}'}(A) = \textbf{true}$.

where $A, B \in \mathsf{PL}(F)$. △

An interpretation $\mathcal{M}$ of A *verifies* A if $v_{\mathcal{M}}(A) = \textbf{true}$; if $v_{\mathcal{M}}(A) = \textbf{false}$, then we say that $\mathcal{M}$ *falsifies* A [34].

The definition of a model is based on [34, Definition 2.1.8].

Definition 2.1.19 (Model). Let A be a formula containing the free variables $x_1, \ldots, x_n$ and $\mathcal{M}$ be an interpretation of A. Then $\mathcal{M}$ is called a *model* of A if all $\mathcal{M}$' that are equivalent to $\mathcal{M}$ modulo $x_1, \ldots, x_n$ verify A. If A is closed, then $\mathcal{M}$ is a model of A iff $\mathcal{M}$ verifies A.

We denote that $\mathcal{M}$ is a model of A by $\mathcal{M} \models A$. △

Example 2.1.20 (cf. [34], Example 2.1.4)**.** Let $F = (\forall x)(P(x,a) \to Q(x, f(a)))$ be a formula in PL. Furthermore, let $\mathcal{M} = (\mathbb{N}, \Phi, I)$ be an interpretation such that $\mathbb{N}$ is the set of natural numbers, $I(x) = 7$ and Φ is defined as follows:

$$\begin{aligned} \Phi(a) &= 0, \\ \Phi(f)(n) &= n + 1, \text{ for all } n \in \mathbb{N}, \\ \Phi(P) &= \leq, \\ \Phi(Q) &= < . \end{aligned}$$

We define $\mathcal{M}_x^* = \{\mathcal{M}' \mid \mathcal{M} \sim_x \mathcal{M}\}$ and compute $v_{\mathcal{M}}(F)$:

$v_{\mathcal{M}}(F) = \textbf{true}$ iff

$$\text{for all } \mathcal{M}' \in \mathcal{M}_x^*\colon v_{\mathcal{M}'}(P(x,a) \to Q(x, f(a))) = \textbf{true}.$$

iff

$$\text{for all } \mathcal{M}' \in \mathcal{M}_x^*\colon v_{\mathcal{M}'}(P(x,a)) = \textbf{false} \text{ or } v_{\mathcal{M}'}(Q(x, f(a))) = \textbf{true}.$$

iff

$$\text{for all } J \sim_x I\colon J(x) > u_{\mathcal{M}'}(a) \text{ or } J(x) < u_{\mathcal{M}'}(f(a)).$$

iff

$$\text{for all } k \in \mathbb{N}\colon k > 0 \text{ or } k < 0 + 1.$$

Because for $k > 0$ the left-hand side and for $k = 0$ the right-hand side (as $0 < 1$) of the "or" holds, we get that $\mathcal{M}$ is indeed a model of F, i.e. $\mathcal{M} \models F$. △

Definition 2.1.21 is an extension of [34, Definition 2.1.9] also containing the definition of unsatisfiability.

Definition 2.1.21 ((Un)satisfiability and Validity). Let $F, G \in \text{PL}$ be arbitrary. Then

- F is called *satisfiable* if F has a model.
- F is called *unsatisfiable* if F is *not* satisfiable.

- F is called *valid* if *every* interpretation of F is a model of F.
- F and G are *logically equivalent* (denoted by $F \equiv G$) if F and G have exactly the same models.
- F and G are called *satisfiability-equivalent* (short: sat-equivalent) if F is satisfiable iff G is satisfiable; we write $F \equiv_{sat} G$.

$\triangle$

Note that with respect to $\equiv_{sat}$ there are only two equivalence classes, namely the satisfiable and the unsatisfiable formulas [34].

Now that we have defined the syntax and semantics of classical first-order logic, we are in the position to introduce a formal proof system.

2.2 Sequent Calculus

The term sequent calculus refers to a formal proof system in the style of Gentzen's original sequent calculi **LK** (logistischer klassischer Kalkül) and **LJ** (logistischer intuitionistischer Kalkül) for classical first-order and intuitionistic logic, respectively [48]. Gentzen's motivation for the definition of his sequent calculi was the fact that they allowed him to investigate properties of the calculi of *natural deduction* in an easier and more elegant way. One of the most important results of this investigation was the so-called "Hauptsatz" (or cut-elimination theorem) [21].

Sequent calculi consist of sets of axioms and inference rules that are applied to so-called sequents. We will define such a sequent calculus for classical first-order logic in the following section.

The definition of sequents is based on [10, Definition 3.1.7] and [9, Definition 2.3].

Definition 2.2.1 (Sequent). Let Γ and Δ be finite (possibly empty) multisets of PL-formulas. Then the expression $S = \Gamma \vdash \Delta$ is called a *sequent*. Γ is called the *antecedent* of S and Δ the *consequent* of S. $\vdash$ is called the *empty sequent*. $\triangle$

Two sequents $\Gamma_1 \vdash \Delta_1$ and $\Gamma_2 \vdash \Delta_2$ are considered equal if $\Gamma_1 = \Gamma_2$ and $\Delta_1 = \Delta_2$ [9].

Multiset union within sequents is denoted by comma: if $S = \Gamma \vdash \Delta$, where Γ is the multiset union of Γ_1, Γ_2 and Δ is the multiset union of Δ_1, Δ_2, then we write $S = \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2$. If $A \in$ PL, then A^n denotes the multiset containing A n times. For instance, we may write $\vdash A^3$ for $\vdash A, A, A$ [9].

Definition 2.2.2 is based on [10, Definition 3.1.8].

Definition 2.2.2 (Semantics of Sequents). Let $S = A_1, \ldots, A_n \vdash B_1, \ldots, B_m$ be a sequent. Then the semantics of S can be expressed by the corresponding PL-formula

$$F(S) = \bigwedge_{i=1}^{n} A_i \to \bigvee_{j=1}^{m} B_j.$$

Let $\mathcal{M}$ be an interpretation. Then $\mathcal{M}$ is an interpretation of S if $\mathcal{M}$ is an interpretation of $F(S)$. If $n = 0$ (i.e. the antecedent of S is empty), we assign $\top$ to $\bigwedge_{i=1}^{n} A_i$; if $m = 0$ (i.e. the consequent of S is empty), we assign $\bot$ to $\bigvee_{j=1}^{m} B_j$. Thus, the formula corresponding to the empty sequent $\vdash$ is given by $\top \to \bot$, which is equivalent to $\bot$.
We say that S is *true* in $\mathcal{M}$ if $F(S)$ is true in $\mathcal{M}$. S is called *valid* if $F(S)$ is valid. △

Example 2.2.3 (cf. [10], Example 3.1.5). Let $S = Q(a), (\forall x)(\neg Q(x) \lor Q(f(x))) \vdash Q(f(a))$ be a sequent. The corresponding formula is given by

$$F(S)\colon (Q(a) \land (\forall x)(\neg Q(x) \lor Q(f(x)))) \to Q(f(a)).$$

Since $F(S)$ is a valid formula, S is a valid sequent. △

The following definition is taken from [10, Definition 3.1.9].

Definition 2.2.4 (Atomic Sequent). A sequent $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$ is called *atomic* if the A_i, B_j are atomic formulas. △

Definition 2.2.5 and Definition 2.2.6 are based on [9, Definition 2.4] and [9, Definition 2.5], respectively.

Definition 2.2.5 (Composition of Sequents). Let $S = \Gamma \vdash \Delta$ and $S' = \Pi \vdash \Lambda$. We define the *composition* of S and S' by $S \circ S'$, where $S \circ S' = \Gamma, \Pi \vdash \Delta, \Lambda$. △

Definition 2.2.6 (Subsequent). Let S, S' be sequents. We say that S' is a *subsequent* of S, denoted by $S' \sqsubseteq S$, if there exists a sequent S'' with $S' \circ S'' = S$. △

By definition of the semantics of sequents, every sequent is implied by all of its subsequents. The empty sequent (which stands for $\bot$) implies every sequent [10].

Example 2.2.7 (cf. [10], Example 3.1.6). Let S be defined as in Example 2.2.3. Then $S' = (\forall x)(\neg Q(x) \lor Q(f(x))) \vdash$ is a subsequent of S. S'' has to be defined as $Q(a) \vdash Q(f(a))$. Then

$$S' \circ S'' = (\forall x)(\neg Q(x) \lor Q(f(x))), Q(a) \vdash Q(f(a))$$

which is equal to S by the definition of sequents via multisets. △

Definition 2.2.8 ([10], Definition 3.1.14). Substitution can be extended to sequents in an obvious way. If $S = A_1, \ldots, A_n \vdash B_1, \ldots, B_m$ and σ is a substitution, then

$$S\sigma = A_1\sigma, \ldots, A_n\sigma \vdash B_1\sigma, \ldots, B_m\sigma.$$

△

The Calculus LK

The notion of an axiom set is defined analogous to [10, Definition 3.2.1].

Definition 2.2.9 (Axiom Set). A (possibly infinite) set $\mathcal{A}$ of 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 $S\sigma \in \mathcal{A}$. If $\mathcal{A}$ consists of atomic sequents only, we speak about an *atomic axiom set*. $\triangle$

Remark. The closure under substitution is required for proof transformations, in particular for cut-elimination [10].

We follow [10, Definition 3.2.2] when defining the standard axiom set:

Definition 2.2.10 (Standard Axiom Set). Let $\mathcal{A}_T$ be the smallest axiom set containing all sequents of the form $A \vdash A$, for arbitrary atomic formulas A. $\mathcal{A}_T$ is called the *standard axiom set*. $\triangle$

Definition 2.2.11 is based on [47, Definition 2.1].

Definition 2.2.11 (Inference Rule). An *inference rule* is an expression of the form

$$\frac{S_1}{S} \qquad \text{or} \qquad \frac{S_1 \qquad S_2}{S},$$

where S_1, S_2 and S are sequents. S_1 and S_2 are called the *upper sequents* (or premises), and S is called the *lower sequent* (or conclusion) of the inference rule. Rules with a single premise are called *unary*; those with two premises are called *binary*. $\triangle$

Intuitively, this means that whenever S_1 (S_1 and S_2) is (are) asserted, we can infer S from it (from them) [47].

The following definition of the sequent calculus **LK** is based on [9, Definition 2.6] and [10, Definition 3.2.3].

Definition 2.2.12 (LK). Basically, we use Gentzen's version of **LK** (as introduced in [21]) adapted to the multiset structure for sequents. For simplification, we do not include implication: as we consider classical logic only, there exists a polynomial cut-homomorphic transformation translating arbitrary **LK**-proofs into proofs in negation normal form (see [6]). Due to the definition of sequents via multisets, we do not need the exchange rules.

In the rules of **LK** we always label the auxiliary formulas (i.e. the formulas in the premis(es) used for the inference) and the principal (i.e. the inferred) formula using different symbols. Thus, in our definition, $\wedge$-introduction to the right takes the form

$$\frac{\Gamma \vdash A^+, \Delta \qquad \Gamma \vdash \Delta, B^+}{\Gamma \vdash A \wedge B^*, \Delta}\, \wedge_r.$$

We usually avoid markings by putting the auxiliary formulas at the leftmost position in the antecedent of sequents and in the rightmost position in the consequent of sequents. The principal formula is mostly identifiable by the context. Thus, the above rule will be written as

$$\frac{\Gamma \vdash \Delta, A \qquad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B}\, \wedge_r.$$

The logical rules of **LK** are the following:

- $\wedge$-introduction:

$$\frac{A, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta}\wedge_{l_1} \qquad \frac{B, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta}\wedge_{l_2} \qquad \frac{\Gamma \vdash \Delta, A \qquad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B}\wedge_r$$

- $\vee$-introduction:

$$\frac{A, \Gamma \vdash \Delta \qquad B, \Gamma \vdash \Delta}{A \vee B, \Gamma \vdash \Delta}\vee_l \qquad \frac{\Gamma \vdash \Delta, A}{\Gamma \vdash \Delta, A \vee B}\vee_{r_1} \qquad \frac{\Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \vee B}\vee_{r_2}$$

- $\neg$-introduction:

$$\frac{\Gamma \vdash \Delta, A}{\neg A, \Gamma \vdash \Delta}\neg_l \qquad \frac{A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg A}\neg_r$$

- $\forall$-introduction:

$$\frac{A(x/t)\Gamma \vdash \Delta}{(\forall x)A(x), \Gamma \vdash \Delta}\forall_l \qquad \frac{\Gamma \vdash \Delta, A(x/y)}{\Gamma \vdash \Delta, (\forall x)A(x)}\forall_r$$

  where t is an arbitrary term containing only free variables, and y in $\forall_r$ is a free variable which may not occur in Γ, Δ. y is called an *eigenvariable*.

- $\exists$-introduction:

$$\frac{A(x/y), \Gamma \vdash \Delta}{(\exists x)A(x), \Gamma \vdash \Delta}\exists_l \qquad \frac{\Gamma \vdash \Delta, A(x/t)}{\Gamma \vdash \Delta, (\exists x)A(x)}\exists_r$$

  where t is an arbitrary term containing only free variables, and y in $\exists_l$ is a free variable which may not occur in Γ, Δ. y is called an *eigenvariable*.

The structural rules of **LK** are the following:

- weakening (A is an arbitrary formula):

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, A}w_r \qquad \frac{\Gamma \vdash \Delta}{A, \Gamma \vdash \Delta}w_l$$

- contraction (A is an arbitrary formula):

$$\frac{A, A, \Gamma \vdash \Delta}{A, \Gamma \vdash \Delta}c_l \qquad \frac{\Gamma \vdash \Delta, A, A}{\Gamma \vdash \Delta, A}c_r$$

- Let us assume that the formula A occurs both in Δ and Π, then the *cut-rule* is defined as follows:

$$\frac{\Gamma \vdash \Delta \qquad \Pi \vdash \Lambda}{\Gamma, \Pi^* \vdash \Delta^*, \Lambda}cut(A)$$

  where Δ^* and Π^* are Δ and Π after deletion of at least one occurrence of A. A is the auxiliary formula of $cut(A)$ and it is also called the *cut-formula*. If A does not occur in Π^*, Δ^*, then the cut is called a *mix*.

△

Definition 2.2.13 ([10], Definition 3.2.4). Let

$$\frac{S_1 \qquad S_2}{S}\ \xi$$

be a binary inference rule of **LK**, and let S', S_1', S_2' be instantiations of the schema variables in S, S_1, S_2. Then (S_1', S_2', S') is called an *instance* of ξ. The instance of a unary rule is defined analogously. △

Example 2.2.14 ([10], Example 3.2.1). Consider the rule

$$\frac{\Gamma \vdash \Delta, A^+ \qquad \Gamma \vdash \Delta, B^+}{\Gamma \vdash \Delta, (A \wedge B)^*}\ \wedge_r.$$

Then

$$\frac{(\forall x)P(x), (\forall x)Q(x) \vdash P(a)^+ \qquad (\forall x)P(x), (\forall x)Q(x) \vdash Q(b)^+}{(\forall x)P(x), (\forall x)Q(x) \vdash (P(a) \wedge Q(b))^*}\ \wedge_r$$

is an instance of $\wedge_r$. △

The following definition is taken from [9, Definition 2.7].

Definition 2.2.15 (LK-derivation). We define an **LK**-*derivation* as a finite directed tree, where the nodes are occurrences of sequents and the edges are defined according to the inference rule applications in **LK** (they are directed from the root to the leaves). The root is the occurrence of the *end-sequent*. The leaves must be occurrences of atomic sequents of the form $A \vdash A$, i.e. elements of the standard axiom set $\mathcal{A}_T$.

Let $\mathcal{A}$ be the set of sequents occurring at the leaves of an **LK**-derivation ψ and S be the sequent occurring at the root (i.e. the end-sequent). Then we say that ψ is an **LK**-derivation of S from $\mathcal{A}$ (notation $\mathcal{A} \vdash_{LK} S$). Note that, in general, complete cut-elimination is only possible in **LK**-derivations, where the leaves are axioms.

We write

$$\begin{matrix}(\psi)\\ S\end{matrix}$$

to express that ψ is a derivation with end-sequent S. △

We follow [10, Definition 3.2.9] for the definition of paths:

Definition 2.2.16 (Path). Let $\pi\colon \mu_1, \ldots, \mu_n$ be a sequence of nodes in an **LK**-derivation ψ such that for all $i \in \{1, \ldots, n-1\}$, (μ_i, μ_{i+1}) is an edge in ψ. Then π is called a *path* from μ_1 to μ_n in ψ of *length* $n-1$ (denoted by $lp(\pi) = n-1$). If $n = 1$ and $\pi = \mu_1$, then π is called a *trivial path*. π is called a *branch* if μ_1 is the root of ψ and μ_n is a leaf in ψ. We use the terms *predecessor* and *successor* contrary to the direction of edges in the tree: if there exists a path from μ_1 to μ_2, then μ_2 is called a *predecessor* of μ_1. The successor relation is defined in an analogous way, e.g. every initial sequent is a predecessor of the end-sequent. △

Definition 2.2.17 is taken from [9, Definition 2.8].

Definition 2.2.17 (Subderivation). A position ν in an **LK**-derivation is defined in the same way as for terms (formally, we may consider a derivation as a term). Here the positions can be identified with the nodes in the derivation tree. If there exists a position ν with $\varphi.\nu = \psi$ (where ν is a node in φ), then we call ψ a *subderivation* of φ. In the same way, we write $\varphi[\rho]_\nu$ for the deduction φ after the replacement of $\varphi.\nu$ by ρ at the position ν in φ. The sequent occurring at the position ν is denoted by $S(\nu)$. △

Example 2.2.18 (cf. [10]). Let φ be the **LK**-derivation

$$\dfrac{\dfrac{\nu_1\colon P(a) \vdash P(a)}{\nu_2\colon (\forall x)P(x) \vdash P(a)}\,\forall_l \qquad \dfrac{\nu_3\colon P(a) \vdash P(a)}{\nu_4\colon P(a) \vdash (\exists x)P(x)}\,\exists_r}{\nu_5\colon (\forall x)P(x) \vdash (\exists x)P(x)}\,cut(P(a)).$$

The ν_i denote the nodes in φ. The leaf nodes (or initial sequents) are ν_1 and ν_3. ν_5 denotes the end-sequent. In practice, the representation of nodes is omitted when writing down **LK**-proofs.

ν_5, ν_4, ν_3 is a path (and also a branch) in φ. ν_3 is predecessor of $\nu 5$, but ν_3 is not a predecessor of ν_2.

$\varphi.\nu_2$ is the subderivation

$$\dfrac{\nu_1\colon P(a) \vdash P(a)}{\nu_2\colon (\forall x)P(x) \vdash P(a)}\,\forall_l.$$

Let ρ be the **LK**-derivation

$$\dfrac{\dfrac{\nu_6\colon P(a) \vdash P(a), P(a)}{\nu_7\colon (\forall x)P(x) \vdash P(a), P(a)}\,\forall_l.}{\nu_8\colon (\forall x)P(x) \vdash P(a)}\,c_r$$

Then $\varphi[\rho]_{\nu_2}$ is the **LK**-derivation

$$\dfrac{\dfrac{\dfrac{\nu_6\colon P(a) \vdash P(a), P(a)}{\nu_7\colon (\forall x)P(x) \vdash P(a), P(a)}\,\forall_l.}{\nu_8\colon (\forall x)P(x) \vdash P(a)}\,c_r \qquad \dfrac{\nu_3\colon P(a) \vdash P(a)}{\nu_4\colon P(a) \vdash (\exists x)P(x)}\,\exists_r}{\nu_5\colon (\forall x)P(x) \vdash (\exists x)P(x)}\,cut(P(a)).$$

Moreover, $\varphi[\rho]_{\nu_2}$ is an **LK**-derivation from the axiom set

$$\mathcal{A} = \{P(a) \vdash P(a), P(a); P(a) \vdash P(a)\}.$$

△

The following two definitions correspond to [10, Definition 3.2.11] and [10, Definition 3.2.12], respectively.

Definition 2.2.19 (Depth). Let φ be an **LK**-derivation and ν be a node in φ. Then the *depth* of ν (denoted by $depth(\nu)$) is defined as the maximal length of a path from ν to a leaf of $\varphi.\nu$. The depth of any leaf in φ is 0. △

Definition 2.2.20 (Regularity). An **LK**-derivation φ is called *regular* if

- all eigenvariables of quantifier introductions $\forall_r$ and $\exists_l$ are mutually different.
- If an eigenvariable y occurs as an eigenvariable in a proof node ν, then y occurs only above ν in the proof tree.

△

There exists a straightforward transformation from **LK**-derivations into regular ones: just rename the eigenvariables in different subderivations (for details we refer to [47]) [9].

Remark. From now on, we assume, without mentioning the fact explicitly, that all considered **LK**-derivations are regular.

The following notions are taken from [9]:

The formulas in sequents on the branch of a deduction tree are connected by a so-called *ancestor relation*. Indeed, if A occurs in a sequent S and A is marked as principal formula of a, let us say binary, inference on the sequents S_1, S_2, then the auxiliary formulas in S_1, S_2 are *immediate ancestors* of A (in S). If A occurs in S_1 and it is not an auxiliary formula of an inference, then A occurs also in S; in this case, A in S_1 is also an immediate ancestor of A in S. The case of unary rules is analogous. General ancestors are defined via the reflexive and transitive closure of the relation.

Let ν be a node in φ, and let S' be a subsequent of $S(\mu)$, for a successor μ of ν. Then we write $S(\nu, (S', \mu))$ for the subsequent S consisting of formulas which are ancestors of formulas in S' (at μ). Let Ω be a set of (S', μ) with $S' \sqsubseteq S(\mu)$, for a successor μ of ν, then $S(\nu, \Omega)$ is the composition of all $S(\nu, \omega)$, for $\omega \in \Omega$. $S(\nu, \Omega)$ is just the subsequent of S consisting of ancestors of some of the formulas in some successors μ.

If Ω consists just of the cut-formulas of cuts which occur "below" ν, then $S(\nu, \Omega)$ is the subsequent consisting of all formulas which are ancestors of a cut. These subsequents are crucial for the definition of the characteristic clause set and thus for the method **CERES** (see Chapter 4).

Definition 2.2.21 ([9], Definition 2.10). The *length* of a proof ω is defined as the number of nodes in ω, and it is denoted by $l(\omega)$. △

Example 2.2.22 (cf. [10], Example 3.2.9)**.** Let φ be the **LK**-derivation from Example 2.2.18, and let α and β be the left and right occurrence of the cut formula in φ, respectively.

Let $\Omega = \{\alpha, \beta\}$, then we have

$$\begin{array}{rcl} S(\nu_1, \Omega) & = & \vdash P(a), \\ S(\nu_3, \Omega) & = & P(a) \vdash. \end{array}$$

φ has length $l(\varphi) = 5$. △

Definition 2.2.23 constitutes an adaptation of [9, Definition 2.11].

Definition 2.2.23 (Cut/Mix Derivation). Let ψ be an **LK**-derivation of the form

$$\frac{\begin{array}{c}(\psi_1) \\ \Gamma_1 \vdash \Delta_1, A\end{array} \qquad \begin{array}{c}(\psi_2) \\ A, \Gamma_2 \vdash \Delta_2\end{array}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\ cut(A).$$

Then ψ is called a *cut-derivation*. If the cut is a mix, we speak about a *mix-derivation*. Let ψ be a mix-derivation. Then we define the *grade* of ψ (denoted by $grade(\psi)$) as $comp(A)$; the *left-rank* of ψ (denoted by $rank_l(\psi)$) is the maximal number of nodes in a branch in ψ_1 such that A occurs in the consequent of a predecessor of $\Gamma_1 \vdash \Delta_1$. If A is "produced" in the last inference of ψ_1, then the left-rank of ψ is 1. The *right-rank* (denoted by $rank_r(\psi)$) is defined in an analogous way. The *rank* of ψ is the sum of left-rank and right-rank, i.e. $rank(\psi) = rank_l(\psi) + rank_r(\psi)$. △

Example 2.2.24 (cf. [10]). Let ψ be the **LK**-derivation

$$\frac{\dfrac{\dfrac{\nu_1\colon P(a) \vdash P(a)}{\nu_2\colon (\forall x)P(x) \vdash P(a)}\ \forall_l \qquad \dfrac{\nu_3\colon P(a) \vdash P(a)}{\nu_4\colon P(a) \vdash (\exists x)P(x)}\ \exists_r}{\nu_5\colon (\forall x)P(x) \vdash (\exists x)P(x)}\ cut(P(a))}{\nu_6\colon (\forall x)P(x) \vdash (\exists x)P(x), Q(b)}\ w_r.$$

Then the only cut-derivation in ψ is φ:

$$\frac{\dfrac{\nu_1\colon P(a) \vdash P(a)}{\nu_2\colon (\forall x)P(x) \vdash P(a)}\ \forall_l \qquad \dfrac{\nu_3\colon P(a) \vdash P(a)}{\nu_4\colon P(a) \vdash (\exists x)P(x)}\ \exists_r}{\nu_5\colon (\forall x)P(x) \vdash (\exists x)P(x)}\ cut(P(a)).$$

The grade of φ is 0, as the cut-formula $P(a)$ is atomic.

Moreover, $rank_l(\varphi) = 2$, $rank_r(\varphi) = 2$ and $rank(\varphi) = rank_l(\varphi) + rank_r(\varphi) = 4$.

△

We use an adapted version of [10, Definition 6.4.3], since our formulation of **LK** is based on sequents defined via multisets of formulas.

Definition 2.2.25 (Context Product). Let C be a sequent and φ be an **LK**-derivation such that no free variable in C occurs as eigenvariable in φ. We define the left context product $C \star \varphi$ of C and φ (which gives a proof of $C \circ S'$) inductively:

- If φ consists only of the root node ν and $S(\nu) = S'$, then $C \star \varphi$ is a proof consisting only of a node μ such that $S(\mu) = C \circ S'$.

- Assume that φ is of the form

$$\dfrac{\begin{matrix}(\varphi')\\ S''\end{matrix}}{S'}\,\xi,$$

where ξ is a unary rule. Assume also that the **LK**-derivation $C \star \varphi'$ of $C \circ S''$ is already defined. Then we define $C \star \varphi$ as

$$\dfrac{\begin{matrix}(C \star \varphi')\\ C \circ S''\end{matrix}}{C \circ S'}\,\xi.$$

$C \star \varphi$ is also well-defined for the rules $\forall_r$ and $\exists_l$, as C does not contain free variables, which are eigenvariables in φ.

- Assume that φ is of the form

$$\dfrac{\begin{matrix}(\varphi_1)\\ S_1\end{matrix} \qquad \begin{matrix}(\varphi_2)\\ S_2\end{matrix}}{S'}\,\xi,$$

where $C \star \varphi_1$ is a proof of $C \circ S_1$ and $C \star \varphi_2$ is a proof of $C \circ S_2$. Then we define the proof $C \star \varphi$ as

$$\dfrac{\dfrac{\begin{matrix}(C \star \varphi_1)\\ C \circ S_1\end{matrix} \qquad \begin{matrix}(C \star \varphi_2)\\ C \circ S_2\end{matrix}}{S''}\,\xi}{C \circ S'}\,s^*$$

Note that if ξ is the cut-rule, restoring the context after application of ξ might require weakening; otherwise s^* stands for applications of contractions.

The right context product $\varphi \star C$ is defined analogously. $\triangle$

Example 2.2.26 (cf. [10], Example 6.4.2)**.** Let φ be the proof

$$\dfrac{\dfrac{\dfrac{\dfrac{R(a) \vdash R(a)}{R(a) \vdash Q(a), R(a)}\,w_r}{\neg R(a), R(a) \vdash Q(a)}\,\neg_l \qquad \dfrac{\dfrac{Q(a) \vdash Q(a) \qquad Q(a) \vdash Q(a)}{Q(a) \vdash Q(a)}\,cut(Q(a))}{Q(a), R(a) \vdash Q(a)}\,w_l}{\neg R(a) \vee Q(a), R(a) \vdash Q(a)}\,\vee_l}{(\forall x)(\neg R(x) \vee Q(x)), R(a) \vdash Q(a)}\,\forall_l$$

and $C = P(y) \vdash Q(y)$. Then the context product $C \star \varphi$ is:

$$
\begin{prooftree}
\AxiomC{$P(y), R(a) \vdash Q(y), R(a)$}
\RightLabel{w_r}
\UnaryInfC{$P(y), R(a) \vdash Q(y), Q(a), R(a)$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P(y), \neg R(a), R(a) \vdash Q(y), Q(a)$}
\AxiomC{(φ_1)}
\noLine
\UnaryInfC{$P(y), Q(a) \vdash Q(y), Q(y), Q(a)$}
\RightLabel{c_r}
\UnaryInfC{$P(y), Q(a) \vdash Q(y), Q(a)$}
\RightLabel{w_l}
\UnaryInfC{$P(y), Q(a), R(a) \vdash Q(y), Q(a)$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P(y), \neg R(a) \vee Q(a), R(a) \vdash Q(y), Q(a)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$P(y), (\forall x)(\neg R(x) \vee Q(x)), R(a) \vdash Q(y), Q(a)$}
\end{prooftree}
$$

where φ_1 is

$$
\begin{prooftree}
\AxiomC{$P(y), Q(a) \vdash Q(y), Q(a)$}
\AxiomC{$P(y), Q(a) \vdash Q(y), Q(a)$}
\RightLabel{$cut(Q(a))$}
\BinaryInfC{$P(y), P(y), Q(a) \vdash Q(y), Q(y), Q(a)$}
\RightLabel{c_l}
\UnaryInfC{$P(y), Q(a) \vdash Q(y), Q(y), Q(a)$}
\end{prooftree}
$$

△

2.3 Resolution Calculus

In 1965, the resolution calculus was introduced by Robinson in his seminal paper *"A Machine-Oriented Logic Based on the Resolution Principle"* [42] and represented an improvement of the works of Gilmore [23] as well as that of Davis and Putnam [16]. As opposed to traditional, "human-oriented" calculi, like **LK**, the "machine-oriented" resolution calculus was specifically designed as a theoretical basis to be used in automated theorem proving[1]. Robinson's principle lead to enormous improvements in performance over prior methods [34]. The resolution calculus is a so-called refutation calculus, i.e. the goal is not to prove that a statement is a theorem, but rather to refute it.

Our formulation of the resolution calculus is based on sets of specific sequents (called clauses) and uses most general unification as well as the rules of resolution, contraction and weakening.

We follow [9, Definition 2.12] by defining:

Definition 2.3.1 (Clause). A *clause* is an atomic sequent, i.e. a sequent of the form $\Gamma \vdash \Delta$, where Γ and Δ are multisets of atomic formulas. △

Remark. Clauses are usually defined as disjunctions of *literals*. A literal is either an atom or a negated atom.

Definition 2.3.2 is taken from [10, Definition 3.3.1].

Definition 2.3.2 ((Most General) Unifier). Let $\mathcal{A}$ be a nonempty set of atoms and σ be a substitution. σ is called a *unifier* of $\mathcal{A}$ if the set $\mathcal{A}\sigma$ contains only one element. σ is called a *most general unifier* (abbreviated as m.g.u.) of $\mathcal{A}$ if σ is a unifier of $\mathcal{A}$ and, for all unifiers τ of $\mathcal{A}$, it holds that $\sigma \leq_s \tau$. △

[1]A research field concerned with algorithmic methods for proving theorems (of e.g. classical first-order logic) [20].

Example 2.3.3 ([10], Example 3.3.1)**.** Let $\mathcal{A} = \{P(x, f(y)), P(x, f(x)), P(x', y')\}$ and

$$\sigma = \{y \leftarrow x, x' \leftarrow x, y' \leftarrow f(x)\},$$
$$\sigma_t = \{x \leftarrow t, y \leftarrow t, x' \leftarrow t, y' \leftarrow f(t)\}.$$

All substitutions σ, σ_t are unifiers of $\mathcal{A}$. Moreover, we see that the unifier σ plays an exceptional role. Indeed,

$$\sigma\{x \leftarrow t\} = \sigma_t, \text{ i.e. } \sigma \leq_s \sigma_t.$$

It is easy to verify that for all unifying substitutions ϑ (including those with $dom(\vartheta) \setminus V(\mathcal{A}) \neq \emptyset$) we obtain $\sigma \leq_s \vartheta$. σ is more general than all other unifiers of $\mathcal{A}$, i.e. it is indeed "most" general. However, σ is not the only most general unifier; for the unifier $\lambda = \{y \leftarrow x', x \leftarrow x', y' \leftarrow f(x')\}$, we get

$$\lambda \leq_s \sigma, \sigma \leq_s \lambda \text{ and } \lambda \leq_s \vartheta, \text{ for all unifiers } \vartheta \text{ of } \mathcal{A}.$$

$\triangle$

The following theorem corresponds to [34, Theorem 2.6.1]:

Theorem 2.3.4 (Unification Theorem). *There exists a decision procedure UAL for the unifiability of two terms. In particular, the following two properties hold:*

(i) *If $\{t_1, t_2\}$ is not unifiable, then UAL stops with failure.*

(ii) *If $\{t_1, t_2\}$ is unifiable, then UAL stops and ϑ (the final substitution constructed by UAL) is a most general unifier of $\{t_1, t_2\}$.*

PROOF. See the proof of Theorem 2.6.1 in [34]. $\square$

The definitions of resolvents and p-resolvents are based on [9, Definition 2.13] and [9, Definition 2.14], respectively.

Definition 2.3.5 (Resolvent). Let C and D be clauses of the form

$$\begin{array}{rcl} C & = & \Gamma \vdash \Delta, A_1, \ldots, A_m, \\ D & = & B_1, \ldots, B_n, \Pi \vdash \Lambda \end{array}$$

such that C and D are variable-disjoint, $n, m \geq 1$, and let σ be a most general unifier of $\{A_1, \ldots, A_m, B_1, \ldots, B_n\}$. Then the clause

$$\Gamma\sigma, \Pi\sigma \vdash \Delta\sigma, \Lambda\sigma$$

is called a *resolvent* of C and D. The resolution rule can thus be represented as follows:

$$\frac{\Gamma \vdash \Delta, A_1, \ldots, A_m \qquad B_1, \ldots, B_n, \Pi \vdash \Lambda}{\Gamma\sigma, \Pi\sigma \vdash \Delta\sigma, \Lambda\sigma}\, R.$$

$\triangle$

Definition 2.3.6 (P-resolvent). Let $C = \Gamma \vdash \Delta, A^m$ and $D = A^n, \Pi \vdash \Lambda$ be clauses[2] with $n, m \geq 1$. Then the clause

$$\Gamma, \Pi \vdash \Delta, \Lambda$$

is called a *p-resolvent* of C and D. $\triangle$

Remark. Note that the p-resolution rule is nothing else than atomic cut [9].

The following definition is based on [34, Definition 2.6.5].

Definition 2.3.7 (Permutation Substitution). A substitution σ is called a *permutation (substitution)* if σ is injective and $rg(\sigma) \subseteq V$. A permutation σ is called a *renaming* of a clause C if the set of variables occurring in C and $rg(\sigma)$ are disjoint. $\triangle$

We follow [10, Definition 3.3.12] by defining:

Definition 2.3.8 (Variant). Let C be a clause, and let π be a permutation substitution (i.e. π is a binary function $V \to V$). Then $C\pi$ is called a *variant* of C. $\triangle$

The following definition is based on [10, Definition 3.3.8].

Definition 2.3.9 (Contraction Normalization). Let $C = \Gamma \vdash \Delta$ be a clause. A *contraction normalization* of C is a clause D obtained from C by omitting multiple occurrences of atoms in Γ and Δ. $\triangle$

Definition 2.3.10 is taken from [10, Definition 3.3.9].

Definition 2.3.10 (Factor). Let $C = \Gamma \vdash \Delta$ be a clause and D be a nonempty subclause of $\Gamma \vdash$ or of $\vdash \Delta$, and let σ be an m.g.u.of the atoms of D. Then a contraction normalization of $C\sigma$ is called a *factor* of C. $\triangle$

Our notion of resolution deduction is based on [9, Definition 2.15] and [10, Definition 3.3.13].

Definition 2.3.11 (Resolution Deduction). A deduction tree having clauses as leaves and resolution, contraction and weakening as rules is called a *resolution deduction*. If, instead of resolution, we have p-resolution as (the only binary) rule, then we call the deduction a *p-resolution deduction*. Let γ be a p-resolution deduction in which all clauses are variable-free. Then we call γ a *ground resolution deduction*. Let $\mathcal{C}$ be a set of clauses. If all leaves in γ are variants of clauses in $\mathcal{C}$ and D is the clause at the root of the deduction tree, then γ is called a *resolution deduction of D from $\mathcal{C}$*. If $D = \vdash$, then γ is called a *resolution refutation of $\mathcal{C}$*. $\triangle$

Remark. A p-resolution deduction γ is an **LK**-deduction with atomic sequents and structural rules only, i.e. the only rules in γ are cut, contraction and weakening [9].

Definition 2.3.12 is taken from [10, Definition 3.3.14].

[2] Note that, here, A^m and A^n are not indexed clauses (see Chapter 6); m, n just denote the number of occurrences of A in the respective clause.

Definition 2.3.12 (Ground Projection). Let γ' be a ground resolution deduction which is an instance of a resolution deduction γ. Then γ' is called a *ground projection* of γ. $\triangle$

Example 2.3.13 ([10], Example 3.3.7). Let

$$\mathcal{C} = \{\vdash P(x), P(a);\ P(y) \vdash P(f(y));\ P(f(f(a))) \vdash\}.$$

Then the following derivation γ is a resolution refutation of $\mathcal{C}$:

$$\cfrac{\cfrac{\cfrac{\vdash P(x), P(a) \qquad P(y) \vdash P(f(y))}{\vdash P(f(a))}\,R \qquad P(z) \vdash P(f(z))}{\vdash P(f(f(a)))}\,R \qquad P(f(f(a))) \vdash}{\vdash}\,R$$

The following instance γ' of γ

$$\cfrac{\cfrac{\cfrac{\vdash P(a), P(a) \qquad P(a) \vdash P(f(a))}{\vdash P(f(a))}\,R \qquad P(f(a)) \vdash P(f(f(a)))}{\vdash P(f(f(a)))}\,R \qquad P(f(f(a))) \vdash}{\vdash}\,R$$

is a ground resolution refutation of $\mathcal{C}$ and a ground projection of γ. $\triangle$

Definition 2.3.14 (cf. [9]). Let Γ be a multiset of atomic formulas, then $set(\Gamma)$ denotes the set of atomic formulas occurring in Γ. Moreover, let $C = \Gamma \vdash \Delta$ be a sequent, then $set(C) = set(\Gamma) \cup set(\Delta)$. $\triangle$

Subsumption is defined as in [9, Definition 2.16].

Definition 2.3.15 (Subsumption). Let $C = \Gamma \vdash \Delta$ and $D = \Pi \vdash \Lambda$ be clauses. Then C *subsumes* D (denoted by $C \leq_{ss} D$) if there exists a substitution θ such that

$$\begin{array}{rcl} set(\Gamma)\theta & \subseteq & set(\Pi) \text{ and} \\ set(\Delta)\theta & \subseteq & set(\Lambda). \end{array}$$

We extend the relation $\leq_{ss}$ to sets of clauses $\mathcal{C}, \mathcal{D}$ in the following way: $\mathcal{C} \leq_{ss} \mathcal{D}$ if for all $D \in \mathcal{D}$, there exists a $C \in \mathcal{C}$ such that $C \leq_{ss} D$. $\triangle$

The subsumption relation can also be extended to resolution deductions [9].

Definition 2.3.16 ([9], Definition 2.17). Let γ and δ be resolution deductions. We define $\gamma \leq_{ss} \delta$ by induction on the number of nodes in δ:

If δ consists of a single node labelled with a clause D, then $\gamma \leq_{ss} \delta$ if γ consists of a single node labelled with C and $C \leq_{ss} D$.

Let δ be

$$\cfrac{\begin{matrix}(\delta_1) & & (\delta_2)\\ D_1 & & D_2\end{matrix}}{D}\, R,$$

and γ_1 be a deduction of C_1 with $\gamma_1 \leq_{ss} \delta_1$, γ_2 be a deduction of C_2 with $\gamma_2 \leq_{ss} \delta_2$. Then we distinguish the following cases:

$$\text{if } C_1 \leq_{ss} D, \text{ then } \gamma_1 \leq_{ss} \delta.$$
$$\text{if } C_2 \leq_{ss} D, \text{ then } \gamma_2 \leq_{ss} \delta.$$

Otherwise, let C be a resolvent of C_1 and C_2 and let $\gamma =$

$$\cfrac{\begin{matrix}(\gamma_1) & & (\gamma_2)\\ C_1 & & C_2\end{matrix}}{C}\, R.$$

Then $\gamma \leq_{ss} \delta$. It can be shown that—in the second case—such a resolvent C of C_1 and C_2 with the above property actually exists (for a proof we refer to [34, Lemma 4.2.1]). $\triangle$

Proposition 2.3.17 ([9], Proposition 2.1). *Let $\mathcal{C}, \mathcal{D}$ be sets of clauses with $\mathcal{C} \leq_{ss} \mathcal{D}$, and let δ be a resolution deduction from $\mathcal{D}$. Then there exists a resolution deduction γ from $\mathcal{C}$ such that $\gamma \leq_{ss} \delta$.*

PROOF. By Lemma 4.2.1 in [34], and by Definition 2.3.16. $\square$

Theorem 2.3.18 (Completeness of Resolution Deduction). *If $\mathcal{C}$ is an unsatisfiable set of clauses, then there exists a resolution refutation of $\mathcal{C}$.*

PROOF. By Theorem 2.7.2 in [34]. $\square$

CHAPTER 3

The Problem of Cut-Elimination

The following chapter is intended to give an overview on the general problem of (reductive) cut-elimination as introduced by Gentzen in his seminal papers *"Untersuchungen über das logische Schließen I + II"* [21]. Particularly, in Section 4.1, we will outline why cut-elimination is such an important method for both mathematical logic and computer science. In Section 4.2, we will consider cut-elimination and some of its most important consequences in a more formal way. This chapter is then concluded in Section 4.3, by defining a rewriting system for cut-elimination based on rules obtained from Gentzen's original proof of the cut-elimination theorem.

3.1 Motivation

Cut-elimination was introduced by Gerhard Gentzen as a constructive method for proving the so-called "Hauptsatz" (or cut-elimination theorem) for both **LK** and **LJ** [10, 21]. Basically, it states that any theorem of first-order logic can be proved without detours, i.e. without the use of cuts [47].

Generally speaking, cut-elimination is concerned with the elimination of all cuts from proofs in order to obtain a cut-free proof of the same statement. Since cuts correspond to the use of lemmas (i.e. intermediary statements) in mathematical proofs, the cut-elimination theorem implies that any statement can be proved without the use of lemmas [9].

Moreover, cut-free proofs are analytic in the sense that all formulas used in the proof are (instances of) subformulas of the end-sequent (i.e. they have the so-called *subformula property*) [9, 35]. The subformula property is just one of many important consequences of the cut-elimination theorem. One of the most important ones is the *consistency* of both **LK** and **LJ**. Indeed, if there would be a proof of the empty sequent, then it would be provable without a cut, which is impossible by the subformula property of cut-free proofs [47].

