DISSERTATION

A General Analysis of Cut-Elimination by CERes

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

Univ.Prof. Dr.phil. Alexander Leitsch
Institut für Computersprachen
Theoretische Informatik und Logik

eingereicht an der

Technischen Universität Wien
Fakultät für Informatik

von

Dipl.-Ing. M.Sc. Bruno Woltzenlogel Paleo
Matrikelnummer: 0626891
Seidengasse 27/17a
A-1070 Wien

Wien, am 28. August 2009 ....

Bruno Woltzenlogel Paleo

TECHNISCHE UNIVERSITÄT WIEN
VIENNA UNIVERSITY OF TECHNOLOGY

DOCTORAL THESIS

A General Analysis of Cut-Elimination by CERes

carried out at the

Institute of Computer Languages
Theory and Logic Group
of the Vienna University of Technology

under the supervision of

Univ.Prof. Dr.phil. Alexander Leitsch

by

Dipl.-Ing. M.Sc. Bruno Woltzenlogel Paleo
Seidengasse 27/17a
A-1070 Wien

To my parents, Claudia and Claudio. . .

Kurzfassung

Das Prinzip der Beweiskompositionalität wird durch die Inferenzregel *Schnitt* formal vertreten. Beweise einfacher Lemmas können durch die Schnittregel zusammengesetzt werden, um damit komplexe Theoremen beweisen zu können. Gentzens Hauptsatz, in welchem die Eliminierbarkeit der Schnittregel bewiesen wird, ist einer der wichtigsten und berühmtesten Sätze der Beweistheorie, denn viele nützliche Korollare, wie zum Beispiel Herbrands Satz und die Teilformeleigenschaft, folgen aus ihm. Um einige von diesen Korollaren in der Praxis benutzen zu können, muss man zuerst Algorithmen entwickeln, welche die Schnittinferenzen aus Beweisen wirklich eliminiert. Die Methode CERes zeichnet sich dabei als eine effiziente und robuste Methode für Schnittelimination aus.

Diese Dissertation enthält eine generelle Untersuchung von CERes durch die Entwicklung vieler verschiedenen Varianten, die in zwei Gruppen eingeteilt werden können. Varianten der ersten Gruppe sind charakterisiert durch Änderungen in der Konstruktion der schnittzugehörigen Klausenmenge und der Projektionen. Sie nutzen die Möglichkeit aus, Inferenzen zu permutieren, und benutzen strukturelle Klauselformtransformationen, um die exponentielle Vergrößerung der Klausenmenge zu vermeiden. Die zweite Gruppe enthält Verfeinerungen des Resolutionskalküls zum Zwecke der Schnittelimination durch CERes. Die Verfeinerungen beschränken die Benutzung der Inferenzregeln des Resolutionskalküls, sodass sich die Varianten in der Mitte, bezüglich kanonischer Refutationen, zwischen unbeschränktem CERes und reduktiven Schnitteliminationsmethoden bewegen. Deswegen können diese Varianten weiter erklären, wodurch diese zwei anscheinend sehr verschiedenen Methoden sich wirklich unterscheiden und trotzdem ähnlich sind.

Schließlich wird in dieser Dissertation auch gezeigt, wie CERes in eine Methode für die Einfürung atomarer Schnitte (CIRes) umgewandelt werden kann. Diese Methode kann Beweise komprimieren, und es wird vermutet, dass exponentiell kleinere Beweise dadurch erhalten werden können.

Abstract

The cut rule formally represents the principle of compositionality of proofs. Proofs of simple lemmas can be composed using the cut rule to form proofs of more complex theorems. The cut-elimination theorem (i.e. the completeness of Sequent Calculus without the cut-rule) is one of the most important and famous theorems of proof theory, mainly because it leads to many useful corollaries, such as the subformula property and the midsequent or Herbrand's theorem. However, in order to exploit these corollaries in practice, it is often necessary to have algorithms for the actual elimination of cuts, and CERes stands out as an efficient and robust cut-elimination method based on the resolution calculus.