In Gentzen's "sharpened Hauptsatz" (or midsequent theorem) it was shown that in a cut-free proof of a sequent containing only formulas in prenex form, there exists a so-called *midsequent*, which splits the proof into an upper part, containing the propositional inferences and into a lower part, containing the quantifier inferences [21,47]. This allows the extraction of *Herbrand sequents*[1].

Furthermore, the cut-elimination theorem can be used to prove *Craig's interpolation theorem* [15] via *Maehara's lemma* [10,47], which gives a method to construct an interpolant of $A \to B$ from a cut-free proof of $A \to B$, where A and B are formulas [47]. Craig's interpolation theorem can also be used to prove *Beth's definability theorem*, which states that implicit definitions can be converted into explicit ones [10,47].

Cut-elimination can also be used as a method of *proof mining* in the sense that hidden mathematical information can be extracted by eliminating lemmas from proofs [35].

Another approach for proof mining, the extraction of functionals from proofs, is based on Gödel's dialectica interpretation [26], and it allows the construction of programs from proofs (see [11,12]) [35]. Moreover, functional interpretation can also be used to extract Herbrand disjunctions from proofs [22]. Originally, Gödel's method was motivated by Hilbert's program[2] [2], in particular by the desire to show the consistency of number theory [26].

As demonstrated by Girard in [24], cut-elimination can actually be applied to "real" proofs in current mathematics. By applying cut-elimination to Fürstenberg and Weiss' topological proof [19] of van der Waerden's theorem[3] [49] and thus eliminating all lemmas used in the proof, Girard was able to obtain van der Waerden's original proof as a result.

The logic programming paradigm represents a more practical application of cut-elimination, as the computation of logic programs is based on a search for cut-free proofs [37].

Gentzen's "Hauptsatz" also plays an important role in automated theorem proving based on backward reasoning in sound and complete calculi without cuts. This is due to the fact that omitting the cut-rule rules out the possibility of infinite branching factors in the search tree, as there are infinitely many possible cut-formulas [17].

Other types of calculi frequently used in automated deduction are so-called *tableau methods*. In these types of calculi, an equivalent form of the cut-elimination theorem does also hold [18,44].

3.2 Cut-Elimination Theorem & Consequences

We have already mentioned that Gentzen's "Hauptsatz" has many important consequences within mathematical logic. In this section we will formulate the cut-elimination

[1]Herbrand sequents are generalizations of *Herbrand disjunctions* (see [27]) for the sequent calculus **LK** [30].

[2]For more information on Hilbert's program, we refer the reader to [52].

[3]Given a partition $\mathbb{N} = C_1, \ldots, C_k$ of the integers, one of the sets C_i contains arbitrarily long arithmetic progressions [24].

theorem and some of its consequences in a more formal way.

Theorem 3.2.1 ([21], Gentzen 1934). *If a sequent is **LK**-provable, then it is **LK**-provable without a cut.*

PROOF. We will only give a general outline of the proof; for the full proof we refer to [21, 47]. Let φ be an **LK**-proof. The proof is then by double induction on $grade(\varphi)$ and $rank(\varphi)$, where the uppermost cut (in fact, a mix) is eliminated by permuting the cut upwards (and thus reducing the rank) until no longer possible (i.e. the cut occurs immediately below the inferences that introduced its cut-formula in both premises); then the grade of the cut-formula A is reduced by replacing this cut by cuts, where the cut-formulas are subformulas of A. Cuts having axioms as premises can then be eliminated completely. Iterating this procedure eventually yields a cut-free proof of the same end-sequent. □

The following corollary corresponds to the above mentioned subformula property.

Corollary 3.2.2 ([47], Theorem 6.3). *In a cut-free proof in **LK** (or **LJ**) all the formulas which occur in it are subformulas of the formulas in the end-sequent.*

PROOF. By mathematical induction on the number of inferences in the cut-free proof. □

Corollary 3.2.3 corresponds to Theorem 6.2 in [47].

Corollary 3.2.3 (Consistency). ***LK** and **LJ** are consistent.*

PROOF. Suppose $\vdash$ were provable in **LK** (or **LJ**). Then, by Theorem 3.2.1, it would be provable in **LK** (or **LJ**) without a cut. But this is impossible by the subformula property of cut-free proofs. □

The following result is a formulation of Gentzen's midsequent theorem.

Theorem 3.2.4 ([47], Theorem 6.4). *Let S be a sequent which consists of prenex formulas only and is provable in **LK**. Then there is a cut-free proof of S which contains a sequent (called a midsequent), say S', which satisfies the following:*

(i) *S' is quantifier-free.*

(ii) *Every inference above S' is either structural or propositional.*

(iii) *Every inference below S' is either structural or a quantifier inference.*

PROOF. See [47]. □

Now, we will state Craig's interpolation theorem; for the full proof based on cut-elimination we refer to [47].

Theorem 3.2.5 ([47], Theorem 6.6). *Let A and B be two formulas such that $A \to B$ is **LK**-provable. If A and B have at least one predicate symbol in common, then there exists a formula C, called an* interpolant *of $A \to B$, such that C contains only those constants,* *predicate symbols and free variables that occur in both A and B, and such $A \rightarrow C$ and $C \rightarrow B$ are **LK**-provable. If A and B have no predicate symbol in common, then either $A \vdash$ or $\vdash B$ is **LK**-provable.*

3.3 Reductive Cut-Elimination

From the method described in Theorem 3.2.1 one can extract an algorithm that actually transforms a proof of a sequent containing cuts into a cut-free proof of the same sequent. These transformation steps can be used to define a proof rewriting system whose normal forms are cut-free proofs [9]. We will describe such a rewriting system (or reduction system) in more detail in the following section.

Closely related to Gentzen's procedure is a method due to Schütte [43] and Tait [46] which eliminates the uppermost cut in a proof whose cut-formula has maximal complexity, i.e. if the cut-formula of the uppermost cut is A, then $comp(B) \leq comp(A)$, for all other cut-formulas B in the proof [9]. We will refer to both of these methods as *reductive cut-elimination methods* in the following; they even share a common rule base when interpreted as proof rewriting systems [10, 41].

We will now give the definitions needed in order to formulate a reduction system for cut-elimination. For a detailed discussion of reduction systems we refer to [3].

The following definition is based on [9, 41].

Definition 3.3.1 (Cut-Reduction System). Let Ψ be the set of all **LK**-derivations. Then the pair $\mathcal{R} = \langle \Psi, >_{\mathcal{R}} \rangle$ is called a *cut-reduction system*, where $>_{\mathcal{R}} \subseteq \Psi \times \Psi$ is a binary relation over **LK**-derivations. If $\varphi, \psi \in \Psi$, then $\varphi >_{\mathcal{R}} \psi$ if and only if φ reduces to ψ according to the cut-reduction rules (without cut-elimination over axioms) specified in Definition 3.3.9. Similarly, we define $\mathcal{R}_{\mathsf{ax}}$ by including all reduction rules from Definition 3.3.9. △

Definition 3.3.2 (cf. [9], Definition 3.1). Let $> \subseteq \Psi \times \Psi$. We say that $>$ is based on $\mathcal{R} = \langle \Psi, >_{\mathcal{R}} \rangle$ if $> \subseteq >_{\mathcal{R}}$ and write $\psi > \chi$ for $(\psi, \chi) \in >$. Analogous for $\mathcal{R}_{\mathsf{ax}}$. △

Definition 3.3.3 (cf. [9], Definition 3.2). Let $\psi, \chi \in \Psi$ with $\psi >_{\mathcal{R}} \chi$, and let $\varphi \in \Psi$ such that $\varphi.\nu = \psi$, for a node ν in φ. Then we define $\varphi >_{\mathcal{R}} \varphi[\chi]_\nu$ (i.e. $>_{\mathcal{R}}$ is closed under contexts). Analogous for $\mathcal{R}_{\mathsf{ax}}$. △

Remark. The reduction relation defined by Gentzen's proof is a subrelation of $\mathcal{R}$ [9].

Definition 3.3.4 is taken from [9, Definition 3.3].

Definition 3.3.4 (Gentzen Reduction). We define $\psi >_G \chi$ if $\psi >_{\mathcal{R}} \chi$ and ψ is a cut-derivation with a single non-atomic cut only, which is the last inference. $>_G$ is extended like $>_{\mathcal{R}}$: $\varphi >_G \varphi'$ if $\varphi' = \varphi[\chi]_\nu$ and $\varphi.\nu >_G \chi$. △

We follow [9, Definition 3.4] by defining:

Definition 3.3.5 (Tait Reduction). We define $\varphi >_T \varphi'$ if the following conditions are fulfilled:

(i) There exists a node ν in φ such that $\varphi.\nu$ is a cut-derivation with a maximal cut-formula (i.e. if the cut-formula of the last cut in $\varphi.\nu$ is A, then $comp(B) \leq comp(A)$, for all other cut-formulas B in φ).

(ii) $\varphi'.\nu$ is strict, i.e. for all other cut-formulas B in $\varphi'.\nu$, we have $comp(B) < comp(A)$.

(iii) $\varphi' = \varphi[\chi]_\nu$, for an **LK**-derivation χ with $\varphi.\nu >_{\mathcal{R}} \chi$.

△

Remark. For the sake of convenience, we will refer to the reduction based on the Schütte-Tait method simply as Tait reduction in the following.

Definition 3.3.6. Let $>$ be a cut-reduction relation based on $\mathcal{R}$. Then we define $\psi >^{\mathsf{top}} \chi$ if $\psi >_{\mathcal{R}} \chi$ and ψ is a cut-derivation with a single cut only (either atomic or non-atomic), which is the last inference. $>^{\mathsf{top}}$ is extended like $>$: $\varphi >^{\mathsf{top}} \varphi'$ if $\varphi' = \varphi[\chi]_\nu$ and $\varphi.\nu >^{\mathsf{top}} \chi$. △

Clearly, both $>_G$, $>_{G^{\mathsf{top}}}$ and $>_T$, $>_{T^{\mathsf{top}}}$ are based on $\mathcal{R}$. The end-products of cut-reductions are **LK**-derivations with atomic cuts only. These derivations are our normal forms [9].

The definition of an atomic cut normal form corresponds to [9, Definition 3.5].

Definition 3.3.7 (ACNF). Let $>$ be a cut-reduction relation based on $\mathcal{R}$. Then an **LK**-derivation ψ is in *atomic cut normal form* (**ACNF**) w.r.t. $>$ if there exists no χ such that $\psi > \chi$.

Let $>^*$ be the reflexive and transitive closure of $>$. We say that ψ is an **ACNF** of φ if ψ is in **ACNF** and $\varphi >^* \psi$. Any method which transforms **LK**-proofs into **ACNF**s is called an *AC-normalization*. △

For $>_{\mathcal{R}}$, $>_G$ and $>_T$, all normal forms are **LK**-proofs without non-atomic cuts [9].

Remark. Let ψ be an **LK**-derivation of a sequent S from a set of sequents $\mathcal{A}$ and ψ be in **ACNF**. If the set $\mathcal{A}$ is closed under cut, then there exists also a cut-free derivation of S from $\mathcal{A}$ [9].

Definition 3.3.8 ($\mathbf{ACNF}^{\mathsf{top}}$). Let $>$ be one of $>_{\mathcal{R}^{\mathsf{top}}}$, $>_{G^{\mathsf{top}}}$ or $>_{T^{\mathsf{top}}}$. Then an **LK**-derivation ψ is in $\mathbf{ACNF}^{\mathsf{top}}$ w.r.t. $>$ if there exists no χ such that $\psi > \chi$.

Let $>^*$ be the reflexive and transitive closure of $>$. We say that ψ is an $\mathbf{ACNF}^{\mathsf{top}}$ of φ if ψ is in $\mathbf{ACNF}^{\mathsf{top}}$ and $\varphi >^* \psi$. Any method which transforms **LK**-proofs into $\mathbf{ACNF}^{\mathsf{top}}$s is called an *AC*$^{\mathsf{top}}$*-normalization*. △

Remark. In **LK**-derivations ψ in $\mathbf{ACNF}^{\mathsf{top}}$, all cuts occurring in ψ are atomic cuts that occur at the top of ψ.

Furthermore, note that since $>_{\mathcal{R}}$, $>_G$ and $>_T$ are subrelations of $>_{\mathcal{R}^{\mathsf{top}}}$, $>_{G^{\mathsf{top}}}$ and $>_{T^{\mathsf{top}}}$, respectively, any **LK**-derivation in $\mathbf{ACNF}^{\mathsf{top}}$ is also in **ACNF**.

Cut-Reduction Rules

In the following, we will define the *cut-reduction rules*, which can be divided into *cut-elimination, grade-reduction* and *rank-reduction rules*. Cut-elimination rules transform a given proof ψ into a proof ψ' in such a way that ψ' is the result of eliminating the uppermost cut from ψ. Grade reductions serve the purpose to replace a cut with non-atomic cut-formulas by new cuts whose cut-formulas are subformulas of the non-atomic cut-formula. Rank reductions, on the other hand, are used to permute cuts over unary or binary rules upwards in the proof.

The following definition is based on the ones given in [9, 40, 51].

Definition 3.3.9 (Cut-reduction Rules). In the following we indicate via the symbol $\Downarrow$ that the proof ψ above $\Downarrow$ can be transformed into the proof ψ' below $\Downarrow$ according to the respective cut-reduction rules, i.e. $\psi >_{\mathcal{R}_{\mathsf{ax}}} \psi'$. Furthermore, $\rho, \sigma, \rho_1, \sigma_1, \ldots, \rho', \sigma', \ldots$ denote subderivations of ψ and ψ'.

Cut-elimination rules:

Over axioms:

$$\dfrac{A \vdash A \qquad \begin{array}{c}(\sigma)\\ A, \Gamma \vdash \Delta\end{array}}{A, \Gamma \vdash \Delta}\ cut(A) \qquad\qquad \dfrac{\begin{array}{c}(\rho)\\ \Gamma \vdash \Delta, A\end{array} \qquad A \vdash A}{\Gamma \vdash \Delta, A}\ cut(A)$$

$$\Downarrow \qquad\qquad\qquad\qquad \Downarrow$$

$$\begin{array}{c}(\sigma)\\ A, \Gamma \vdash \Delta\end{array} \qquad\qquad\qquad \begin{array}{c}(\rho)\\ \Gamma \vdash \Delta, A\end{array}$$

Over weakening:

$$\dfrac{\dfrac{\begin{array}{c}(\rho')\\ \Gamma \vdash \Delta\end{array}}{\Gamma \vdash \Delta, A}\ w_r \qquad \begin{array}{c}(\sigma)\\ A, \Pi \vdash \Lambda\end{array}}{\Gamma, \Pi \vdash \Delta, \Lambda}\ cut(A) \qquad\qquad \dfrac{\begin{array}{c}(\rho)\\ \Gamma \vdash \Delta, A\end{array} \qquad \dfrac{\begin{array}{c}(\sigma')\\ \Pi \vdash \Lambda\end{array}}{A, \Pi \vdash \Lambda}\ w_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\ cut(A)$$

$$\Downarrow \qquad\qquad\qquad\qquad \Downarrow$$

$$\begin{array}{c}(\rho')\\ \Gamma \vdash \Delta \\ \hline\hline \Gamma, \Pi \vdash \Delta, \Lambda\end{array}\ w_r^*, w_l^* \qquad\qquad \begin{array}{c}(\sigma')\\ \Pi \vdash \Lambda \\ \hline\hline \Gamma, \Pi \vdash \Delta, \Lambda\end{array}\ w_r^*, w_l^*$$

Grade-reduction rules:

If the cut-formula has $\neg$ as top-level connective:

$$\dfrac{\dfrac{\begin{matrix}(\rho')\\ A,\Gamma\vdash\Delta\end{matrix}}{\Gamma\vdash\Delta,\neg A}\,\neg_r \qquad \dfrac{\begin{matrix}(\sigma')\\ \Pi\vdash\Lambda,A\end{matrix}}{\neg A,\Pi\vdash\Lambda}\,\neg_l}{\Gamma,\Pi\vdash\Delta,\Lambda}\,cut(\neg A)$$

$$\Downarrow$$

$$\dfrac{\begin{matrix}(\sigma')\\ \Pi\vdash\Lambda,A\end{matrix} \qquad \begin{matrix}(\rho')\\ A,\Gamma\vdash\Delta\end{matrix}}{\Gamma,\Pi\vdash\Delta,\Lambda}\,cut(A)$$

If the cut-formula has $\wedge$ as top-level connective:

$$\dfrac{\dfrac{\begin{matrix}(\rho_1)\\ \Gamma\vdash\Delta,A_1\end{matrix} \qquad \begin{matrix}(\rho_2)\\ \Gamma\vdash\Delta,A_2\end{matrix}}{\Gamma\vdash\Delta,A_1\wedge A_2}\,\wedge_r \qquad \dfrac{\begin{matrix}(\sigma')\\ A_i,\Pi\vdash\Lambda\end{matrix}}{A_1\wedge A_2,\Pi\vdash\Lambda}\,\wedge_{l_i}}{\Gamma,\Pi\vdash\Delta,\Lambda}\,cut(A_1\wedge A_2)$$

$$\Downarrow$$

$$\dfrac{\dfrac{\begin{matrix}(\rho_1)\\ \Gamma\vdash\Delta,A_1\end{matrix} \qquad \dfrac{\begin{matrix}(\rho_2)\\ \Gamma\vdash\Delta,A_2\end{matrix} \qquad \dfrac{\begin{matrix}(\sigma')\\ A_i,\Pi\vdash\Lambda\end{matrix}}{A_1,A_2,\Pi\vdash\Lambda}\,w_l}{A_1,\Gamma,\Pi\vdash\Delta,\Lambda}\,cut(A_2)}{\Gamma,\Gamma,\Pi\vdash\Delta,\Delta,\Lambda}\,cut(A_1)}{\Gamma,\Pi\vdash\Delta,\Lambda}\,c_l^*,c_r^*$$

If the cut-formula has $\vee$ as top-level connective:

$$\dfrac{\dfrac{\begin{matrix}(\rho)\\ \Gamma\vdash\Delta,A_i\end{matrix}}{\Gamma\vdash\Delta,A_1\vee A_2}\,\vee_{r_i} \qquad \dfrac{\begin{matrix}(\sigma_1)\\ A_1,\Pi\vdash\Lambda\end{matrix} \qquad \begin{matrix}(\sigma_2)\\ A_2,\Pi\vdash\Lambda\end{matrix}}{A_1\vee A_2,\Pi\vdash\Lambda}\,\vee_l}{\Gamma,\Pi\vdash\Delta,\Lambda}\,cut(A_1\vee A_2)$$

$$\Downarrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{matrix}(\rho)\\ \Gamma\vdash\Delta,A_i\end{matrix}}{\Gamma\vdash\Delta,A_1,A_2}\,w_r \qquad \begin{matrix}(\sigma_2)\\ A_2,\Pi\vdash\Lambda\end{matrix}}{\Gamma,\Pi\vdash\Delta,\Lambda,A_1}\,cut(A_2) \qquad \begin{matrix}(\sigma_1)\\ A_1,\Pi\vdash\Lambda\end{matrix}}{\Gamma,\Pi,\Pi\vdash\Delta,\Lambda,\Lambda}\,cut(A_1)}{\Gamma,\Pi\vdash\Delta,\Lambda}\,c_l^*,c_r^*$$

If the cut-formula has $\exists$ as top-level connective:

$$\dfrac{\dfrac{\overset{(\rho')}{\Gamma \vdash \Delta, A(x/t)}}{\Gamma \vdash \Delta, (\exists x)A(x)}\,\exists_r \qquad \dfrac{\overset{(\sigma'(x/y))}{A(x/y), \Pi \vdash \Lambda}}{(\exists x)A(x), \Pi \vdash \Lambda}\,\exists_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut((\exists x)A)$$

$$\Downarrow$$

$$\dfrac{\overset{(\rho')}{\Gamma \vdash \Delta, A(x/t)} \qquad \overset{(\sigma'(x/t))}{A(x/t), \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(A(x/t))$$

If the cut-formula has $\forall$ as top-level connective:

$$\dfrac{\dfrac{\overset{(\rho'(x/y))}{\Gamma \vdash \Delta, A(x/y)}}{\Gamma \vdash \Delta, (\forall x)A(x)}\,\forall_r \qquad \dfrac{\overset{(\sigma')}{A(x/t), \Pi \vdash \Lambda}}{(\forall x)A(x), \Pi \vdash \Lambda}\,\forall_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut((\forall x)A)$$

$$\Downarrow$$

$$\dfrac{\overset{(\rho'(x/t))}{\Gamma \vdash \Delta, A(x/t)} \qquad \overset{(\sigma')}{A(x/t), \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(A(x/t))$$

Rank-reduction rules:

Over a unary rule ξ:

$$\dfrac{\dfrac{\overset{(\rho')}{\Gamma' \vdash \Delta', A}}{\Gamma \vdash \Delta, A}\,\xi \qquad \overset{(\sigma)}{A, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(A) \qquad\qquad \dfrac{\overset{(\rho)}{\Gamma \vdash \Delta, A} \qquad \dfrac{\overset{(\sigma')}{A, \Pi' \vdash \Lambda'}}{A, \Pi \vdash \Lambda}\,\xi}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(A)$$

$$\Downarrow \qquad\qquad\qquad\qquad \Downarrow$$

$$\dfrac{\dfrac{\overset{(\rho')}{\Gamma' \vdash \Delta', A} \qquad \overset{(\sigma)}{A, \Pi \vdash \Lambda}}{\Gamma', \Pi \vdash \Delta', \Lambda}\,cut(A)}{\Gamma, \Pi \vdash \Delta, \Lambda}\,\xi \qquad\qquad \dfrac{\dfrac{\overset{(\rho)}{\Gamma \vdash \Delta, A} \qquad \overset{(\sigma')}{A, \Pi' \vdash \Lambda'}}{\Gamma, \Pi' \vdash \Delta, \Lambda'}\,cut(A)}{\Gamma, \Pi \vdash \Delta, \Lambda}\,\xi$$

Over a binary rule ξ:

$$\dfrac{\dfrac{\begin{matrix}(\rho_1)\\ \Gamma_1 \vdash \Delta_1, A\end{matrix} \qquad \begin{matrix}(\rho_2)\\ \Gamma_2 \vdash \Delta_2\end{matrix}}{\Gamma \vdash \Delta, A}\,\xi \qquad \begin{matrix}(\sigma)\\ A, \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(A)$$

$$\Downarrow$$

$$\dfrac{\dfrac{\dfrac{\begin{matrix}(\rho_1)\\ \Gamma_1 \vdash \Delta_1, A\end{matrix} \qquad \begin{matrix}(\sigma)\\ A, \Pi \vdash \Lambda\end{matrix}}{\Gamma_1, \Pi \vdash \Delta_1, \Lambda}\,cut(A) \qquad \dfrac{\dfrac{\begin{matrix}(\rho_2)\\ \Gamma_2 \vdash \Delta_2\end{matrix}}{\Gamma_2 \vdash \Delta_2, A}\,w_r^* \qquad \begin{matrix}(\sigma')\\ A, \Pi \vdash \Lambda\end{matrix}}{\Gamma_2, \Pi \vdash \Delta_2, \Lambda}\,cut(A)}{\Gamma, \Pi, \Pi \vdash \Delta, \Lambda, \Lambda}\,\xi}{\Gamma, \Pi \vdash \Delta, \Lambda}\,c_l^*, c_r^*$$

$$\dfrac{\dfrac{\begin{matrix}(\rho_1)\\ \Gamma_1 \vdash \Delta_1\end{matrix} \qquad \begin{matrix}(\rho_2)\\ \Gamma_2 \vdash \Delta_2, A\end{matrix}}{\Gamma \vdash \Delta, A}\,\xi \qquad \begin{matrix}(\sigma)\\ A, \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(A)$$

$$\Downarrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{matrix}(\rho_1)\\ \Gamma_1 \vdash \Delta_1\end{matrix}}{\Gamma_1 \vdash \Delta_1, A}\,w_r^* \qquad \begin{matrix}(\sigma)\\ A, \Pi \vdash \Lambda\end{matrix}}{\Gamma_1, \Pi \vdash \Delta_1, \Lambda}\,cut(A) \qquad \dfrac{\begin{matrix}(\rho_2)\\ \Gamma_2 \vdash \Delta_2, A\end{matrix} \qquad \begin{matrix}(\sigma')\\ A, \Pi \vdash \Lambda\end{matrix}}{\Gamma_2, \Pi \vdash \Delta_2, \Lambda}\,cut(A)}{\Gamma, \Pi, \Pi \vdash \Delta, \Lambda, \Lambda}\,\xi}{\Gamma, \Pi \vdash \Delta, \Lambda}\,c_l^*, c_r^*$$

where in the above two reductions σ' is obtained from σ by renaming the eigenvariables in such a way that the regularity of ψ' is ensured. Moreover, in these particular cases, w_r^* stands for at most one application of w_r.

$$\dfrac{\begin{matrix}(\rho)\\ \Gamma \vdash \Delta, A\end{matrix} \qquad \dfrac{\begin{matrix}(\sigma_1)\\ A, \Pi_1 \vdash \Lambda_1\end{matrix} \qquad \begin{matrix}(\sigma_2)\\ \Pi_2 \vdash \Lambda_2\end{matrix}}{A, \Pi \vdash \Lambda}\,\xi}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(A)$$

$$\Downarrow$$

$$\dfrac{\dfrac{\dfrac{(\rho)}{\Gamma \vdash \Delta, A} \quad \dfrac{(\sigma_1)}{A, \Pi_1 \vdash \Lambda_1}}{\Gamma, \Pi_1 \vdash \Delta, \Lambda_1}\, cut(A) \quad \dfrac{\dfrac{(\rho')}{\Gamma \vdash \Delta, A} \quad \dfrac{\dfrac{(\sigma_2)}{\Pi_2 \vdash \Lambda_2}}{A, \Pi_2 \vdash \Lambda_2}\, w_l^*}{\Gamma, \Pi_2 \vdash \Delta, \Lambda_2}\, cut(A)}{\dfrac{\Gamma, \Gamma, \Pi \vdash \Delta, \Delta, \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda}\, c_l^*, c_r^*}\, \xi$$

$$\dfrac{\dfrac{(\rho)}{\Gamma \vdash \Delta, A} \quad \dfrac{\dfrac{(\sigma_1)}{\Pi_1 \vdash \Lambda_1} \quad \dfrac{(\sigma_2)}{A, \Pi_2 \vdash \Lambda_2}}{A, \Pi \vdash \Lambda}\, \xi}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut(A)$$

$$\Downarrow$$

$$\dfrac{\dfrac{\dfrac{(\rho)}{\Gamma \vdash \Delta, A} \quad \dfrac{\dfrac{(\sigma_1)}{\Pi_1 \vdash \Lambda_1}}{A, \Pi_1 \vdash \Lambda_1}\, w_l^*}{\Gamma, \Pi_1 \vdash \Delta, \Lambda_1}\, cut(A) \quad \dfrac{\dfrac{(\rho')}{\Gamma \vdash \Delta, A} \quad \dfrac{(\sigma_2)}{A, \Pi_2 \vdash \Lambda_2}}{\Gamma, \Pi_2 \vdash \Delta, \Lambda_2}\, cut(A)}{\dfrac{\Gamma, \Gamma, \Pi \vdash \Delta, \Delta, \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda}\, c_l^*, c_r^*}\, \xi$$

where in the above two reductions ρ' is obtained from ρ by renaming the eigenvariables in such a way that the regularity of ψ' is ensured. Moreover, in these particular cases, w_l^* stands for at most one application of w_l.

Over contraction rules:

Contraction right c_r:

$$\dfrac{\dfrac{\dfrac{(\rho')}{\Gamma \vdash \Delta, A, A}}{\Gamma \vdash \Delta, A}\, c_r \quad \dfrac{(\sigma)}{A, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut(A)$$

$$\Downarrow$$

$$
\dfrac{\dfrac{\overset{(\rho')}{\Gamma \vdash \Delta, A, A} \quad \overset{(\sigma)}{A, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda, A}\, cut(A) \qquad \overset{(\sigma')}{A, \Pi \vdash \Lambda}}{\dfrac{\Gamma, \Pi, \Pi \vdash \Delta, \Lambda, \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda}\, c_l^*, c_r^*}\, cut(A)
$$

where σ' is obtained from σ by renaming the eigenvariables in such a way that the regularity of ψ' is ensured.

Contraction left c_l:

$$
\dfrac{\overset{(\rho)}{\Gamma \vdash \Delta, A} \qquad \dfrac{\overset{(\sigma')}{A, A, \Pi \vdash \Lambda}}{A, \Pi \vdash \Lambda}\, c_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut(A)
$$

$$\Downarrow$$

$$
\dfrac{\overset{(\rho')}{\Gamma \vdash \Delta, A} \qquad \dfrac{\overset{(\rho)}{\Gamma \vdash \Delta, A} \quad \overset{(\sigma')}{A, A, \Pi \vdash \Lambda}}{A, \Gamma, \Pi \vdash \Delta, \Lambda}\, cut(A)}{\dfrac{\Gamma, \Gamma, \Pi \vdash \Delta, \Delta, \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda}\, c_l^*, c_r^*}\, cut(A)
$$

where ρ' is obtained from ρ by renaming the eigenvariables in such a way that the regularity of ψ' is ensured.

$\triangle$

Example 3.3.10. Let φ be the derivation (where u, v are free variables and a, b constant symbols)

$$
\dfrac{\overset{(\varphi_1)}{P(a) \lor Q(b) \vdash (\exists y)(P(y) \lor Q(y))} \qquad \overset{(\varphi_2)}{(\exists y)(P(y) \lor Q(y)), (\forall x)\neg P(x) \vdash (\exists z)Q(z)}}{P(a) \lor Q(b), (\forall x)(\neg P(x)) \vdash (\exists z)Q(z)}\, cut,
$$

where φ_1 is the **LK**-derivation:

$$
\dfrac{\dfrac{\dfrac{P(a) \vdash P(a)}{P(a) \vdash (P(a) \lor Q(a))}\, \lor_{r_1}}{P(a) \vdash (\exists y)(P(y) \lor Q(y))}\, \exists_r \qquad \dfrac{\dfrac{Q(b) \vdash Q(b)}{Q(b) \vdash (P(b) \lor Q(b))}\, \lor_{r_2}}{Q(b) \vdash (\exists y)(P(y) \lor Q(y))}\, \exists_r}{\underbrace{P(a) \lor Q(b) \vdash (\exists y)(P(y) \lor Q(y))}_{S_1}}\, \lor_l
$$

and φ_2 is the **LK**-derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(u)\vdash P(u)}{P(u),\neg P(u)\vdash}\,\neg_l}{P(u),\neg P(u)\vdash Q(u)}\,w_r \qquad \dfrac{Q(u)\vdash Q(u)}{Q(u),\neg P(u)\vdash Q(u)}\,w_l}{((P(u)\lor Q(u)),\neg P(u)\vdash Q(u)}\,\lor_l}{((P(u)\lor Q(u)),\neg P(u)\vdash(\exists z)Q(z)}\,\exists_r}{((P(u)\lor Q(u)),(\forall x)\neg P(x)\vdash(\exists z)Q(z)}\,\forall_l}{\underbrace{(\exists y)(P(y)\lor Q(y)),(\forall x)\neg P(x)\vdash(\exists z)Q(z)}_{S_2}}\,\exists_l$$

In the following, we will indicate via the colour purple, which inference rules and cuts are the target of the cut-reduction rules.

For φ, we obtain the following cut-reduction sequence:

$$\dfrac{\dfrac{\dfrac{\dfrac{P(a)\vdash P(a)}{P(a)\vdash(P(a)\lor Q(a))}\,\lor_{r_1}}{P(a)\vdash(\exists y)(P(y)\lor Q(y))}\,\exists_r \qquad \dfrac{\dfrac{Q(b)\vdash Q(b)}{Q(b)\vdash(P(b)\lor Q(b))}\,\lor_{r_2}}{Q(b)\vdash(\exists y)(P(y)\lor Q(y))}\,\exists_r}{P(a)\lor Q(b)\vdash(\exists y)(P(y)\lor Q(y))}\,\lor_l \qquad \begin{matrix}(\varphi_2)\\ S_2\end{matrix}}{P(a)\lor Q(b),(\forall x)(\neg P(x))\vdash(\exists z)Q(z)}\,cut$$

$$\text{rank-reduction over } \lor_l$$
$$\Downarrow$$

$$\dfrac{\begin{matrix}(\varphi_1')\\ P(a),(\forall x)\neg P(x)\vdash(\exists z)Q(z)\end{matrix} \qquad \begin{matrix}(\varphi_2')\\ Q(b),(\forall x)\neg P(x)\vdash(\exists z)Q(z)\end{matrix}}{P(a)\lor Q(b),(\forall x)(\neg P(x))\vdash(\exists z)Q(z)}\,\lor_l$$

where φ_1' is the **LK**-derivation:

$$\dfrac{\dfrac{\dfrac{P(a)\vdash P(a)}{P(a)\vdash P(a)\lor Q(a)}\,\lor_{r_1}}{P(a)\vdash(\exists y)(P(y)\lor Q(y))}\,\exists_r \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(u)\vdash P(u)}{P(u),\neg P(u)\vdash}\,\neg_l}{P(u),\neg P(u)\vdash Q(u)}\,w_r \qquad \dfrac{Q(u)\vdash Q(u)}{Q(u),\neg P(u)\vdash Q(u)}\,w_l}{P(u)\lor Q(u),\neg P(u)\vdash Q(u)}\,\lor_l}{P(u)\lor Q(u),\neg P(u)\vdash(\exists z)Q(z)}\,\exists_r}{P(u)\lor Q(u),(\forall x)\neg P(x)\vdash(\exists z)Q(z)}\,\forall_l}{(\exists y)(P(y)\lor Q(y)),(\forall x)\neg P(x)\vdash(\exists z)Q(z)}\,\exists_l}{P(a),(\forall x)\neg P(x)\vdash(\exists z)Q(z)}\,cut$$

and φ_2' is the **LK**-derivation:

$$
\begin{prooftree}
\AxiomC{$Q(b) \vdash Q(b)$}
\RightLabel{$\vee_{r_2}$}
\UnaryInfC{$Q(b) \vdash P(b) \vee Q(b)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$Q(b) \vdash (\exists y)(P(y) \vee Q(y))$}
\AxiomC{(φ_2'')}
\noLine
\UnaryInfC{S_2}
\RightLabel{$cut,$}
\BinaryInfC{$Q(b), (\forall x)\neg P(x) \vdash (\exists z)Q(z)$}
\end{prooftree}
$$

where φ_2'' is the **LK**-derivation obtained form φ_2 by replacing the eigenvariable u by the eigenvariable v.

We will eliminate the cut occurring in φ_1' first:

$$
\begin{prooftree}
\AxiomC{$P(a) \vdash P(a)$}
\RightLabel{$\vee_{r_1}$}
\UnaryInfC{$P(a) \vdash P(a) \vee Q(a)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$P(a) \vdash (\exists y)(P(y) \vee Q(y))$}
\AxiomC{$P(u) \vdash P(u)$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P(u), \neg P(u) \vdash$}
\RightLabel{w_r}
\UnaryInfC{$P(u), \neg P(u) \vdash Q(u)$}
\AxiomC{$Q(u) \vdash Q(u)$}
\RightLabel{w_l}
\UnaryInfC{$Q(u), \neg P(u) \vdash Q(u)$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P(u) \vee Q(u), \neg P(u) \vdash Q(u)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$P(u) \vee Q(u), \neg P(u) \vdash (\exists z)Q(z)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$P(u) \vee Q(u), (\forall x)\neg P(x) \vdash (\exists z)Q(z)$}
\RightLabel{$\exists_l$}
\UnaryInfC{$(\exists y)(P(y) \vee Q(y)), (\forall x)\neg P(x) \vdash (\exists z)Q(z)$}
\RightLabel{cut}
\BinaryInfC{$P(a), (\forall x)\neg P(x) \vdash (\exists z)Q(z)$}
\end{prooftree}
$$

grade-reduction of $(\exists y)(P(y) \vee Q(y))$

$\Downarrow$

$$
\begin{prooftree}
\AxiomC{$P(a) \vdash P(a)$}
\RightLabel{$\vee_{r_1}$}
\UnaryInfC{$P(a) \vdash P(a) \vee Q(a)$}
\AxiomC{$P(a) \vdash P(a)$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P(a), \neg P(a) \vdash$}
\RightLabel{w_r}
\UnaryInfC{$P(a), \neg P(a) \vdash Q(a)$}
\AxiomC{$Q(a) \vdash Q(a)$}
\RightLabel{w_l}
\UnaryInfC{$Q(a), \neg P(a) \vdash Q(a)$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P(a) \vee Q(a), \neg P(a) \vdash Q(a)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$P(a) \vee Q(a), \neg P(a) \vdash (\exists z)Q(z)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$P(a) \vee Q(a), (\forall x)\neg P(x) \vdash (\exists z)Q(z)$}
\RightLabel{cut}
\BinaryInfC{$P(a), (\forall x)\neg P(x) \vdash (\exists z)Q(z)$}
\end{prooftree}
$$

rank-reduction over $\forall_l$

$\Downarrow$

$$
\begin{prooftree}
\AxiomC{$P(a) \vdash P(a)$}
\RightLabel{$\vee_{r_1}$}
\UnaryInfC{$P(a) \vdash P(a) \vee Q(a)$}
\AxiomC{$P(a) \vdash P(a)$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P(a), \neg P(a) \vdash$}
\RightLabel{w_r}
\UnaryInfC{$P(a), \neg P(a) \vdash Q(a)$}
\AxiomC{$Q(a) \vdash Q(a)$}
\RightLabel{w_l}
\UnaryInfC{$Q(a), \neg P(a) \vdash Q(a)$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P(a) \vee Q(a), \neg P(a) \vdash Q(a)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$P(a) \vee Q(a), \neg P(a) \vdash (\exists z)Q(z)$}
\RightLabel{cut}
\BinaryInfC{$P(a), \neg P(a) \vdash (\exists z)Q(z)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$P(a), (\forall x)\neg P(x) \vdash (\exists z)Q(z)$}
\end{prooftree}
$$

rank-reduction over $\exists_r$

$\Downarrow$

$$
\dfrac{\dfrac{\dfrac{\dfrac{P(a)\vdash P(a)}{P(a)\vdash P(a)\vee Q(a)}\,\vee_{r_1}\qquad \dfrac{\dfrac{\dfrac{P(a)\vdash P(a)}{P(a),\neg P(a)\vdash}\,\neg_l}{P(a),\neg P(a)\vdash Q(a)}\,w_r\quad \dfrac{Q(a)\vdash Q(a)}{Q(a),\neg P(a)\vdash Q(a)}\,w_l}{P(a)\vee Q(a),\neg P(a)\vdash Q(a)}\,\vee_l}{P(a),\neg P(a)\vdash Q(a)}\,cut}{P(a),\neg P(a)\vdash(\exists z)Q(z)}\,\exists_r}{P(a),(\forall x)\neg P(x)\vdash(\exists z)Q(z)}\,\forall_l
$$

grade-reduction of $P(a)\vee Q(a)$

$\Downarrow$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(a)\vdash P(a)}{P(a)\vdash P(a),Q(a)}\,w_r\quad \dfrac{Q(a)\vdash Q(a)}{Q(a),\neg P(a)\vdash Q(a)}\,w_l}{P(a),\neg P(a)\vdash P(a),Q(a)}\,cut\qquad \dfrac{\dfrac{P(a)\vdash P(a)}{P(a),\neg P(a)\vdash}\,\neg_l}{P(a),\neg P(a)\vdash Q(a)}\,w_r}{P(a),\neg P(a),\neg P(a)\vdash Q(a),Q(a)}\,cut}{P(a),\neg P(a)\vdash Q(a),Q(a)}\,c_l}{P(a),\neg P(a)\vdash Q(a)}\,c_r}{P(a),\neg P(a)\vdash(\exists z)Q(z)}\,\exists_r}{P(a),(\forall x)\neg P(x)\vdash(\exists z)Q(z)}\,\forall_l
$$

cut-elimination over w_r

$\Downarrow$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(a)\vdash P(a)}{P(a)\vdash P(a),Q(a)}\,w_r}{P(a),\neg P(a)\vdash P(a),Q(a)}\,w_l\qquad \dfrac{\dfrac{P(a)\vdash P(a)}{P(a),\neg P(a)\vdash}\,\neg_l}{P(a),\neg P(a)\vdash Q(a)}\,w_r}{P(a),\neg P(a),\neg P(a)\vdash Q(a),Q(a)}\,cut}{P(a),\neg P(a)\vdash Q(a),Q(a)}\,c_l}{P(a),\neg P(a)\vdash Q(a)}\,c_r}{P(a),\neg P(a)\vdash(\exists z)Q(z)}\,\exists_r}{P(a),(\forall x)\neg P(x)\vdash(\exists z)Q(z)}\,\forall_l
$$

rank-reduction over w_r

$\Downarrow$

$$
\begin{prooftree}
\AxiomC{$P(a) \vdash P(a)$}
\RightLabel{w_r}
\UnaryInfC{$P(a) \vdash P(a), Q(a)$}
\RightLabel{w_l}
\UnaryInfC{$P(a), \neg P(a) \vdash P(a), Q(a)$}
\AxiomC{$P(a) \vdash P(a)$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P(a), \neg P(a) \vdash$}
\RightLabel{cut}
\BinaryInfC{$P(a), \neg P(a), \neg P(a) \vdash Q(a)$}
\RightLabel{w_r}
\UnaryInfC{$P(a), \neg P(a), \neg P(a) \vdash Q(a), Q(a)$}
\RightLabel{c_l}
\UnaryInfC{$P(a), \neg P(a) \vdash Q(a), Q(a)$}
\RightLabel{c_r}
\UnaryInfC{$P(a), \neg P(a) \vdash Q(a)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$P(a), \neg P(a) \vdash (\exists z) Q(z)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$P(a), (\forall x) \neg P(x) \vdash (\exists z) Q(z)$}
\end{prooftree}
$$

rank-reduction over $\neg_l$

$\Downarrow$

$$
\begin{prooftree}
\AxiomC{$P(a) \vdash P(a)$}
\RightLabel{w_r}
\UnaryInfC{$P(a) \vdash P(a), Q(a)$}
\RightLabel{w_l}
\UnaryInfC{$P(a), \neg P(a) \vdash P(a), Q(a)$}
\AxiomC{$P(a) \vdash P(a)$}
\RightLabel{cut}
\BinaryInfC{$P(a), \neg P(a) \vdash P(a), Q(a)$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P(a), \neg P(a), \neg P(a) \vdash Q(a)$}
\RightLabel{w_r}
\UnaryInfC{$P(a), \neg P(a), \neg P(a) \vdash Q(a), Q(a)$}
\RightLabel{c_l}
\UnaryInfC{$P(a), \neg P(a) \vdash Q(a), Q(a)$}
\RightLabel{c_r}
\UnaryInfC{$P(a), \neg P(a) \vdash Q(a)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$P(a), \neg P(a) \vdash (\exists z) Q(z)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$P(a), (\forall x) \neg P(x) \vdash (\exists z) Q(z)$}
\end{prooftree}
$$

cut-elimination over axioms

$\Downarrow$

$$
\begin{prooftree}
\AxiomC{$P(a) \vdash P(a)$}
\RightLabel{w_r}
\UnaryInfC{$P(a) \vdash P(a), Q(a)$}
\RightLabel{w_l}
\UnaryInfC{$P(a), \neg P(a) \vdash P(a), Q(a)$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P(a), \neg P(a), \neg P(a) \vdash Q(a)$}
\RightLabel{w_r}
\UnaryInfC{$P(a), \neg P(a), \neg P(a) \vdash Q(a), Q(a)$}
\RightLabel{c_l}
\UnaryInfC{$P(a), \neg P(a) \vdash Q(a), Q(a)$}
\RightLabel{c_r}
\UnaryInfC{$P(a), \neg P(a) \vdash Q(a)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$P(a), \neg P(a) \vdash (\exists z) Q(z)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$P(a), (\forall x) \neg P(x) \vdash (\exists z) Q(z)$}
\end{prooftree}
$$

Next, we will eliminate the cut occurring in φ_2':

$$
\begin{prooftree}
\AxiomC{$Q(b) \vdash Q(b)$}
\RightLabel{$\vee_{r_2}$}
\UnaryInfC{$Q(b) \vdash P(b) \vee Q(b)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$Q(b) \vdash (\exists y)(P(y) \vee Q(y))$}
\AxiomC{$P(v) \vdash P(v)$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P(v), \neg P(v) \vdash$}
\RightLabel{w_r}
\UnaryInfC{$P(v), \neg P(v) \vdash Q(v)$}
\AxiomC{$Q(v) \vdash Q(v)$}
\RightLabel{w_l}
\UnaryInfC{$Q(v), \neg P(v) \vdash Q(v)$}
\RightLabel{$\vee_l$}
\BinaryInfC{$((P(v) \vee Q(v)), \neg P(v) \vdash Q(v)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$((P(v) \vee Q(v)), \neg P(v) \vdash (\exists z)Q(z)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$((P(v) \vee Q(v)), (\forall x)\neg P(x) \vdash (\exists z)Q(z)$}
\RightLabel{$\exists_l$}
\UnaryInfC{$(\exists y)(P(y) \vee Q(y)), (\forall x)\neg P(x) \vdash (\exists z)Q(z)$}
\RightLabel{cut}
\BinaryInfC{$Q(b), (\forall x)\neg P(x) \vdash (\exists z)Q(z)$}
\end{prooftree}
$$

applying the the same intermediate reduction steps as for φ_1'

$\Downarrow_*$

$$
\begin{prooftree}
\AxiomC{$Q(b) \vdash Q(b)$}
\RightLabel{w_r}
\UnaryInfC{$Q(b) \vdash P(b), Q(b)$}
\AxiomC{$Q(b) \vdash Q(b)$}
\RightLabel{w_l}
\UnaryInfC{$Q(b), \neg P(b) \vdash Q(b)$}
\RightLabel{cut}
\BinaryInfC{$Q(b), \neg P(b) \vdash P(b), Q(b)$}
\AxiomC{$P(b) \vdash P(b)$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P(b), \neg P(b) \vdash$}
\RightLabel{w_r}
\UnaryInfC{$P(b), \neg P(b) \vdash Q(b)$}
\RightLabel{cut}
\BinaryInfC{$Q(b), \neg P(b), \neg P(b) \vdash Q(b), Q(b)$}
\RightLabel{c_l}
\UnaryInfC{$Q(b), \neg P(b) \vdash Q(b), Q(b)$}
\RightLabel{c_r}
\UnaryInfC{$Q(b), \neg P(b) \vdash Q(b)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$Q(b), \neg P(b) \vdash (\exists z)Q(z)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$Q(b), (\forall x)\neg P(x) \vdash (\exists z)Q(z)$}
\end{prooftree}
$$

rank-reduction over w_l

$\Downarrow$

$$
\begin{prooftree}
\AxiomC{$Q(b) \vdash Q(b)$}
\RightLabel{w_r}
\UnaryInfC{$Q(b) \vdash P(b), Q(b)$}
\AxiomC{$Q(b) \vdash Q(b)$}
\RightLabel{cut}
\BinaryInfC{$Q(b) \vdash P(b), Q(b)$}
\RightLabel{w_l}
\UnaryInfC{$Q(b), \neg P(b) \vdash P(b), Q(b)$}
\AxiomC{$P(b) \vdash P(b)$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P(b), \neg P(b) \vdash$}
\RightLabel{w_r}
\UnaryInfC{$P(b), \neg P(b) \vdash Q(b)$}
\RightLabel{cut}
\BinaryInfC{$Q(b), \neg P(b), \neg P(b) \vdash Q(b), Q(b)$}
\RightLabel{c_l}
\UnaryInfC{$Q(b), \neg P(b) \vdash Q(b), Q(b)$}
\RightLabel{c_r}
\UnaryInfC{$Q(b), \neg P(b) \vdash Q(b)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$Q(b), \neg P(b) \vdash (\exists z)Q(z)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$Q(b), (\forall x)\neg P(x) \vdash (\exists z)Q(z)$}
\end{prooftree}
$$

cut-elimination over axioms

$\Downarrow$

$$
\begin{prooftree}
\AxiomC{$Q(b) \vdash Q(b)$}
\RightLabel{w_r}
\UnaryInfC{$Q(b) \vdash P(b), Q(b)$}
\RightLabel{w_l}
\UnaryInfC{$Q(b), \neg P(b) \vdash {\color{red}P(b)}, Q(b)$}
\AxiomC{$P(b) \vdash P(b)$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P(b), \neg P(b) \vdash$}
\RightLabel{${\color{red}w_r}$}
\UnaryInfC{${\color{red}P(b)}, \neg P(b) \vdash Q(b)$}
\RightLabel{${\color{red}cut}$}
\BinaryInfC{$Q(b), \neg P(b), \neg P(b) \vdash Q(b), Q(b)$}
\RightLabel{c_l}
\UnaryInfC{$Q(b), \neg P(b) \vdash Q(b), Q(b)$}
\RightLabel{c_r}
\UnaryInfC{$Q(b), \neg P(b) \vdash Q(b)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$Q(b), \neg P(b) \vdash (\exists z)Q(z)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$Q(b), (\forall x)\neg P(x) \vdash (\exists z)Q(z)$}
\end{prooftree}
$$

applying the the same intermediate reduction steps as for φ_1'

$$\Downarrow_*$$

$$
\begin{prooftree}
\AxiomC{$Q(b) \vdash Q(b)$}
\RightLabel{w_r}
\UnaryInfC{$Q(b) \vdash P(b), Q(b)$}
\RightLabel{w_l}
\UnaryInfC{$Q(b), \neg P(b) \vdash P(b), Q(b)$}
\RightLabel{$\neg_l$}
\UnaryInfC{$Q(b), \neg P(b), \neg P(b) \vdash Q(b)$}
\RightLabel{w_r}
\UnaryInfC{$Q(b), \neg P(b), \neg P(b) \vdash Q(b), Q(b)$}
\RightLabel{c_l}
\UnaryInfC{$Q(b), \neg P(b) \vdash Q(b), Q(b)$}
\RightLabel{c_r}
\UnaryInfC{$Q(b), \neg P(b) \vdash Q(b)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$Q(b), \neg P(b) \vdash (\exists z)Q(z)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$Q(b), (\forall x)\neg P(x) \vdash (\exists z)Q(z)$}
\end{prooftree}
$$

Finally, we obtain the following cut-free **LK**-proof φ' with the same end-sequent as φ:

$$
\begin{prooftree}
\AxiomC{$P(a) \vdash P(a)$}
\RightLabel{w_r}
\UnaryInfC{$P(a) \vdash P(a), Q(a)$}
\RightLabel{w_l}
\UnaryInfC{$P(a), \neg P(a) \vdash P(a), Q(a)$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P(a), \neg P(a), \neg P(a) \vdash Q(a)$}
\RightLabel{w_r}
\UnaryInfC{$P(a), \neg P(a), \neg P(a) \vdash Q(a), Q(a)$}
\RightLabel{c_l}
\UnaryInfC{$P(a), \neg P(a) \vdash Q(a), Q(a)$}
\RightLabel{c_r}
\UnaryInfC{$P(a), \neg P(a) \vdash Q(a)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$P(a), \neg P(a) \vdash (\exists z)Q(z)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$P(a), (\forall x)\neg P(x) \vdash (\exists z)Q(z)$}
\AxiomC{$Q(b) \vdash Q(b)$}
\RightLabel{w_r}
\UnaryInfC{$Q(b) \vdash P(b), Q(b)$}
\RightLabel{w_l}
\UnaryInfC{$Q(b), \neg P(b) \vdash P(b), Q(b)$}
\RightLabel{$\neg_l$}
\UnaryInfC{$Q(b), \neg P(b), \neg P(b) \vdash Q(b)$}
\RightLabel{w_r}
\UnaryInfC{$Q(b), \neg P(b), \neg P(b) \vdash Q(b), Q(b)$}
\RightLabel{c_l}
\UnaryInfC{$Q(b), \neg P(b) \vdash Q(b), Q(b)$}
\RightLabel{c_r}
\UnaryInfC{$Q(b), \neg P(b) \vdash Q(b)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$Q(b), \neg P(b) \vdash (\exists z)Q(z)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$Q(b), (\forall x)\neg P(x) \vdash (\exists z)Q(z)$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P(a) \vee Q(b), (\forall x)(\neg P(x)) \vdash (\exists z)Q(z)$}
\end{prooftree}
$$

$\triangle$

CHAPTER 4

Cut-Elimination by Resolution

In this chapter, we will introduce the cut-elimination method **CERES** (*cut-elimination by resolution*), which takes a different approach than reductive methods in additionally using the resolution principle. Section 4.1 outlines the motivation for the development of **CERES** and presents the main steps of the procedure. In Section 4.2, we will define clause terms and prove some of their most important properties. Finally, in Section 4.3, we will formally define the method **CERES** and state some of its most crucial properties.

4.1 Motivation & Overview

In Chapter 3, we have seen how reductive cut-elimination methods work, namely, by eliminating cuts by a stepwise reduction of the complexity of the cut-formula. More precisely, these methods always identify the top-level connective of the cut-formula and either eliminate it directly (grade reduction) or indirectly (rank reduction) [10]. Reductive methods are local in the sense that they focus on a small part of the whole proof only, namely, on the derivation corresponding to the introduction of the top-level connective of the cut-formula [10, 35]. The major drawback of these methods is that they do not take the general structure of the proof into account. Thus, reductive methods have a bad computational behaviour, as many types of redundancies in proofs remain undetected [10].

Baaz and Leitsch introduced the method **CERES** [7], which, as opposed to reductive methods, analyzes the global structure of an **LK**-proof φ (i.e. all cut-derivations in φ are analyzed simultaneously) [10,40]. The most important part of **CERES** is the characteristic clause set—extracted from an **LK**-proof—which depends on the interplay between binary rules that produce ancestors of cut-formulas and those that do not [10].

In [9, 10], it was shown that **CERES** can achieve a nonelementary speed-up over reductive methods. Furthermore, in [9], it was shown that there always exists a resolution refutation γ of the characteristic clause set $\mathrm{CL}(\varphi)$ (obtained by **CERES**), which subsumes the canonic resolution refutation $\mathrm{RES}(\psi)$, where ψ is an **LK**-deduction in **ACNF** obtained from φ by reductive methods. A consequence of the fact that the number of nodes in RES(ψ) is at most exponential in the number of nodes in ψ is that **CERES** yields a deduction χ in **ACNF** that is exponentially bounded by reductive methods [9]. In Chapter 5, we will take a closer look at the complexity of **CERES**.

Moreover, **CERES** may also be used to prove negative results about cut-elimination, e.g. that a certain cut-free proof is not obtainable by a given one. For instance, let ψ be an **ACNF** of a proof φ such that $\mathrm{CL}(\psi)$ is not subsumed by $\mathrm{CL}(\varphi)$, then ψ cannot be obtained by any sequence of cut-reduction rules using reductive methods [10].

CERES was originally developed for classical logic, but—in the meantime—it has also been successfully extended to finitely valued logics [8], Gödel logic [4] and, more recently, to higher-order logic [29, 50] as well as to subclasses of intuitionistic logic [36, 40].

We will now describe the main steps of the method **CERES** in the following:

Let φ be an **LK**-derivation with end-sequent S. Then the method **CERES** consists of the following steps [40, Chapter 3]:

(1) **Skolemization of φ.**
The method requires that the end-sequent contains no occurrence of $\forall$ on the right and $\exists$ on the left (due to the eigenvariable conditions of the corresponding inference rules). In order to achieve this, φ needs to be skolemized, i.e. eigenvariables need to be replaced by so-called Skolem terms. After AC-normalization, the final derivation is transformed into a derivation of the original (unskolemized) end-sequent [9].

(2) **Construction of the characteristic clause set $\mathbf{CL}(\varphi)$.**
Each instance of the cut-rule introduces two copies of a (potentially) new formula (in the bottom-up interpretation) into φ. These two formulas are then gradually decomposed into their atomic subformulas. Some of these atoms may end up in initial sequents of the form $C = C_i \circ C_i'$, where C_i denotes the part of C consisting of atomic cut-ancestors, and C_i' denotes the part of C consisting of ancestors of formulas occurring in the end-sequent. Starting from initial sequents, a set of clauses $\mathrm{CL}(\varphi)$, consisting solely of clauses composed of the C_i, is then constructed in a particular way according to the structure of φ [40].

(3) **Computation of a projection $\varphi(C_i)$ for each $C_i \in \mathbf{CL}(\varphi)$.**
Due to the fact that each $C_i \in \mathrm{CL}(\varphi)$ is a subsequent of some initial sequent in φ, one can obtain a cut-free derivation of a sequent $S \circ C_i$, where S is the end-sequent of φ, for each $C_i \in \mathrm{CL}(\varphi)$. This is achieved by skipping all inferences that operate on cut-ancestors and possibly introducing some additional weakenings in order to obtain all formulas of S. As a consequence, the atoms of C_i remain unchanged throughout φ. The projection $\varphi(C_i)$ of C_i is then given by the derivation of the sequent $S \circ C_i$ [9, 40].

(4) **Construction of a resolution refutation γ of $\mathbf{CL}(\varphi)$.**
One can show that the characteristic clause set $\mathrm{CL}(\varphi)$ is always unsatisfiable [10].

This means that there always exists a resolution refutation γ of $\mathrm{CL}(\varphi)$ by the completeness of the resolution calculus [34, 42]. More precisely, such a refutation corresponds to a derivation of the empty sequent $\vdash$ from the clauses in $\mathrm{CL}(\varphi)$. By applying a ground projection to γ, a ground resolution refutation γ' of $\mathrm{CL}(\varphi)$ is obtained [9].

(5) **Merging the projections $\varphi(C_i)$ and the ground resolution refutation γ'.**
The last step of **CERES** consists of bringing the projections $\varphi(C_i)$ and the ground resolution refutation γ' of $\mathrm{CL}(\varphi)$ together. This is done by applying the ground substitution σ (which defines the ground projection γ') to each projection $\varphi(C_i)$ and placing $\varphi(C_i)\sigma$ immediately above the initial sequents in γ' that correspond to the same $C_i \in \mathrm{CL}(\varphi)$. After merging the projections and γ', we obtain an **LK**-derivation of S that contains only atomic cuts, as the resolution steps in γ' can be considered as atomic cuts in **LK**. Note that some contractions might be necessary in order to obtain an **LK**-derivation of S [9, 40].

4.2 Clause Terms

The information present in the axioms refuted by the cuts will be represented by a set of clauses. Every proof φ with cuts can be transformed into a proof φ' of the empty sequent by skipping inferences going into the end-sequent. The axioms of this refutation φ' can be compactly represented by clause terms [10].

The following definition is based on [10, Definition 6.3.1].

Definition 4.2.1 (Clause Term). Clause terms are $\{\oplus, \otimes\}$-terms over clause sets.
More formally:

- (Finite) sets of clauses are clause terms.
- If X and Y are clause terms, then $X \oplus Y$ is a clause term.
- If X and Y are clause terms, then $X \otimes Y$ is a clause term.

$\triangle$

Definition 4.2.2 is taken from [10, Definition 6.3.2].

Definition 4.2.2 (Semantics of Clause Terms). We define a mapping $|\cdot|$ from clause terms to sets of clauses in the following way:

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

where $\mathcal{C} \times \mathcal{D} = \{C \circ D \mid C \in \mathcal{C}, D \in \mathcal{D}\}$. $\triangle$

We follow [10] in defining:

Two clause terms X and Y are said to be *equivalent* if the corresponding sets of clauses are equal, i.e. $X \sim Y$ iff $|X| = |Y|$.

Clause terms are binary trees whose nodes are finite sets of clauses (instead of constants or variables). Therefore, term occurrences are defined in the same way as for ordinary terms. When speaking about occurrences in clause terms, we only consider nodes in *this* term tree, but not occurrences inside the leaves, i.e. within the sets of clauses on the leaves. In contrast, we consider the internal structure of leaves in the concept of substitution:

Definition 4.2.3 ([10], Definition 6.3.3). Let θ be a substitution. We define the application of θ to clause terms as follows:

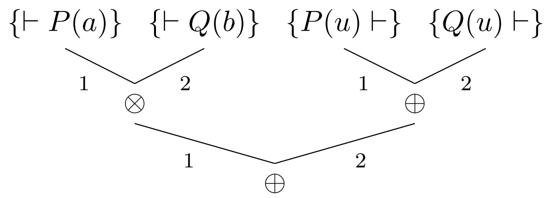
$$\begin{aligned} X\theta &= \mathcal{C}\theta \text{ if } X = \mathcal{C} \text{ for a set of clauses } \mathcal{C}, \\ (X \oplus Y)\theta &= X\theta \oplus Y\theta, \\ (X \otimes Y)\theta &= X\theta \otimes Y\theta. \end{aligned}$$

△

Example 4.2.4. Let X be the following clause term:

$$(\{\vdash P(a)\} \otimes \{\vdash Q(b)\}) \oplus (\{P(u) \vdash\} \oplus \{Q(u) \vdash\}).$$

Then X can be represented in tree form (the numbers on the edges indicate the path from the root to each node in the tree, i.e. the position of each subterm):



Consider, for instance, the following positions in X:

$$\begin{aligned} X.0 &= X, \\ X.(0,1) &= \{\vdash P(a)\} \otimes \{\vdash Q(b)\}, \\ X.(0,2) &= \{P(u) \vdash\} \oplus \{Q(u) \vdash\}, \\ X.(0,1,1) &= \{\vdash P(a)\}, \\ X.(0,1,2) &= \{\vdash Q(b)\}, \\ X.(0,2,1) &= \{P(u) \vdash\}, \\ X.(0,2,2) &= \{Q(u) \vdash\}. \end{aligned}$$

△

There are five binary relations on clause terms, which will play an important role in the subsequent parts of this thesis:

Definition 4.2.5 (cf. [9], Definition 4.4). Let X and Y be clause terms. We define

$$
\begin{array}{rll}
X \subseteq Y & \text{iff} & |X| \subseteq |Y|, \\
X \sqsubseteq Y & \text{iff} & \text{for all } C \in |Y| \text{ there exists a } D \in |X| \text{ such that } D \sqsubseteq C, \\
X \leq_i Y & \text{iff} & \text{there exist clause terms } X_1, X_2 \text{ and a renaming substitution } \vartheta \text{ s.t.} \\
 & & (X_1 \oplus X_2) \subseteq X \text{ and } |(X_1 \oplus X_2) \oplus X_2\vartheta| = |Y|, \\
X \leq_s Y & \text{iff} & \text{there exists a substitution } \theta \text{ such that } X\theta = Y^1, \\
X \leq_{ss} Y & \text{iff} & |X| \leq_{ss} |Y|.
\end{array}
$$

△

Remark. If $Y \subseteq X$, then $X \sqsubseteq Y$. Indeed, assume that $|Y| \subseteq |X|$ (i.e. every clause in $|Y|$ is also a clause in $|X|$); then, for every $C \in |Y|$, there exists a $D \in |X|$ (namely C itself) such that $D \sqsubseteq C$ [10].

The operators $\oplus$ and $\otimes$ are compatible with the relations $\subseteq$ and $\sqsubseteq$. This is formally proved in the following lemmas [9]:

Lemma 4.2.6 ([9], Lemma 4.1). *Let X, Y, Z be clause terms and $X \subseteq Y$. Then*

(i) $X \oplus Z \subseteq Y \oplus Z$,

(ii) $Z \oplus X \subseteq Z \oplus Y$,

(iii) $X \otimes Z \subseteq Y \otimes Z$,

(iv) $Z \otimes X \subseteq Z \otimes Y$.

PROOF. (ii) follows from (i) because $\oplus$ is commutative, i.e. $X \oplus Z \sim Z \oplus X$. The cases (iii) and (iv) are analogous. Thus, we only prove (i) and (iii).

(i) $|X \oplus Z| = |X| \cup |Z| \subseteq |Y| \cup |Z| = |Y \oplus Z|$.

(iii) Let $C \in |X \otimes Z|$. Then there exist clauses D, E with $D \in |X|, E \in |Z|$ and $C = D \circ E$. Clearly, D is also in $|Y|$ and thus $C \in |Y \otimes Z|$.

□

Lemma 4.2.7 ([9], Lemma 4.2). *Let X, Y, Z be clause terms and $X \sqsubseteq Y$. Then*

(i) $X \oplus Z \sqsubseteq Y \oplus Z$,

(ii) $Z \oplus X \sqsubseteq Z \oplus Y$,

(iii) $X \otimes Z \sqsubseteq Y \otimes Z$,

(iv) $Z \otimes X \sqsubseteq Z \otimes Y$.

PROOF. (i) and (ii) are trivial, (iii) and (iv) are analogous. Thus, we only prove (iv): Let $C \in |Z \otimes Y|$. Then $C \in |Z| \times |Y|$ and there exist $D \in |Z|$ and $E \in |Y|$ such that $C = D \circ E$.

[1] Note that $\leq_s$ is defined directly on the syntax of clause terms and not on the semantics [9].

By definition of $\sqsubseteq$, there exists an $E' \in |X|$ with $E' \sqsubseteq E$. This implies $D \circ E' \in |Z \otimes X|$ and $D \circ E' \sqsubseteq D \circ E$. So, $Z \otimes X \sqsubseteq Z \otimes Y$. $\square$

Lemma 4.2.8. *Let X, Y, Z be clause terms and $X \leq_i Y$. Then*

(i) $X \oplus Z \leq_i Y \oplus Z$,

(ii) $Z \oplus X \leq_i Z \oplus Y$,

PROOF. (ii) follows from (i) because $\oplus$ is commutative, i.e. $X \oplus Z \sim Z \oplus X$. Thus, we only prove (i):

Assume $X \leq_i Y$, i.e. there are clause terms X_1, X_2 and a renaming substitution ϑ such that $(X_1 \oplus X_2) \subseteq X$ and $|(X_1 \oplus X_2) \oplus X_2\vartheta| = |Y|$. Then, clearly, $(X_1 \oplus X_2) \oplus Z \subseteq X \oplus Z$. By associativity and commutativity of $\oplus$, we get

$$(X_1 \oplus X_2) \oplus Z \sim (X_1 \oplus Z) \oplus X_2.$$

Therefore, $|((X_1 \oplus Z) \oplus X_2) \oplus X_2\vartheta| = |Y \oplus Z|$, i.e. $X \oplus Z \leq_i Y \oplus Z$. $\square$

The original proof of the following (unpublished) lemma is due to Alexander Leitsch:

Lemma 4.2.9. *Let $X \leq_i Y$ via a renaming substitution ϑ such that the variables in the domain and range of ϑ do not occur in Z. Then $X \otimes Z \leq_i Y \otimes Z$ via ϑ.*

PROOF. Assume $X \leq_i Y$ via a renaming substitution ϑ such that the variables in the domain and range of ϑ do not occur in Z. Then there are clause terms X_1, X_2 such that $(X_1 \oplus X_2) \subseteq X$ and $|(X_1 \oplus X_2) \oplus X_2\vartheta| = |Y|$. Observe that $(X_1 \oplus X_2) \otimes Z \subseteq X \otimes Z$ and

$$(X_1 \oplus X_2) \otimes Z \sim (X_1 \otimes Z) \oplus (X_2 \otimes Z),$$
$$((X_1 \oplus X_2) \oplus X_2\vartheta) \otimes Z \sim (X_1 \otimes Z) \oplus (X_2 \otimes Z) \oplus (X_2\vartheta \otimes Z).$$

But, by the restrictions on ϑ w.r.t. Z, we have $(X_2 \otimes Z)\vartheta = (X_2\vartheta \otimes Z)$, i.e.

$$|(X_1 \otimes Z) \oplus (X_2 \otimes Z) \oplus (X_2 \otimes Z)\vartheta| = |Y \otimes Z|.$$

Therefore, $X \otimes Z \leq_i Y \otimes Z$ via ϑ. $\square$

Remark. In general, $X \leq_i Y$ does not imply $X \otimes Z \leq_i Y \otimes Z$, i.e. $\leq_i$ is not compatible with $\otimes$ for arbitrary substitutions such that $X \leq_i Y$. Consider, for instance, the terms

$$\begin{aligned} X &= \{\vdash Q(y)\} \oplus \{\vdash P(x)\}, \\ Y &= \{\vdash Q(y)\} \oplus \{\vdash P(x)\} \oplus \{\vdash P(v)\}, \\ Z &= \{R(x) \vdash\}. \end{aligned}$$

Then we get

$$\begin{aligned} X \otimes Z &\sim \{R(x) \vdash Q(y)\} \oplus \{R(x) \vdash P(x)\}, \\ Y \otimes Z &\sim \{R(x) \vdash Q(y)\} \oplus \{R(x) \vdash P(x)\} \oplus \{R(x) \vdash P(v)\}. \end{aligned}$$

Let $\vartheta = \{x \leftarrow v\}$, then, clearly, $X \leq_i Y$ because $\{\vdash P(x)\}\vartheta = \{\vdash P(v)\}$. Thus,

$$|\{\vdash Q(y)\} \oplus \{\vdash P(x)\} \oplus \{\vdash P(x)\}\vartheta| = |\{\vdash Q(y)\} \oplus \{\vdash P(x)\} \oplus \{\vdash P(v)\}| = |Y|.$$

But

$$|\{R(x) \vdash Q(y)\} \oplus \{R(x) \vdash P(x)\} \oplus \{R(x) \vdash P(x)\}\vartheta| \neq |Y \otimes Z|$$

because $\{R(x) \vdash P(x)\}\vartheta = \{R(v) \vdash P(v)\} \neq \{R(x) \vdash P(v)\}$. Hence, $X \otimes Z \not\leq_i Y \otimes Z$. Note that this problem occurs for all substitutions ϑ, where variables in the domain or the range of ϑ also occur in Z.

Definition 4.2.10. We define $X \preceq Y$ if at least one of the following relations hold:

- $Y \subseteq X$,
- $X \sqsubseteq Y$,
- $X \leq_i Y$.

$\triangle$

We are now able to show that replacing subterms in a clause term preserves the relations $\subseteq$ and $\sqsubseteq$ (cf. [9]).

Lemma 4.2.11 ([9], Lemma 4.3). *Let λ be a position in a clause term X, and $Y \lesssim X.\lambda$, for $\lesssim \in \{\subseteq, \sqsubseteq\}$. Then $X[Y]_\lambda \lesssim X$.*

PROOF. We proceed by induction on the term-complexity n (i.e. the number of nodes) of X.

BASE CASE: $n = 1$. Then X is a set of clauses and λ is the top position, i.e. $X.\lambda = X$. Consequently, $X[Y]_\lambda = Y$ and thus $X[Y]_\lambda \lesssim X$.

INDUCTION HYPOTHESIS (IH): The claim holds for all clause terms X with term-complexity $k \leq n$.

INDUCTION STEP: Let X be $X_1 \odot X_2$, for $\odot \in \{\oplus, \otimes\}$. If λ is the top position in X, then the lemma trivially holds. Thus, we may assume that λ is a position in X_1 or in X_2. We consider the case that λ is in X_1 (the other one is completely symmetric): then there exists a position μ in X_1 such that $X.\lambda = X_1.\mu$. Since $Y \lesssim X_1.\mu$, we get by **(IH)** that $X_1[Y]_\mu \lesssim X_1$. By Lemmas 4.2.6 and 4.2.7 we obtain

$$X_1[Y]_\mu \odot X_2 \lesssim X_1 \odot X_2.$$

But

$$X_1[Y]_\mu \odot X_2 = (X_1 \odot X_2)[Y]_\lambda = X[Y]_\lambda,$$

and therefore $X[Y]_\lambda \lesssim X$. $\square$

Lemma 4.2.12. *Let λ be an occurrence in a clause term X and $X.\lambda \lesssim Y$, for $\lesssim \in \{\subseteq, \sqsubseteq\}$. Then $X \lesssim X[Y]_\lambda$.*

PROOF. We proceed by induction on the term-complexity n (i.e. the number of nodes) of X.

BASE CASE: $n = 1$. Then X is a set of clauses and λ is the top position, i.e. $X.\lambda = X$. Consequently, $X[Y]_\lambda = Y$, and thus $X \subseteq Y$ and $X \sqsubseteq Y$.

INDUCTION HYPOTHESIS (IH): The claim holds for all clause terms X with term-complexity $k \leq n$.

INDUCTION STEP: Let X be $X_1 \odot X_2$, for $\odot \in \{\oplus, \otimes\}$. If λ is the top position in X, then the lemma trivially holds. Thus, we may assume that λ is a position in X_1 or in X_2. We consider the case that λ is in X_1 (the other one is completely symmetric): then there exists a position μ in X_1 such that $X.\lambda = X_1.\mu$. Since $X_1.\mu \lesssim Y$, we get by **(IH)** that $X_1 \lesssim X_1[Y]_\mu$. By Lemmas 4.2.6 and 4.2.7 we obtain

$$X_1 \odot X_2 \lesssim X_1[Y]_\mu \odot X_2.$$

But

$$X_1[Y]_\mu \odot X_2 = (X_1 \odot X_2)[Y]_\lambda = X[Y]_\lambda.$$

and therefore, $X \lesssim X[Y]_\lambda$. □

The replacement of subterms in a clause term preserves the relation $\leq_i$ only in specific cases:

Lemma 4.2.13. *Let λ be an occurrence in a clause term $X = X_1 \odot X_2$, for $\odot \in \{\oplus, \otimes\}$ and $X.\lambda \leq_i Y$ (if $X = X_1 \otimes X_2$, then we assume $X.\lambda \leq_i Y$ via a renaming substitution ϑ such that the variables in the domain and range of ϑ do not occur in X_2). Then $X \leq_i X[Y]_\lambda$.*

PROOF. We proceed by induction on the term-complexity n (i.e. the number of nodes) of X.

BASE CASE: $n = 1$. Then X is a set of clauses and λ is the top position, i.e. $X.\lambda = X$. Consequently, $X[Y]_\lambda = Y$ and thus $X \leq_i Y$.

INDUCTION HYPOTHESIS (IH): The claim holds for all clause terms X with term-complexity $k \leq n$.

INDUCTION STEP: Let X be $X_1 \odot X_2$, for $\odot \in \{\oplus, \otimes\}$. If λ is the top position in X, then the lemma trivially holds. Thus, we may assume that λ is a position in X_1 or in X_2. We consider the case that λ is in X_1 (the other one is completely symmetric): then there exists a position μ in X_1 such that $X.\lambda = X_1.\mu$. Since $X_1.\mu \leq_i Y$, we get by **(IH)** that $X_1 \leq_i X_1[Y]_\mu$. By assumption we have $X_2\vartheta = X_2$ (if $X = X_1 \otimes X_2$), thus Lemmas 4.2.8

and 4.2.9 yield

$$X_1 \odot X_2 \leq_i X_1[Y]_\mu \odot X_2.$$

But

$$X_1[Y]_\mu \odot X_2 = (X_1 \odot X_2)[Y]_\lambda = X[Y]_\lambda.$$

and therefore, $X \leq_i X[Y]_\lambda$. □

The relations $\preceq$ and $\leq_s$ together define a relation $\rhd$:

Definition 4.2.14 (cf. [9], Definition 4.5). Let X and Y be two clause terms. We define $X \rhd Y$ if (at least) one of the following properties is fulfilled:

(i) $X \preceq Y$, or

(ii) $X \leq_s Y$.

△

The following remark is due to Alexander Leitsch:

Remark. In general $Y \leq_s Z$ does not imply $X[Y]_\lambda \leq_{ss} X[Z]_\lambda$, i.e. $\leq_s$ is not compatible with $\oplus$ and $\otimes$ for subsumption. Consider, for instance, the terms

$$Y = \{\vdash P(x)\}, Z = \{\vdash P(f(x))\} \text{ and}$$
$$X = \{\vdash Q(x)\} \otimes \{\vdash R(x)\}, X.\lambda = \{\vdash Q(x)\}.$$

Clearly, $Y \leq_s Z$. By replacement and evaluation, we obtain

$$|X[Y]_\lambda| = \{\vdash P(x), R(x)\}, |X[Z]_\lambda| = \{\vdash P(f(x)), R(x)\}.$$

Obviously, $X[Y]_\lambda \not\leq_{ss} X[Z]_\lambda$.

The transitive closure $\rhd^*$ of $\rhd$ can be considered as a weak form of subsumption [9]:

Proposition 4.2.15 (cf. [9], Proposition 4.1). *Let X and Y be clause terms such that $X \rhd^* Y$. Then $X \leq_{ss} Y$.*

PROOF. As the relation $\leq_{ss}$ is reflexive and transitive (see [34, Proposition 4.2.1]), it suffices to show that $\rhd$ is a sub-relation of $\leq_{ss}$.

(i) Assume $Y \subseteq X$. Then $X \leq_{ss} Y$ is trivial.

(ii) Assume $X \sqsubseteq Y$, i.e. for all $C \in |Y|$ there exists a $D \in |X|$ with $D \sqsubseteq C$. But then also $D \leq_{ss} C$. The definition of $\leq_{ss}$ for sets finally yields $X \leq_{ss} Y$.

(iii) Assume $X \leq_i Y$, i.e. there are clause terms X_1, X_2 and a renaming substitution ϑ such that $X_1 \oplus X_2 \subseteq X$ and $|(X_1 \oplus X_2) \oplus X_2\vartheta| = |Y|$. By semantics of $\oplus$, we get $|X_1 \oplus X_2| \cup |X_2\vartheta| = |Y|$ and $|X_1 \oplus X_2| = |X_1| \cup |X_2| \subseteq |X|$.

Now, let $C \in |Y|$ be arbitrary. We distinguish the following cases:

(a) If $C \in |X_1 \oplus X_2|$, then $C \in |X|$ and $C \leq_{ss} C$ by reflexivity[2] of $\leq_{ss}$.

(b) If $C \notin |X_1 \oplus X_2|$, then, since $|X_1 \oplus X_2| \cup |X_2\vartheta| = |Y|$, $C \in |X_2\vartheta|$. But this means that there is some $C' \in |X_2| \subseteq |X|$ such that $C'\vartheta = C$ and thus $C' \leq_{ss} C$.

Therefore, in both cases, $X \leq_{ss} Y$.

(iv) Assume $X \leq_s Y$. Then $X \leq_{ss} Y$ is trivial.

□

4.3 The Method CERES

The following two definitions correspond to [10, Definition 3.1.15] and [10, Definition 3.1.16].

Definition 4.3.1 (Polarity). Let λ be an occurrence of a formula A in B.

- If $A = B$, then λ is a positive occurrence in B.
- If $B = (C \odot D)$ or $B = (Qx)C$, for $\odot \in \{\wedge, \vee\}, Q \in \{\forall, \exists\}$ and λ is a positive (negative) occurrence of A in C (or in D, respectively), then the corresponding occurrence λ' of A in B is positive (negative).
- If $B = \neg C$ and λ is a positive (negative) occurrence of A in C, then the corresponding occurrence λ' of A in B is negative (positive).

If there exists a positive (negative) occurrence of a formula A in B, we say that A is of positive (negative) *polarity* in B. △

Definition 4.3.2 (Strong and Weak Quantifierss). If $(\forall x)$ occurs positively (negatively) in B, then $(\forall x)$ is called a *strong (weak) quantifier*.

If $(\exists x)$ occurs positively (negatively) in B, then $(\exists x)$ is called a *weak (strong) quantifier*. △

We define skolemization as in [10, Definition 6.2.1]:

Definition 4.3.3 (Skolemization). The function sk maps closed formulas into closed formulas; it is defined in the following way:

$$\mathrm{sk}(F) = F \text{ if } F \text{ does not contain strong quantifiers.}$$

Otherwise, assume that (Qy) is the first strong quantifier in F (in a tree ordering), which is in the scope of the weak quantifiers $(Q_1x_1), \dots, (Q_nx_n)$ (appearing in this order). Let f be an n-ary function symbol not occurring in F (f is a constant symbol for $n = 0$). Then $\mathrm{sk}(F)$ is inductively defined as

$$\mathrm{sk}(F) = \mathrm{sk}(F_{(Qy)}\{y \leftarrow f(x_1, \dots, x_n)\}),$$

[2] For reflexivity, it suffices to choose the empty substitution.

where $F_{(Qy)}$ is F after omission of (Qy). We call $\mathrm{sk}(F)$ the (structural) *Skolemization* of F. $\triangle$

In model theory and automated deduction the definition of Skolemization is mostly dual to Definition 4.3.3, i.e. in the case of prenex forms, the existential quantifiers are eliminated instead of the universal ones. We call this kind of Skolemization *refutational Skolemization* [10]. The dual kind of Skolemization (elimination of universal quantifiers) is frequently called "Herbrandization" [32]. The Skolemization of sequents, defined below, yields a more general framework, covering both concepts [10].

The following definition constitutes a modification of [10, Definition 6.2.2].

Definition 4.3.4 (Skolemization of Sequents). Let $S = A_1, \ldots, A_n \vdash B_1, \ldots, B_m$ be a sequent consisting of closed formulas only and

$$\mathrm{sk}(\neg(A_1 \wedge \ldots \wedge A_n) \vee (B_1 \vee \ldots \vee B_m)) = \neg(A'_1 \wedge \ldots \wedge A'_n) \vee (B'_1 \vee \ldots \vee B'_m).$$

Then the sequent

$$S' = A'_1, \ldots, A'_n \vdash B'_1, \ldots, B'_m$$

is called the *Skolemization* of S. $\triangle$

Example 4.3.5 ([10], Example 6.2.1)**.** Let $S = (\forall x)(\exists y)P(x,y) \vdash (\forall x)(\exists y)P(x,y)$ be a sequent. Then the Skolemization of S is $S' = (\forall x)P(x, f(x)) \vdash (\exists y)P(c,y)$, for a unary function symbol f and a constant symbol c. Note that the Skolemization of the left-hand side of the sequent corresponds to the refutational Skolemization concept of formulas. $\triangle$

By a *skolemized proof*, we mean a proof of the skolemized end-sequent. Also proofs with cuts can be skolemized, but the cut-formulas themselves cannot. Only the strong quantifiers, which are ancestors of the end-sequent, are eliminated [10].

We restrict AC-normalizations (and $\mathrm{AC}^{\mathsf{top}}$-normalizations) to derivations with skolemized end-sequents (cf. [9]). It is always possible to construct derivations of skolemized end-sequents from the original ones without increase of length (see [5, 6, 10]).

After AC-normalization (or $\mathrm{AC}^{\mathsf{top}}$-normalization) the derivation can be transformed into a derivation of the original (unskolemized) sequent (cf. [9]).

Definition 4.3.6 ([9], Definition 5.1). Let $\mathcal{SK}$ be the set of all **LK**-derivations with skolemized end-sequents. $\mathcal{SK}_\emptyset$ is the set of all cut-free proofs in $\mathcal{SK}$, and, for all $i \geq 0$, $\mathcal{SK}^i$ is the subset of $\mathcal{SK}$ containing all derivations with cut-formulas of complexity $\leq i$. $\triangle$

The aim of **CERES** is to transform a proof in $\mathcal{SK}$ into a proof in $\mathcal{SK}^0$. As already mentioned in the overview, the first step consists of the definition of a clause term corresponding to the subderivations of an **LK**-proof ending in a cut. In particular, we focus on derivations of the cut-formulas themselves, i.e. on the derivation of formulas having no successors in the end-sequent [9].

Definition 4.3.7 is based on [9, Definition 5.2] and [10, Definition 6.4.1].

Definition 4.3.7 (Characteristic Term). Let φ be an **LK**-derivation of S, and let Ω be the set of all occurrences of cut-formulas in φ. We define the *characteristic (clause) term* $\Theta(\varphi)$ inductively as follows:

Let ν be the occurrence of an initial sequent S' in φ. Moreover, let S'' be the subsequent of S' consisting of all atoms, which are ancestors of an occurrence in Ω, i.e. $S'' = S(\nu, \Omega)$. Then $\Theta(\varphi)/\nu = \{S''\}$.

Let us assume that the clause terms $\Theta(\varphi)/\nu$ are already constructed for all sequent occurrences ν in φ with $depth(\nu) \leq k$. Now, let ν be an occurrence with $depth(\nu) = k+1$. We distinguish the following cases:

(a) ν is the conclusion of μ, i.e. a unary rule applied to μ gives ν. Here, we simply define $\Theta(\varphi)/\nu = \Theta(\varphi)/\mu$.

(b) ν is the conclusion of μ_1 and μ_2, i.e. a binary rule ξ applied to μ_1 and μ_2 gives ν.

  (b1) The auxiliary formulas of ξ are ancestors of Ω, i.e. the formulas occur in $S(\mu_1, \Omega)$ and $S(\mu_2, \Omega)$, respectively. Then $\Theta(\varphi)/\nu = \Theta(\varphi)/\mu_1 \oplus \Theta(\varphi)/\mu_2$.

  (b2) The auxiliary formulas of ξ are *not* ancestors of Ω. In this case, we define $\Theta(\varphi)/\nu = \Theta(\varphi)/\mu_1 \otimes \Theta(\varphi)/\mu_2$.

Note that, in a binary inference, either both auxiliary formulas are ancestors of Ω or none of them.

Finally, the characteristic term $\Theta(\varphi)$ is defined as $\Theta(\varphi)/\nu$, where ν is the occurrence of the end-sequent. $\triangle$

Remark. If φ is a *cut-free* proof, then there are no occurrences of cut-formulas in φ, and thus $|\Theta(\varphi)| = \{\vdash\}$ [9].

The definition of a characteristic clause set is taken from [9].