This thesis contains a general investigation of CERes through the development of several variants of the method, which can be distinguished in two groups. The first group consists of variants obtained by modifying the construction of cut-pertinent clause sets and projections. They exploit the possibility of swapping inferences in sequent calculus proofs and use structural clause form transformation to avoid the exponential blow-up in the size of the clause sets. The second group consists of refinements of the resolution calculus that are specific for cut-elimination by CERes. A few refinements are defined by increasingly restricting the applicability of the inference rules of the resolution calculus in such a way that the variants are intermediary, regarding simulation with respect to canonic refutations, between the unrestricted CERes and reductive methods of cut-elimination (i.e. methods based on local proof rewriting rules). Consequently, this group of variants further clarifies the differences and similarities between these two kinds of methods, which appear to be so distinct from each other.

Furthermore, this thesis shows how CERes can be transformed into a method of atomic-cut-introduction (CIRes), which is capable of compressing proofs. Asymptotically, an exponential compression in the size of proofs is conjectured to be achievable by the method.

Contents

Contents i

1 Introduction 5
1.1 Introduction 5

2 Basics of Logic 9
2.1 The Language 10
2.2 Sequent Calculus 12
2.3 Resolution Calculus 17

3 Cut-Elimination 23
3.1 The Importance of Cut-Elimination for Logic 23
3.1.1 Related Approaches 26
3.2 Reductive Cut-elimination Proof Rewriting Rules 27
3.2.1 Preliminaries about Rewriting Relations 28
3.2.2 The Rewriting Rules 29
3.2.3 Examples of Reductive Cut-Elimination 35
3.2.4 Some Properties of Reductive Cut-elimination 40

4 Cut-Elimination by Resolution 43
4.1 Cut-Pertinent Struct 45
4.2 CERes_S 48
4.2.1 Cut-Pertinent Standard Clause Set 48
4.2.2 Projections 50
4.3 CERes_W 67
4.3.1 Inference Swapping 67
4.3.2 Cut-Pertinent Swapped Clause Sets 80
4.3.3 Projections 89
4.4 CERes_P 90

4.4.1 Cut-Pertinent Profile Clause Set 91
4.4.2 Projections 94
4.5 CERes$_D$ 94
4.5.1 Cut-Pertinent Definitional Clause Set 95
4.5.2 Projections 97
4.6 CERes$_{DW}$ 104
4.6.1 Cut-Pertinent Swapped Definitional Clause Set . . . 104
4.6.2 Projections 107
4.7 CERes-Normal-Forms 109
4.8 CERes Ignoring Atomic Cuts 118
4.9 CERes Ignoring Quantifier-Free Cuts 121

5 Comparison of Cut-Elimination Methods 123
5.1 Canonic Refutations 124
5.2 CR-Simulation 128
5.2.1 CR-Simulation between Two Reductive Methods . . 128
5.2.2 CR-Simulation by CERes 129

6 Resolution Refinements for Cut-Elimination 145
6.1 Refuting Clause Sets versus Reproving End-Sequents 146
6.2 Refinements based on Reductive Methods 152
6.2.1 Cut-Linkage and Cut-Side 153
6.2.2 Cut-Linkage Preservation 156
6.2.3 Cut-Side Preservation 163
6.2.4 Using Cut-Linkage 163
6.2.5 Using Cut-Linkage (More Strictly) 165
6.2.6 Using Cut-Side and Cut-Linkage 166
6.2.7 Using Atomic Cut-Linkage 168
6.2.8 Refined Refutability 169
6.2.9 CR-Simulation among Refinements 172

7 Cut-Introduction 173
7.1 Cut-Introduction by Resolution 174
7.2 Remarks 183

8 Complexity 185
8.1 Number of Clauses in Clause Sets 185
8.2 Search Space 190
8.3 Proof Sizes and Lengths 192
8.4 Proof Compression by Cut-Introduction 195