Definition 4.3.8 (Characteristic Clause Set). Let φ be an **LK**-derivation and $\Theta(\varphi)$ be the characteristic term of φ. Then $\mathrm{CL}(\varphi)$, for $\mathrm{CL}(\varphi) = |\Theta(\varphi)|$, is called the *characteristic clause set* of φ. $\triangle$

Example 4.3.9 (cf. [35], Example 4)**.** The structure of the following example is based on [9, Example 5.1]. Let φ be the derivation (for u a free variable, a, b constant symbols)

$$\dfrac{\begin{matrix}(\varphi_1) \\ P(a) \lor Q(b) \vdash (\exists y)(P(y) \lor Q(y))^*\end{matrix} \qquad \begin{matrix}(\varphi_2) \\ (\exists y)(P(y) \lor Q(y))^*, (\forall x)\neg P(x) \vdash (\exists z)Q(z)\end{matrix}}{P(a) \lor Q(b), (\forall x)(\neg P(x)) \vdash (\exists z)Q(z)}\ cut$$

where φ_1 is the **LK**-derivation:

$$\dfrac{\dfrac{\dfrac{P(a) \vdash P(a)^*}{P(a) \vdash (P(a) \vee Q(a))^*}\,\vee_{r_1}}{P(a) \vdash (\exists y)(P(y) \vee Q(y))^*}\,\exists_r \qquad \dfrac{\dfrac{Q(b) \vdash Q(b)^*}{Q(b) \vdash (P(b) \vee Q(b))^*}\,\vee_{r_2}}{Q(b) \vdash (\exists y)(P(y) \vee Q(y))^*}\,\exists_r}{P(a) \vee Q(b) \vdash (\exists y)(P(y) \vee Q(y))^*}\,\vee_l$$

and φ_2 is the **LK**-derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(u)^* \vdash P(u)}{P(u)^*, \neg P(u) \vdash}\,\neg_l}{P(u)^*, \neg P(u) \vdash Q(u)}\,w_r \qquad \dfrac{Q(u)^* \vdash Q(u)}{Q(u)^*, \neg P(u) \vdash Q(u)}\,w_l}{(P(u) \vee Q(u))^*, \neg P(u) \vdash Q(u)}\,\vee_l}{(P(u) \vee Q(u))^*, \neg P(u) \vdash (\exists z)Q(z)}\,\exists_r}{(P(u) \vee Q(u))^*, (\forall x)\neg P(x) \vdash (\exists z)Q(z)}\,\forall_l}{(\exists y)(P(y) \vee Q(y))^*, (\forall x)\neg P(x) \vdash (\exists z)Q(z)}\,\exists_l$$

Let Ω be the set of the two occurrences of the cut-formula in φ. The ancestors of Ω are labelled with $*$. We compute the characteristic clause term $\Theta(\varphi)$:

From the $*$-labels in φ, we first get the clause terms corresponding to the initial sequents:

$$X_1 = \{\vdash P(a)\}, X_2 = \{\vdash Q(b)\}, X_3 = \{P(u) \vdash\} \text{ and } X_4 = \{Q(u) \vdash\}.$$

As unary inferences do not change the clause term, the first (= uppermost) binary inference that we consider in φ_1 is $\vee_l$. Since the auxiliary formulas of this inference are not ancestors of Ω (the auxiliary formulas are not labelled with $*$), we obtain the term

$$Y_1 = X_1 \otimes X_2 = \{\vdash P(a)\} \otimes \{\vdash Q(b)\}.$$

Furthermore, since $\vee_l$ is the last inference in φ_1, we get

$$\Theta(\varphi)/\nu_1 = Y_1,$$

where ν_1 is the position of the end-sequent of φ_1 in φ.

Since X_3 and X_4 are not changed by the unary inferences above $\vee_l$ in φ_2 (i.e. the uppermost binary inference in φ_2) and the auxiliary formulas of $\vee_l$ are ancestors of Ω (they are labelled with $*$), we have to apply $\oplus$ to X_3 and X_4:

$$Y_2 = X_3 \oplus X_4 = \{P(u) \vdash\} \oplus \{Q(u) \vdash\}.$$

As all inferences below $\vee_l$ are unary, the clause term remains unchanged. Consequently, if ν_2 is the position of the end-sequent of φ_2 in φ, then the corresponding clause term is

$$\Theta(\varphi)/\nu_2 = Y_2.$$

The last inference in φ is a cut, and since the auxiliary formulas of the cut-rule are always ancestors of Ω, we have to apply $\oplus$ to Y_1 and Y_2. This eventually gives the characteristic clause term of φ:

$$\Theta(\varphi) = \Theta(\varphi)/\nu = Y_1 \oplus Y_2 = (\{\vdash P(a)\} \otimes \{\vdash Q(b)\}) \oplus (\{P(u) \vdash\} \oplus \{Q(u) \vdash\}),$$

where ν is the position of the end-sequent of φ.

The corresponding characteristic clause set is the given by

$$\mathrm{CL}(\varphi) = |\Theta(\varphi)| = \{\vdash P(a), Q(b);\ P(u) \vdash;\ Q(u) \vdash\}.$$

$\triangle$

It is easy to see that—in the above example—the characteristic clause set $\mathrm{CL}(\varphi)$ is unsatisfiable. The following proposition shows that this is not a coincidence, but a general property of characteristic clause sets constructed in this way [9]:

Proposition 4.3.10. *Let $\varphi \in \mathcal{SK}$ be an **LK**-derivation. Then $\mathrm{CL}(\varphi)$ is unsatisfiable.*

PROOF. In [7, 10]. $\square$

We follow [9]:

Let $\varphi \in \mathcal{SK}$ be a deduction of $S = \Gamma \vdash \Delta$ and $\mathrm{CL}(\varphi)$ be the characteristic clause set of φ. Then $\mathrm{CL}(\varphi)$ is unsatisfiable, and, by Theorem 2.3.18, there exists a resolution refutation γ of $\mathrm{CL}(\varphi)$. By applying a ground projection to γ, we obtain a ground resolution refutation γ' of $\mathrm{CL}(\varphi)$; by our definition of resolution, γ' is also an AC-deduction of $\vdash$ from (ground instances of) $\mathrm{CL}(\varphi)$. This deduction γ' may serve as a skeleton of an AC-deduction ψ of $\Gamma \vdash \Delta$ itself. The construction of ψ from γ' is based on so-called *projections* replacing φ by cut-free deductions $\varphi(C)$ of $\overline{P}, \Gamma \vdash \Delta, \overline{Q}$, for clauses $C = \overline{P} \vdash \overline{Q}$ in $\mathrm{CL}(\psi)$. We merely give an informal description of the projections, for details we refer to [7, 10]. Roughly speaking, the projections of the proof φ are obtained by skipping all inferences leading to a cut. As a "residue", we obtain a characteristic clause in the end-sequent. Thus, a projection is a cut-free derivation of the end-sequent S + some atomic formulas in S. For the application of projections, it is vital to have a skolemized end-sequent, as otherwise eigenvariable conditions could be violated.

Definition 4.3.11 ([9], Definition 5.4). A sequent $\overline{P}' \vdash \overline{Q}'$ is called a *contraction variant* of $\overline{P} \vdash \overline{Q}$ if $set(\overline{P}') = set(\overline{P})$ and $set(\overline{Q}') = set(\overline{Q})$ (i.e. the sequents would be equal if they were defined via sets instead of multisets). $\triangle$

Lemma 4.3.12 ([9], Lemma 5.1). *Let $\varphi \in \mathcal{SK}$ be a deduction of a sequent $S = \Gamma \vdash \Delta$. Furthermore, let $C = \overline{P} \vdash \overline{Q}$ be a clause in $\mathrm{CL}(\varphi)$. Then there exists a deduction $\varphi(C)$ of $\overline{P}', \Gamma \vdash \Delta, \overline{Q}'$ such that $\overline{P}' \vdash \overline{Q}'$ is a contraction variant of $\overline{P} \vdash \overline{Q}$, $\varphi(C) \in \mathcal{SK}_\emptyset$ and $l(\varphi(C)) \leq l(\varphi)$.*

PROOF. In [7]. $\square$

In the following example, we will illustrate how the projection $\varphi(C)$ is constructed.

Example 4.3.13 (cf. [35], Example 4)**.** The structure of the following example is based on [9, Example 5.2]. Let φ be the **LK**-proof of the sequent

$$S = P(a) \vee Q(b), (\forall x)(\neg P(x)) \vdash (\exists z)Q(z)$$

as defined in Example 4.3.9. We have already seen that

$$\mathrm{CL}(\varphi) = \{\underbrace{\vdash P(a), Q(b);}_{C_1}\ \underbrace{P(u) \vdash;}_{C_2}\ \underbrace{Q(u) \vdash}_{C_3}\}.$$

Now, we are able to define $\varphi(C_1)$, the "projection" of φ to C_1:

The problem can be reduced to a projection in φ_1 because the last inference in φ is a cut and

$$|\Theta(\varphi)/\nu_1| = \{\vdash P(a), Q(b)\} = C_1.$$

By skipping all inferences in φ_1 leading to cut-formulas, we obtain the deduction

$$\dfrac{\dfrac{P(a) \vdash P(a)}{P(a) \vdash P(a), Q(b)}\, w_r \qquad \dfrac{Q(b) \vdash Q(b)}{Q(b) \vdash P(a), Q(b)}\, w_r}{P(a) \vee Q(b) \vdash P(a), Q(b)}\, \vee_l$$

In order to obtain the end-sequent S, we only need additional weakenings, thus $\varphi(C_1) =$

$$\dfrac{\dfrac{\dfrac{P(a) \vdash P(a)}{P(a) \vdash P(a), Q(b)}\, w_r \qquad \dfrac{Q(b) \vdash Q(b)}{Q(b) \vdash P(a), Q(b)}\, w_r}{P(a) \vee Q(b) \vdash P(a), Q(b)}\, \vee_l}{\overline{P(a) \vee Q(b), (\forall x)(\neg P(x)) \vdash (\exists z)Q(z), P(a), Q(b)}}\, w_l, w_r$$

For C_2, we obtain the projection $\varphi(C_2)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(u) \vdash P(u)}{P(u), \neg P(u) \vdash}\, \neg_l}{P(u), \neg P(u) \vdash Q(u)}\, w_r}{P(u), \neg P(u) \vdash (\exists z)Q(z)}\, \exists_r}{P(u), (\forall x)(\neg P(x)) \vdash (\exists z)Q(z)}\, \forall_l}{P(u), P(a) \vee Q(b), (\forall x)(\neg P(x)) \vdash (\exists z)Q(z)}\, w_l$$

Similarly, we obtain the projection $\varphi(C_3)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{Q(u) \vdash Q(u)}{Q(u), \neg P(u) \vdash Q(u)}\, w_l}{Q(u), \neg P(u) \vdash (\exists z)Q(z)}\, \exists_r}{Q(u), (\forall x)(\neg P(x)) \vdash (\exists z)Q(z)}\, \forall_l}{Q(u), P(a) \vee Q(b), (\forall x)(\neg P(x)) \vdash (\exists z)Q(z)}\, w_l$$

$\triangle$

In the projections, only inferences on non-ancestors of cut-formulas are performed; if the auxiliary formulas of a binary inference are ancestors of a cut-formula, then we have to apply weakening in order to obtain the required formulas from the other premise [9].

Let $\varphi \in \mathcal{SK}$ be a proof of S, and let γ be a resolution refutation of the (unsatisfiable) set of clauses $\mathrm{CL}(\varphi)$. Then γ can be transformed into a deduction $\varphi(\gamma)$ of S such that $\varphi(\gamma) \in \mathcal{SK}^0$. Moreover, $\varphi(\gamma)$ is a proof containing only atomic cuts, i.e. an AC-normal form of φ. We construct $\varphi(\gamma)$ from γ simply by replacing the resolution steps by the corresponding proof projections. The construction of $\varphi(\gamma)$ is the essential part of the method **CERES** (the final elimination of atomic cuts is inessential). The resolution refutation γ can be considered as the characteristic part of $\varphi(\gamma)$ representing the essential result of AC-normalization. Below, we give an example of the construction of $\varphi(\gamma)$ (for details, we refer to [7, 10]) [9].

Example 4.3.14. We follow the structure of [9, Example 5.3]. Let φ be the **LK**-proof of the sequent

$$S = P(a) \vee Q(b), (\forall x)(\neg P(x)) \vdash (\exists z)Q(z)$$

as defined in Examples 4.3.9 and 4.3.13. Then

$$\mathrm{CL}(\varphi) = \{\underbrace{\vdash P(a), Q(b);}_{C_1}\ \underbrace{P(u) \vdash;}_{C_2}\ \underbrace{Q(u) \vdash}_{C_3}\}.$$

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

$$\dfrac{\dfrac{\vdash P(a), Q(b) \qquad P(u) \vdash}{\vdash Q(b)}\,R \qquad Q(v) \vdash}{\vdash}\,R.$$

Note that we have renamed the variable u to v in $Q(u)$ in order to obtain that P and Q are variable-disjoint.

The corresponding ground resolution refutation γ' is then given by:

$$\dfrac{\dfrac{\vdash P(a), Q(b) \qquad P(a) \vdash}{\vdash Q(b)}\,R \qquad Q(b) \vdash}{\vdash}\,R.$$

The ground substitution defining the ground projection is

$$\sigma = \{u \leftarrow a, v \leftarrow b\}.$$

Now, let $\chi_1 = \varphi(C_1)\sigma$, $\chi_2 = \varphi(C_2)\sigma$ and $\chi_3 = \varphi(C_3)\sigma$. Moreover, let us denote $P(a) \vee Q(b)$ by B, $(\forall x)(\neg P(x))$ by C and $(\exists z)(Q(z))$ by D.

Then $\varphi(\gamma')$ is of the form

$$
\dfrac{\dfrac{\overset{(\chi_1)}{B, C \vdash D, P(a), Q(b)} \qquad \overset{(\chi_2)}{P(a), B, C \vdash D}}{B, B, C, C \vdash D, D, Q(b)}\,cut(P(a)) \qquad \overset{(\chi_3)}{Q(b), B, C \vdash D}}{\dfrac{B, B, B, C, C, C \vdash D, D, D}{B, C \vdash D}\,c_l^*, c_r^*}\,cut(Q(b)).
$$

$\triangle$

CHAPTER 5

Complexity Analysis of CERES

This chapter is concerned with analyzing the computational complexity of the method **CERES** and with an asymptotic comparison of reductive methods and **CERES**. Section 5.1 introduces the concept of a canonic resolution refutation as a means for comparing normal forms under reductive methods and under **CERES**. In Section 5.2, we will show that reductive methods (based on $\mathcal{R}$) are redundant w.r.t. **CERES**, as the characteristic clause set of an **ACNF** under reductive methods is subsumed by the one of the original proof. We conclude the chapter in Section 5.3, where the asymptotic comparison of reductive methods and **CERES** will show that the latter has a nonelementary speed-up over the former, but not the other way round.

5.1 Canonic Resolution Refutations

In this section, we define the notion of a *canonic resolution refutation*. If ψ is a deduction in AC-normal form, then there exists a "canonic" resolution refutation $\mathsf{RES}(\psi)$ of the set of clauses $\mathrm{CL}(\psi)$. $\mathsf{RES}(\psi)$ is "the" resolution proof corresponding to ψ. Indeed, as ψ is a deduction with atomic cuts only, the part of ψ ending in the cut-formulas is nothing else than a p-resolution refutation. For the construction of $\mathsf{RES}(\psi)$ we need some technical definitions [9]:

Definition 5.1.1 ([9], Definition 5.5). Let γ be a p-resolution deduction of a clause C from a set of clauses $\mathcal{C}$, and let D be a clause. We define a p-resolution deduction $\gamma(D)$ of $D \circ C$ from $\{D\} \times \mathcal{C}$ in the following way:

(i) Construct a deduction γ' by replacing all initial clauses S in γ by $D \circ S$, and leave the inference nodes unchanged.

(ii) Apply contractions and weakenings to the end-clause of γ' (if necessary) in order to obtain a deduction $\gamma(D)$ of $D \circ C$ from $\{D\} \times \mathcal{C}$.

$\triangle$

Remark. Contractions may become necessary, as the occurrence of D in clauses may be multiplied by resolutions γ'. Weakenings are required if atoms in D are cut out by resolutions in γ' [9].

Example 5.1.2 (cf. [10], Example 6.7.1). Let $\gamma =$

$$\dfrac{P(a) \vdash R(x) \qquad R(x) \vdash Q(x)}{P(a) \vdash Q(x)}\ R$$

and $D = R(x) \vdash S(x)$. Then $\gamma(D) =$

$$\dfrac{\dfrac{\dfrac{R(x), P(a) \vdash S(x), R(x) \qquad R(x), R(x) \vdash S(x), Q(x)}{R(x), P(a), R(x) \vdash S(x), S(x), Q(x)}\ R}{R(x), P(a) \vdash S(x), S(x), Q(x)}\ c_l}{R(x), P(a) \vdash S(x), Q(x)}\ c_r$$

$\triangle$

Definition 5.1.3 ([9], Definition 5.6). Let γ be a p-resolution deduction of C from $\mathcal{C}$, and let δ be a p-resolution deduction of D from $\mathcal{D}$. We define a p-resolution deduction $\gamma \odot \delta$ of $C \circ D$ from $\mathcal{C} \times \mathcal{D}$ in the following way:

(i) Construct a deduction η by replacing all initial clauses S in γ by the deductions $\delta(S)$ of $D \circ S$, and leave the inference nodes in γ unchanged.

(ii) Apply contractions and weakenings to the end-clause of η (if needed) in order to obtain the deduction $\gamma \odot \delta$ of $D \circ C$.

$\triangle$

Remark. $\gamma \odot \delta$ is indeed a p-resolution deduction from $\mathcal{C} \times \mathcal{D}$, as the initial clauses are of the form $S \circ S'$, for $S \in \mathcal{C}$ and $S' \in \mathcal{D}$ [9].

If ψ is in **ACNF**, then there exists something like a *canonic* resolution refutation of $\mathrm{CL}(\psi)$. The definition of this refutation follows the steps of the definition of the characteristic clause term [9].

Definition 5.1.4 (cf. [9], Definition 5.7). Let ψ be an **LK**-derivation in **ACNF**, Ω be the set of occurrences of the (atomic) cut-formulas in ψ and $\mathcal{C} = \mathrm{CL}(\psi)$. For the sake of convenience, we write $\mathcal{C}_\nu$ for the set of clauses $|\Theta(\psi)/\nu|$ defined by the characteristic terms as in Definition 4.3.7. Clearly, $\mathcal{C} = \mathcal{C}_{\nu_0}$, for the root node ν_0 in ψ.

We proceed by induction and define a p-resolution deduction γ_ν for every deduction node ν in ψ such that γ_ν is a deduction of $S(\nu, \Omega)$ from $\mathcal{C}_\nu$.

If ν is a leaf in ψ, then we define γ_ν as $S(\nu, \Omega)$. By definition of $\mathcal{C}$, we have $\mathcal{C}_\nu = S(\nu, \Omega)$. Clearly, γ_ν is a p-resolution deduction of $S(\nu, \Omega)$ from $\mathcal{C}_\nu$.

(1) Let γ_μ be already defined for a node μ in ψ such that γ_μ is a p-resolution deduction of $S(\mu, \Omega)$ from $\mathcal{C}_\mu$. Moreover, let ξ be a unary inference in ψ with premise μ and conclusion ν. We distinguish two cases:

(1a) The auxiliary formulas of ξ are in $S(\mu, \Omega)$.
Then ξ is a weakening or a contraction[1], and we define $\gamma_\nu =$

$$\dfrac{\begin{array}{c}(\gamma_\mu)\\ S(\mu, \Omega)\end{array}}{S(\nu, \Omega).}\ \xi$$

(1b) The auxiliary formulas of ξ are not in $S(\mu, \Omega)$.
Then we define $\gamma_\nu = \gamma_\mu$.

In both cases γ_ν is a p-resolution deduction of $S(\nu, \Omega)$ from $\mathcal{C}_\mu$. But by definition of the characteristic clause term, we have $\mathcal{C}_\nu = \mathcal{C}_\mu$.

(2) Assume that γ_{μ_i} are p-resolution deductions of $S(\mu_i, \Omega)$ from $\mathcal{C}_{\mu_i}$, for $i = 1, 2$. Let ν be an inference node in ψ with premises μ_1, μ_2 and the corresponding binary rule ξ. Again, we distinguish two cases:

(2a) The auxiliary formulas of ξ are in $S(\mu_1, \Omega)$ and $S(\mu_2, \Omega)$.
Then ξ must be a cut (there are no other binary inferences leading to Ω than atomic cuts), and we define $\gamma_\nu =$

$$\dfrac{\begin{array}{c}(\gamma_{\mu_1})\\ S(\mu_1, \Omega)\end{array} \qquad \begin{array}{c}(\gamma_{\mu_2})\\ S(\mu_2, \Omega)\end{array}}{S(\nu, \Omega)}\ cut.$$

By definition γ_ν is a p-resolution deduction of $S(\nu, \Omega)$ from $\mathcal{C}_{\mu_1} \cup \mathcal{C}_{\mu_2}$. Furthermore, we have $\mathcal{C}_\nu = \mathcal{C}_{\mu_1} \cup \mathcal{C}_{\mu_2}$, by definition of the characteristic clause term, and therefore, γ_ν is a p-resolution deduction of $S(\nu, \Omega)$ from $\mathcal{C}_\nu$.

(2b) The auxiliary formulas of ξ are not in $S(\mu_1, \Omega)$ and $S(\mu_2, \Omega)$.
In this case, we define

$$\gamma_\nu = \gamma_{\mu_1} \odot \gamma_{\mu_2}.$$

By definition of $\odot$, the deduction γ_ν is a p-resolution deduction of $S(\mu_1, \Omega) \circ S(\mu_2, \Omega)$ from $\mathcal{C}_{\mu_1} \times \mathcal{C}_{\mu_2}$. But $S(\nu, \Omega) = S(\mu_1, \Omega) \circ S(\mu_2, \Omega)$ and, by definition of the characteristic clause term, $\mathcal{C}_\nu = \mathcal{C}_{\mu_1} \times \mathcal{C}_{\mu_2}$.

Finally, we define $\mathsf{RES}(\psi) = \gamma_{\nu_0}$, where ν_0 is the root node in ψ. $\triangle$

Remark. The root node does not contain any ancestors of cut-occurrences Ω, this means $S(\nu_0, \Omega) = \vdash$ and γ_{ν_0}, as defined above, is also a *refutation* of $\mathrm{CL}(\psi)$ [9].

[1] This is due to the fact that ψ only contains *atomic* cuts, i.e. all ancestors of cuts must be atomic as well.

For an AC-deduction ψ, the number of nodes in $\mathsf{RES}(\psi)$ may be exponential in the number of nodes in ψ. But, in general, resolution refutations of $\mathrm{CL}(\psi)$ are of nonelementary length (see Section 5.3). Thus, the proofs $\mathsf{RES}(\psi)$ for AC-deductions ψ can be considered as "small" [9].

Proposition 5.1.5 ([9], Proposition 5.2). *Let ψ be an **LK**-derivation in **ACNF**. Then*

$$l(\mathsf{RES}(\psi)) \leq l(\psi) * 2^{2*l(\psi)}.$$

PROOF. By induction on the definition of the γ_ν in Definition 5.1.4 and [7, Proposition 4.2]. For the full proof see [9]. □

5.2 Characteristic Terms and Cut-Reduction

In this section, we show that a cut-reduction step on a derivation (based on the set $\mathcal{R}$) corresponds to a reduction step w.r.t. $\rhd$ on the corresponding clause term [9]. It will turn out that the characteristic clause set $\mathrm{CL}(\varphi')$ of an **ACNF** φ' of a proof φ is subsumed by the original characteristic clause set $\mathrm{CL}(\varphi)$. In this sense, reductive methods based on $\mathcal{R}$ are redundant w.r.t. the results of the method **CERES** [10].

Lemma 5.2.1 ([9], Lemma 6.1). *Let φ, φ' be **LK**-derivations with $\varphi >_\mathcal{R} \varphi'$, for a cut-reduction relation $>_\mathcal{R}$ based on $\mathcal{R}$. Then $\Theta(\varphi) \rhd \Theta(\varphi')$.*

PROOF. Analogous to the proof of Lemma 6.2.19, but without atom indexing. For the full proof see also [9]. □

Theorem 5.2.2 ([9], Theorem 6.1). *Let φ be an **LK**-deduction and ψ an **ACNF** of φ under a cut-reduction relation $>_\mathcal{R}$ based on $\mathcal{R}$. Then $\Theta(\varphi) \leq_{ss} \Theta(\psi)$.*

PROOF. Suppose $\varphi >^*_\mathcal{R} \psi$. Then by Lemma 5.2.1 we get $\Theta(\varphi) \rhd^* \Theta(\psi)$. By Proposition 4.2.15 we obtain $\Theta(\varphi) \leq_{ss} \Theta(\psi)$. □

Theorem 5.2.3 ([9], Theorem 6.2). *Let φ be an **LK**-derivation and ψ be an **ACNF** of φ under a cut-reduction relation $>_\mathcal{R}$ based on $\mathcal{R}$. Then there exists a resolution refutation γ of $\mathrm{CL}(\varphi)$ such that $\gamma \leq_{ss} \mathsf{RES}(\psi)$.*

PROOF. Suppose $\varphi >^*_\mathcal{R} \psi$. Then by Theorem 5.2.2 we have $\Theta(\varphi) \leq_{ss} \Theta(\psi)$, and therefore $\mathrm{CL}(\varphi) \leq_{ss} \mathrm{CL}(\psi)$. By Definition 5.1.4, $\mathsf{RES}(\psi)$ is a resolution refutation of $\mathrm{CL}(\psi)$. Thus, by Proposition 2.3.17, there exists a resolution refutation γ of $\mathrm{CL}(\varphi)$ such that $\gamma \leq_{ss} \mathsf{RES}(\psi)$. □

Corollary 5.2.4 ([9], Corollary 6.1). *Let φ be an **LK**-derivation and ψ be an **ACNF** of φ under a cut-reduction relation $>_\mathcal{R}$ based on $\mathcal{R}$. Then there exists a resolution refutation γ of $\mathrm{CL}(\varphi)$ such that*

$$l(\gamma) \leq l(\mathsf{RES}(\psi)) \leq l(\psi) * 2^{2*l(\psi)}.$$

PROOF. By Theorem 5.2.3, there exists a resolution refutation γ with $\gamma \leq_{ss} \mathsf{RES}(\psi)$. Thus, by definition of subsumption of proofs (see Definition 2.3.16), we have $l(\gamma) \leq l(\mathsf{RES}(\psi))$. Finally, the result follows from Proposition 5.1.5. □

Corollary 5.2.5 (cf. [9], Corollary 6.2). *Let φ be an **LK**-derivation and ψ be an* **ACNF** *of φ under a cut-reduction relation $>_{\mathcal{R}}$ based on $\mathcal{R}$. Then there exists an* **ACNF** *χ of φ under **CERES** and a $k \in \mathbb{N}$ such that*

$$l(\chi) \leq l(\varphi) * l(\psi) * 2^{2 * l(\psi)} + k.$$

PROOF. If γ is a resolution refutation of $\mathrm{CL}(\varphi)$, then an **ACNF** χ of φ can be obtained by **CERES** using projection. As the **LK**-derivations in the projections are not longer than φ itself, we get $l(\chi) \leq l(\varphi) * l(\gamma) + k$ (the term "$+k$" comes from the number of final contractions c_l^*, c_r^*). Then the inequality follows from Corollary 5.2.4. □

Corollary 5.2.6 (cf. [9], Corollary 6.3). *Let φ be an **LK**-derivation and ψ be an* **ACNF** *of φ under Gentzen's or Tait's method. Then there exists an* **ACNF** *χ of φ under **CERES** and a $k \in \mathbb{N}$ such that*

$$l(\chi) \leq l(\varphi) * l(\psi) * 2^{2 * l(\psi)} + k.$$

PROOF. Follows from the fact that both Gentzen's and Tait's method are based on $\mathcal{R}$. □

In the next section, we will see that **CERES** can achieve a nonelementary speed-up over reductive methods. Corollary 5.2.6 tells us that the computational expense of **CERES** is exponentially (and thus elementarily) bounded by that of Gentzen's or Tait's method. As a consequence, **CERES** is never "much slower" than the traditional methods, but there are sequences of derivations where it is substantially faster [9].

It can also be shown that Theorems 5.2.2 and 5.2.3 do not hold in general, i.e. not all cut-reduction methods based on a set of rules yield characteristic terms which are subsumed by the characteristic term of the original proof [9, 10].

5.3 SPEED-UP RESULTS

This section serves the purpose to compare reductive methods and **CERES** from a computational point of view. More specifically, we will show that **CERES** NE[2]-improves both Gentzen and Tait reductions. Moreover, we will also show that no reductive cut-elimination method (based on $\mathcal{R}$) NE-improves **CERES**. In this sense, **CERES** is uniformly better than $>_G$ and $>_T$ [10]. For the comparison in an asymptotic sense, we will use the notion of nonelementary improvement, which is a natural measure for comparing cut-elimination methods, as the complexity of cut-elimination itself is nonelementary [35, 38, 45].

Definition 5.3.1 ([35], Definition 8). Let $\mathbb{N}^2 \to \mathbb{N}$ be the following function:

$$e(0, m) = m,$$
$$e(n+1, m) = 2^{e(n,m)}.$$

[2] NE is an abbreviation for nonelementarily.

A function $f\colon \mathbb{N}^k \to \mathbb{N}^m$, for $k, m \geq 1$, is called *elementary* if there exists an $n \in \mathbb{N}$ and a Turing machine[3] T computing f such that the computing time of T on input $(l_1, \ldots, l_k)$ is less than or equal to $e(n, norm_{max}(l_1, \ldots, l_k))$, where $norm_{max}$ denotes the maximum norm on $\mathbb{N}^k$ (see [14]).

The function $s\colon \mathbb{N} \to \mathbb{N}$ is defined as $s(n) = e(n, 1)$, for $n \in \mathbb{N}$.

A function, which is not elementary, is called *nonelementary*. △

Remark. The notion of elementary function is robust under the use of different models of Turing machines. In fact, it does not matter whether we consider machines with just one tape or with several ones, or machines with unary or k-ary alphabets, for $k > 1$ [35].

Note that the functions e and s are nonelementary. In general, any function f which grows "too fast", i.e. for which there exists no number k such that

$$f(n) \leq e(k, n),$$

is nonelementary [35].

Definition 5.3.2 ([35], Definition 9). Let $\zeta\colon (x_n)_{n\in\mathbb{N}}$ and $\eta\colon (y_n)_{n\in\mathbb{N}}$ be two sequences of natural numbers. We say that ζ *is elementary in* η if there exists a number k s.t. for all $n \in \mathbb{N}$: $x_n \leq e(k, y_n)$; otherwise ζ is called *nonelementary in* η. △

For complexity analysis, we use the following two measures [35]:

- the symbolic complexity $||\cdot||$ (i.e. the number of symbol occurrences) and
- $l(\psi)$ (see Definition 2.2.21).

Statman [45] and Orevkov [38] have independently shown that the complexity of cut-elimination is nonelementary by giving proof sequences that encode the principle of iterated exponentiation [35, 39].

The following theorem, due to Statman and Orevkov, is formulated as in [35].

Theorem 5.3.3 (Statman, Orevkov). *There exists a sequence S_n of sequents with the following properties:*

- *There is a constant a such that for every n there exists an **LK**-proof φ_n of S_n with $||\varphi_n|| \leq 2^{a*n}$.*
- *For every n, let $c(n) = min\{||\psi|| \mid \psi$ is a cut-free proof of $S_n\}$.*
  Then $(c_n)_{n\in\mathbb{N}}$ is nonelementary in $(||\varphi_n||)_{n\in\mathbb{N}}$.

PROOF. In [45] and [38]. □

The following definition gives a basis for comparing reductive cut-elimination methods and **CERES**. Thereby, reductive cut-elimination is described as a sequence of proofs θ obtained via a proof-reduction relation $>_x$ based on $\mathcal{R}$, starting with a proof φ and ending

[3]For a detailed discussion of Turing machines we refer the reader to [1, 31].

in a proof φ' with at most atomic cuts. Such a sequence θ is called an $>_x$-cut-elimination sequence on φ [35].

The following definition is based on [35, Definition 10] and [10, Definition 6.10.1].

Definition 5.3.4. Let $>_x$ be a proof-reduction relation based on $\mathcal{R}$. We say that **CERES** NE-improves $>_x$ if there exists a sequence of proofs $(\varphi_n)_{n\in\mathbb{N}}$ s.t.

- there exists a sequence of resolution refutations $(\gamma_n)_{n\in\mathbb{N}}$ of the sequence of the corresponding characteristic clause sets $(\mathrm{CL}(\varphi_n))_{n\in\mathbb{N}}$ s.t. $(l(\gamma_n))_{n\in\mathbb{N}}$ is elementary in $(||\varphi_n||)_{n\in\mathbb{N}}$.
- For every n, let $g(n) = min\{||\theta|| \mid \theta$ is a $>_x$ -cut-elimination sequence on $\varphi_n\}$. Then $(g(n))_{n\in\mathbb{N}}$ is nonelementary in $(||\varphi_n||)_{n\in\mathbb{N}}$.

Similarly, we define that $>_x$ NE-improves **CERES** if there exists a sequence of proofs $(\varphi_n)_{n\in\mathbb{N}}$ such that

- there exists a sequence of $>_x$-cut-elimination sequences $(\theta_n)_{n\in\mathbb{N}}$ on $(\varphi_n)_{n\in\mathbb{N}}$ such that $(||\theta_n||)_{n\in\mathbb{N}}$ is elementary in $(||\varphi_n||)_{n\in\mathbb{N}}$.
- For every n, let $h(n) = min\{l(\gamma) \mid \gamma$ is a resolution refutation of $\mathrm{CL}(\varphi_n)\}$. Then $(h(n))_{n\in\mathbb{N}}$ is nonelementary in $(||\varphi_n||)_{n\in\mathbb{N}}$.

$\triangle$

Remark. Comparing the size of the resolution refutations in **CERES** with the total size of cut-elimination sequences is justified by the fact that the resolution refutations of characteristic clause sets are the main source of complexity in **CERES** (for details see [10]); in fact, the computation time of a sequence of **CERES** normal forms grows nonelementarily in the size of the input proofs iff this holds for the computation of the resolution refutations. So, for this asymptotic comparison, the computation of the characteristic clause sets and the projections do not matter. Also, mathematically, the core of the **CERES**-method is the resolution refutation of the characteristic clause set [35]. Furthermore, it may seem a bit odd that, in the above definition, for **CERES** we use the length l and for the reductive methods the symbolic complexity $||\cdot||$. This, however, is justified by [10, Proposition 6.5.3], which states that it does not matter whether we use $l(\gamma_n)$ or $||\varphi_n^*||$ for **CERES** normal forms φ_n^* based on γ_n for measuring the asymptotic complexity [10].

Theorem 5.3.5 (cf. [35], Theorem 4). ***CERES** NE-improves $>_G$.*

PROOF. We give a modified version of the proof given in [35] (which itself is a modified version of the one given in [10]). Let $(\psi_n)_{n\in\mathbb{N}}$ be a sequence of proofs for $\psi_n =$

$$\dfrac{\dfrac{\dfrac{A \vdash A^*}{A, \Delta_n \vdash A^*}\,w_l \quad \dfrac{\overset{(\gamma_n)}{\Delta_n \vdash D_n^*}}{A, \Delta_n \vdash D_n^*}\,w_l}{A, \Delta_n \vdash (A \wedge D_n)^*}\,\wedge_r \quad \dfrac{A^* \vdash A}{(A \wedge D_n)^* \vdash A}\,\wedge_{l_1}}{A, \Delta_n \vdash A}\,cut(A \wedge D_n),$$

where formulas marked with $*$ are ancestors of a cut-formula and γ_n is Statman's worst-case sequence admitting only nonelementary cut-elimination (no matter which method is applied); for details concerning the definition of γ_n see [10]. In Gentzen's method, we always select an uppermost cut. As all cuts in γ_n are above the cut with $A \wedge D_n$ as cut-formula, Gentzen's method eliminates all the cuts in γ_n before eliminating the cut with cut-formula $A \wedge D_n$; thus it constructs a cut-free proof of $\Delta_n \vdash D_n$, which is of nonelementary size in $||\gamma_n||$ and also in $||\psi_n||$.

The application of **CERES** to ψ_n yields the following characteristic clause set:

$$\begin{aligned} \mathrm{CL}(\psi_n) &= \{\vdash A\} \cup \mathrm{CL}(\gamma_n) \cup \{A \vdash\} \\ &= \{\vdash A;\ A \vdash\} \cup \mathrm{CL}(\gamma_n). \end{aligned}$$

As a consequence, every $\mathrm{CL}(\psi_n)$ has the resolution refutation $\rho =$

$$\frac{\vdash A \qquad A \vdash}{\vdash}\ R,$$

which is of constant length, and, by defining $\rho_n = \rho$ for all n, we get $||\rho_n|| = 5$. Trivially, $(||\rho_n||)_{n\in\mathbb{N}}$ is elementary in $(||\psi_n||)_{n\in\mathbb{N}}$. Since $l(\rho_n) = 3$, we have $l(\rho_n) \leq ||\rho_n||$ for all n, and thus it follows that $(l(\rho_n))_{n\in\mathbb{N}}$ is also elementary in $(||\psi_n||)_{n\in\mathbb{N}}$. □

Theorem 5.3.6 ([10], Theorem 6.10.2). ***CERES*** *NE-improves* $>_T$.

PROOF. In [10]. □

The following theorem shows that a nonelementary speed-up in the other direction is impossible—for every method based on $\mathcal{R}$ [35]:

Theorem 5.3.7 ([10], Theorem 6.10.3). *No reductive method based on $\mathcal{R}$ NE-improves* ***CERES****; in particular $>_{\mathcal{R}}$ does not NE-improve* ***CERES****.*

PROOF. Towards contradiction assume that $>_x$ is a reduction relation based on $\mathcal{R}$ which NE-improves **CERES**. By Definitions 5.3.2 and 5.3.4, there exists a sequence of proofs $(\varphi_n)_{n\in\mathbb{N}}$ such that there exists $k \in \mathbb{N}$ and a sequence of $>_x$-normal forms $(\varphi_n^*)_{n\in\mathbb{N}}$ with

(i) $||\varphi_n^*|| \leq e(k, ||\varphi_n||)$ and

(ii) for all k there exists an m such that for all $n \geq m$ we have $h(n) > e(k, ||\varphi_n||)$, where $h(n) = min\{l(\gamma) \mid \gamma$ is a resolution refutation of $\mathrm{CL}(\varphi_n)\}$.

By Corollary 5.2.4, we know that there exists a sequence $(\rho_n)_{n\in\mathbb{N}}$ of resolution refutations ρ_n of $\mathrm{CL}(\varphi_n)$ such that

$$l(\rho_n) \leq g(l(\varphi_n^*)),\ \text{for } g = \lambda n.n * 2^{2*n}.$$

But $l(\varphi_n^*) \leq ||\varphi_n^*||$, and therefore, by (i),

$$l(\rho_n) \leq g(e(k, ||\varphi_n||)).$$

As

$$n * 2^{2*n} \leq e(3, n) \text{ and } e(3, e(k, n)) \leq e(k + 3, n),$$

we have that

$$l(\rho_n) \leq e(k + 3, ||\varphi_n||) \text{ for all } n,$$

which contradicts (ii). □

CHAPTER 6

A More General Analysis of Characteristic Clause Sets and Cut-Elimination

In this chapter, we prove our main result, namely that **CERES** still simulates reductive cut-elimination methods if we also eliminate atomic cuts from proofs in atomic cut normal form. To this end, we will first introduce the method of *term resolution* in Section 6.1, which will serve as an auxiliary means in the completeness proof of *indexed resolution* w.r.t. clause sets obtained from clause terms in a special normal form. After proving some important properties of term resolution, we will define the resolution refinement of indexed resolution and prove some of its most important properties in Section 6.2. The final section of this chapter is then devoted to proving the completeness of both term and indexed resolution w.r.t. certain normal forms. Moreover, we will use these completeness results in order to show that each atomic cut-elimination step by reductive methods on a proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ corresponds to some specific indexed resolution steps on the corresponding characteristic clause sets. Finally, we will use this fact in order to show that we can always obtain an indexed clause set from the characteristic clause set of the original proof by indexed resolution, which subsumes the characteristic clause set of the proof after atomic cut-elimination by reductive methods.

6.1 Term Resolution

In this section, we will introduce a new method for removing atoms A from clause terms of the form $X_1 \oplus X_2$ such that A occurs in the consequent of some clause in a subterm of X_1 and in the antecedent of some clause in a subterm of X_2 (or vice versa). We will call this method *term resolution*, even though it is not a resolution method in the classical sense (since it operates on clause terms and not directly on clauses). Nevertheless, we decided to call it term resolution simply because—for clause terms in a special normal form—each term resolution deduction of a clause term Y from a clause term X (which operates on the syntactic level of clause terms) corresponds to a specific resolution deduction from the clause set $|X|$ of X. Such a resolution deduction only resolves upon clauses that contain atoms, which have been eliminated from X in order to obtain Y. This way, one can obtain a clause set that corresponds to $|Y|$. In this sense—for a certain class of clause terms—term resolution on the syntax of clause terms corresponds to a special case of resolution (called *indexed resolution*, see Section 6.2) on the semantics of clause terms. In Section 6.3, it will turn out that term resolution is complete w.r.t. the above mentioned normal forms, i.e. if the clause set of a clause term (in normal form) is unsatisfiable, then there exists a term resolution deduction of the clause term $\{\vdash\}$. However, term resolution on arbitrary clause terms is not complete, but this is not needed for our purposes, namely to use term resolution as an auxiliary means to prove the completeness of indexed resolution w.r.t. indexed clause sets extracted from proofs in a special normal form.

The following two rules of term factoring w.r.t. $\vdash$ are needed in order to get rid of superfluous subterms that might be the result of applications of the rule R_t of term resolution (see Definition 6.1.3).

Definition 6.1.1 (Term factoring w.r.t. $\vdash$). Let X be a clause term such that there exists a position λ in X with $X.\lambda = \{\vdash\} \odot B$ or $X.\lambda = A \odot \{\vdash\}$, where A and B are clause terms and $\odot \in \{\oplus, \otimes\}$.

Furthermore, let λ_1 and λ_2 be positions in X such that $X.\lambda_1 = A$ or $X.\lambda_1 = B$ and $X.\lambda_2 = \{\vdash\}$, respectively. Then we define the following rules of term factoring w.r.t. $\vdash$:

$$\frac{X}{X[X.\lambda_1]_\lambda}\, ft_\otimes \qquad\qquad \frac{X}{X[X.\lambda_2]_\lambda}\, ft_\oplus$$

$\triangle$

Remark. The intention behind the definitions of $ft_\otimes$ and $ft_\oplus$ is that it indeed holds that $|\{\vdash\} \otimes B| = B$ (or $|A \otimes \{\vdash\}| = A$), by semantics of $\otimes$. Moreover, for $\{\vdash\} \oplus B$ (or $A \oplus \{\vdash\}$), we get $|\{\vdash\} \oplus B| = \{\vdash\} \cup B$ ($|A \oplus \{\vdash\}| = A \cup \{\vdash\}$), by semantics of $\oplus$, i.e. $\vdash$ occurs in the corresponding characteristic clause set. But this means that the characteristic clause set is unsatisfiable. Therefore, replacing both $\{\vdash\} \oplus B$ and $A \oplus \{\vdash\}$ by $\{\vdash\}$ is justified.

In order to be able to replace specific subterms on the syntactic level, which coincide semantically, we introduce the rules of *equivalent replacement w.r.t.* $\otimes$:

Definition 6.1.2 (Equivalent replacement w.r.t. $\otimes$). Let X, Y be clause terms s.t. there exist positions λ_1 and λ_2 in X and Y, respectively, with $X.\lambda_1 = \{\Gamma \vdash \Delta\} \otimes \{\Pi \vdash \Lambda\}$ and $Y.\lambda_2 = \{\Gamma, \Pi \vdash \Delta, \Lambda\}$, where Γ, Δ, Π and Λ are (possibly empty) multisets of atoms (possibly containing some indexed atoms). Then we define the following rules of equivalent replacement w.r.t. $\otimes$:

$$\frac{X}{X[\{\Gamma, \Pi \vdash \Delta, \Lambda\}]_{\lambda_1}}\, ert^-_\otimes \qquad\qquad \frac{Y}{Y[\{\Gamma \vdash \Delta\} \otimes \{\Pi \vdash \Lambda\}]_{\lambda_2}}\, ert^+_\otimes$$

$\triangle$

Remark. The soundness of $ert^{+}_{\otimes}$ and $ert^{-}_{\otimes}$ follows from the fact that

$$|\{\Gamma \vdash \Delta\} \otimes \{\Pi \vdash \Lambda\}| = \{\Gamma \vdash \Delta \circ \Pi \vdash \Lambda\} = \{\Gamma, \Pi \vdash \Delta, \Lambda\} = |\{\Gamma, \Pi \vdash \Delta, \Lambda\}|,$$

by semantics of $\otimes$ and the definition of $\circ$. In particular, the rules of equivalent replacement w.r.t. $\otimes$ are needed in order to obtain the characteristic clause term by term resolution that would result from the extraction from the proof after atomic cut-elimination.

More formally, eliminating an atomic cut (with cut-formula A^i) from a proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ (containing so-called chains of atomic cuts, see Section 6.3) amounts to replacing the subterm $\{A^l \vdash A^i\} \otimes \{A^i \vdash A^j\}$ by $\{A^l \vdash A^j\}$ (where the indexed atoms A^l or A^j might be omitted) in the corresponding characteristic clause terms. Without the rules of equivalent replacement w.r.t. $\otimes$, term resolution would yield $\{A^l \vdash\} \otimes \{\vdash A^j\}$. Despite the semantic equivalence of the two results, term resolution would produce a syntactically different clause term than the actual characteristic clause term of the proof after atomic cut-elimination.

The rule R_t of term resolution is the core part of the term resolution principle:

Definition 6.1.3 (Term resolution). Let X be a clause term such that there is a position λ in X with $X.\lambda = X_1 \oplus X_2$, where X_1 and X_2 are clause terms. We define the following rule of term resolution:

$$\frac{X}{X[X_1[\{\Gamma \vdash \Delta\}]_{\lambda_1} \otimes X_2[\{\Pi \vdash \Lambda\}]_{\lambda_2}]_{\lambda}}\ R_t$$

which is only applicable if there are positions λ_1 and λ_2 in X_1 and X_2, respectively, s.t.

$$X_1.\lambda_1 = \{\Gamma \vdash \Delta, A\} \text{ and } X_2.\lambda_2 = \{A, \Pi \vdash \Lambda\}$$

or

$$X_1.\lambda_1 = \{A, \Gamma \vdash \Delta\} \text{ and } X_2.\lambda_2 = \{\Pi \vdash \Lambda, A\},$$

for some (possibly indexed[1]) atom A and (possibly empty) multisets of atoms $\Gamma, \Delta, \Pi, \Lambda$ (possibly containing some indexed atoms). $\triangle$

Remark. For clause terms in **TACNF** (see Definition 6.1.6), the following rule

$$\frac{X}{X[X_1[\{\Gamma \vdash \Delta\}]_{\lambda_1} \oplus X_2[\{\Pi \vdash \Lambda\}]_{\lambda_2}]_{\lambda}}\ R_t'$$

is sound, but this does not hold in general, as the following example demonstrates:

Let $X = \{A \vdash B\} \oplus \{B \vdash A\}$. Then $|X| = \{A \vdash B; B \vdash A\}$ is clearly satisfiable. But applying R_t' to X yields:

$$\frac{\{A \vdash B\} \oplus \{B \vdash A\}}{\{A \vdash\} \oplus \{\vdash A\}}\ R_t',$$

[1] See Definition 6.2.1.

where $\Gamma = \Lambda = A$, $X.\lambda = X$, $X_1.\lambda_1 = \{A \vdash B\}$ and $X_2.\lambda_2 = \{B \vdash A\}$.

But $|\{A \vdash\} \oplus \{\vdash A\}| = \{A \vdash; \vdash A\}$ is unsatisfiable, as it is equivalent to $\neg A \wedge A$, by semantics of sequents and clause sets. On the other hand, $|\{A \vdash\} \otimes \{\vdash A\}| = \{A \vdash A\}$ is clearly satisfiable (and in this case even valid).

Definition 6.1.4 (Term Resolution Deduction). A deduction tree having clause terms as leaves and term resolution (R_t), term factoring w.r.t. $\vdash$ ($ft_\otimes$, $ft_\oplus$) and equivalent replacement w.r.t. $\otimes$ ($ert^-_\otimes, ert^+_\otimes$) as rules is called a *term resolution deduction*.

Let γ be a term resolution deduction where X is the clause term at the leaf-node of γ. If Y is the clause term at the root of the deduction tree, then γ is called a *term resolution deduction* of Y from X. If $Y = \{\vdash\}$, then γ is called a *term resolution refutation of* X. $\triangle$

Remark. Note that since all rules in a term resolution deduction are unary, each deduction tree has exactly one leaf-node.

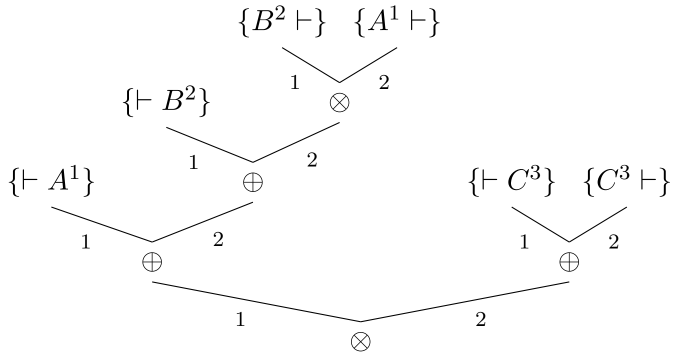
To illustrate how term resolution works, we will give a concrete example of a term resolution refutation:

Example 6.1.5. Let X be the following clause term

$$[\{\vdash A^1\} \oplus (\{\vdash B^2\} \oplus (\{B^2 \vdash\} \otimes \{A^1 \vdash\}))] \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\}),$$

where A^1, B^2 and C^3 are indexed atoms.

First, consider the term tree of X:



Then, we apply R_t to X in order to resolve upon B^2:

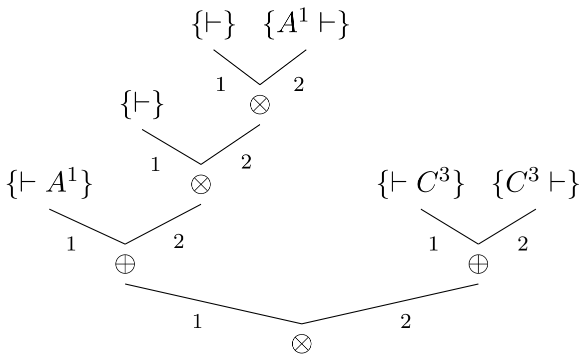
$$\frac{[\{\vdash A^1\} \oplus (\{\vdash B^2\} \oplus (\{B^2 \vdash\} \otimes \{A^1 \vdash\}))] \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})}{\underbrace{X[\{\vdash B^2\}[\{\vdash\}]_0 \otimes (\{B^2 \vdash\} \otimes \{A^1 \vdash\})[\{\vdash\}]_{(0,1)}]_{(0,1,2)}}_{X'}} R_t,$$

where

$$X.(0,1,2) = \{\vdash B^2\} \oplus (\{B^2 \vdash\} \otimes \{A^1 \vdash\}),$$
$$\{\vdash B^2\}.0 = \{\vdash B^2\},$$
$$(\{B^2 \vdash\} \otimes \{A^1 \vdash\}).(0,1) = \{B^2 \vdash\}.$$

Consequently, $X' = [(\{\vdash A^1\} \oplus (\{\vdash\} \otimes (\{\vdash\} \otimes \{A^1 \vdash\})))] \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})$.

Term tree of X':

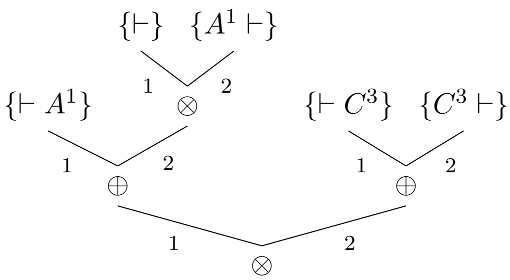


In order to simplify the clause term X', we apply term factoring w.r.t. $\vdash$:

$$\frac{[(\{\vdash A^1\} \oplus (\{\vdash\} \otimes (\{\vdash\} \otimes \{A^1 \vdash\})))] \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})}{\underbrace{X'[X'.(0,1,2,2)]_{(0,1,2)}}_{X''}}\, ft_{\otimes},$$

where $X'.(0,1,2) = \{\vdash\} \otimes (\{\vdash\} \otimes \{A^1 \vdash\})$ and $X'.(0,1,2,2) = \{\vdash\} \otimes \{A^1 \vdash\}$.

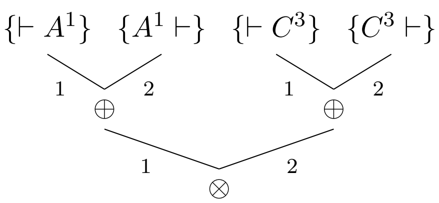
As a consequence, $X'' = [\{\vdash A^1\} \oplus (\{\vdash\} \otimes \{A^1 \vdash\})] \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})$.
Term tree of X'':



An analogous application of $ft_{\otimes}$ to X'' yields:

$$X''' = (\{\vdash A^1\} \oplus \{A^1 \vdash\}) \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\}).$$

Term tree of X''':

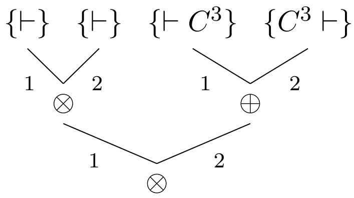


Now, we can apply R_t to X''':

$$\frac{(\{\vdash A^1\} \oplus \{A^1 \vdash\}) \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})}{\underbrace{X'''[\{\vdash A^1\}[\{\vdash\}]_0 \otimes \{A^1 \vdash\}[\{\vdash\}]_0]_{(0,1)}}_{Y}} \; R_t,$$

where $X'''.(0,1) = \{\vdash A^1\} \oplus \{A^1 \vdash\}$, $\{\vdash A^1\}.0 = \{\vdash A^1\}$ and $\{A^1 \vdash\}.0 = \{A^1 \vdash\}$. Thus, $Y = (\{\vdash\} \otimes \{\vdash\}) \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})$.

Term tree of Y:



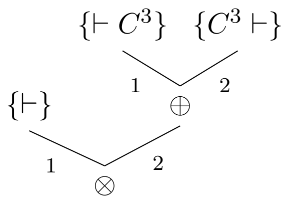
An application of $ft_\otimes$ to Y yields

$$\frac{(\{\vdash\} \otimes \{\vdash\}) \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})}{\underbrace{Y[Y.(0,1,1)]_{(0,1)}}_{Y'}} \; ft_\otimes,$$

where $Y.(0,1) = \{\vdash\} \otimes \{\vdash\}$ and $Y.(0,1,1) = \{\vdash\}$.

Consequently, $Y' = \{\vdash\} \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})$.

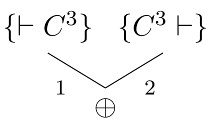
Term tree of Y':



An analogous application of $ft_\otimes$ to Y' yields:

$$Y'' = \{\vdash C^3\} \oplus \{C^3 \vdash\}.$$

Term tree of Y'':

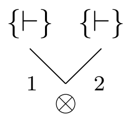


Next, we can resolve upon C^3:

$$\dfrac{\{\vdash C^3\} \oplus \{C^3 \vdash\}}{\underbrace{Y''[\{\vdash C^3\}[\{\vdash\}]_0 \otimes \{C^3 \vdash\}[\{\vdash\}]_0]_0}_{Z}}\ R_t$$

where $Y''.0 = Y''$, $\{\vdash C^3\}.0 = \{\vdash C^3\}$ and $\{C^3 \vdash\}.0 = \{C^3 \vdash\}$.
As a consequence, $Z = \{\vdash\} \otimes \{\vdash\}$.

Term tree of Z:



Finally, an application of $ft_\otimes$ to Z yields

$$\dfrac{\{\vdash\} \otimes \{\vdash\}}{\underbrace{Z[Z.(0,1)]_0}_{Z'}}\ ft_\otimes$$

where $Z.0 = \{\vdash\} \otimes \{\vdash\}$ and $Z.(0,1) = \{\vdash\}$. Thus, $Z' = \{\vdash\}$, i.e. there exists a term resolution refutation γ of X, which can be represented more compactly as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\{\vdash A^1\} \oplus (\{\vdash B^2\} \oplus (\{B^2 \vdash\} \otimes \{A^1 \vdash\}))] \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})}{[(\{\vdash A^1\} \oplus (\{\vdash\} \otimes (\{\vdash\} \otimes \{A^1 \vdash\}))] \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})}\ R_t}{[\{\vdash A^1\} \oplus (\{\vdash\} \otimes \{A^1 \vdash\})] \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})}\ ft_\otimes}{(\{\vdash A^1\} \oplus \{A^1 \vdash\}) \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})}\ ft_\otimes}{(\{\vdash\} \otimes \{\vdash\}) \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})}\ R_t}{\{\vdash\} \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})}\ ft_\otimes}{\{\vdash C^3\} \oplus \{C^3 \vdash\}}\ ft_\otimes}{\{\vdash\} \otimes \{\vdash\}}\ R_t}{\{\vdash\}}\ ft_\otimes$$

$\triangle$

The following definitions of normal forms for clause terms will be useful for the proofs in Section 6.3.

Definition 6.1.6 (TACNF). We say that a clause term X is in **TACNF** (*top atomic cut normal form*) if $X = \{\vdash\}$ or

$$X = (\{\vdash A_1\} \oplus \{A_1 \vdash\}) \otimes \ldots \otimes (\{\vdash A_k\} \oplus \{A_k \vdash\}),$$

where A_i is an atomic formula, for $1 \leq i \leq k$. $\triangle$

Definition 6.1.7 ($\textbf{TACNF}_{\textbf{ai}}$). We say that a clause term X is in $\textbf{TACNF}_{\textbf{ai}}$ if $X = \{\vdash\}$, or

$$X = (\{\vdash A_1^1\} \oplus \{A_1^1 \vdash\}) \otimes \ldots \otimes (\{\vdash A_k^k\} \oplus \{A_k^k \vdash\}),$$

where A_i^i is an indexed atomic formula, for $1 \leq i \leq k$. △

Remark. Note that, in general, characteristic clause terms extracted from proofs φ in $\textbf{ACNF}_{\textbf{ai}}^{\textbf{top}}$ (see Definition 6.2.12) may contain subterms of the form $\{\vdash A_i^j\} \oplus \{A_i^j \vdash\}$ with $i \neq j$. But this is no real problem, as we can reassign the numbers i in such a way that the respective characteristic clause term conforms with Definition 6.1.7.

Definition 6.1.8 ($\textbf{TACNF}^{\textbf{ext}}$). We say that a clause term X is in $\textbf{TACNF}^{\textbf{ext}}$ (extended top atomic cut normal form) if X is in **TACNF** or

$$X = \{\vdash A\} \oplus \{A \vdash A\} \oplus \ldots \oplus \{A \vdash A\} \oplus \{A \vdash\},$$

where A is an arbitrary atomic formula.
If X and Y are in $\textbf{TACNF}^{\textbf{ext}}$, then $X \otimes Y$ is in $\textbf{TACNF}^{\textbf{ext}}$. △

Definition 6.1.9 ($\textbf{TACNF}_{\textbf{ai}}^{\textbf{ext}}$). We say that a clause term X is in $\textbf{TACNF}_{\textbf{ai}}^{\textbf{ext}}$ if X is in $\textbf{TACNF}_{\textbf{ai}}$ or

$$X = \{\vdash A^1\} \oplus \{A^1 \vdash A^2\} \oplus \{A^2 \vdash A^3\} \oplus \ldots \oplus \{A^{k-1} \vdash A^k\} \oplus \{A^k \vdash\},$$

where A^i is an indexed atomic formula, for $1 \leq i \leq k$.
If X and Y are in $\textbf{TACNF}_{\textbf{ai}}^{\textbf{ext}}$, then $X \otimes Y$ is in $\textbf{TACNF}_{\textbf{ai}}^{\textbf{ext}}$. △

Remark. Note that, if $k = 1$ in Definition 6.1.9, then we have $X = \{\vdash A^1\} \oplus \{A^1 \vdash\}$, i.e. X is in $\textbf{TACNF}_{\textbf{ai}}$. Thus, the definition of $\textbf{TACNF}_{\textbf{ai}}^{\textbf{ext}}$ includes clause terms in $\textbf{TACNF}_{\textbf{ai}}$ implicitly, but nevertheless, for the sake of convenience, we still keep the separate definition of $\textbf{TACNF}_{\textbf{ai}}$.

Moreover, in all of the above four versions of the top atomic cut normal form, parentheses in a clause term can be set in any possible way. As a consequence, the application of the resolution rule depends on the way parentheses are set in a clause term. For instance, consider the two different settings of parentheses in the clause term X:

$$X = (\{\vdash A^1\} \oplus \{A^1 \vdash A^2\}) \oplus (\{A^2 \vdash A^3\} \oplus \ldots \oplus \{A^{k-1} \vdash A^k\}) \oplus \{A^k \vdash\},$$
$$X = \{\vdash A^1\} \oplus (\{A^1 \vdash A^2\} \oplus \{A^2 \vdash A^3\}) \oplus \ldots \oplus (\{A^{k-1} \vdash A^k\} \oplus \{A^k \vdash\}).$$

This is especially important for clause terms in $\textbf{TACNF}^{\textbf{ext}}$ or $\textbf{TACNF}_{\textbf{ai}}^{\textbf{ext}}$.

The relation $X \vdash_{\mathsf{res_t}} Y$ serves the purpose to compactly represent term resolution deductions of Y from X, where Y was obtained from X by eliminating at most one subterm of the form $\{\vdash A\} \oplus \{A \vdash\}$ from X. Note that A might be an indexed atom.

Definition 6.1.10 ($\vdash_{\mathsf{res_t}}$). Let X and Y be two clause terms in **TACNF** (or $\textbf{TACNF}_{\textbf{ai}}$). Then we write $X \vdash_{\mathsf{res_t}} Y$ if $X = Y$ or the subterm $\{\vdash A\} \oplus \{A \vdash\}$ of X (where A is a (possibly indexed) atom) was resolved by a term resolution deduction from X in order to obtain Y. If A is an indexed atom, then we require that both occurrences of A in the indexed clauses $\vdash A$ and $A \vdash$ have the same index. △

Similarly, $X \vdash_{\mathsf{res_{te}}} Y$ is used to compactly represent term resolution deductions of Y from X, where Y was obtained from X by replacing a subterm $\{A^l \vdash A^i\} \oplus \{A^i \vdash A^j\}$ in X by $\{A^l \vdash A^j\}$.

Definition 6.1.11 ($\vdash_{\mathsf{res_{te}}}$). Let X and Y be two clause terms in $\mathsf{TACNF}^{\mathsf{ext}}_{\mathsf{ai}}$ (or $\mathsf{TACNF}^{\mathsf{ext}}$). Then we write $X \vdash_{\mathsf{res_{te}}} Y$ if $X = Y$ or the subterm $\{A^l \vdash A^i\} \oplus \{A^i \vdash A^j\}$ in X was replaced by $\{A^l \vdash A^j\}$ (where A^i, A^j and A^l are indexed atoms; A^l or A^j may be missing) by a term resolution deduction from X in order to obtain Y.
For terms in $\mathsf{TACNF}^{\mathsf{ext}}$, we simply omit the indices in the above definition. $\triangle$

Definition 6.1.12 ($\vdash_{\mathsf{res_{ta}}}$). Let X and Y be two clause terms in $\mathsf{TACNF}^{\mathsf{ext}}_{\mathsf{ai}}$ (or $\mathsf{TACNF}^{\mathsf{ext}}$). Then we write $X \vdash_{\mathsf{res_{ta}}} Y$ if $X \vdash_{\mathsf{res_{t}}} Y$ or $X \vdash_{\mathsf{res_{te}}} Y$ holds. $\triangle$

The following two lemmas show that we can always obtain a clause term Y in $\mathsf{TACNF}_{\mathsf{ai}}$ from a clause term X in $\mathsf{TACNF}^{\mathsf{ext}}_{\mathsf{ai}}$ by a term resolution deduction on X that uses a finite number of intermediary $\vdash_{\mathsf{res_{te}}}$-steps.

Lemma 6.1.13. *Let X be a clause term in $\mathsf{TACNF}^{\mathsf{ext}}_{\mathsf{ai}}$ of the following form:*

$$\{\vdash A^1\} \oplus \{A^1 \vdash A^2\} \oplus \{A^2 \vdash A^3\} \oplus \ldots \oplus \{A^{k-1} \vdash A^k\} \oplus \{A^k \vdash\},$$

where A^i is an indexed atomic formula, for $1 \leq i \leq k$. Then there exists a clause term X_{k-1} in $\mathsf{TACNF}_{\mathsf{ai}}$ such that $X \vdash_{\mathsf{res_{te}}} X_1 \vdash_{\mathsf{res_{te}}} \ldots \vdash_{\mathsf{res_{te}}} X_{k-1}$, where X_i is obtained from X_{i-1} by replacing the subterm $\{A^l \vdash A^i\} \oplus \{A^i \vdash A^j\}$ by $\{A^l \vdash A^j\}$, for $1 \leq i, j, l \leq k$.

PROOF. Let X be a clause term in $\mathsf{TACNF}^{\mathsf{ext}}_{\mathsf{ai}}$. We proceed by induction on the number n of distinct indexed atoms in X.

BASE CASE: $n = 0$. Then $X = \{\vdash\}$, i.e. $X \vdash_{\mathsf{res_{te}}} X$, and since X is already in $\mathsf{TACNF}_{\mathsf{ai}}$ we are done.

INDUCTION HYPOTHESIS (IH): Assume the claim holds for all clause terms X in $\mathsf{TACNF}^{\mathsf{ext}}_{\mathsf{ai}}$ containing k distinct indexed atoms, for $0 \leq k \leq n$.

INDUCTION STEP: Suppose w.l.o.g. that X is a clause term in $\mathsf{TACNF}^{\mathsf{ext}}_{\mathsf{ai}}$ containing $n+1$ distinct indexed atoms, i.e.

$$X = \{\vdash A^1\} \oplus (\bigoplus_{1 \leq i \leq n} \{A^i \vdash A^{i+1}\}) \oplus \{A^{n+1} \vdash\}.$$

Furthermore, assume w.l.o.g. that X_1 is the clause term in $\mathsf{TACNF}^{\mathsf{ext}}_{\mathsf{ai}}$ obtained from X by replacing the subterm

$$\{A^{j-1} \vdash A^j\} \oplus \{A^j \vdash A^{j+1}\}$$

in X by $\{A^{j-1} \vdash A^{j+1}\}$, for some j with $1 \leq j \leq n$. The clause term X_1 can be obtained from X by term resolution as follows:

Let $X.\lambda = \{A^{j-1} \vdash A^j\} \oplus \{A^j \vdash A^{j+1}\}$ and

$$\{A^{j-1} \vdash A^j\}.\lambda_1 = \{A^{j-1} \vdash A^j\},$$
$$\{A^j \vdash A^{j+1}\}.\lambda_2 = \{A^j \vdash A^{j+1}\}.$$

Then we get

$$\frac{X}{\underbrace{X[\{A^{j-1} \vdash A_j\}[\{A^{j-1} \vdash\}]_{\lambda_1} \otimes \{A_j \vdash A^{j+1}\}[\{\vdash A^{j+1}\}]_{\lambda_2}]_\lambda}_{X'}} R_t,$$

where $X'.\lambda = \{A^{j-1} \vdash\} \otimes \{\vdash A^{j+1}\}$. An application of $ert_\otimes^-$ to X' yields X_1 with $X_1.\lambda = \{A^{j-1} \vdash A^{j+1}\}$. As a consequence, $X \vdash_{\mathsf{res_{te}}} X_1$ and since X_1 contains fewer distinct indexed atoms than X, we get, by **(IH)**, that $X_1 \vdash_{\mathsf{res_{te}}} X_2 \vdash_{\mathsf{res_{te}}} \ldots \vdash_{\mathsf{res_{te}}} X_n$, where X_n is in $\mathbf{TACNF_{ai}}$. Putting things together, we obtain $X \vdash_{\mathsf{res_{te}}} X_1 \vdash_{\mathsf{res_{te}}} \ldots \vdash_{\mathsf{res_{te}}} X_n$. □

Proposition 6.1.14. *Let X be a clause term in $\mathbf{TACNF_{ai}^{ext}}$. Then there exists a clause term X_{k-1} in $\mathbf{TACNF_{ai}}$ such that $X \vdash_{\mathsf{res_{te}}} X_1 \vdash_{\mathsf{res_{te}}} \ldots \vdash_{\mathsf{res_{te}}} X_{k-1}$, where X_i is obtained from X_{i-1} by replacing the subterm $\{A^l \vdash A^i\} \oplus \{A^i \vdash A^j\}$ by $\{A^l \vdash A^j\}$, for $1 \leq i, j, l \leq k$.*

Proof. We proceed by induction on the structure of X.

BASE CASE:

- If X is in $\mathbf{TACNF_{ai}}$, we trivially have $X \vdash_{\mathsf{res_{te}}} X$.
- $X = \{\vdash A^1\} \oplus (\bigoplus_{1 \leq i < k} \{A^i \vdash A^{i+1}\}) \oplus \{A^k \vdash\}$. Then there exists a clause term X_{k-1} in $\mathbf{TACNF_{ai}}$ such that $X \vdash_{\mathsf{res_{te}}} X_1 \vdash_{\mathsf{res_{te}}} \ldots \vdash_{\mathsf{res_{te}}} X_{k-1}$, by Lemma 6.1.13.

INDUCTION STEP: Suppose $X = X' \otimes X''$ is a clause term in $\mathbf{TACNF_{ai}^{ext}}$. Then, by **(IH)**, there exist clause terms X'_l, X''_m in $\mathbf{TACNF_{ai}}$ such that $X' \vdash_{\mathsf{res_{te}}} X'_1 \vdash_{\mathsf{res_{te}}} \ldots \vdash_{\mathsf{res_{te}}} X'_l$ and $X'' \vdash_{\mathsf{res_{te}}} X''_1 \vdash_{\mathsf{res_{te}}} \ldots \vdash_{\mathsf{res_{te}}} X''_m$, with $l + m = k - 1$. This means there are corresponding term resolution deductions δ_1 of X'_l from X' and δ_2 of X''_m from X'' of a specific form. Since $X' \otimes X''$ is in $\mathbf{TACNF_{ai}^{ext}}$, we can combine δ_1 and δ_2 to a term resolution deduction δ of $X'_l \otimes X''_m$ from X as follows:

$$\begin{array}{c} X' \otimes X'' \\ (\delta_1 \otimes X'') \\ X'_l \otimes X'' \\ (X'_l \otimes \delta_2) \\ X'_l \otimes X''_m \end{array}$$

where $\delta_1 \otimes X''$ is an abbreviation for extending each node in the deduction tree of δ_1 with "$\otimes X''$" (and similarly for $X'_l \otimes \delta_2$). Note that the positions in the clause terms used in the deduction δ have to be adapted, since the structure of the corresponding term trees has changed.

Clearly, $X'_l \otimes X''_m$ is in **TACNF$_{\mathbf{ai}}$**, as both X'_l and X''_m are in **TACNF$_{\mathbf{ai}}$**. As a consequence, $X \vdash_{\mathsf{res_{te}}} X'_1 \otimes X'' \vdash_{\mathsf{res_{te}}} \ldots \vdash_{\mathsf{res_{te}}} X'_l \otimes X'' \vdash_{\mathsf{res_{te}}} X'_l \otimes X''_1 \vdash_{\mathsf{res_{te}}} \ldots \vdash_{\mathsf{res_{te}}} X'_l \otimes X''_m$, with $l + m = k - 1$. □

6.2 Indexed Resolution

Term resolution, as stated in the previous section, is just an auxiliary means for proving the completeness (w.r.t. a specific normal form) of a resolution refinement[2] called *indexed resolution*. This section is thus devoted to introducing the indexed resolution principle and proving some important properties w.r.t. proofs having an atom indexing. What we call indexed resolution is basically a modified version of the resolution refinement of *atomic cut-linkage* as defined in [51]. Indexed resolution admits only resolving upon indexed atoms that coincide when omitting the indices and whose assigned indices are exactly the same. To use this method in conjunction with **CERES**, we have to assign indices to the atoms occurring in the cut-formulas of **LK**-proofs. As a consequence, the characteristic clause sets extracted from proofs with atom indexing will be composed of indexed atoms only. Moreover, it will turn out that the cut-reduction rules in $\mathcal{R}_{\mathsf{ax}}$ preserve the fact that in all cuts in a proof, both auxiliary formulas have exactly the same atom indexing. Since the characteristic clause set of a proof with atom indexing contains now additional information in the form of indices, the resulting **ACNF$_{\mathbf{ai}}$** will be based on resolution refutations that only resolve upon atoms with the same index. This way, not only the search space for resolution refutations of the characteristic clause set reduces but also the difference between normal forms under **CERES** and reductive methods will be minimized [40, 51]. Another important property of proofs with atom indexing is the fact that each cut-reduction step under $>_{\mathcal{R}}$ corresponds to a reduction step w.r.t. $\triangleright$ on the corresponding clause term. The completeness result (w.r.t. a special normal form) for indexed resolution (see Theorems 6.3.14 and 6.3.16) will also be a key result towards answering the conjecture posed by Reis in [40] whether **CERES** for intuitionistic logic, using indexed resolution and joining projections (for details see [40]), applied to a skolemized **LJ**-proof with cuts yields an intuitionistic proof.

Definition 6.2.1 (Indexed Formula). Let $F \in \mathsf{PL}$ and $i \in \mathbb{N}$, then the pair (F, i) is called an *indexed* formula. If F is an atomic formula, then we call (F, i) an *indexed atomic formula (or indexed atom)*. We denote an indexed formula (F, i) by F^i. △

Definition 6.2.2. Let σ be a substitution and (F, i) an indexed formula. We define the application of σ to indexed formulas as follows: $(F, i)\sigma = (F\sigma, i)$. △

Definition 6.2.3 is based on [40, Definition 34].

Definition 6.2.3 (Atom Indexing). Let φ be an **LK**-proof containing k cuts and F_k be the cut-formula of the k-th cut. Then we assign to each atomic subformula of F_k an

[2] Intuitively, a resolution refinement can be seen as a subset of the set of all resolution deductions. In fact, the actual definition has to fulfil some specific properties, but for our purposes it suffices to consider refinements as restricted forms of resolution [34].

index $i \geq 1$ such that no two atomic subformulas of F_k get assigned the same index. Moreover, all atomic cut-ancestors of F_k will get the same index as the corresponding atomic subformula of F_k. All other atomic subformulas of formulas that are not cut-formulas get assigned the index 0.

Furthermore, there is no index $j \geq 1$ such that A^j is an atomic subformula of both F_k and F_l, where $k \neq l$ (i.e. atomic subformulas occurring in different cuts must not have the same index). $\triangle$

Remark. A consequence of the above definition is that both auxiliary occurrences of the cut-formula in a cut have exactly the same indexing.

Example 6.2.4. Let φ be the proof of Example 4.3.9, then the application of atom indexing to φ yields:

$$\dfrac{\begin{matrix}(\varphi_1)\\ P(a) \vee Q(b) \vdash (\exists y)(P(y)^1 \vee Q(y)^2)\end{matrix} \qquad \begin{matrix}(\varphi_2)\\ (\exists y)(P(y)^1 \vee Q(y)^2), (\forall x)\neg P(x) \vdash (\exists z)Q(z)\end{matrix}}{P(a) \vee Q(b), (\forall x)(\neg P(x)) \vdash (\exists z)Q(z)}\ cut$$

where φ_1 is the **LK**-derivation:

$$\dfrac{\dfrac{\dfrac{P(a) \vdash P(a)^1}{P(a) \vdash (P(a)^1 \vee Q(a)^2)}\ \vee_{r_1}}{P(a) \vdash (\exists y)(P(y)^1 \vee Q(y)^2)}\ \exists_r \qquad \dfrac{\dfrac{Q(b) \vdash Q(b)^2}{Q(b) \vdash (P(b)^1 \vee Q(b)^2)}\ \vee_{r_2}}{Q(b) \vdash (\exists y)(P(y)^1 \vee Q(y)^2)}\ \exists_r}{P(a) \vee Q(b) \vdash (\exists y)(P(y)^1 \vee Q(y)^2)}\ \vee_l$$

and φ_2 is the **LK**-derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(u)^1 \vdash P(u)}{P(u)^1, \neg P(u) \vdash}\ \neg_l}{P(u)^1, \neg P(u) \vdash Q(u)}\ w_r \qquad \dfrac{Q(u)^2 \vdash Q(u)}{Q(u)^2, \neg P(u) \vdash Q(u)}\ w_l}{((P(u)^1 \vee Q(u)^2), \neg P(u) \vdash Q(u)}\ \vee_l}{((P(u)^1 \vee Q(u)^2), \neg P(u) \vdash (\exists z)Q(z)}\ \exists_r}{((P(u)^2 \vee Q(u)^1), (\forall x)\neg P(x) \vdash (\exists z)Q(z)}\ \forall_l}{(\exists y)(P(y)^1 \vee Q(y)^2), (\forall x)\neg P(x) \vdash (\exists z)Q(z)}\ \exists_l$$

$\triangle$

In order to have a means to denote that two formulas (that are composed of indexed atoms only) coincide (i.e. they have the same structure and their corresponding indexed atoms have the same indices), we will define the relation $\simeq$ as follows:

Definition 6.2.5. Let F and G be formulas composed of indexed atoms only. Then we define $F \simeq G$ inductively as follows:

- $A^i \doteq B^j$ (for atoms A and B) iff $A = B$ and $i = j$.
- $\neg F_1 \doteq \neg G_1$ iff $F_1 \doteq G_1$.
- $F_1 \odot F_2 \doteq G_1 \odot G_2$ (for $\odot \in \{\wedge, \vee\}$) iff $F_1 \doteq G_1$ and $F_2 \doteq G_2$.
- $(Qx)F_1 \doteq (Qx)G_1$ (for $Q \in \{\forall, \exists\}$ and $x \in V$) iff $F_1 \doteq G_1$.

$\triangle$

Definition 6.2.6 (Indexed Clause). Let $S = \Gamma \vdash \Delta$ be a clause, where Γ, Δ are (possibly empty) multisets of atoms such that $\Gamma \cup \Delta$ contains at least one indexed atom (provided that $\Gamma \cup \Delta \neq \emptyset$). Then we call S an *indexed clause*. We also define that $\vdash$ is an indexed clause. $\triangle$

Remark. The notion of subsequent is applicable to indexed clauses without further ado. Moreover, with the above definition of substitution applied to indexed formulas, substitution applied to indexed clauses works in the same way as for ordinary clauses.

Now, we extend the concept of subsumption to indexed clauses. We just need to adapt Definition 2.3.14 to indexed clauses in order to use Definition 2.3.15 also for (sets of) indexed clauses:

Definition 6.2.7. Let Γ be a multiset of indexed atomic formulas (possibly containing some atoms without indices—those are assumed to have index 0), then $set(\Gamma)$ denotes the set of indexed and index-free atomic formulas occurring in Γ. If $C = \Gamma \vdash \Delta$ is an indexed clause, then we define $set_{\mathsf{ant}}(C) = set(\Gamma)$ and $set_{\mathsf{cons}}(C) = set(\Delta)$. $\triangle$

Remark. Note that we can treat atoms without indices as indexed atoms by implicitly assuming that they have index 0.

Thus, the definitions of $\subseteq, \sqsubseteq, \leq_i, \leq_s, \leq_{ss}, \preceq$ and $\rhd$ for clause terms can be used as defined in Definitions 4.2.5, 4.2.10 and 4.2.14, respectively, for clause terms built from sets of indexed clauses. As a consequence, Lemmas 4.2.6–4.2.13 and Proposition 4.2.15 also hold for clause terms containing sets of indexed clauses.

Definition 6.2.8 (Indexed Resolution). Let C and D be indexed clauses of the form

$$\begin{array}{rcl} C & = & \Gamma \vdash \Delta, A_1^j, \ldots, A_m^j, \\ D & = & B_1^j, \ldots, B_n^j, \Pi \vdash \Lambda \end{array}$$

such that C and D are variable-disjoint, $n, m \geq 1$, and σ be a most general unifier of $\{A_1^j, \ldots, A_m^j, B_1^j, \ldots, B_n^j\}$. Then the clause

$$\Gamma\sigma, \Pi\sigma \vdash \Delta\sigma, \Lambda\sigma$$

is called an *indexed resolvent* of C and D.

The indexed resolution rule can thus be represented as follows:

$$\frac{\Gamma \vdash \Delta, A_1^j, \ldots, A_m^j \qquad B_1^j, \ldots, B_n^j, \Pi \vdash \Lambda}{\Gamma\sigma, \Pi\sigma \vdash \Delta\sigma, \Lambda\sigma}\ R_i.$$

$\triangle$

Remark. By definition, substitution applied to indexed formulas does not change the index, thus, only a set of indexed atoms in which all indexed atoms have the same index is actually unifiable. Note that indexed resolution, as opposed to general resolution, is a restricted form of resolution that only allows to resolve upon atoms that have the same index. For instance, the clauses $\vdash A^1$ and $A^2 \vdash$ cannot be resolved using indexed resolution, although—when omitting the indices—they can be resolved using the unrestricted form of general resolution.

Definition 6.2.9 (Indexed Resolution Deduction). An *indexed resolution deduction* is defined analogous to a general resolution deduction; we only replace the rules of resolution and contraction by the rules of indexed resolution and indexed contraction, respectively. *Indexed contractions* are defined analogous to contractions, but atoms can only be contracted if they have the same index. $\triangle$

The next step is to extend the indexed subsumption relation to indexed resolution deductions. We thus define, analogously to Definition 2.3.16:

Definition 6.2.10. Let γ and δ be indexed resolution deductions. We define $\gamma \leq_{ss} \delta$ by induction on the number of nodes in δ:

If δ consists of a single node labelled with an indexed clause D, then $\gamma \leq_{ss} \delta$ if γ consists of a single node labelled with an indexed clause C and $C \leq_{ss} D$.

Let δ be

$$\frac{\begin{array}{c}(\delta_1)\\ D_1\end{array} \qquad \begin{array}{c}(\delta_2)\\ D_2\end{array}}{D}\ R_i,$$

and γ_1 be an indexed resolution deduction of C_1 with $\gamma_1 \leq_{ss} \delta_1$, γ_2 be an indexed resolution deduction of C_2 with $\gamma_2 \leq_{ss} \delta_2$. Then we distinguish the following cases:

$$\text{if } C_1 \leq_{ss} D, \text{ then } \gamma_1 \leq_{ss} \delta.$$
$$\text{if } C_2 \leq_{ss} D, \text{ then } \gamma_2 \leq_{ss} \delta.$$

Otherwise, let C be a resolvent of C_1 and C_2, and let $\gamma =$

$$\frac{\begin{array}{c}(\gamma_1)\\ C_1\end{array} \qquad \begin{array}{c}(\gamma_2)\\ C_2\end{array}}{C}\ R_i.$$

Then $\gamma \leq_{ss} \delta$. The existence of such an indexed resolvent C of C_1 and C_2 having the above property follows from Lemma 6.2.24. $\triangle$

Definition 6.2.11 ($\mathsf{ACNF_{ai}}$). We say that an **LK**-proof φ is in $\mathsf{ACNF_{ai}}$ if φ is in ACNF and φ has an atom indexing. $\triangle$

Definition 6.2.12 ($\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$). We say that an **LK**-proof φ is in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ if φ is in $\mathbf{ACNF}^{\mathbf{top}}$ and φ has an atom indexing. $\triangle$

Definition 6.2.13 ($\vdash_{\mathsf{res_i}}$). Let $\mathcal{C}$ be a set of indexed clauses such that some clauses in $\mathcal{C}$ contain an indexed atom A^i. Furthermore, let D be an indexed clause such that A^i does not occur in D. Moreover, let

$$\mathcal{C}^i = \{C \mid A^i \in set(C), C \in \mathcal{C}\}.$$

Then we define $\mathcal{C} \vdash_{\mathsf{res_i}} D$ if $D \in \mathcal{C}$ or there exists an indexed resolution deduction γ of D from $\mathcal{C}^i$ such that all applications of R_i resolve upon A^i.

We extend $\vdash_{\mathsf{res_i}}$ to sets of indexed clauses as follows: Let $\mathcal{C}, \mathcal{D}$ be sets of indexed clauses such that some clauses in $\mathcal{C}$ contain an indexed atom A^i, but no clause in $\mathcal{D}$ contains A^i. Then we write $\mathcal{C} \vdash_{\mathsf{res_i}} \mathcal{D}$ if for all indexed clauses $D \in \mathcal{D}$, it holds that $\mathcal{C} \vdash_{\mathsf{res_i}} D$.

$\triangle$

Remark. The intention behind Definition 6.2.13 is to have a means to speak about the effect of term resolution w.r.t. $\mathbf{TACNF_{ai}}$ and $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$ (which operates on the syntactic level of clause terms) on the semantic level. For instance, if we have $X \vdash_{\mathsf{res_{ta}}} Y$ such that Y was obtained from X by a term resolution deduction resolving upon the subterm $\{\vdash A^i\} \oplus \{A^i \vdash\}$ or by replacing the subterm $\{A^l \vdash A^i\} \oplus \{A^i \vdash A^j\}$ in X by $\{A^l \vdash A^j\}$. Then all indexed clauses C in $|Y|$ can be obtained by an indexed resolution deduction of C from $|X|$, where in each indexed resolution step, A^i is the indexed atom resolved upon[3].