9 Applications of Cut **197**
9.1 Physics: Energy Conservation as Cut 197
9.2 Robotics and Agent Theory: Plan Reuse as Cut 206
9.3 Machine Learning: Decision Tree as Cut 211

10 Conclusion **217**

Bibliography **221**

List of Notations **229**
Notations for Reductive Cut-Elimination 229
Notations for Cut-Elimination by Resolution 230
Notations for Cut-Introduction by Resolution 231
Notations for Auxiliary Proof Transformations 231

Index **233**

Preface and Acknowledgments

For so many reasons, this thesis would not exist without the support of my supervisor, Prof. Alexander Leitsch. Firstly, he offered me a Ph.D. position in his group, giving me the opportunity to research in his challenging project in the intersection of elegant and exciting areas such as proof theory, automated deduction and mathematics. Then he gave me a clear initial goal (i.e. to develop resolution refinements for cut-elimination) and a good starting direction toward the goal (i.e. to investigate more deeply the differences and similarities between reductive methods of cut-elimination and cut-elimination by resolution). Due to my relative lack of experience at the beginning of my Ph.D. studies, I would never have thought of such an interesting, relevant and useful initial goal only by myself at that time. As usual, the initial goal eventually gave rise to many other goals, some of them not so related to the initial goal (e.g. the search for a more intuitive definition of Stefan Hetzl's profile clause sets, my concerns about the in principle unnecessary duplications (exponentially-many in the worst case) that occur in the construction of some clause sets and my search for a solution to this problem). At these moments, Prof. Alexander Leitsch gave me enough freedom to explore, combined with good guidance to keep me on track. He also gave me hints (e.g. structural clause form transformations) about how to tackle some problems posed by the additional goals. But his support was not only academic. He was, for example, immensely comprehensive when a conjunction of housing and health problems caused me to be almost totally unproductive for a few months. For all this, I would like to thank him.

Other professors of the Theory and Logic group of Vienna also contributed to my general knowledge of Logic. I would like to thank especially: Prof. Maathias Baaz, for his lectures on Proof Theory; and Prof. Agata Ciabattoni and Prof. Chris Fermüller, for informal discussions about non-classical logics.

During my Ph.D. I was accompanied by great colleagues to whom I am also very thankful. Stefan Hetzl's profile clause set (and its improvement, the swapped clause set, developed in this thesis) allowed me to state and prove many theorems much more elegantly than if I had to use the standard clause set. Moreover, my discussions with him were very helpful in my search for an alternative definition of profile clause set, especially when he found a "bug" in one of my first attempts. Daniel Weller and I started our Ph.D. studies almost simultaneously and had to struggle to understand and improve (in different ways) the method of cut-elimination by resolution together. Being able to discuss the method with him, and especially being able to learn from him about the difficulties of extending the method to second-order logic, was very helpful to broaden and deepen my understanding. Tomer Libal's flexible, interactive and automated theorem prover (ATP), developed during his M.Sc. studies, will serve as a basis for the future implementation of the resolution refinements described in this thesis. Conversely, I wish the refinements will be helpful to him in his Ph.D. studies concerning theorem proving and the refutation of clause sets in second-order logic. Mikheil Rukhaia implemented Gentzen's method of cut-elimination during his M.Sc. studies [105], thus allowing empirical comparisons between cut-elimination by resolution and Gentzen's method and hence complementing the theoretical comparison present in this thesis. Cvetan Dunchev, in his M.Sc. thesis [38], continued and improved the work on Herbrand sequent extraction contained in my M.Sc. thesis. We all have extensively used **CERes**, **ProofTool** and **HLK**, which were initially programmed by the former Ph.D. students Clemens Richter and Hendrik Spohr. I am therefore also grateful to them for paving our way and freeing us from the burden of building the software infra-structure from scratch, even though we sometimes do not agree with all implementation choices that have been made.