We illustrate the relation between $\vdash_{\mathsf{res_{ta}}}$ and $\vdash_{\mathsf{res_i}}$ in the following example:

Example 6.2.14. Let

$$\begin{aligned}
X &= [(\{\vdash B^2\} \oplus \{B^2 \vdash\}) \otimes (\{\vdash A^1\} \oplus (\{A^1 \vdash A^4\} \oplus \{A^4 \vdash\}))] \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\}),\\
Y &= [(\{\vdash B^2\} \oplus \{B^2 \vdash\}) \otimes (\{\vdash A^1\} \oplus \{A^1 \vdash\})] \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\}) \text{ and}\\
Z &= (\{\vdash A^1\} \oplus \{A^1 \vdash\}) \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\}).
\end{aligned}$$

Clearly, X is in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$ and Y as well as Z are in $\mathbf{TACNF_{ai}}$, and it holds that $X \vdash_{\mathsf{res_{te}}} Y$ and $Y \vdash_{\mathsf{res_t}} Z$, i.e. there are term resolution deductions of Y from X and of Z from Y, by replacing the subterm $\{A^1 \vdash A^4\} \oplus \{A^4 \vdash\}$ by $\{A^1 \vdash\}$ and by resolving upon the subterm $\{\vdash B^2\} \oplus \{B^2 \vdash\}$, respectively. We can even combine the two deductions to a single one, i.e. $X \vdash_{\mathsf{res_{te}}} Y \vdash_{\mathsf{res_t}} Z$:

[3]Recall that, by definition of $\mathbf{TACNF_{ai}}$ and semantics of clause terms, all indexed clauses in $|X|$ contain A^i.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[(\{\vdash B^2\} \oplus \{B^2 \vdash\}) \otimes (\{\vdash A^1\} \oplus (\{A^1 \vdash A^4\} \oplus \{A^4 \vdash\}))] \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})}{[(\{\vdash B^2\} \oplus \{B^2 \vdash\}) \otimes (\{\vdash A^1\} \oplus \{A^1 \vdash\} \otimes \{\vdash\})] \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})} R_t}{[(\{\vdash B^2\} \oplus \{B^2 \vdash\}) \otimes (\{\vdash A^1\} \oplus \{A^1 \vdash\})] \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})} ft_\otimes}{[(\{\vdash\} \otimes \{\vdash\}) \otimes (\{\vdash A^1\} \oplus \{A^1 \vdash\})] \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})} R_t}{[\{\vdash\} \otimes (\{\vdash A^1\} \oplus \{A^1 \vdash\})] \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})} ft_\otimes}{(\{\vdash A^1\} \oplus \{A^1 \vdash\}) \otimes (\{\vdash C^3\} \oplus \{C^3 \vdash\})} ft_\otimes.$$

By semantics of clause terms we have

$$\begin{aligned}
|X| &= \{\vdash A^1, B^2, C^3;\ A^1 \vdash B^2, C^3, A^4;\ A^4 \vdash B^2, C^3;\ B^2 \vdash A^1, C^3;\ A^1, B^2 \vdash C^3, A^4\} \\
&\quad \cup \{B^2, A^4 \vdash C^3;\ C^3 \vdash A^1, B^2;\ A^1, C^3 \vdash B^2, A^4;\ C^3, A^4 \vdash B^2;\ B^2, C^3 \vdash A^1\} \\
&\quad \cup \{A^1, B^2, C^3 \vdash A^4;\ B^2, C^3, A^4 \vdash\}, \\
|Y| &= \{\vdash A^1, B^2, C^3;\ A^1 \vdash B^2, C^3;\ B^2 \vdash A^1, C^3;\ C^3 \vdash A^1, B^2\} \\
&\quad \cup \{A^1, B^2 \vdash C^3;\ A^1, C^3 \vdash B^2;\ B^2, C^3 \vdash A^1;\ A^1, B^2, C^3 \vdash\} \text{ and} \\
|Z| &= \{\vdash A^1, C^3;\ A^1 \vdash C^3;\ C^3 \vdash A^1;\ A^1, C^3 \vdash\}.
\end{aligned}$$

Then for each indexed clause in $|Y|$, there exists an indexed resolution deduction from $|X|$ such that we only resolve upon A^4:

$$\dfrac{\dfrac{\dfrac{A^1 \vdash B^2, C^3, A^4 \qquad A^4 \vdash B^2, C^3}{A^1 \vdash B^2, B^2, C^3, C^3} R_i}{A^1 \vdash B^2, C^3, C^3} c_r}{A^1 \vdash B^2, C^3} c_r \qquad \dfrac{\dfrac{\dfrac{A^1, B^2 \vdash C^3, A^4 \qquad B^2, A^4 \vdash C^3}{A^1, B^2, B^2 \vdash C^3, C^3} R_i}{A^1, B^2 \vdash C^3, C^3} c_l}{A^1, B^2 \vdash C^3} c_r$$

$$\dfrac{\dfrac{\dfrac{A^1, C^3 \vdash B^2, A^4 \qquad C^3, A^4 \vdash B^2}{A^1, C^3, C^3 \vdash B^2, B^2} R_i}{A^1, C^3, C^3 \vdash B^2} c_r}{A^1, C^3 \vdash B^2} c_l \qquad \dfrac{\dfrac{\dfrac{A^1, B^2, C^3 \vdash A^4 \qquad B^2, C^3, A^4 \vdash}{A^1, B^2, B^2, C^3, C^3 \vdash} R_i}{A^1, B^2, C^3, C^3 \vdash} c_l}{A^1, B^2, C^3 \vdash} c_l$$

The remaining indexed clauses in $|Y|$ are already contained in $|X|$, thus they have a trivial indexed resolution deduction. Furthermore, we have

$$\begin{aligned}
|X|^4 &= \{C \mid A^4 \in set(C), C \in |X|\} \\
&= \{A^1 \vdash B^2, C^3, A^4;\ A^4 \vdash B^2, C^3;\ A^1, B^2 \vdash C^3, A^4;\ B^2, A^4 \vdash C^3\} \\
&\quad \cup \{A^1, C^3 \vdash B^2, A^4;\ C^3, A^4 \vdash B^2;\ A^1, B^2, C^3 \vdash A^4;\ B^2, C^3, A^4 \vdash\}.
\end{aligned}$$

Consequently, $|X| \vdash_{\mathsf{res}_4} D$ for all indexed clauses $D \in |Y|$, i.e. $|X| \vdash_{\mathsf{res}_4} |Y|$.

Moreover, for each indexed clause in $|Z|$, there exists an indexed resolution deduction from $|Y|$ such that we only resolve upon B^2:

$$\dfrac{\dfrac{\dfrac{\vdash A^1, B^2, C^3 \qquad B^2 \vdash A^1, C^3}{\vdash A^1, A^1, C^3, C^3}\,R_i}{\vdash A^1, C^3, C^3}\,c_r}{\vdash A^1, C^3}\,c_r \qquad\qquad \dfrac{\dfrac{\dfrac{A^1 \vdash B^2, C^3 \qquad A^1, B^2 \vdash C^3}{A^1, A^1 \vdash C^3, C^3}\,R_i}{A^1 \vdash C^3, C^3}\,c_l}{A^1 \vdash C^3}\,c_r$$

$$\dfrac{\dfrac{\dfrac{C^3 \vdash A^1, B^2 \qquad B^2, C^3 \vdash A^1}{C^3, C^3 \vdash A^1, A^1}\,R_i}{C^3, C^3 \vdash A^1}\,c_r}{C^3 \vdash A^1}\,c_l \qquad\qquad \dfrac{\dfrac{\dfrac{A^1, C^3 \vdash B^2 \qquad A^1, B^2, C^3 \vdash}{A^1, A^1, C^3, C^3 \vdash}\,R_i}{A^1, C^3, C^3 \vdash}\,c_l}{A^1, C^3 \vdash}\,c_l$$

Since all indexed clauses in $|Y|$ contain B^2, we have

$$|Y|^2 = \{C \mid B^2 \in set(C), C \in |Y|\} = |Y|.$$

Consequently, $|Y| \vdash_{\mathsf{res}_2} D$ for all indexed clauses $D \in |Z|$, i.e. $|Y| \vdash_{\mathsf{res}_2} |Z|$. Putting things together, we have $X \vdash_{\mathsf{res}_{\mathsf{ta}}} Y \vdash_{\mathsf{res}_{\mathsf{ta}}} Z$ and $|X| \vdash_{\mathsf{res}_4} |Y| \vdash_{\mathsf{res}_2} |Z|$. $\triangle$

Lemma 6.2.15 and Proposition 6.2.16 show that there is indeed a correspondence between $\vdash_{\mathsf{res}_{\mathsf{i}}}$ on the semantical and $\vdash_{\mathsf{res}_{\mathsf{te}}}$ on the syntactical level of clause terms in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$ (cf. Lemma 6.1.13 and Proposition 6.1.14).

Lemma 6.2.15. *Let X be a clause term in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$ of the following form:*

$$\{\vdash A^1\} \oplus \{A^1 \vdash A^2\} \oplus \{A^2 \vdash A^3\} \oplus \ldots \oplus \{A^{k-1} \vdash A^k\} \oplus \{A^k \vdash\},$$

where A^i is an indexed atomic formula, for $1 \leq i \leq k$. Then there exists a clause term X_{k-1} in $\mathbf{TACNF}_{\mathbf{ai}}$ such that $|X| \vdash_{\mathsf{res}_{\mathsf{i}}} |X_1| \vdash_{\mathsf{res}_{\mathsf{i}}} \ldots \vdash_{\mathsf{res}_{\mathsf{i}}} |X_{k-1}|$, where X_i is obtained from X_{i-1} by replacing the subterm $\{A^l \vdash A^i\} \oplus \{A^i \vdash A^j\}$ in X_{i-1} by $\{A^l \vdash A^j\}$, for $1 \leq i, j, l \leq k$.

Proof. Let X be a clause term in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$. We proceed by induction on the number n of indexed clauses in $|X|$.

BASE CASE: $n = 1$. Then we have $|X| = \{\vdash\}$. Trivially, $X \vdash_{\mathsf{res}_{\mathsf{i}}} X$, and since X is already in $\mathbf{TACNF}_{\mathbf{ai}}$ we are done.

INDUCTION HYPOTHESIS (IH): Assume the claim holds for all $|X|$ containing k indexed clauses with $1 \leq k \leq n$, where X is a clause term in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$.

INDUCTION STEP: Suppose w.l.o.g. that $|X|$ is a clause term in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$ containing $n+1$ indexed clauses, where

$$X = \{\vdash A^1\} \oplus (\bigoplus_{1 \leq i \leq n} \{A^i \vdash A^{i+1}\}) \oplus \{A^{n+1} \vdash\}.$$

Furthermore, assume w.l.o.g. that X_1 is the clause term in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$ obtained from X by replacing the subterm

$$\{A^{j-1} \vdash A^j\} \oplus \{A^j \vdash A^{j+1}\}$$

in X by $\{A^{j-1} \vdash A^{j+1}\}$, for some j with $1 \leq j \leq n$.

Clearly, $\{A^{j-1} \vdash A^j;\ A^j \vdash A^{j+1}\} \subseteq |X|$, but neither $A^{j-1} \vdash A^j \in |X_1|$ nor $A^j \vdash A^{j+1} \in |X_1|$. Furthermore, observe that, by semantics of clause terms, $|X_1| \setminus \{A^{j-1} \vdash A^{j+1}\} \subseteq |X|$, i.e. $A^{j-1} \vdash A^{j+1} \notin |X|$. Thus, it suffices to show that $|X| \vdash_{\mathsf{res_i}} A^{j-1} \vdash A^{j+1}$ in order to show that $|X| \vdash_{\mathsf{res_i}} |X_1|$. We can obtain an indexed resolution deduction of $A^{j-1} \vdash A^{j+1}$ from $|X|$ as follows:

$$\frac{A^{j-1} \vdash A^j \qquad A^j \vdash A^{j+1}}{A^{j-1} \vdash A^{j+1}}\ R_i.$$

Hence, $|X| \vdash_{\mathsf{res_i}} |X_1|$ and since $|X_1|$ contains fewer indexed clauses than $|X|$, we can apply the **(IH)** and obtain $|X_1| \vdash_{\mathsf{res_i}} |X_2| \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} |X_n|$, where X_n is a clause term in $\mathbf{TACNF_{ai}}$. Putting things together, we get $|X| \vdash_{\mathsf{res_i}} |X_1| \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} |X_n|$. $\square$

Proposition 6.2.16. *Let X be a clause term in $\mathbf{TACNF^{ext}_{ai}}$. Then there exists a term X_{k-1} in $\mathbf{TACNF_{ai}}$ such that $|X| \vdash_{\mathsf{res_i}} |X_1| \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} |X_{k-1}|$, where X_i is obtained from X_{i-1} by replacing the subterm $\{A^l \vdash A^i\} \oplus \{A^i \vdash A^j\}$ in X_{i-1} by $\{A^l \vdash A^j\}$, for $1 \leq i, j, l \leq k$.*

PROOF. We proceed by induction on the structure of X.

BASE CASE:

- X is in $\mathbf{TACNF_{ai}}$, we trivially have $|X| \vdash_{\mathsf{res_i}} |X|$.
- $X = \{\vdash A^1\} \oplus (\bigoplus_{1 \leq i < k}(\{A^i \vdash\} \otimes \{\vdash A^{i+1}\}) \oplus \{A^k \vdash\}$. Then there exists a clause term X_{k-1} in $\mathbf{TACNF_{ai}}$ s.t. $|X| \vdash_{\mathsf{res_i}} |X_1| \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} |X_{k-1}|$, by Lemma 6.2.15.

INDUCTION STEP: Suppose $X = X' \otimes X''$ is a clause term in $\mathbf{TACNF^{ext}_{ai}}$. Then, by **(IH)**, there exist clause terms X'_l, X'_m in $\mathbf{TACNF_{ai}}$ s.t. $|X'| \vdash_{\mathsf{res_i}} |X'_1| \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} |X'_l|$ and $|X''| \vdash_{\mathsf{res_i}} |X''_1| \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} |X''_m|$ with $l + m = k - 1$. This means there are corresponding indexed resolution deductions δ^p_i of each $S^p_i \in |X'_i|$ from $|X'_{i-1}|$ and δ^q_j of each $S^q_j \in |X''_j|$ from $|X''_{j-1}|$ of a specific form, where $X'_0 = X'$ and $X''_0 = X''$. Since $X' \otimes X''$ is in $\mathbf{TACNF^{ext}_{ai}}$, we can obtain an indexed resolution deduction of each clause $S^p_i \circ S^q_j \in |X'_i \otimes X''_j|$ from $|X'_{i-1} \otimes X''_j|$ (or from $|X'_i \otimes X''_{j-1}|$) by taking the context products of δ^p_i with S^q_j, i.e. $S^q_j \star \delta^p_i$ (or $S^p_i \star \delta^q_j$). Note that in each $\vdash_{\mathsf{res}_n}$-step on $|X'_i \otimes X''_j|$, only $|X'_i|$ or $|X''_j|$ can change, as we only resolve upon a single indexed atom with a fixed index n; moreover, we assume that S^q_j and δ^p_i as well as S^p_i and δ^q_j are both variable-disjoint.

Therefore, $|X| \vdash_{\mathsf{res_i}} |X'_1 \otimes X''| \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} |X'_l \otimes X''| \vdash_{\mathsf{res_i}} |X'_l \otimes X''_1| \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} |X'_l \otimes X''_m|$, where $X'_l \otimes X''_m$ is clearly in $\mathbf{TACNF_{ai}}$, as both X'_l and X''_m are in $\mathbf{TACNF_{ai}}$. $\square$

Remark. Note that whenever we write some sequence of the form

$$|X| \vdash_{\mathsf{res_i}} |X_1| \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} |X_{k-1}|,$$

we do not mean that in each step we resolve upon the same index i; this is just a matter of convenience in order to indicate that some $\vdash_{\mathsf{res_i}}$-step has been carried out, where the indexed atom resolved upon may have an arbitrary index occurring in the respective clause term.

The proof of Proposition 6.2.17 is inspired by the proof of Theorem 6.3 in [51].

Proposition 6.2.17 (Preservation of Cut-Indices under $>_{\mathcal{R}_{\mathsf{ax}}}$). *Let φ be an **LK**-proof with atom indexing containing at least one cut, and let φ^* be an **LK**-proof s.t. $\varphi >^*_{\mathcal{R}_{\mathsf{ax}}} \varphi^*$. Then for all cuts with auxiliary formulas F_l and F_r occurring in φ^* it holds that $F_l \simeq F_r$.*

PROOF. We proceed by induction on the number n of reduction steps

$$\varphi >_{\mathcal{R}_{\mathsf{ax}}} \varphi^1 >_{\mathcal{R}_{\mathsf{ax}}} \ldots >_{\mathcal{R}_{\mathsf{ax}}} \varphi^n = \varphi^*.$$

BASE CASE: $n = 0$. Then $\varphi = \varphi^0$. Trivially, $F_l \simeq F_r$ by Definition 6.2.3 for all cuts with auxiliary formulas F_l and F_r occurring in φ.

INDUCTION HYPOTHESIS (IH): The claim holds for all **LK**-proofs φ with atom indexing containing at least one cut such that $\varphi >_{\mathcal{R}_{\mathsf{ax}}} \varphi^k$, for $k \leq n$.

INDUCTION STEP: Suppose $\varphi >_{\mathcal{R}_{\mathsf{ax}}} \varphi^1 >_{\mathcal{R}_{\mathsf{ax}}} \ldots >_{\mathcal{R}_{\mathsf{ax}}} \varphi^n >_{\mathcal{R}_{\mathsf{ax}}} \varphi^{n+1}$. We distinguish cases according to the cut-reduction rule used in the reduction step $\varphi^n >_{\mathcal{R}_{\mathsf{ax}}} \varphi^{n+1}$. In all cases, it suffices to analyze the effect of the corresponding reduction rule on the uppermost cut in φ^n. This is so because each cut-reduction rule only modifies one cut at a time, all other cuts in the proof remain unchanged (and thus $F_l \simeq F_r$ for all cuts that are not modified).

Cut-elimination rules:

Over axioms: it might hold that $i \neq j$ or that A^j is an atom without index.

$$\dfrac{A^j \vdash A^i \qquad \overset{(\sigma)}{A^i, \Gamma \vdash \Delta}}{A^j, \Gamma \vdash \Delta}\, cut(A^i) \qquad\qquad \dfrac{\overset{(\rho)}{\Gamma \vdash \Delta, A^i} \qquad A^i \vdash A^j}{\Gamma \vdash \Delta, A^j}\, cut(A^i)$$

$$\Downarrow \qquad\qquad\qquad\qquad \Downarrow$$

$$\overset{(\sigma')}{A^j, \Gamma \vdash \Delta} \qquad\qquad\qquad\qquad \overset{(\rho')}{\Gamma \vdash \Delta, A^j}$$

Note that we have to replace A^i in σ and ρ by A^j in order to obtain the end-sequents $A^j, \Gamma \vdash \Delta$ and $\Gamma \vdash \Delta, A^j$, respectively. But this causes no problems, as there are no cuts in σ and ρ. We denote the result of replacing A^i in σ and ρ by A^j by σ' and ρ', respectively.

Over weakening:

$$\dfrac{\dfrac{\overset{(\rho')}{\Gamma \vdash \Delta}}{\Gamma \vdash \Delta, F_l}\,w_r \qquad \overset{(\sigma)}{F_r, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(F) \qquad\qquad \dfrac{\overset{(\rho)}{\Gamma \vdash \Delta, F_l} \qquad \dfrac{\overset{(\sigma')}{\Pi \vdash \Lambda}}{F_r, \Pi \vdash \Lambda}\,w_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(F)$$

$$\Downarrow \qquad\qquad\qquad\qquad \Downarrow$$

$$\dfrac{\overset{(\rho')}{\Gamma \vdash \Delta}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,w_r^*, w_l^* \qquad\qquad \dfrac{\overset{(\sigma')}{\Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,w_r^*, w_l^*$$

Since in the above four cases a cut is eliminated from φ^n, all other cuts in φ^n (that also occur in φ^{n+1}) remain unchanged. Therefore, by **(IH)**, $F_l \simeq F_r$ for all cuts in φ^{n+1} with auxiliary formulas F_l and F_r.

Grade-reduction rules:

If the cut-formula has $\neg$ as top-level connective:

$$\dfrac{\dfrac{\overset{(\rho')}{F_l, \Gamma \vdash \Delta}}{\Gamma \vdash \Delta, \neg F_l}\,\neg_r \qquad \dfrac{\overset{(\sigma')}{\Pi \vdash \Lambda, F_r}}{\neg F_r, \Pi \vdash \Lambda}\,\neg_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(\neg F)$$

$$\Downarrow$$

$$\dfrac{\overset{(\sigma')}{\Pi \vdash \Lambda, F_r} \qquad \overset{(\rho')}{F_l, \Gamma \vdash \Delta}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(F)$$

By **(IH)**, we know that $\neg F_l \simeq \neg F_r$, and thus, by definition of $\simeq$, we must have $F_l \simeq F_r$. Therefore, $F_l \simeq F_r$ for all cuts in φ^{n+1} with auxiliary formulas F_l and F_r.

If the cut-formula has $\wedge$ as top-level connective:

$$\dfrac{\dfrac{\overset{(\rho_1)}{\Gamma \vdash \Delta, F_{l_1}} \qquad \overset{(\rho_2)}{\Gamma \vdash \Delta, F_{l_2}}}{\Gamma \vdash \Delta, F_{l_1} \wedge F_{l_2}}\,\wedge_r \qquad \dfrac{\overset{(\sigma')}{F_{r_i}, \Pi \vdash \Lambda}}{F_{r_1} \wedge F_{r_2}, \Pi \vdash \Lambda}\,\wedge_{l_i}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(F_1 \wedge F_2)$$

$$\Downarrow$$

$$\dfrac{\begin{matrix}(\rho_1)\\ \Gamma \vdash \Delta, F_{l_1}\end{matrix} \qquad \dfrac{\begin{matrix}(\rho_2)\\ \Gamma \vdash \Delta, F_{l_2}\end{matrix} \qquad \dfrac{\begin{matrix}(\sigma')\\ F_{r_i}, \Pi \vdash \Lambda\end{matrix}}{F_{r_1}, F_{r_2}, \Pi \vdash \Lambda}\, w_l}{F_{r_1}, \Gamma, \Pi \vdash \Delta, \Lambda}\, cut(F_2)}{\dfrac{\Gamma, \Gamma, \Pi \vdash \Delta, \Delta, \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda}\, c_l^*, c_r^*}\, cut(F_1)$$

By **(IH)**, $F_{l_1} \wedge F_{l_2} \doteq F_{r_1} \wedge F_{r_2}$, and thus, by definition of $\doteq$, we must have $F_{l_1} \doteq F_{r_1}$ and $F_{l_2} \doteq F_{r_2}$. Therefore, $F_l \doteq F_r$ for all cuts in φ^{n+1} with auxiliary formulas F_l and F_r.

If the cut-formula has $\vee$ as top-level connective: symmetric to the case of $\wedge$.

If the cut-formula has $\forall$ as top-level connective:

$$\dfrac{\dfrac{\begin{matrix}(\rho'(x/y))\\ \Gamma \vdash \Delta, F_l(x/y)\end{matrix}}{\Gamma \vdash \Delta, (\forall x)F_l(x)}\, \forall_r \qquad \dfrac{\begin{matrix}(\sigma')\\ F_r(x/t), \Pi \vdash \Lambda\end{matrix}}{(\forall x)F_r(x), \Pi \vdash \Lambda}\, \forall_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut((\forall x)F)$$

$$\Downarrow$$

$$\dfrac{\begin{matrix}(\rho'(x/t))\\ \Gamma \vdash \Delta, F_l(x/t)\end{matrix} \qquad \begin{matrix}(\sigma')\\ F_r(x/t), \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut(F(x/t))$$

By **(IH)**, $(\forall x)F_l(x) \doteq (\forall x)F_r(x)$, and thus, by definition of $\doteq$, we must have $F_l(x) \doteq F_r(x)$. Hence, replacing x by t at the same positions in both F_l and F_r results in $F_l(x/t) \doteq F_r(x/t)$, as the indices of all atoms remain unchanged. Therefore, $F_l \doteq F_r$ for all cuts in φ^{n+1} with auxiliary formulas F_l and F_r.

If the cut-formula has $\exists$ as top-level connective: symmetric to the case of $\forall$.

Rank-reduction rules:

Over a unary rule ξ:

$$\dfrac{\dfrac{\overset{(\rho')}{\Gamma' \vdash \Delta', F_l}}{\Gamma \vdash \Delta, F_l}\,\xi \qquad \overset{(\sigma)}{F_r, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(F) \qquad\qquad \dfrac{\overset{(\rho)}{\Gamma \vdash \Delta, F_l} \qquad \dfrac{\overset{(\sigma')}{F_r, \Pi' \vdash \Lambda'}}{F_r, \Pi \vdash \Lambda}\,\xi}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(F)$$

$$\Downarrow \qquad\qquad\qquad\qquad\qquad\qquad \Downarrow$$

$$\dfrac{\dfrac{\overset{(\rho')}{\Gamma' \vdash \Delta', F_l} \qquad \overset{(\sigma)}{F_r, \Pi \vdash \Lambda}}{\Gamma', \Pi \vdash \Delta', \Lambda}\,cut(F)}{\Gamma, \Pi \vdash \Delta, \Lambda}\,\xi \qquad\qquad \dfrac{\dfrac{\overset{(\rho)}{\Gamma \vdash \Delta, F_l} \qquad \overset{(\sigma')}{F_r, \Pi' \vdash \Lambda'}}{\Gamma, \Pi' \vdash \Delta, \Lambda'}\,cut(F)}{\Gamma, \Pi \vdash \Delta, \Lambda}\,\xi$$

In both cases, by **(IH)**, $F_l \simeq F_r$, and since the transformation only shifted the cut upwards, the cut-formula (and thus the auxiliary formulas F_l and F_r) did not change, we still have $F_l \simeq F_r$. Therefore, $F_l \simeq F_r$ for all cuts in φ^{n+1} with auxiliary formulas F_l and F_r.

Over a binary rule ξ:

$$\dfrac{\dfrac{\overset{(\rho_1)}{\Gamma_1 \vdash \Delta_1, F_l} \qquad \overset{(\rho_2)}{\Gamma_2 \vdash \Delta_2}}{\Gamma \vdash \Delta, F_l}\,\xi \qquad \overset{(\sigma)}{F_r, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(F)$$

$$\Downarrow$$

$$\dfrac{\dfrac{\dfrac{\overset{(\rho_1)}{\Gamma_1 \vdash \Delta_1, F_l} \qquad \overset{(\sigma)}{F_r, \Pi \vdash \Lambda}}{\Gamma_1, \Pi \vdash \Delta_1, \Lambda}\,cut(F) \qquad \dfrac{\dfrac{\overset{(\rho_2)}{\Gamma_2 \vdash \Delta_2}}{\Gamma_2 \vdash \Delta_2, F_l}\,w_r^* \qquad \overset{(\sigma')}{F_r, \Pi \vdash \Lambda}}{\Gamma_2, \Pi \vdash \Delta_2, \Lambda}\,cut(F)}{\Gamma, \Pi, \Pi \vdash \Delta, \Lambda, \Lambda}\,\xi}{\Gamma, \Pi \vdash \Delta, \Lambda}\,c_l^*, c_r^*$$

where σ' is the regularized version of σ. Note that regularization has no effect on indices, thus the regularization causes no problems w.r.t. $\simeq$.
In both cases, by **(IH)**, $F_l \simeq F_r$, and since the transformation only shifted the cut upwards, the cut-formula (and thus the auxiliary formulas F_l and F_r) did not change, we still have $F_l \simeq F_r$. Therefore, $F_l \simeq F_r$ for all cuts in φ^{n+1} with auxiliary formulas F_l and F_r. The argument for the other binary cases is analogous.

Over contraction rules:

Contraction right c_r:

$$\cfrac{\cfrac{\begin{matrix}(\rho')\\ \Gamma \vdash \Delta, F_l, F_l\end{matrix}}{\Gamma \vdash \Delta, F_l}\,c_r \qquad \begin{matrix}(\sigma)\\ F_r, \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(F)$$

$$\Downarrow$$

$$\cfrac{\cfrac{\cfrac{\begin{matrix}(\rho')\\ \Gamma \vdash \Delta, F_l, F_l\end{matrix} \qquad \begin{matrix}(\sigma)\\ F_r, \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi \vdash \Delta, \Lambda, F_l}\,cut(F) \qquad \begin{matrix}(\sigma')\\ F_r, \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi, \Pi \vdash \Delta, \Lambda, \Lambda}\,cut(F)}{\Gamma, \Pi \vdash \Delta, \Lambda}\,c_l^*, c_r^*$$

where σ' is the regularized version of σ. Note that regularization has no effect on indices, thus the regularization causes no problems w.r.t. $\doteq$.
By **(IH)**, $F_l \doteq F_r$, and since the cut and thus the cut-formula together with the indices are duplicated in φ^{n+1}, we have for both cuts on F in φ^{n+1} that $F_l \doteq F_r$ holds. Therefore, $F_l \doteq F_r$ for all cuts in φ^{n+1} with auxiliary formulas F_l and F_r.

Contraction left c_l: analogous to contraction right.

□

Corollary 6.2.18 (Preservation of Cut-Indices under $>_{\mathcal{R}^{\mathsf{top}}}$). *Let φ be an **LK**-proof with atom indexing containing at least one cut, and let φ^* be an **LK**-proof such that $\varphi >^*_{\mathcal{R}^{\mathsf{top}}} \varphi^*$. Then for all cuts with auxiliary formulas F_l and F_r occurring in φ^* it holds that $F_l \doteq F_r$.*

PROOF. Follows immediately from the fact that $>_{\mathcal{R}^{\mathsf{top}}} \subseteq >_{\mathcal{R}_{\mathsf{ax}}}$. □

Next, we show that Lemma 5.2.1 also holds for **LK**-proofs with atom indexing:

Lemma 6.2.19. *Let φ, φ' be **LK**-proofs with atom indexing such that $\varphi >_{\mathcal{R}} \varphi'$ for a cut-reduction relation $>_{\mathcal{R}}$ based on $\mathcal{R}$. Then $\Theta(\varphi) \rhd \Theta(\varphi')$.*

PROOF. The proof is basically a modified version of the proof of Lemma 6.1 in [9].
We construct a proof by cases on the definition of $>_{\mathcal{R}}$. To this end, we consider subderivations ψ of φ of the form

$$\cfrac{\begin{matrix}(\rho, X)\\ \Gamma \vdash \Delta, F\end{matrix} \qquad \begin{matrix}(\sigma, Y)\\ F, \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(F)$$

where $X = \Theta(\varphi)/\lambda$, for the position λ corresponding to the deduction ρ and $Y = \Theta(\varphi)/\mu$, for the position μ corresponding to the deduction σ. By ν we denote the position of ψ in φ. That means we do not only indicate the subderivations ending in the cut but also the corresponding clause terms. Note that by definition of the characteristic clause term, we have $\Theta(\varphi)/\nu = X \oplus Y$.

If $\psi >_{\mathcal{R}} \chi$, then, by definition of the reduction relation $>_{\mathcal{R}}$, we get $\varphi = \varphi[\psi]_\nu >_{\mathcal{R}} \varphi[\chi]_\nu$. For the remaining part of the proof, we denote $\varphi[\chi]_\nu$ by φ'. Our aim is to prove that $\Theta(\varphi) \rhd \Theta(\varphi')$. By Proposition 6.2.17, for all cuts with auxiliary formulas F_l and F_r the indices of the atoms of F_l and F_r coincide, i.e. $F_l \doteqdot F_r$.
In the following, we represent that $\psi >_{\mathcal{R}} \chi$ by $\Downarrow$, where ψ occurs above and χ below $\Downarrow$.

Cut-elimination rules:

Since $>_{\mathcal{R}}$ contains no cut-elimination rules over axioms, we only have to consider the ones over weakening.

Over weakening:

$$\dfrac{\dfrac{\begin{matrix}(\rho', X)\\ \Gamma \vdash \Delta\end{matrix}}{\Gamma \vdash \Delta, F}\, w_r \qquad \begin{matrix}(\sigma, Y)\\ F, \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut(F)$$

$$\Downarrow$$

$$\dfrac{\begin{matrix}(\rho', X)\\ \Gamma \vdash \Delta\end{matrix}}{\Gamma, \Pi \vdash \Delta, \Lambda}\, w_r^*, w_l^*$$

Therefore, also $\varphi[\psi]_\nu >_{\mathcal{R}} \varphi[\chi]_\nu$, i.e. $\varphi >_{\mathcal{R}} \varphi'$. But $\Theta(\varphi')/\nu = X$ and $\Theta(\varphi)/\nu = X \oplus Y$. Clearly, $X \oplus Y \rhd X$, as $|X| \subseteq |X \oplus Y| = |X| \cup |Y|$, and, by Lemma 4.2.11, $\Theta(\varphi) \rhd \Theta(\varphi')$.

The other weakening case is symmetric to the one above.

Grade-reduction rules:

If the cut-formula has $\neg$ as top-level connective, i.e. $F = \neg G$:

$$\dfrac{\dfrac{\begin{matrix}(\rho', X)\\ G, \Gamma \vdash \Delta\end{matrix}}{\Gamma \vdash \Delta, \neg G}\, \neg_r \qquad \dfrac{\begin{matrix}(\sigma', Y)\\ \Pi \vdash \Lambda, G\end{matrix}}{\neg G, \Pi \vdash \Lambda}\, \neg_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut(F)$$

$$\Downarrow$$

$$\dfrac{\dfrac{(\sigma', Y)}{\Pi \vdash \Lambda, G} \qquad \dfrac{(\rho', X)}{G, \Gamma \vdash \Delta}}{\Gamma, \Pi \vdash \Delta, \Lambda}\; cut(G)$$

In this case we have

$$\Theta(\varphi)/\nu = X \oplus Y,$$
$$\Theta(\varphi')/\nu = Y \oplus X.$$

Clearly, $X \oplus Y \rhd Y \oplus X$ (we even have $X \oplus Y \sim Y \oplus X$) and, by Lemma 4.2.11, we obtain $\Theta(\varphi) \rhd \Theta(\varphi')$.

If the cut-formula has $\wedge$ as top-level connective, i.e. $F = F_1 \wedge F_2$:

$$\dfrac{\dfrac{\dfrac{(\rho_1, X_1)}{\Gamma \vdash \Delta, F_1} \qquad \dfrac{(\rho_2, X_2)}{\Gamma \vdash \Delta, F_2}}{\Gamma \vdash \Delta, F_1 \wedge F_2}\;\wedge_r \qquad \dfrac{\dfrac{(\sigma', Y)}{F_i, \Pi \vdash \Lambda}}{F_1 \wedge F_2, \Pi \vdash \Lambda}\;\wedge_{l_i}}{\Gamma, \Pi \vdash \Delta, \Lambda}\; cut(F)$$

$$\Downarrow$$

$$\dfrac{\dfrac{\dfrac{(\rho_1, X_1)}{\Gamma \vdash \Delta, F_1} \qquad \dfrac{\dfrac{(\rho_2, X_2)}{\Gamma \vdash \Delta, F_2} \qquad \dfrac{\dfrac{(\sigma', Y)}{F_i, \Pi \vdash \Lambda}}{F_1, F_2, \Pi \vdash \Lambda}\; w_l}{F_1, \Gamma, \Pi \vdash \Delta, \Lambda}\; cut(F_2)}{\Gamma, \Gamma, \Pi \vdash \Delta, \Delta, \Lambda}\; cut(F_1)}{\Gamma, \Pi \vdash \Delta, \Lambda}\; c_l^*, c_r^*$$

In this case we have

$$\Theta(\varphi)/\nu = (X_1 \oplus X_2) \oplus Y,$$
$$\Theta(\varphi')/\nu = X_1 \oplus (X_2 \oplus Y).$$

Clearly, $(X_1 \oplus X_2) \oplus Y \sim X_1 \oplus (X_2 \oplus Y)$, by elementary properties of $\cup$. Thus, $\Theta(\varphi)/\nu \rhd \Theta(\varphi')/\nu$ and, by Lemma 4.2.11, we obtain $\Theta(\varphi) \rhd \Theta(\varphi')$.

If the cut-formula has $\vee$ as top-level connective, i.e. $F = F_1 \vee F_2$: symmetric to $\wedge$.

If the cut-formula has $\forall$ as top-level connective, i.e. $F = (\forall x)G$:

$$\dfrac{\dfrac{\begin{matrix}(\rho'(x/y), X(x/y))\\ \Gamma \vdash \Delta, G(x/y)\end{matrix}}{\Gamma \vdash \Delta, (\forall x)G(x)}\,\forall_r \qquad \dfrac{\begin{matrix}(\sigma', Y)\\ G(x/t), \Pi \vdash \Lambda\end{matrix}}{(\forall x)G(x), \Pi \vdash \Lambda}\,\forall_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(F)$$

$$\Downarrow$$

$$\dfrac{\begin{matrix}(\rho'(x/t), X(x/t))\\ \Gamma \vdash \Delta, G(x/t)\end{matrix} \qquad \begin{matrix}(\sigma', Y)\\ G(x/t), \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(G(x/t))$$

In this case we have

$$\Theta(\varphi)/\nu = X(x/y) \oplus Y,$$
$$\Theta(\varphi')/\nu = X(x/t) \oplus Y.$$

By assumption, φ is regular and so the variable y only occurs in the subderivation ρ. Therefore, $\Theta(\varphi')/\nu = (X(x/y) \oplus Y)\{y \leftarrow t\}$, and even $\Theta(\varphi') = \Theta(\varphi)\{y \leftarrow t\}$. But this means $\Theta(\varphi) \leq_s \Theta(\varphi')$, and therefore $\Theta(\varphi) \rhd \Theta(\varphi')$.

If the cut-formula has $\exists$ as top-level connective, i.e. $F = (\exists x)G$: symmetric to $\forall$.

Rank-reduction rules:

Over a unary rule ξ:

$$\dfrac{\dfrac{\begin{matrix}(\rho', X)\\ \Gamma' \vdash \Delta', F\end{matrix}}{\Gamma \vdash \Delta, F}\,\xi \qquad \begin{matrix}(\sigma, Y)\\ F, \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(F)$$

$$\Downarrow$$

$$\dfrac{\dfrac{\begin{matrix}(\rho', X)\\ \Gamma' \vdash \Delta', F\end{matrix} \qquad \begin{matrix}(\sigma, Y)\\ F, \Pi \vdash \Lambda\end{matrix}}{\Gamma', \Pi \vdash \Delta', \Lambda}\,cut(F)}{\Gamma, \Pi \vdash \Delta, \Lambda}\,\xi$$

In this case we have

$$\Theta(\varphi)/\nu = X \oplus Y,$$
$$\Theta(\varphi')/\nu = X \oplus Y.$$

Since $\Theta(\varphi)/\nu = \Theta(\varphi')/\nu$, we also have $\Theta(\varphi) = \Theta(\varphi')$, and thus $\Theta(\varphi) \rhd \Theta(\varphi')$. Analogous for the other unary case.

Over a binary rule ξ:

$$\dfrac{\dfrac{\dfrac{(\rho_1, X_1)}{\Gamma_1 \vdash \Delta_1, F} \quad \dfrac{(\rho_2, X_2)}{\Gamma_2 \vdash \Delta_2}}{\Gamma \vdash \Delta, F}\,\xi \qquad \dfrac{(\sigma, Y)}{F, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(F)$$

$$\Downarrow$$

$$\dfrac{\dfrac{\Gamma, \Pi, \Pi \vdash \Delta, \Lambda, \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda}\,c_l^*, c_r^*}{}$$

$$\dfrac{\dfrac{\dfrac{(\rho_1, X_1)}{\Gamma_1 \vdash \Delta_1, F} \quad \dfrac{(\sigma, Y)}{F, \Pi \vdash \Lambda}}{\Gamma_1, \Pi \vdash \Delta_1, \Lambda}\,cut(F) \qquad \dfrac{\dfrac{\dfrac{(\rho_2, X_2)}{\Gamma_2 \vdash \Delta_2}}{\Gamma_2 \vdash \Delta_2, F}\,w_r^* \quad \dfrac{(\sigma', Y')}{F, \Pi \vdash \Lambda}}{\Gamma_2, \Pi \vdash \Delta_2, \Lambda}\,cut(F)}{\dfrac{\Gamma, \Pi, \Pi \vdash \Delta, \Lambda, \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda}\,c_l^*, c_r^*}\,\xi$$

where σ' and Y' are obtained by regularization from σ and Y, respectively. Now, we have to distinguish two cases:

(i) The principal formula of ξ is an ancestor of another cut in φ:
In this case we have

$$\Theta(\varphi)/\nu = (X_1 \oplus X_2) \oplus Y,$$
$$\Theta(\varphi')/\nu = (X_1 \oplus Y) \oplus (X_2 \oplus Y').$$

Clearly, $(X_1 \oplus X_2) \oplus Y \sim (X_1 \oplus Y) \oplus (X_2 \oplus Y)$, and thus, it follows that

$$(X_1 \oplus Y) \oplus (X_2 \oplus Y) \subseteq (X_1 \oplus X_2) \oplus Y.$$

Let ϑ be the renaming substitution that only maps the eigenvariables of Y to the eigenvariables of Y', i.e. $Y\vartheta = Y'$. By the regularity of φ and φ', it holds that the eigenvariables of Y and Y' do neither occur in X_1 nor in X_2. As a consequence, $X_1\vartheta = X_1$ and $X_2\vartheta = X_2$, and thus,

$$(X_2 \oplus Y)\vartheta = (X_2\vartheta \oplus Y\vartheta) = (X_2 \oplus Y').$$

But this means

$$|(X_1 \oplus Y) \oplus (X_2 \oplus Y) \oplus (X_2 \oplus Y)\vartheta| = |(X_1 \oplus Y) \oplus (X_2 \oplus Y) \oplus (X_2 \oplus Y')|.$$

Furthermore, by elementary properties of $\cup$, we have

$$|(X_1 \oplus Y) \oplus (X_2 \oplus Y) \oplus (X_2 \oplus Y')| = |(X_1 \oplus Y) \oplus (X_2 \oplus Y')| = |\Theta(\varphi')/\nu|.$$

Therefore, $\Theta(\varphi)/\nu \leq_i \Theta(\varphi')/\nu$ and, by Lemma 4.2.13, $\Theta(\varphi) \leq_i \Theta(\varphi')$, i.e. we have $\Theta(\varphi) \rhd \Theta(\varphi')$.

(ii) The principal formula of ξ is not an ancestor of a cut in φ:
In this case we have

$$\begin{aligned} \Theta(\varphi)/\nu &= (X_1 \otimes X_2) \oplus Y, \\ \Theta(\varphi')/\nu &= (X_1 \oplus Y) \otimes (X_2 \oplus Y). \end{aligned}$$

By elementary properties of $\cup$ and $\times$, we obtain

$$(X_1 \otimes X_2) \oplus Y \sqsubseteq (X_1 \oplus Y) \otimes (X_2 \oplus Y).$$

This means $\Theta(\varphi)/\nu \sqsubseteq \Theta(\varphi')/\nu$ and, by Lemma 4.2.12, we get $\Theta(\varphi) \sqsubseteq \Theta(\varphi')$. Therefore, also $\Theta(\varphi) \rhd \Theta(\varphi')$.

The other binary cases are analogous/symmetric.

Over contraction rules:

Contraction right c_r:

$$\dfrac{\dfrac{\overset{(\rho', X)}{\Gamma \vdash \Delta, F, F}}{\Gamma \vdash \Delta, F}\,c_r \qquad \overset{(\sigma, Y)}{F, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut(F)$$

$$\Downarrow$$

$$\dfrac{\dfrac{\dfrac{\overset{(\rho', X)}{\Gamma \vdash \Delta, F, F} \qquad \overset{(\sigma, Y)}{F, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda, F}\,cut(F) \qquad \overset{(\sigma', Y')}{F, \Pi \vdash \Lambda}}{\Gamma, \Pi, \Pi \vdash \Delta, \Lambda, \Lambda}\,cut(F)}{\Gamma, \Pi \vdash \Delta, \Lambda}\,c_l^*, c_r^*$$

where σ' and Y' are obtained by regularization from σ and Y, respectively. In this case we have

$$\Theta(\varphi)/\nu = X \oplus Y,$$
$$\Theta(\varphi')/\nu = (X \oplus Y) \oplus Y'.$$

Let ϑ be the renaming substitution that only maps the eigenvariables of Y to the eigenvariables of Y', i.e. $Y\vartheta = Y'$. Clearly, $X \oplus Y \subseteq X \oplus Y$, and thus

$$|(X \oplus Y) \oplus Y\vartheta| = |(X \oplus Y) \oplus Y'| = |\Theta(\varphi')/\nu|.$$

Therefore, $\Theta(\varphi)/\nu \leq_i \Theta(\varphi')/\nu$ and, by Lemma 4.2.13, $\Theta(\varphi) \leq_i \Theta(\varphi')$, i.e. we have $\Theta(\varphi) \rhd \Theta(\varphi')$.

Contraction left c_l: analogous to contraction right.

□

Corollary 6.2.20. *Let φ, φ' be **LK**-proofs with atom indexing such that $\varphi >_{\mathcal{R}^{\mathsf{top}}} \varphi'$ for a cut-reduction relation $>_{\mathcal{R}^{\mathsf{top}}}$ based on $\mathcal{R}$. Then $\Theta(\varphi) \rhd \Theta(\varphi')$.*

PROOF. Holds by Corollary 6.2.18 and the fact that $>_{\mathcal{R}^{\mathsf{top}}}$ is based on $\mathcal{R}$. □

Theorem 6.2.21 shows that the characteristic clause set $\mathrm{CL}(\psi)$ of an $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ of an **LK**-proof φ with atom indexing is subsumed by the original characteristic clause set $\mathrm{CL}(\varphi)$. As a consequence, reductive methods are still redundant w.r.t. the results of **CERES** if we consider indexed proofs up to the point where all atomic cuts are shifted to the top, i.e. only non-atomic cuts have been eliminated.

Theorem 6.2.21. *Let φ be an **LK**-proof with atom indexing and ψ be an $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ of φ under a cut-reduction relation $>_{\mathcal{R}^{\mathsf{top}}}$ based on $\mathcal{R}$. Then $\Theta(\varphi) \leq_{ss} \Theta(\psi)$.*

PROOF. Suppose $\varphi >^*_{\mathcal{R}^{\mathsf{top}}} \psi$. Then by Corollary 6.2.20 we get $\Theta(\varphi) \rhd^* \Theta(\psi)$. By Proposition 4.2.15 we obtain $\Theta(\varphi) \leq_{ss} \Theta(\psi)$. □

The following result constitutes the indexed version of [34, Proposition 4.2.1].

Proposition 6.2.22. *$\leq_{ss}$ for indexed clauses fulfills the following properties:*

(i) *Reflexivity,*

(ii) *Transitivity,*

(iii) *If $C \leq_{ss} D$, then $C \leq_{ss} D\vartheta$ for all substitutions ϑ.*

PROOF.

(i) Just take the empty substitution.

(ii) If

$$set_{\mathsf{ant}}(C)\vartheta \subseteq set_{\mathsf{ant}}(D) \text{ and } set_{\mathsf{cons}}(C)\vartheta \subseteq set_{\mathsf{cons}}(D) \text{ as well as}$$
$$set_{\mathsf{ant}}(D)\eta \subseteq set_{\mathsf{ant}}(E) \text{ and } set_{\mathsf{cons}}(D)\eta \subseteq set_{\mathsf{cons}}(E).$$

Then also $set_{\mathsf{ant}}(C)\vartheta\eta \subseteq set_{\mathsf{ant}}(E)$ and $set_{\mathsf{cons}}(C)\vartheta\eta \subseteq set_{\mathsf{cons}}(E)$.

(iii) If $set_{\mathsf{ant}}(C)\vartheta \subseteq set_{\mathsf{ant}}(D)$ and $set_{\mathsf{cons}}(C)\vartheta \subseteq set_{\mathsf{cons}}(D)$.
Then also $set_{\mathsf{ant}}(C)\vartheta\eta \subseteq set_{\mathsf{ant}}(D\eta)$ and $set_{\mathsf{cons}}(C)\vartheta\eta \subseteq set_{\mathsf{cons}}(D\eta)$.

□

Lemma 6.2.23. *Let C and D be indexed clauses with $C \leq_{ss} D$, and let D' be an arbitrary factor of D. Then $C \leq_{ss} D'$.*

PROOF. Let $C = \Gamma \vdash \Delta$ and $D = \Pi \vdash \Lambda$ be arbitrary indexed clauses such that $C \leq_{ss} D$, i.e. there exists a substitution ϑ such that $set(\Gamma)\vartheta \subseteq set(\Pi)$ and $set(\Delta)\vartheta \subseteq set(\Lambda)$.

Furthermore, let D' be an arbitrary factor of D, i.e. there exists an m.g.u. σ of some nonempty subset $\Pi' \subseteq \Pi$ or $\Lambda' \subseteq \Lambda$ and D' is a contraction normalization of $D\sigma$.

We distinguish two cases:

(i) σ is an m.g.u. of Π'.

This means $\Pi'\sigma = \{B^i\}$ for some indexed atom B^i. Assume $\Pi' = \{B^i_1, \ldots, B^i_n\}$, for indexed atoms B^i_k with $1 \leq k \leq n$ (the B^i_k are only unifiable if they have the same index). Furthermore, assume w.l.o.g. that the arguments of the elements in Π' are pairwise distinct. Then

$$D' = B^i, \Pi\sigma \setminus \{B^i\} \vdash \Lambda\sigma.$$

Since $C \leq_{ss} D$, we know that

$$\begin{array}{rcl} set(\Gamma)\vartheta & \subseteq & set(\{B^i_1, \ldots, B^i_n\}) \cup set((\Pi \setminus \{B^i_1, \ldots, B^i_n\}) \text{ and} \\ set(\Delta)\vartheta & \subseteq & set(\Lambda). \end{array}$$

Let $P^j \in set(\Gamma)$ be arbitrary, then $P^j\vartheta \in set(\{B^i_1, \ldots, B^i_n\}) \cup set((\Pi \setminus \{B^i_1, \ldots, B^i_n\})$.
Again, we distinguish cases:

(a) $P^j\vartheta \in set(\{B^i_1, \ldots, B^i_n\})$.
In this case, there is some B^i_k with $1 \leq k \leq n$ such that $P^j\vartheta = B^i_k$ (since $set(\{B^i_1, \ldots, B^i_n\}) = \{B^i_1, \ldots, B^i_n\}$), i.e. $P\vartheta = B_k$ and $i = j$. Moreover, since $\{B^i_1, \ldots, B^i_n\}\sigma = \{B^i\}$, it follows that $P^j\vartheta\sigma = B^i_k\sigma = B^i$, where $P\vartheta\sigma = B$ and $i = j$. Hence, we have that $P^j\vartheta\sigma \in set(\{B^i\} \cup \Pi\sigma \setminus \{B^i\})$.
Therefore, $set(\Gamma)\vartheta\sigma \subseteq set(\{B^i\} \cup (\Pi\sigma \setminus \{B^i\}))$, and since Λ in D' coincides with Λ in D, we have $set(\Delta)\vartheta\sigma \subseteq set(\Lambda)\sigma$. Putting things together, we obtain $C \leq_{ss} D'$.

(b) $P^j\vartheta \notin set(\{B^i_1, \dots, B^i_n\})$.
In this case, we must have $P^j\vartheta \in set((\Pi \setminus \{B^i_1, \dots, B^i_n\}))$. But then it holds that $P^j\vartheta\sigma \in set((\Pi \setminus \{B^i_1, \dots, B^i_n\}))\sigma = set(\Pi\sigma \setminus \{B^i\})$.
Therefore, $set(\Gamma)\vartheta\sigma \subseteq set(\{B^i\} \cup (\Pi\sigma \setminus \{B^i\}))$, and since Λ in D' coincides with Λ in D, we have $set(\Delta)\vartheta\sigma \subseteq set(\Lambda)\sigma$. Putting things together, we obtain $C \leq_{ss} D'$.

(ii) σ is an m.g.u. of Λ'. Analogous to (i).

□

We now prove [34, Lemma 4.2.1] for indexed resolution.

Lemma 6.2.24. *Let C_1, D_1, C_2 and D_2 be indexed clauses such that $C_1 \leq_{ss} D_1$ and $C_2 \leq_{ss} D_2$. Furthermore, let D be an indexed resolvent of D_1 and D_2. Then one of the following properties holds:*

(a) $C_1 \leq_{ss} D$, or

(b) $C_2 \leq_{ss} D$, or

(c) there exists an indexed resolvent C of C_1 and C_2 such that $C \leq_{ss} D$.

PROOF. Let C_1, D_1, C_2 and D_2 be indexed clauses such that $C_1 \leq_{ss} D_1$ and $C_2 \leq_{ss} D_2$, and let D be an indexed resolvent of D_1 and D_2. Since D is an indexed resolvent of D_1 and D_2, there exist (variable-disjoint variants of) factors $D_1' = \Gamma_1 \vdash \Delta_1, M^i$ and $D_2' = N^i, \Gamma_2 \vdash \Delta_2$ of D_1 and D_2 (or $D_1' = M^i, \Gamma_1 \vdash \Delta_1$ and $D_2' = \Gamma_2 \vdash \Delta_2, N^i$), respectively, such that $\{M^i, N^i\}$ is unifiable by some m.g.u. σ. Thus,

$$D = \Gamma_1\sigma, \Gamma_2\sigma \vdash \Delta_1\sigma, \Delta_2\sigma.$$

By assumption $C_1 \leq_{ss} D_1$ and $C_2 \leq_{ss} D_2$, and since D_1' and D_2' are factors of D_1 and D_2, respectively, we obtain by Lemma 6.2.23:

$$C_1 \leq_{ss} \Gamma_1 \vdash \Delta_1, M^i \text{ and } C_2 \leq_{ss} N^i, \Gamma_2 \vdash \Delta_2. \qquad (*)$$

From $(*)$ and the definition of $\leq_{ss}$, we know that there must be a substitution ϑ_1 such that

$$set_{\mathsf{ant}}(C_1)\vartheta_1 \subseteq set(\Gamma_1) \text{ and } set_{\mathsf{cons}}(C_1)\vartheta_1 \subseteq set(\Delta_1 \cup \{M^i\}).$$

Similarly, there is a substitution ϑ_2 such that

$$set_{\mathsf{ant}}(C_2)\vartheta_2 \subseteq set(\{N^i\} \cup \Gamma_2) \text{ and } set_{\mathsf{cons}}(C_2)\vartheta_2 \subseteq set(\Delta_2).$$

We distinguish cases:

(i) $M^i \notin set_{\mathsf{cons}}(C_1)\vartheta_1$ or $N^i \notin set_{\mathsf{ant}}(C_2)\vartheta_2$. Suppose $M^i \notin set_{\mathsf{cons}}(C_1)\vartheta_1$. Then $set_{\mathsf{cons}}(C_1)\vartheta_1 \subseteq set(\Delta_1)$, and thus, since $set_{\mathsf{ant}}(C_1)\vartheta_1 \subseteq set(\Gamma_1)$, we get that $C_1 \leq_{ss} \Gamma_1 \vdash \Delta_1$. Moreover, we have

$$\Gamma_1 \vdash \Delta_1 \leq_{ss} \Gamma_1\sigma \vdash \Delta_1\sigma \leq_{ss} \Gamma_1\sigma, \Gamma_2\sigma \vdash \Delta_1\sigma, \Delta_2\sigma = D.$$

By transitivity of $\leq_{ss}$, we get $C_1 \leq_{ss} D$, and thus property (a) holds.

For $N^i \notin set_{\mathsf{ant}}(C_2)\vartheta_2$, a completely analogous argument yields $C_2 \leq_{ss} \Gamma_2 \vdash \Delta_2$ and $C_2 \leq_{ss} D$, i.e. property (b) holds.

(ii) $M^i \in set_{\mathsf{cons}}(C_1)\vartheta_1$ and $N^i \in set_{\mathsf{ant}}(C_2)\vartheta_2$. Let $\mathcal{L}_1 = \{L_1^i, \ldots, L_m^i\}$ be the set of all indexed atoms L^i in $set_{\mathsf{cons}}(C_1)$ with $L^i\vartheta_1 = M^i$.

Similarly, we define $\mathcal{L}_2$ for C_2 and ϑ_2. Then ϑ_1 and ϑ_2 are unifiers of $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively. By the unification theorem (Theorem 2.3.4), there exist m.g.u.'s λ_1 and λ_2 of $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively. Applying contraction normalization to the clauses $C_1\lambda_1$ and $C_2\lambda_2$ yields the factors

$$\Gamma_1' \vdash \Delta_1', R^i \text{ and } S^i, \Gamma_2' \vdash \Delta_2'$$

fulfilling the following properties: $R^i \leq_s M^i$ and $S^i \leq_s N^i$ as well as

$$\Gamma_1' \vdash \Delta_1' \leq_{ss} \Gamma_1 \vdash \Delta_1 \text{ and } \Gamma_2' \vdash \Delta_2' \leq_{ss} \Gamma_2 \vdash \Delta_2.$$

We even know that $R^i \leq_s M^i$ and $\Gamma_1' \vdash \Delta_1' \leq_{ss} \Gamma_1 \vdash \Delta_1$ via a common substitution (the same holds for $S^i \leq_s N^i$ and $\Gamma_2' \vdash \Delta_2' \leq_{ss} \Gamma_2 \vdash \Delta_2$).

So, let η_1, η_2 be defined as follows:

$$\lambda_1\eta_1 = \vartheta_1 \text{ and } \lambda_2\eta_2 = \vartheta_2.$$

By definition of the indexed resolvent D, σ is an m.g.u. of $\{M^i, N^i\}$.

Let $\eta = \eta_1 \cup \eta_2$ (note that such a definition is possible, as $\Gamma_1 \vdash \Delta_1, M^i$ and $N^i, \Gamma_2 \vdash \Delta_2$ are variable-disjoint). By $R^i\eta = M^i$ and $S^i\eta = N^i$, the set $\{R^i, S^i\}$ is unifiable by the substitution $\eta\sigma$. Moreover, by the unification theorem (Theorem 2.3.4), there exists an m.g.u. τ of $\{R^i, S^i\}$.

Since τ is an m.g.u., there must be a substitution ρ such that $\tau\rho = \eta\sigma$.

We thus obtain

$$R^i\tau\rho = M^i\sigma \text{ and } S^i\tau\rho = N^i\sigma,$$

and

$$set(\Gamma_1')\tau\rho \subseteq set(\Gamma_1)\sigma \text{ and } set(\Delta_1')\tau\rho \subseteq set(\Delta_1)\sigma$$
$$set(\Gamma_2')\tau\rho \subseteq set(\Gamma_2)\sigma \text{ and } set(\Delta_2')\tau\rho \subseteq set(\Delta_2)\sigma.$$

But the clause

$$C = \Gamma_1'\tau, \Gamma_2'\tau \vdash \Delta_1'\tau, \Delta_2'\tau$$

is an indexed resolvent of C_1 and C_2, and

$$set(\Gamma_1' \cup \Gamma_2')\tau\rho \subseteq set(\Gamma_1 \cup \Gamma_2)\sigma \text{ and } set(\Delta_1' \cup \Delta_2')\tau\rho \subseteq set(\Delta_1 \cup \Delta_2)\sigma.$$

But this means that $C \leq_{ss} D$, and thus property (c) holds.

□

The following result, which is an indexed version of Proposition 2.3.17, will be important for proving Theorem 6.3.15.

Proposition 6.2.25. *Let $\mathcal{C}, \mathcal{D}$ be sets of indexed clauses with $\mathcal{C} \leq_{ss} \mathcal{D}$, and let δ be an indexed resolution deduction from $\mathcal{D}$. Then there exists an indexed resolution deduction γ from $\mathcal{C}$ such that $\gamma \leq_{ss} \delta$.*

PROOF. By Definition 6.2.10 and Lemma 6.2.24. □

6.3 COMPLETENESS AND ALL THAT

Finally, we will use the methods and results from the previous sections in order to show that **CERES** indeed simulates reductive cut-elimination methods if we eliminate atomic cuts too. To this end, we will first show that term resolution applied to characteristic clause terms extracted from a proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ is complete. This is done by showing that, from such a clause term, we can always obtain a sequence of $\vdash_{\mathsf{res}_{\mathsf{ta}}}$-steps that eventually leads to the clause term $\{\vdash\}$. Subsequently, this completeness result will help us to show that indexed resolution on characteristic clause sets obtained from a proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ is also complete. As in the case of term resolution, this is done by showing that, from such a clause set, we can always obtain a sequence of $\vdash_{\mathsf{res}_{\mathsf{i}}}$-steps that eventually leads to the clause set $\{\vdash\}$.

We will then combine these results with Proposition 6.2.25 in order to obtain that we can derive each characteristic clause set of a proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ after reductive elimination of an atomic cut by a sequence of $\vdash_{\mathsf{res}_{\mathsf{i}}}$-steps from the characteristic clause set of the original proof, i.e. the proof before any reductive cut-elimination step had been applied.

The previous results can then be used to show that each atomic cut-elimination step (by reductive methods), on a proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$, corresponds to a $\vdash_{\mathsf{res}_{\mathsf{i}}}$-step on the characteristic clause sets of the corresponding proofs. Moreover, it will follow that the characteristic clause set obtained from a proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ is always refutable by indexed resolution. This section is then concluded by showing that we can always obtain an indexed clause set from the characteristic clause set of the original proof by indexed resolution, which subsumes the characteristic clause set of the proof after reductive elimination of an atomic cut. In other words, **CERES** simulates reductive cut-elimination methods up to the elimination of atomic cuts.

The following results show that proofs in specific normal forms yield characteristic clause terms in different term normal forms.

Lemma 6.3.1. *Let φ be a chain of atomic cuts with atom indexing, i.e. an **LK**-proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ consisting only of atomic cuts. Then $\Theta(\varphi)$ is $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$.*

PROOF. We proceed by induction on the number n of atomic cuts in φ.

BASE CASE: $n = 1$. Then φ is of the form

$$\dfrac{A \vdash A^1 \qquad A^1 \vdash A}{A \vdash A}\ cut(A^1)$$

where A^1 is some indexed atom.

Clearly, $\Theta(\varphi) = \{\vdash A^1\} \oplus \{A^1 \vdash\}$ is in $\mathsf{TACNF}^{\mathsf{ext}}_{\mathsf{ai}}$.

INDUCTION HYPOTHESIS (IH): Assume the claim holds for all chains of atomic cuts φ with atom indexing containing k atomic cuts, for $1 \leq k \leq n$.

INDUCTION STEP: Suppose ρ_1 and ρ_2 are chains of atomic cuts with atom indexing and end-sequents $A \vdash A^{l+1}$ and $A^{l+1} \vdash A$, respectively. Furthermore, suppose ρ_1 and ρ_2 contain l and m atomic cuts, respectively, such that $l + m = n$. Then, by **(IH)**, both $\Theta(\rho_1)$ and $\Theta(\rho_2)$ are in $\mathsf{TACNF}^{\mathsf{ext}}_{\mathsf{ai}}$. Assume w.l.o.g. that we have

$$\Theta(\rho_1) = \{\vdash A^1\} \oplus \{A^1 \vdash A^2\} \oplus \ldots \oplus \{A^{l-1} \vdash A^l\} \oplus \{A^l \vdash\} \text{ and}$$
$$\Theta(\rho_2) = \{\vdash A^{l+2}\} \oplus \{A^{l+2} \vdash A^{l+3}\} \oplus \ldots \oplus \{A^{m-1} \vdash A^m\} \oplus \{A^m \vdash\}.$$

Let us now construct a proof φ from ρ_1 and ρ_2 as follows:

$$\dfrac{\begin{array}{c}(\rho_1)\\ A \vdash A^{l+1}\end{array} \qquad \begin{array}{c}(\rho_2)\\ A^{l+1} \vdash A\end{array}}{A \vdash A}\ cut(A^{l+1})$$

such that A^{l+1} occurs both in ρ_1 and ρ_2 at the appropriate positions. Clearly, φ is a chain of atomic cuts with atom indexing containing $n + 1$ atomic cuts, and, since A^{l+1} is now a cut-ancestor, the characteristic clause terms of ρ_1 and ρ_2 change as follows:

$$\Theta(\rho_1) = \{\vdash A^1\} \oplus \{A^1 \vdash A^2\} \oplus \ldots \oplus \{A^{l-1} \vdash A^l\} \oplus \{A^l \vdash A^{l+1}\}, \text{ and}$$
$$\Theta(\rho_2) = \{A^{l+1} \vdash A^{l+2}\} \oplus \{A^{l+2} \vdash A^{l+3}\} \oplus \ldots \oplus \{A^{m-1} \vdash A^m\} \oplus \{A^m \vdash\}.$$

The characteristic clause term $\Theta(\varphi) = \Theta(\rho_1) \oplus \Theta(\rho_2)$ is clearly in $\mathsf{TACNF}^{\mathsf{ext}}_{\mathsf{ai}}$. □

Remark. By a *chain of atomic cuts*, we mean a situation like the following:

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}(\sigma_1)\\ A \vdash A^3\end{array} \qquad \begin{array}{c}(\sigma_2)\\ A^3 \vdash A^2\end{array}}{A \vdash A^2} \qquad \begin{array}{c}(\sigma_3)\\ A^2 \vdash A^1\end{array}}{A \vdash A^1} \qquad \dfrac{\begin{array}{c}(\rho_1)\\ A^1 \vdash A^4\end{array} \qquad \begin{array}{c}(\rho_2)\\ A^4 \vdash A\end{array}}{A^1 \vdash A}}{A \vdash A}$$

where $\sigma_1, \ldots, \sigma_3$ and ρ_1, ρ_2 are subderivations which consist of atomic cuts on some indexed atoms A^k.

Proposition 6.3.2. *Let ψ be an **LK**-proof in* $\mathsf{ACNF}^{\mathsf{top}}_{\mathsf{ai}}$ *containing k instances of binary inferences. Then $\Theta(\psi)$ is in* $\mathsf{TACNF}^{\mathsf{ext}}_{\mathsf{ai}}$.

PROOF. Let ψ be an arbitrary **LK**-proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$. We proceed by induction on the number n of instances of binary inferences in ψ.

BASE CASE: $n = 0$. Then ψ does not contain cuts, thus $\Theta(\psi) = \{\vdash\}$ is trivially in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$.

INDUCTION HYPOTHESIS (IH): Assume the claim holds for all **LK**-proofs ψ in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ containing k instances of binary inferences, for $1 \leq k \leq n$.

INDUCTION STEP: Suppose ψ is an **LK**-proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ containing $n+1$ instances of binary inferences. Then ψ is of the following form:

$$\cfrac{\cfrac{\begin{matrix}(\rho_1) \\ S_1\end{matrix} \qquad \begin{matrix}(\rho_2) \\ S_2\end{matrix}}{S'}\ \xi}{\cfrac{\text{some unary inferences}}{S}}$$

where ρ_1 and ρ_2 are subderivations of ψ with end-sequents S_1 and S_2, respectively. Furthermore, ξ is the lowermost instance of a binary inference in ψ; this means all inferences below ξ are unary.

Since ψ is in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$, so are ρ_1 and ρ_2. Neither ρ_1 nor ρ_2 contain ξ, thus, the numbers of instances of binary inferences k_1 and k_2 in ρ_1 and ρ_2, respectively, are strictly less than $n+1$. Hence, we can apply the **(IH)** to both ρ_1 and ρ_2, and obtain that both $\Theta(\rho_1)$ and $\Theta(\rho_2)$ are in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$.

We have to distinguish the following cases:

(i) ξ is a binary inference different from the cut-rule. Then ξ does not operate on cut-ancestors, as all cuts appear above ξ in ψ. As a consequence, $\Theta(\psi) = \Theta(\rho_1) \otimes \Theta(\rho_2)$. Hence, by definition of $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$, $\Theta(\psi)$ is in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$.

(ii) ξ is a cut. As all cuts have been shifted to the top, there can only be cuts above ξ that operate on the same atomic formula. Consequently, the subproof of ψ rooted in S' must be a chain of atomic cuts.

Observe that $\Theta(\psi) = \Theta(\rho_1) \oplus \Theta(\rho_2)$, as all inferences below ξ are unary. Finally, Lemma 6.3.1 yields that $\Theta(\psi)$ is in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$, as the subproof of ψ rooted in S' is a chain of atomic cuts.

□

Corollary 6.3.3. *Let ψ be an **LK**-proof in* $\mathbf{ACNF}^{\mathbf{top}}$ *containing k instances of binary inferences. Then $\Theta(\psi)$ is in* $\mathbf{TACNF}^{\mathbf{ext}}$.

PROOF. The proof is essentially that of Proposition 6.3.2; the only difference is that we consider atoms instead of indexed atoms. □

Proposition 6.3.4. *Let ψ be an **LK**-proof in* $\mathbf{ACNF^{top}}$ *(without chains of atomic cuts) containing k instances of binary inferences (different from the atomic cut-rule). Then the number l of atomic cuts in ψ is $l = k + 1$ (provided that ψ contains at least one atomic cut) and $\Theta(\psi)$ is in* **TACNF**.

PROOF. Let ψ be an arbitrary **LK**-proof in $\mathbf{ACNF^{top}}$. We proceed by induction on the number n of instances of binary inferences (different from the atomic cut-rule) in ψ.

BASE CASE: $n = 0$. Then ψ either consists of a single atomic cut only or all inferences below the only atomic cut in ψ are unary. We distinguish cases:

(i) Let ψ consist of a single atomic cut only, i.e. ψ is of the form

$$\dfrac{A \vdash A \qquad A \vdash A}{A \vdash A}\, cut(A)$$

where A is some atomic formula.

Clearly, $l = n + 1 = 0 + 1 = 1$ and $\Theta(\psi) = \{\vdash A\} \oplus \{A \vdash\}$, by definition of characteristic clause terms.

(ii) Let ψ be of the form

$$\dfrac{\dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A \vdash A}\, cut(A)}{\text{some unary inferences}}}{S}$$

where A is some atomic formula and S is the end-sequent of ψ.

Clearly, $l = n + 1 = 0 + 1 = 1$ and, since unary inference rules do not change the characteristic clause term defined for the subderivations above them, we get $\Theta(\psi) = \{\vdash A\} \oplus \{A \vdash\}$, by definition of characteristic clause terms.

INDUCTION HYPOTHESIS (IH): Assume the claim holds for all **LK**-proofs ψ in $\mathbf{ACNF^{top}}$ containing k instances of binary inferences (different from the atomic cut-rule), for $1 \leq k \leq n$.

INDUCTION STEP: Suppose ψ is an **LK**-proof in $\mathbf{ACNF^{top}}$ containing $n + 1$ instances of binary inferences (different from the atomic cut-rule). Then ψ is of the following form:

$$\dfrac{\dfrac{\dfrac{\begin{matrix}(\rho_1)\\ S_1\end{matrix} \qquad \begin{matrix}(\rho_2)\\ S_2\end{matrix}}{S'}\, \xi}{\text{some unary inferences}}}{S}$$

where ρ_1 and ρ_2 are subderivations of ψ with end-sequents S_1 and S_2, respectively. Furthermore, ξ is the lowermost instance of a binary inference in ψ; this means all inferences below ξ are unary.

Since ψ is in $\mathbf{ACNF^{top}}$, so are ρ_1 and ρ_2. Neither ρ_1 nor ρ_2 contain ξ, thus, the numbers of instances of binary inferences k_1 and k_2 in ρ_1 and ρ_2, respectively, are strictly less than $n+1$. Hence, we can apply the **(IH)** to both ρ_1 and ρ_2, and obtain that $l_1 = k_1 + 1$ and $l_2 = k_2 + 1$, where l_1 and l_2 are the numbers of atomic cuts in ρ_1 and ρ_2, respectively. Moreover, the **(IH)** also yields that both $\Theta(\rho_1)$ and $\Theta(\rho_2)$ are in **TACNF**.

Since the number of instances of binary inferences in ψ is $n+1$, we get that $n+1 = k_1 + k_2 + 1$. As all atomic cuts of ψ exclusively occur in ρ_1 and ρ_2, the number l of atomic cuts in ψ is given by

$$l = l_1 + l_2 = (k_1 + 1) + (k_2 + 1) = (k_1 + k_2 + 1) + 1 = (n+1) + 1 = n + 2.$$

Moreover, assume w.l.o.g. that

$$\Theta(\rho_1) = (\{\vdash A_1\} \oplus \{A_1 \vdash\}) \otimes \ldots \otimes (\{\vdash A_{l_1}\} \oplus \{A_{l_1} \vdash\})$$

and

$$\Theta(\rho_2) = (\{\vdash A_{l_1+1}\} \oplus \{A_{l_1+1} \vdash\}) \otimes \ldots \otimes (\{\vdash A_l\} \oplus \{A_l \vdash\}).$$

Due to the fact that ξ occurs below all cuts in ψ, none of the auxiliary formulas of ξ is an ancestor of some formula in Ω. Therefore, by definition of $\Theta(\psi)$, it follows that

$$\begin{array}{rcl}\Theta(\psi) & = & \Theta(\rho_1) \otimes \Theta(\rho_2) \\ & = & (\{\vdash A_1\} \oplus \{A_1 \vdash\}) \otimes \ldots \otimes (\{\vdash A_l\} \oplus \{A_l \vdash\}),\end{array}$$

i.e. $\Theta(\psi)$ is in **TACNF**. □

Corollary 6.3.5. *Let ψ be an **LK**-proof in $\mathbf{ACNF_{ai}^{top}}$ (without chains of atomic cuts) containing k instances of binary inferences (different from the atomic cut-rule). Then the number l of atomic cuts in ψ is $l = k + 1$ (given that ψ contains at least one atomic cut) and $\Theta(\psi)$ is in $\mathbf{TACNF_{ai}}$.*

PROOF. Due to Proposition 6.2.17, the proof is essentially that of Proposition 6.3.4; the only difference is that we consider indexed atoms instead of atoms. □

Remark. If ψ is just $\vdash$, then $\Theta(\psi)$ is trivially in **TACNF** (or $\mathbf{TACNF_{ai}}$), but $l = k+1$ does not hold in this case, as $0 \neq 0 + 1$. For this reason, we require that ψ in Proposition 6.3.4 and Corollary 6.3.5 contains at least one atomic cut.

We now show that if a clause term in **TACNF** (or $\mathbf{TACNF_{ai}}$) can be obtained from another clause term in the same normal form by removing a single subterm of the form $\{\vdash A_i\} \oplus \{A_i \vdash\}$, then the former clause term can be obtained from the latter by term resolution via a single $\vdash_{\mathsf{res_t}}$-step.

Lemma 6.3.6. *Let φ be an **LK**-proof and ψ be the **LK**-proof in* $\mathbf{ACNF^{top}}$ *(without chains of atomic cuts) containing k atomic cuts such that $\varphi >^*_{\mathcal{R}^{\mathsf{top}}} \psi$ and $\Theta(\psi)$ is in* **TACNF**. *Furthermore, let ψ_1 be obtained from ψ by eliminating a single atomic cut by reductive methods, i.e. $\psi >_{\mathcal{R}_{\mathsf{ax}}} \psi_1$. Then $\Theta(\psi) \vdash_{\mathsf{res}_t} \Theta(\psi_1)$.*

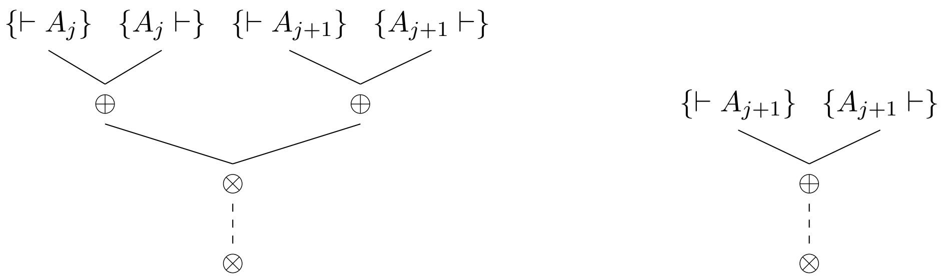
PROOF. Assume that ψ_1 was obtained from ψ by eliminating the atomic cut containing the atom A_j by reductive methods, for some j with $1 \leq j \leq k$. Thus, both ψ and ψ_1 are in $\mathbf{ACNF^{top}}$; this means, by Proposition 6.3.4, that both $\Theta(\psi)$ and $\Theta(\psi_1)$ are in **TACNF**. Therefore we have

$$\Theta(\psi) = \bigotimes_{1 \leq i \leq k} (\{\vdash A_i\} \oplus \{A_i \vdash\}),$$

and

$$\Theta(\psi_1) = \bigotimes_{1 \leq i \leq k, i \neq j} (\{\vdash A_i\} \oplus \{A_i \vdash\}).$$

W.l.o.g. assume that the term trees of $\Theta(\psi)$ and $\Theta(\psi_1)$ are the following:



(a) Term tree of $\Theta(\psi)$

(b) Term tree of $\Theta(\psi_1)$

Application of term resolution on $\Theta(\psi)$:

$$\frac{\Theta(\psi)}{\underbrace{\Theta(\psi)[\{\vdash A_j\}[\{\vdash\}]_0 \otimes \{A_j \vdash\}[\{\vdash\}]_0]_\lambda}_{\Theta(\psi)'}}\ R_t$$

where

$$\begin{aligned} \Theta(\psi).\lambda &= \{\vdash A_j\} \oplus \{A_j \vdash\}, \\ \{\vdash A_j\}.0 &= \{\vdash A_j\} \text{ and} \\ \{A_j \vdash\}.0 &= \{A_j \vdash\}. \end{aligned}$$

Furthermore,

$$\Theta(\psi)' = (\{\vdash A_1\} \oplus \{A_1 \vdash\}) \otimes \ldots \otimes (\{\vdash\} \otimes \{\vdash\}) \otimes \ldots \otimes (\{\vdash A_k\} \oplus \{A_k \vdash\}).$$

See Figure 6.2 for the term tree of $\Theta(\psi)'$.

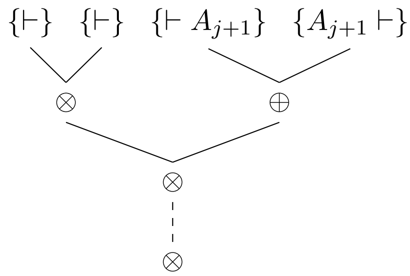


Figure 6.2: Term tree of $\Theta(\psi)'$

Application of term factoring on $\Theta(\psi)'$:

$$\frac{\Theta(\psi)'}{\underbrace{\Theta(\psi)'[\Theta(\psi)'.\lambda_2]_\lambda}_{\Theta(\psi)''}}\, ft_\otimes$$

where

$$\Theta(\psi)'.\lambda = \{\vdash\} \otimes \{\vdash\} \text{ and}$$
$$\Theta(\psi)'.\lambda_2 = \{\vdash\}.$$

Term factoring yields the term

$$\Theta(\psi)'' = (\{\vdash A_1\} \oplus \{A_1 \vdash\}) \otimes \ldots \otimes \{\vdash\} \otimes \ldots \otimes (\{\vdash A_k\} \oplus \{A_k \vdash\}).$$

Term tree of $\Theta(\psi)''$:

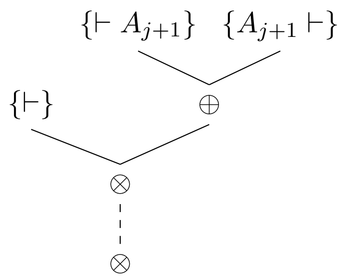


Figure 6.3: Term tree of $\Theta(\psi)''$

Application of term factoring on $\Theta(\psi)''$:

$$\frac{\Theta(\psi)''}{\underbrace{\Theta(\psi)''[\Theta(\psi)''.\lambda'']_{\lambda'}}_{\Theta(\psi)'''}}\, ft_\otimes$$

where

$$\Theta(\psi)''.\lambda' = \{\vdash\} \otimes (\{\vdash A_{j+1}\} \oplus \{A_{j+1} \vdash\}) \text{ and}$$
$$\Theta(\psi)''.\lambda'' = (\{\vdash A_{j+1}\} \oplus \{A_{j+1} \vdash\}).$$

Thus, we finally obtain that

$\Theta(\psi)''' = (\{\vdash A_1\} \oplus \{A_1 \vdash\}) \otimes \ldots \otimes (\{\vdash A_{j-1}\} \oplus \{A_{j-1} \vdash\}) \otimes (\{\vdash A_{j+1}\} \oplus \{A_{j+1} \vdash \}) \otimes \ldots \otimes (\{\vdash A_k\} \oplus \{A_k \vdash\})$.

Term tree of $\Theta(\psi)'''$:

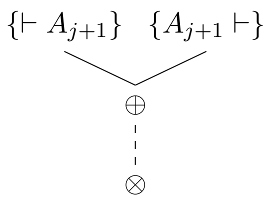


Figure 6.4: Term tree of $\Theta(\psi)'''$.

Since the term tree of $\Theta(\psi)'''$ coincides with the term tree of $\Theta(\psi_1)$, we obtain that $\Theta(\psi)''' = \Theta(\psi_1)$ and thus, $\Theta(\psi) \vdash_{\mathsf{res}_\mathsf{t}} \Theta(\psi_1)$. □

The following corollary states that Lemma 6.3.6 also holds for **LK**-proofs with atom indexing.

Corollary 6.3.7. *Let φ be an **LK**-proof with atom indexing and ψ be the **LK**-proof in $\mathsf{ACNF}^{\mathsf{top}}_{\mathsf{ai}}$ (without chains of atomic cuts) containing k atomic cuts such that $\varphi >^*_{\mathcal{R}^{\mathsf{top}}} \psi$. Furthermore, let ψ_1 be obtained from ψ by eliminating a single atomic cut by reductive methods, i.e. $\psi >_{\mathcal{R}_{\mathsf{ax}}} \psi_1$. Then $\Theta(\psi) \vdash_{\mathsf{res}_\mathsf{t}} \Theta(\psi_1)$.*

PROOF. Due to Proposition 6.2.17, the proof is essentially that of Lemma 6.3.6; the only difference is that we consider indexed atoms instead of atoms. □

In the following proposition, we will show that we can always reduce a characteristic clause term of a proof in $\mathsf{ACNF}^{\mathsf{top}}_{\mathsf{ai}}$ (containing chains of atomic cuts) to the characteristic clause term of a proof after elimination of these chains of atomic cuts using the relation $\vdash_{\mathsf{res}_{\mathsf{te}}}$.

Proposition 6.3.8. *Let φ be an **LK**-proof and ψ be an **LK**-proof in $\mathsf{ACNF}^{\mathsf{top}}_{\mathsf{ai}}$ such that $\varphi >^*_{\mathcal{R}^{\mathsf{top}}} \psi$. Then*

$$\Theta(\psi) \vdash_{\mathsf{res}_{\mathsf{te}}} \Theta(\psi^1) \vdash_{\mathsf{res}_{\mathsf{te}}} \ldots \vdash_{\mathsf{res}_{\mathsf{te}}} \Theta(\psi^{k-1}) \vdash_{\mathsf{res}_{\mathsf{te}}} \Theta(\psi^k),$$

*where $\Theta(\psi^k)$ is in $\mathsf{TACNF}_{\mathsf{ai}}$, but $\Theta(\psi^{k-1})$ is not and ψ^i is the **LK**-proof in $\mathsf{ACNF}^{\mathsf{top}}_{\mathsf{ai}}$ obtained from ψ after eliminating i cuts by reductive methods (i.e. $\psi >^i_{\mathcal{R}_{\mathsf{ax}}} \psi^i$).*

PROOF. Let φ be an **LK**-proof and ψ be an **LK**-proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ such that $\varphi >^*_{\mathcal{R}^{\mathsf{top}}} \psi$. By Proposition 6.3.2, $\Theta(\psi)$ is in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$. Moreover, let ψ^k be an $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ with $\psi >^k_{\mathcal{R}_{\mathsf{ax}}} \psi^k$ such that $\Theta(\psi^{k-1})$ is not in $\mathbf{TACNF}_{\mathbf{ai}}$, but $\Theta(\psi^k)$ is. By definition of $\vdash_{\mathsf{res_{te}}}$ and Proposition 6.1.14, we get $\Theta(\psi) \vdash_{\mathsf{res_{te}}} \Theta(\psi^1) \vdash_{\mathsf{res_{te}}} \ldots \vdash_{\mathsf{res_{te}}} \Theta(\psi^k)$, since each atomic cut-elimination step corresponds to replacing a subterm of the form $\{A^l \vdash A^i\} \oplus \{A^i \vdash A^j\}$ by $\{A^l \vdash A^j\}$, where A^i is the atomic cut-formula of the eliminated cut and the indexed atoms A^j or A^l might not occur in the above subterms. □

The following theorem shows that we can always perform a stepwise refutation of a characteristic clause term extracted from a proof in $\mathbf{ACNF}^{\mathbf{top}}$ (without chains of atomic cuts) using term resolution via the relation $\vdash_{\mathsf{res_t}}$.

Theorem 6.3.9 (Completeness of Term Resolution Deduction w.r.t. TACNF). *Let φ be an **LK**-proof and ψ be the **LK**-proof in $\mathbf{ACNF}^{\mathbf{top}}$ (without chains of atomic cuts) containing k atomic cuts such that $\varphi >^*_{\mathcal{R}^{\mathsf{top}}} \psi$. Then*

$$\Theta(\psi) \vdash_{\mathsf{res_t}} \Theta(\psi^1) \vdash_{\mathsf{res_t}} \ldots \vdash_{\mathsf{res_t}} \Theta(\psi^k) = \{\vdash\},$$

*where ψ^i is the **LK**-proof obtained from ψ after eliminating i cuts by reductive methods, i.e. $\psi >^i_{\mathcal{R}_{\mathsf{ax}}} \psi^i$.*

PROOF. Let φ be an **LK**-proof and ψ be the **LK**-proof in $\mathbf{ACNF}^{\mathbf{top}}$ containing k atomic cuts obtained from φ by reductive methods.

We proceed by induction on the number n of subterms of the form $\{\vdash A_i\} \oplus \{A_i \vdash\}$ occurring in $\Theta(\psi)$, for $1 \leq i \leq n$.

BASE CASE: $n = 0$. Then we have $\psi = \varphi$, i.e. φ (and thus ψ) is cut-free. By definition of characteristic clause terms, it follows that $\Theta(\psi) = \{\vdash\}$. Hence, $\Theta(\psi) \vdash_{\mathsf{res_t}} \{\vdash\}$.

INDUCTION HYPOTHESIS (IH): Assume the claim holds for all $\Theta(\psi)$ in **TACNF** containing k subterms of the form $\{\vdash A_i\} \oplus \{A_i \vdash\}$ for $1 \leq i \leq k \leq n$.

INDUCTION STEP: Suppose ψ is an **LK**-proof in $\mathbf{ACNF}^{\mathbf{top}}$ without chains of atomic cuts containing $n+1$ atomic cuts. Then, by Proposition 6.3.4, $\Theta(\psi)$ is a characteristic clause term in **TACNF** containing $n+1$ subterms of the form $\{\vdash A_i\} \oplus \{A_i \vdash\}$, for $1 \leq i \leq n+1$.

Furthermore, suppose ψ^1 is the **LK**-proof obtained from ψ by eliminating the atomic cut containing some atom A_j, for $1 \leq j \leq n+1$. Since ψ is in $\mathbf{ACNF}^{\mathbf{top}}$, so is ψ^1, and thus, again by Proposition 6.3.4, $\Theta(\psi^1)$ is in **TACNF**. We denote $\Theta(\psi^1)$ as C^1 in the following.

Since both $\Theta(\psi)$ and C^1 are in **TACNF**, we can apply Lemma 6.3.6. Consequently, we have that $\Theta(\psi) \vdash_{\mathsf{res_t}} C^1$. Moreover, since ψ^1 contains fewer atomic cuts than ψ, it follows that $C^1 = \Theta(\psi^1)$ contains fewer subterms of the form $\{\vdash A_i\} \oplus \{A_i \vdash\}$ than $\Theta(\psi)$. Therefore, we can apply the **(IH)** to C^1, and obtain that $C^1 \vdash_{\mathsf{res_t}} \ldots \vdash_{\mathsf{res_t}} \Theta(\psi^{n+1}) = \{\vdash\}$.

Putting things together, we get $\Theta(\psi) \vdash_{\mathsf{res_t}} \Theta(\psi^1) \vdash_{\mathsf{res_t}} \ldots \vdash_{\mathsf{res_t}} \Theta(\psi^{n+1}) = \{\vdash\}$. □

Theorem 6.3.9 also holds for **LK**-proofs with atom indexing:

Corollary 6.3.10. *Let φ be an **LK**-proof with atom indexing and ψ be the **LK**-proof in* $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ *(without chains of atomic cuts) containing k atomic cuts such that $\varphi >^{*}_{\mathcal{R}^{\mathrm{top}}} \psi$. Then*

$$\Theta(\psi) \vdash_{\mathsf{res_t}} \Theta(\psi^1) \vdash_{\mathsf{res_t}} \ldots \vdash_{\mathsf{res_t}} \Theta(\psi^k) = \{\vdash\},$$

*where ψ^i is the **LK**-proof obtained from ψ after eliminating i cuts by reductive methods, i.e. $\psi >^{i}_{\mathcal{R}_{\mathrm{ax}}} \psi^i$.*

PROOF. Due to Proposition 6.2.17, the proof is essentially that of Theorem 6.3.9; the only difference is that we consider indexed atoms instead of atoms. □

Theorem 6.3.11 tells us that we can also perform a stepwise refutation of a characteristic clause term extracted from an arbitrary **LK**-proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ using term resolution via the relation $\vdash_{\mathsf{res_{ta}}}$.

Theorem 6.3.11 (Completeness of Term Resolution Deduction w.r.t. $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$). *Let φ be an **LK**-proof with atom indexing and ψ be the **LK**-proof in* $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ *containing k atomic cuts such that $\varphi >^{*}_{\mathcal{R}^{\mathrm{top}}} \psi$. Then*

$$\Theta(\psi) \vdash_{\mathsf{res_{ta}}} \Theta(\psi^1) \vdash_{\mathsf{res_{ta}}} \ldots \vdash_{\mathsf{res_{ta}}} \Theta(\psi^k) = \{\vdash\},$$

*where ψ^i is the **LK**-proof obtained from ψ after eliminating i cuts by reductive methods, i.e. $\psi >^{i}_{\mathcal{R}_{\mathrm{ax}}} \psi^i$.*

PROOF. By Proposition 6.3.8, Corollary 6.3.10 and the definition of $\vdash_{\mathsf{res_{ta}}}$. □

The following lemma shows that we can always perform a stepwise refutation of a characteristic clause set extracted from a proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ (without chains of atomic cuts) using indexed resolution via the relation $\vdash_{\mathsf{res_i}}$.

Lemma 6.3.12. *Let φ be an **LK**-proof with atom indexing and ψ be the **LK**-proof in* $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ *(without chains of atomic cuts) containing k atomic cuts such that $\varphi >^{*}_{\mathcal{R}^{\mathrm{top}}} \psi$. Furthermore, let ψ_1 be obtained from ψ by eliminating a single atomic cut by reductive methods, i.e. $\psi >_{\mathcal{R}_{\mathrm{ax}}} \psi_1$. Then there exists an indexed resolution refutation of $\mathrm{CL}(\psi)$ with*

$$\mathrm{CL}(\psi) \vdash_{\mathsf{res_i}} \mathrm{CL}(\psi_1) \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} \{\vdash\}.$$

PROOF. Let φ be an **LK**-proof with atom indexing, and let ψ be the **LK**-proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ containing k atomic cuts obtained from φ by reductive methods. Furthermore, let ψ_1 be obtained from ψ by eliminating a single atomic cut by reductive methods. Then, by Corollary 6.3.5, both $\Theta(\psi)$ and $\Theta(\psi_1)$ are in $\mathbf{TACNF}_{\mathbf{ai}}$. In the following we let $C^1 = \Theta(\psi_1)$. We proceed by induction on the number n of indexed clauses in $\mathrm{CL}(\psi)$.

BASE CASE: $n = 1$. Then we have $\mathrm{CL}(\psi) = \{\vdash\}$[4], and thus trivially $\mathrm{CL}(\psi) \vdash_{\mathsf{res_i}} \{\vdash\}$.

[4]Since $\Theta(\psi)$ is in **TACNF**, the only possibility in which $\mathrm{CL}(\psi)$ contains one indexed clause, is that it only contains $\vdash$.

INDUCTION HYPOTHESIS (IH): Assume the claim holds for all $\mathrm{CL}(\psi)$ containing k indexed clauses with $1 \leq k \leq n$, where $\Theta(\psi)$ is in $\mathbf{TACNF_{ai}}$.

INDUCTION STEP: Suppose $\mathrm{CL}(\psi)$ is a clause set containing $n+1$ indexed clauses, where $\Theta(\psi)$ is in $\mathbf{TACNF_{ai}}$. Furthermore, suppose ψ' was obtained from ψ by eliminating the atomic cut with cut-formula A_j^j, for $1 \leq j \leq n+1$. Then we have that $\Theta(\psi) \vdash_{\mathsf{res_t}} C^1$, by Corollary 6.3.7, where the subterm $\{\vdash A_j^j\} \oplus \{A_j^j \vdash\}$ was resolved in order to obtain C^1 from $\Theta(\psi)$. By Definitions 6.1.3 and 6.1.7, it follows that

$$\Theta(\psi) = \bigotimes_{1 \leq i \leq n+1} (\{\vdash A_i^i\} \oplus \{A_i^i \vdash\}),$$

and

$$C^1 = \bigotimes_{1 \leq i \leq n+1, i \neq j} (\{\vdash A_i^i\} \oplus \{A_i^i \vdash\}).$$

Since C^1 is in $\mathbf{TACNF_{ai}}$ and $\otimes$ is commutative for (indexed) clauses defined via multisets, there is a characteristic clause term $\Theta(\psi)' = C^1 \otimes (\{\vdash A_j^j\} \oplus \{A_j^j \vdash\})$ s.t. $\Theta(\psi) \sim \Theta(\psi)'$, i.e. $\mathrm{CL}(\psi) = |\Theta(\psi)'|$.

Therefore, we have that $|\Theta(\psi)'| = |C^1| \times \{\vdash A_j^j; A_j^j \vdash\}$. As a consequence, $|C^1|$ contains $k \leq n$ indexed clauses; thus, we can apply the **(IH)** to $|C^1|$ and obtain an indexed resolution refutation δ of $|C^1|$ of the following form (w.l.o.g.):

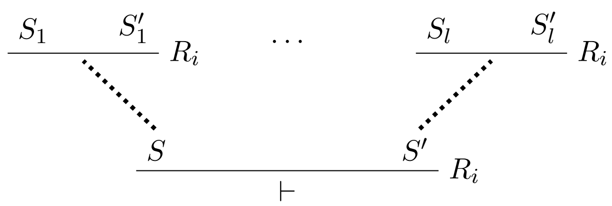


Figure 6.5: Indexed resolution refutation δ of $|C^1|$.

where $S_1, S_1', \ldots, S_l, S_l'$ are all the indexed clauses in $|C^1|$, and $2 * l = k$, i.e. if $|C^1|$ contains more than one element, then there is always a pair (S_m, S_m') with $1 \leq m \leq l$ that can be resolved using indexed resolution. This means each leaf of δ corresponds to an indexed clause in $|C^1|$.

Moreover, since $\Theta(\psi), C^1$ are in $\mathbf{TACNF_{ai}}$, by semantics of clause terms, it follows that $\mathrm{CL}(\psi)$ contains all possible indexed clauses of the form $\Gamma \vdash \Delta$ with $\Gamma, \Delta \subseteq \{A_1^1, \ldots, A_{n+1}^{n+1}\}$ such that $\Gamma \cap \Delta = \emptyset$, for $1 \leq j \leq n+1$.

Hence, $|C^1|$ contains all possible indexed clauses of the form $\Gamma \setminus \{A_j^j\} \vdash \Delta \setminus \{A_j^j\}$, where Γ, Δ are defined as for the indexed clauses in $\mathrm{CL}(\psi)$. Consequently, for each indexed clause $S \in |C^1|$, there are two indexed clauses $S \circ \vdash A_j^j$ and $A_j^j \vdash \circ\, S$ in $\mathrm{CL}(\psi)$. Thus, we can construct an indexed resolution refutation γ of $\mathrm{CL}(\psi)$ by extending δ as shown in Figure 6.6.

$$
\dfrac{\dfrac{\dfrac{S_1 \circ\vdash A_j^j \qquad A_j^j \vdash\circ S_1}{S_1 \circ S_1}\,R_i}{S_1}\,c_r^*, c_l^* \qquad \dfrac{\dfrac{S'_1 \circ\vdash A_j^j \qquad A_j^j \vdash\circ S'_1}{S'_1 \circ S'_1}\,R_i}{S'_1}\,c_r^*, c_l^*}{}\,R_i
\qquad \cdots \qquad
\dfrac{\dfrac{\dfrac{S_l \circ\vdash A_j^j \qquad A_j^j \vdash\circ S_l}{S_l \circ S_l}\,R_i}{S_l}\,c_r^*, c_l^* \qquad \dfrac{\dfrac{S'_l \circ\vdash A_j^j \qquad A_j^j \vdash\circ S'_l}{S'_l \circ S'_l}\,R_i}{S'_l}\,c_r^*, c_l^*}{}\,R_i
$$

$$
\dfrac{S \qquad\qquad S'}{\vdash}\,R_i
$$

Figure 6.6: Indexed resolution refutation γ of $\mathrm{CL}(\psi)$.

Furthermore, as shown in Figure 6.6, the leaves of γ correspond to the elements of $\mathrm{CL}(\psi)$. Therefore, there exists a resolution deduction of each sequent $S_1, S_1', \ldots, S_l, S_l'$ of $|C^1|$ from $\mathrm{CL}(\psi)$ that resolves solely upon A_j^j. Hence, we also have $\mathrm{CL}(\psi) \vdash_{\mathsf{res_i}} |C^1|$.

Putting things together, we have $\mathrm{CL}(\psi) \vdash_{\mathsf{res_i}} \mathrm{CL}(\psi_1) \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} \{\vdash\}$.

□

Proposition 6.3.13. *Let φ be an **LK**-proof with atom indexing and ψ be an **LK**-proof in $\mathbf{ACNF}_{\mathbf{ai}}^{\mathbf{top}}$ such that $\varphi >^*_{\mathcal{R}^{\mathsf{top}}} \psi$. Then*

$$\mathrm{CL}(\psi) \vdash_{\mathsf{res_i}} \mathrm{CL}(\psi^1) \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} \mathrm{CL}(\psi^{k-1}) \vdash_{\mathsf{res_i}} \mathrm{CL}(\psi^k),$$

*where $\Theta(\psi^k)$ is in $\mathbf{TACNF}_{\mathbf{ai}}$, but $\Theta(\psi^{k-1})$ is not and ψ^i is the **LK**-proof in $\mathbf{ACNF}_{\mathbf{ai}}^{\mathbf{top}}$ obtained from ψ after eliminating i cuts by reductive methods (i.e. $\psi >^i_{\mathcal{R}_{\mathsf{ax}}} \psi^i$).*

PROOF. Let φ be an **LK**-proof and ψ be an **LK**-proof in $\mathbf{ACNF}_{\mathbf{ai}}^{\mathbf{top}}$ such that $\varphi >^*_{\mathcal{R}^{\mathsf{top}}} \psi$. By Proposition 6.3.2, $\Theta(\psi)$ is in $\mathbf{TACNF}_{\mathbf{ai}}^{\mathbf{ext}}$. Moreover, let ψ^k be an $\mathbf{ACNF}_{\mathbf{ai}}^{\mathbf{top}}$ with $\psi >^k_{\mathcal{R}_{\mathsf{ax}}} \psi^k$ such that $\Theta(\psi^{k-1})$ is not in $\mathbf{TACNF}_{\mathbf{ai}}$, but $\Theta(\psi^k)$ is. By definition of $\vdash_{\mathsf{res_i}}$ and Proposition 6.2.16, we get $\mathrm{CL}(\psi) \vdash_{\mathsf{res_i}} \mathrm{CL}(\psi^1) \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} \mathrm{CL}(\psi^k)$, since each atomic cut-elimination step corresponds to replacing a subterm of the form $\{A^l \vdash A^i\} \oplus \{A^i \vdash A^j\}$ by $\{A^l \vdash A^j\}$, where A^i is the atomic cut-formula of the eliminated cut and the indexed atoms A^j or A^l might not occur in the above subterms. □

Theorem 6.3.14 tells us that we can also perform a stepwise refutation of a characteristic clause set extracted from an arbitrary **LK**-proof in $\mathbf{ACNF}_{\mathbf{ai}}^{\mathbf{top}}$ using indexed resolution via the relation $\vdash_{\mathsf{res_i}}$.

Theorem 6.3.14. *Let φ be an **LK**-proof with atom indexing and ψ be the **LK**-proof in $\mathbf{ACNF}_{\mathbf{ai}}^{\mathbf{top}}$ containing k atomic cuts such that $\varphi >^*_{\mathcal{R}^{\mathsf{top}}} \psi$. Then*

$$\mathrm{CL}(\psi) \vdash_{\mathsf{res_i}} \mathrm{CL}(\psi^1) \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} \mathrm{CL}(\psi^k) = \{\vdash\},$$

*where ψ^i is the **LK**-proof in $\mathbf{ACNF}_{\mathbf{ai}}^{\mathbf{top}}$ obtained from ψ after eliminating i cuts by reductive methods (i.e. $\psi >^i_{\mathcal{R}_{\mathsf{ax}}} \psi^i$).*

PROOF. By Proposition 6.3.13 and Lemma 6.3.12. □

Theorem 6.3.15. *Let φ be an **LK**-proof with atom indexing, and let ψ be the **LK**-proof in $\mathbf{ACNF}_{\mathbf{ai}}^{\mathbf{top}}$ containing k atomic cuts such that $\varphi >^*_{\mathcal{R}^{\mathsf{top}}} \psi$. Then*

$$\mathrm{CL}(\psi) \vdash_{\mathsf{res_i}} \mathrm{CL}(\psi^1) \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} \mathrm{CL}(\psi^k) = \{\vdash\} \text{ and}$$
$$\mathrm{CL}(\varphi) \vdash_{\mathsf{res_i}} \mathcal{D}^1 \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} \mathcal{D}^k = \{\vdash\}$$

*such that $\mathcal{D}^i \leq_{ss} \mathrm{CL}(\psi^i)$, for $1 \leq i \leq k$, where ψ^i is the **LK**-proof obtained from ψ after eliminating i cuts by reductive methods (i.e. $\psi >^i_{\mathcal{R}_{\mathsf{ax}}} \psi^i$), and $\mathcal{D}^i$ is a set of clauses obtained from $\mathcal{D}^{i-1}$ by indexed resolution (i.e. $\mathcal{D}^{i-1} \vdash_{\mathsf{res_i}} \mathcal{D}^i$), where $\mathcal{D}^0 = \mathrm{CL}(\varphi)$.*

PROOF. Let φ be an **LK**-proof with atom indexing, and let ψ be the **LK**-proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ containing k atomic cuts obtained from φ by reductive methods.

Since ψ is in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$, $\Theta(\psi)$ is in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$, by Proposition 6.3.2. Moreover, by Theorem 6.3.11, it follows that

$$\Theta(\psi) \vdash_{\mathsf{res_{ta}}} \Theta(\psi^1) \vdash_{\mathsf{res_{ta}}} \ldots \vdash_{\mathsf{res_{ta}}} \Theta(\psi^k) = \{\vdash\},$$

where ψ^i is the **LK**-proof obtained from ψ after eliminating i atomic cuts by reductive methods (i.e. $\psi >^i_{\mathcal{R}_{\mathsf{ax}}} \psi^i$).

Furthermore, Theorem 6.3.14 yields

$$\mathrm{CL}(\psi) \vdash_{\mathsf{res_i}} \mathrm{CL}(\psi^1) \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} \mathrm{CL}(\psi^k) = \{\vdash\}.$$

Since $\varphi >^*_{\mathcal{R}^{\mathsf{top}}} \psi$ and ψ is in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$, we can apply Theorem 6.2.21 and obtain that $\Theta(\varphi) \leq_{ss} \Theta(\psi)$, i.e. $\mathrm{CL}(\varphi) \leq_{ss} \mathrm{CL}(\psi)$ by definition of $\leq_{ss}$ and CL.
Moreover, since $\mathrm{CL}(\varphi) \leq_{ss} \mathrm{CL}(\psi)$ and $\mathrm{CL}(\psi) \vdash_{\mathsf{res_i}} \mathrm{CL}(\psi^1) \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} \mathrm{CL}(\psi^k) = \{\vdash\}$, there are indexed resolution deductions δ^i_j from $\mathrm{CL}(\psi^{i-1})$ for each clause $S^i_j \in \mathrm{CL}(\psi^i)$, where $\psi^0 = \psi$ for $1 \leq i \leq k$ and $1 \leq j \leq size(\mathrm{CL}(\psi^i))$(here $size(\mathrm{CL}(\psi^i))$ denotes the number of elements in $\mathrm{CL}(\psi^i)$).

Thus, by repeatedly applying Proposition 6.2.25, there are indexed resolution deductions γ^i_j from $\mathcal{D}^{i-1}$ s.t. $\gamma^i_j \leq_{ss} \delta^i_j$, for $1 \leq i \leq k$ and $1 \leq j \leq size(\mathrm{CL}(\psi^i))$, where $\mathcal{D}^0 = \mathrm{CL}(\varphi)$ and $\mathcal{D}^i =$

$$\{S'^i_j \mid \gamma^i_j \text{ res. ded. of } S'^i_j \text{ from } \mathcal{D}^{i-1},\ \delta^i_j \text{ res. ded. of } S^i_j \text{ from } \mathrm{CL}(\psi^{i-1}) \text{ s.t. } \gamma^i_j \leq_{ss} \delta^i_j\}.$$

As a consequence, $\mathcal{D}^i \leq_{ss} \mathrm{CL}(\psi^i)$ for $1 \leq i \leq k$. □

We will now show that indexed resolution is complete w.r.t. to characteristic clause sets obtained from an **LK**-proof φ with atom indexing, i.e. that there always exists an indexed resolution refutation of an unsatisfiable characteristic clause set $\mathrm{CL}(\varphi)$.

Theorem 6.3.16. *Let φ be an **LK**-proof with atom indexing and $\mathrm{CL}(\varphi)$ the indexed characteristic clause set of φ. Then there exists an indexed resolution refutation of $\mathrm{CL}(\varphi)$.*

PROOF. Let φ be an **LK**-proof with atom indexing and $\mathrm{CL}(\varphi)$ the indexed characteristic clause set of φ. Furthermore, let ψ be an $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ of φ containing k atomic cuts such that $\varphi >^*_{\mathcal{R}^{\mathsf{top}}} \psi$. By Theorem 6.3.15, we obtain

$$\mathrm{CL}(\varphi) \vdash_{\mathsf{res_i}} \mathcal{D}^1 \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} \mathcal{D}^k = \{\vdash\} \qquad (*)$$

such that $\mathcal{D}^i$ is a set of indexed clauses with $\mathcal{D}^{i-1} \vdash_{\mathsf{res}_{i-1}} \mathcal{D}^i$, where $\mathcal{D}^0 = \mathrm{CL}(\varphi)$.

By definition of $\vdash_{\mathsf{res_i}}$, each indexed clause $S^i_j \in \mathcal{D}^i \setminus \mathcal{D}^{i-1}$ can only be obtained from clauses $S^i_l \circ \vdash A^{i-1}_{i-1}$ and $A^{i-1}_{i-1} \vdash \circ S^i_m$ in $\mathcal{D}^{i-1}$, where the contraction normalization of $S^i_l \circ S^i_m$ equals S^i_j and A^{i-1}_{i-1} is the atom resolved upon in $\mathcal{D}^{i-1} \vdash_{\mathsf{res}_{i-1}} \mathcal{D}^i$. If on the other hand, $S^i_j \in \mathcal{D}^i \cap \mathcal{D}^{i-1}$, then we trivially have that $\mathcal{D}^{i-1} \vdash_{\mathsf{res}_{i-1}} S^i_j$. By combining all applications of R_i and contractions (in the correct form: $\mathcal{D}^k$ is the root and all clauses in $\mathrm{CL}(\varphi)$ are axioms) used in the sequence (*), we obtain an indexed resolution retutation of $\mathrm{CL}(\varphi)$. □

Theorem 6.3.17 shows that each atomic cut-elimination step (on a proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$) corresponds to a $\vdash_{\mathsf{res_i}}$-step on the corresponding characteristic clause terms. In other words, indexed resolution on characteristic clause sets (obtained from characteristic clause terms in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$) derives the characteristic clause set after the elimination of atomic cuts by reductive methods.

Theorem 6.3.17. *Let φ and φ' be **LK**-proofs in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ with $\varphi >_{\mathcal{R}_{\mathsf{ax}}} \varphi'$. Then $\mathrm{CL}(\varphi) \vdash_{\mathsf{res_i}} \mathrm{CL}(\varphi')$.*

PROOF. Immediate consequence of Theorem 6.3.14. □

Finally, Theorem 6.3.18 shows that we can always obtain a clause set $\mathcal{D}$ from the characteristic clause set of an arbitrary skolemized **LK**-proof φ by indexed resolution such that $\mathcal{D}$ subsumes the characteristic clause set of an $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ of φ after each atomic cut-elimination step. In other words, indexed resolution on the original characteristic clause set of φ can simulate the elimination of atomic cuts by reductive methods.

Theorem 6.3.18. *Let φ be an arbitrary skolemized **LK**-proof and ψ be an $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ of φ. If $\psi >_{\mathcal{R}_{\mathsf{ax}}} \psi'$, then there exists a clause set $\mathcal{D}$ such that $\mathrm{CL}(\varphi) \vdash_{\mathsf{res_i}} \mathcal{D}$ and $\mathcal{D} \leq_{ss} \mathrm{CL}(\psi')$.*

PROOF. Let φ be an arbitrary skolemized **LK**-proof and ψ be an $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ of φ (in fact, an $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ of the result of applying atom indexing to φ), i.e. $\varphi >^*_{\mathcal{R}^{\mathsf{top}}} \psi$. W.l.o.g. assume that ψ contains k atomic cuts. Then, by Theorem 6.3.15, we get

$$\begin{aligned}&\mathrm{CL}(\psi) \vdash_{\mathsf{res_i}} \mathrm{CL}(\psi^1) \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} \mathrm{CL}(\psi^k) = \{\vdash\} \text{ and}\\ &\mathrm{CL}(\varphi) \vdash_{\mathsf{res_i}} \mathcal{D}^1 \vdash_{\mathsf{res_i}} \ldots \vdash_{\mathsf{res_i}} \mathcal{D}^k = \{\vdash\}\end{aligned}$$

such that $\mathcal{D}^i \leq_{ss} \mathrm{CL}(\psi^i)$, for $1 \leq i \leq k$, where ψ^i is the **LK**-proof obtained from ψ after eliminating i cuts by reductive methods (i.e. $\psi >^i_{\mathcal{R}_{\mathsf{ax}}} \psi^i$), and $\mathcal{D}^i$ is a set of clauses obtained from $\mathcal{D}^{i-1}$ by indexed resolution (i.e. $\mathcal{D}^{i-1} \vdash_{\mathsf{res_i}} \mathcal{D}^i$), where $\mathcal{D}^0 = \mathrm{CL}(\varphi)$.

Thus, in particular, $\mathrm{CL}(\varphi) \vdash_{\mathsf{res_i}} \mathcal{D}^1$ and $\mathcal{D}^1 \leq_{ss} \mathrm{CL}(\psi^1)$. But since we have $\psi >_{\mathcal{R}_{\mathsf{ax}}} \psi^1$, setting $\mathcal{D} = \mathcal{D}^1$ and $\psi' = \psi^1$ yields

$$\mathrm{CL}(\varphi) \vdash_{\mathsf{res_i}} \mathcal{D} \text{ and } \mathcal{D} \leq_{ss} \mathrm{CL}(\psi').$$

□

Remark. Note that Theorem 6.3.18 actually tells us that indexed resolution on the original characteristic clause set can only simulate the elimination of atomic cuts by reductive methods if a specific order in which the atomic cuts are eliminated is maintained. This follows from the fact that we first reduce a clause term in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$ to a clause term in $\mathbf{TACNF}_{\mathbf{ai}}$. The general simulation should, however, be obtainable by extending the proofs of Proposition 6.1.14 and Proposition 6.2.16 in such a way that the $\vdash_{\mathsf{res_{te}}}$- and $\vdash_{\mathsf{res_i}}$-steps, respectively, eventually arrive at $\{\vdash\}$.

CHAPTER 7

Conclusion

This thesis was set out to answer the question whether the method **CERES** is able to simulate *reductive cut-elimination methods* a là Gentzen. Building on a previously established result by Baaz and Leitsch [9] (i.e. that **CERES** simulates reductive methods up to the elimination of *non-atomic cuts*), the goal was, more precisely, to show that the characteristic clause set after reductive elimination of *atomic cuts* can be obtained by a restricted form of resolution, namely *indexed resolution*, from the characteristic clause set of the *original* proof (i.e. before any form of cut-elimination). Since this approach relies on the *unsatisfiability* of indexed characteristic clause sets, answering the above question also involves proving the *completeness of indexed resolution* w.r.t. ordinary characteristic clause sets.

In Chapter 6, we presented and established the vast majority of methods and results needed in order to answer the above question positively. The method of *term resolution*, introduced in Section 6.1, operates on the syntax of clause terms and its intended use is the elimination of atoms A from a clause term of the form $X_1 \oplus X_2$ such that A has complementary positions in X_1 and X_2. As a first result towards proving the completeness of term resolution w.r.t. clause terms in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$, we have shown that we can always obtain a clause term in $\mathbf{TACNF}_{\mathbf{ai}}$ from a clause term in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$ using term resolution via the $\vdash_{\mathsf{res}_{\mathsf{te}}}$-relation.

Our formulation of *indexed resolution* w.r.t. ordinary characteristic clause sets was defined in Section 6.2 (and has similarities to the complete resolution refinement of *atomic cut-linkage* as introduced by Bruno Woltzenlogel Paleo for a different notion of characteristic clause sets called *swapped clause sets* [51]). After defining the notion of a proof with *atom indexing* (i.e. a proof whose atomic subformulas of all cut-formulas have been assigned a unique index), we have shown that applying the proof rewriting rules defined in Chapter 3 preserves the fact that the atomic subformulas of the auxiliary formulas of each cut in an indexed proof have exactly the same indices. Furthermore, we have shown that a proof rewriting step on proofs with atom indexing corresponds to a specific class of transformation steps on the corresponding characteristic clause terms.

This result was then used to show that the characteristic clause term of an $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ (i.e. a proof with atom indexing in which all cuts are atomic and appear at the top) of an indexed proof is *subsumed* by the characteristic clause term thereof. As in the case of term resolution, towards proving the completeness of indexed resolution w.r.t. clause sets obtained from clause terms in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$, we have shown that we can always obtain a clause set of a term in $\mathbf{TACNF}_{\mathbf{ai}}$ from a clause set of a term in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$ using indexed resolution via the $\vdash_{\mathsf{res_i}}$-relation. We have also seen that the relations $\vdash_{\mathsf{res_{ta}}}$ and $\vdash_{\mathsf{res_i}}$ have a specific correspondence on the syntactical and semantical level of clause terms, respectively. In particular, if we obtain a clause term Y from a clause term X using the $\vdash_{\mathsf{res_{ta}}}$-relation, then we can obtain $|Y|$ from $|X|$ using the $\vdash_{\mathsf{res_i}}$-relation. Section 6.2 was then concluded by showing that if a set of indexed clauses subsumes another set of indexed clauses, then there always exists an indexed resolution deduction from the former that subsumes an indexed resolution deduction from the latter.

As a first step in Section 6.3, we have shown that a proof φ in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ either yields a clause term in $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$ or in $\mathbf{TACNF}_{\mathbf{ai}}$ depending on the fact whether φ contains so-called *chains of atomic cuts* or not. Subsequently, this fact was then used to prove the completeness of both term and indexed resolution w.r.t. characteristic clause terms and sets obtained from a proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$. In the case of term resolution, this was done by showing that each atomic cut-elimination step corresponds to a specific $\vdash_{\mathsf{res_{ta}}}$-step on the corresponding characteristic clause terms, where term resolution only operates on subterms $X_1 \oplus X_2$ in which the atomic cut-formula of the eliminated cut occurs both in a clause in X_1 and in X_2, but in complementary positions. The first part of the completeness-proof for indexed resolution has taken a similar approach w.r.t. the $\vdash_{\mathsf{res_i}}$-relation. Particularly, we used the information provided by the corresponding $\vdash_{\mathsf{res_{ta}}}$-sequence, which gives us insight into the structural changes that come with the elimination of an atomic cut from a proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$. On the term-level this amounts to cutting out or replacing a subterm containing the cut-formula of the eliminated cut. Since these structural changes are always of a specific form, the semantical changes can be obtained by $\vdash_{\mathsf{res_i}}$-steps in which the corresponding indexed resolution deductions always have a specific shape. More precisely, each application of the rule R_i in a single $\vdash_{\mathsf{res_i}}$-step is restricted to a fixed indexed atom, namely to the one corresponding to the atomic cut-formula of the eliminated cut. As a consequence, we have seen that eliminating a single atomic cut by reductive methods (from a proof in $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$) corresponds to a $\vdash_{\mathsf{res_i}}$-step on the corresponding characteristic clause sets. The final part of the completeness-proof was the fact that we can always obtain an indexed clause set from the characteristic clause set of the proof φ before applying any kind of reductive cut-elimination via the $\vdash_{\mathsf{res_i}}$-relation, which subsumes the characteristic clause set of an $\mathbf{ACNF}^{\mathbf{top}}_{\mathbf{ai}}$ of φ after eliminating an atomic cut by reductive methods. We have also seen that, by iterating the elimination of atomic cuts, we will eventually end up with a cut-free proof, i.e. the corresponding $\vdash_{\mathsf{res_i}}$-steps will lead to the clause set $\{\vdash\}$. Finally, these $\vdash_{\mathsf{res_i}}$-steps have been combined to a single resolution refutation of the characteristic clause set of the original proof. Since the characteristic clause set of the original proof equals the characteristic clause set of a **CERES** normal form of this proof, this means that **CERES** in conjunction with indexed resolution can actually simulate the elimination of atomic cuts by reductive methods. Thus, **CERES** (for classical first-order logic) applied to proofs with atom indexing is complete.

However, it should be noted that the obtained result is not fully general in the sense that **CERES** simulates arbitrary orders of cut-elimination steps, as the simulation result depends on the order in which the atomic cuts have been eliminated. This is due to the fact that we first transform a $\mathbf{TACNF}^{\mathbf{ext}}_{\mathbf{ai}}$ into a $\mathbf{TACNF}_{\mathbf{ai}}$.

7.1 Applications

We have already mentioned that the results obtained in this thesis are not only important in order to reduce the search space for resolution refutations of characteristic clause sets obtained by **CERES** but they do also have important implications regarding **CERES** for *intuitionistic logic*. In particular, the completeness of a specific **CERES**-variant for intuitionistic logic relies on the fact that indexed resolution is complete w.r.t. characteristic clause sets obtained from intuitionistic proofs with atom indexing. Since each **LJ**-proof is also an **LK**-proof [47], the completeness of indexed resolution w.r.t. clause sets obtained from an **LK**-proof with atom indexing also includes the case that we consider proofs in **LJ**. As a consequence, this thesis provides a partial answer to the following conjecture posed by Reis [40, Conjecture 1]:

Conjecture. *Let φ be a skolemized **LJ**-proof with cuts. Then the computation of the following steps yields an intuitionistic proof:*

- *Compute the atom indexing of φ.*
- *Extract the characteristic clause set* $\mathrm{CL}(\varphi)$.
- *Find a resolution refutation γ of* $\mathrm{CL}(\varphi)$ *using indexed resolution.*
- *Compute the projections.*
- *Join the projections (see [40, Definition 32]) using γ to guide the joining order.*

Our result provides a positive answer to the step coloured in green, i.e. whether it is always possible to find an indexed resolution refutation of the characteristic clause set obtained from an **LJ**-proof with atom indexing. However, the step coloured in red is still open, i.e. the final step towards proving the completeness of **CERES** for intuitionistic logic consists of showing that the indexed resolution refutation of the characteristic clause set guides the joining of the projections in a way that the result is an intuitionistic proof [40].

7.2 Possible Future Work

Apart from answering the remaining parts of the conjecture and thus to prove the completeness of **CERES** for intuitionistic logic, there are still a few other future research possibilities that are more directly linked to the results obtained in this thesis.

Since we now know that **CERES** can simulate reductive cut-elimination methods up to the elimination of atomic cuts, an obvious question that arises in this context is:

*How does atomic cut-elimination effect the complexity of the **CERES**-method?*

One has to keep in mind, however, that this does not only include investigating the expense of the indexed resolution steps, but also the preprocessing step in the form of computing the atom indexing of the input-proof on the one hand and the additional cut-reduction steps that are needed in order to shift the atomic cuts to the top on the other hand. In particular, this knowledge would be useful in order to compare **CERES** and reductive methods complexity-wise in a more general setting.

On a more practical level, the **CERES**-implementation could be extended to include atomic cut-elimination based on the theoretical basis obtained in this thesis. This could prove very useful, as the search space for resolution refutations—when using indexed resolution—is strongly reduced.

Since the method of term resolution is still more or less in the early stages of its development, there is probably still some potential to extend or modify the method in order to explore whether there are more general classes of clause terms for which term resolution is complete.

Due to the fact that our simulation result only covers the case that the reductive elimination of atomic cuts was done according to a specific order, it would definitely be desired to prove the results in such a way that the simulation covers an arbitrary order in which the cuts have been eliminated.

Bibliography

[1] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[2] Jeremy Avigad and Solomon Feferman. Chapter V – Gödel's Functional ("Dialectica") Interpretation. *Studies in Logic and the Foundations of Mathematics*, 137:337–405, 1998.

[3] Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, first edition, 1998.

[4] Matthias Baaz, Agata Ciabattoni, and Christian G. Fermüller. Cut-Elimination for First-Order Gödel Logic by Hyperclause Resolution. In *Logic for Programming, Artificial Intelligence, and Reasoning*, pages 451–466. Springer, 2008.

[5] Matthias Baaz and Alexander Leitsch. On Skolemization and Proof Complexity. *Fundamenta Informaticae*, 20(4):353–379, 1994.

[6] Matthias Baaz and Alexander Leitsch. Cut Normal Forms and Proof Complexity. *Annals of Pure and Applied Logic*, 97(1):127–177, 1999.

[7] Matthias Baaz and Alexander Leitsch. Cut-Elimination and Redundancy-Elimination by Resolution. *Journal of Symbolic Computation*, 29(2):149–177, 2000.

[8] Matthias Baaz and Alexander Leitsch. Ceres in Many-valued Logics. In *Logic for Programming, Artificial Intelligence, and Reasoning*, pages 1–20. Springer, 2005.

[9] Matthias Baaz and Alexander Leitsch. Towards a Clausal Analysis of Cut-Elimination. *Journal of Symbolic Computation*, 41(3-4):381–410, 2006.

[10] Matthias Baaz and Alexander Leitsch. *Methods of Cut-Elimination*, volume 34. Springer, 2011.

[11] Ulrich Berger, Stefan Berghofer, Pierre Letouzey, and Helmut Schwichtenberg. Program Extraction from Normalization Proofs. *Studia Logica*, 82(1):25–49, 2006.

[12] Ulrich Berger, Wilfried Buchholz, and Helmut Schwichtenberg. Refined Program Extraction from Classical Proofs. *Annals of Pure and Applied Logic*, 114(1):3–25, 2002.

[13] George Boolos, John P. Burgess, and Richard C. Jeffrey. *Computability and Logic*. Cambridge University Press, fifth edition, 2007.

[14] Walter S. Brainerd and Lawrence H. Landweber. Theory of Computation. 1974.

[15] William Craig. Three Uses of the Herbrand-Gentzen Theorem in Relating Model Theory and Proof Theory. *The Journal of Symbolic Logic*, 22(03):269–285, 1957.

[16] Martin Davis and Hilary Putnam. A Computing Procedure for Quantification Theory. *Journal of the ACM (JACM)*, 7(3):201–215, 1960.

[17] Elmar Eder. The Cut Rule in Theorem Proving. In *Intellectics and Computational Logic*, pages 101–123. Springer, 2000.

[18] Melvin Fitting. *First-Order Logic and Automated Theorem Proving*. Springer, 1996.

[19] Hillel Fürstenberg and Benji Weiss. Topological Dynamics and Combinatorial Number Theory. *Journal d'Analyse Mathématique*, 34(1):61–85, 1978.

[20] Jean H. Gallier. *Logic For Computer Science: Foundations of Automatic Theorem Proving*. Harper & Row Publishers, Inc., 1985.

[21] Gerhard Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210,405–431, 1934-35.

[22] Philipp Gerhardy and Ulrich Kohlenbach. Extracting Herbrand Disjunctions by Functional Interpretation. *Archive for Mathematical Logic*, 44(5):633–644, 2005.

[23] Paul C. Gilmore. A Proof Method for Quantification Theory: Its Justification and Realization. *IBM Journal of Research and Development*, 4(1):28–35, 1960.

[24] Jean-Yves Girard. *Proof Theory and Logical Complexity*, volume 1. Bibliopolis, Napoli, 1987.

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

[26] Kurt Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica*, 12(3-4):280–287, 1958.

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

[28] Stefan Hetzl. *Characteristic Clause Sets and Proof Transformations*. PhD thesis, Vienna University of Technology, 2007.

[29] Stefan Hetzl, Alexander Leitsch, and Daniel Weller. Ceres in Higher-Order Logic. *Annals of Pure and Applied Logic*, 162(12):1001–1034, 2011.

[30] Stefan Hetzl, Alexander Leitsch, Daniel Weller, and Bruno Woltzenlogel Paleo. Herbrand Sequent Extraction. In *Intelligent Computer Mathematics*, pages 462–477. Springer, 2008.

[31] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. Introduction to Automata Theory, Languages, and Computation. *ACM SIGACT News*, 32(1):60–65, 2001.

[32] Ulrich Kohlenbach. Applied Proof Theory: Proof Interpretations and their Use in Mathematics. *Springer Monographs in Mathematics*, 2008.

[33] Steven G. Krantz. The History and Concept of Mathematical Proof. *Accessed via http://www.math.wustl.edu/%7Esk/eolss.pdf (15 March 2015)*, 2007.

[34] Alexander Leitsch. *The Resolution Calculus*. Springer-Verlag New York, Inc., 1997.

[35] Alexander Leitsch. On Proof Mining by Cut-Elimination. Invited talk, July 18, All about Proofs, Proofs for all (APPA), Vienna Summer of Logic, Vienna, Austria, 2014.

[36] Alexander Leitsch, Giselle Reis, and Bruno Woltzenlogel Paleo. Towards Ceres in Intuitionistic Logic. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 16. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2012.

[37] Dale Miller and Gopalan Nadathur. *Programming with Higher-Order Logic*. Cambridge University Press, 2012.

[38] Vladimir P. Orevkov. Lower Bounds for Increasing Complexity of Derivations after Cut Elimination. *Journal of Soviet Mathematics*, 20(4):2337–2350, 1982.

[39] Pavel Pudlák. The Lengths of Proofs. In Samuel R. Buss, editor, *Handbook of Proof Theory*, pages 547–637. Elsevier, 1998.

[40] Giselle Reis. *Cut-Elimination by Resolution in Intuitionistic Logic*. PhD thesis, Vienna University of Technology, 2014.

[41] Clemens Richter. *Proof Transformation by Resolution*. PhD thesis, Vienna University of Technology, 2006.

[42] John A. Robinson. A Machine-Oriented Logic based on the Resolution Principle. *Journal of the ACM (JACM)*, 12(1):23–41, 1965.

[43] Kurt Schütte. *Beweistheorie*, volume 103. Springer, Berlin, 1960.

[44] Raymond M. Smullyan. *First-Order Logic*. Courier Dover Publications, 1995.

[45] Richard Statman. Lower Bounds on Herbrand's Theorem. *Proceedings of the American Mathematical Society*, pages 104–107, 1979.

[46] William W. Tait. Normal Derivability in Classical Logic. In *The Syntax and Semantics of Infinitary Languages*, pages 204–236. Springer, 1968.

[47] Gaisi Takeuti. *Proof Theory*. Courier Dover Publications, 2013.

[48] Anne S. Troelstra and Helmut Schwichtenberg. *Basic Proof Theory*. Number 43. Cambridge University Press, 2000.

[49] Bartel L. Van der Waerden. Beweis einer Baudetschen Vermutung. *Nieuw Archiv Wiskunde*, 15(2):212–216, 1927.

[50] Daniel Weller. *CERES in Higher-Order Logic*. PhD thesis, Vienna University of Technology, 2010.

[51] Bruno Woltzenlogel Paleo. *A General Analysis of Cut-Elimination by CERes*. PhD thesis, Vienna University of Technology, 2009.

[52] Richard Zach. Hilbert's Program Then and Now. *Philosophy of Logic. Handbook of the Philosophy of Science*, 5:411–447, 2006.