I must also thank my friends (and it must be remarked that the set of my friends has a large intersection with the set of previously mentioned colleagues) for the leisure activities (e.g. the basketball and board games, biking, dinners . . .) that we did together and for all the help provided when I needed it (e.g. when I was having housing problems).

My fiancee Katya (Ekaterina Lebedeva) supported me in so many different ways that are hard to describe, analyze and list, but so easy to feel as love. Exploring science with her has been a most joyful experience.

I find her work in computational linguistics great, and my attempts to understand it during our discussions are always expanding the limits of my knowledge of logic. She has always praised my work, and even though her opinion might be biased by her love, her nice words about my work have always motivated me a lot. The unfortunate geographical distance that separated us for two years was a major source of motivation for me to complete my thesis in a record time, in order to join her in France for a post-doc research job that she helped me to find.

In my opinion, the study of logic, and especially of proof theory, requires a mindset characterized by a balanced combination of diverse and almost mutually contradictory traits such as curiosity, creativity and discipline. Considering this, I could say that the work that culminated in this thesis is not limited to the approximately 22 months during which I have officially worked on it. This thesis is actually just a landmark fragment of a lifelong ongoing process of perfecting and refining this mindset. I owe the start and maintenance of this process during my childhood and youth to my parents, Claudia Zambello Woltzenlogel Paleo and Claudio Mastrocolla Paleo. They have always done their best to provide me the best education and environment, where I could develop my interests without worries at all.

If I had only supporters, things would be so much easier. Unfortunately there were also enemies, whom, I believe, deserve to be acknowledged as well, for their obstacles made me stronger. Firstly, I would like to acknowledge the coordinators of the Erasmus Mundus M.Sc. Programme in Computational Logic (EMCL), especially Steffen Hoeldobler. Never in my life I had encountered so much corruption, condescending arrogance and false advertisement of a low-quality programme as I encountered in the EMCL. I certainly do not recommend this programme to any prospective student with a true interest in logic, and I find it shameful that the European Commission funds it. I used to think that the European Commission had been a victim of the EMCL's false advertisement too, but later I realized that it is all just politics. I therefore extend my dislike to all people who prioritize politics instead of scientific merit, especially those who have the power to fix the corruption, but for political reasons do not.

Chapter 1

Introduction

1.1 Introduction

Cut-elimination theorems and algorithms that actually perform the elimination of cuts from proofs are among the most prominent results and techniques of proof theory and of logic in general. Originally devised as a way to prove consistency [48, 49], cut-elimination also has many important applications (discussed in Section 3.1).

Gentzen's demonstration of the cut-elimination theorem implicitly defines a method (described in Chapter 3) that actually eliminates cuts from proofs. The method is based on certain proof rewriting rules, which are very convenient for demonstrating the theorem by induction, because they reduce (hence the name *reductive methods*) the grade and rank measures of the proof. However, the use of the method in practice is jeopardized by the fact that it is unable to exploit redundancies in the proofs, as it is a very local method, and hence is not very efficient.

Therefore, an alternative and substantially different method of cut-elimination, known as CERes (*Cut-Elimination by Resolution*, defined in detail in Chapter 4), has recently been developed with the aim of being an efficient method of cut-elimination to be used in practice, and particularly for the automated analysis and transformation of formalized mathematical proofs. In contrast to reductive methods, CERes is a global method. It extracts an unsatisfiable clause set from the whole input proof with cuts, and refutes it by resolution. The leaf clauses of the resolution refutation can be replaced by so-called projections, which are cut-free parts of the input proof. In this way, a normalized proof in which all cuts are atomic is created.

In spite of its superiority, CERes is constantly under development and there is still much room for improvement. Among the possible improvements that were clear at the start of this thesis, refinements of the resolution calculus specifically for CERes stood out as a main goal to be pursued. Indeed, experiments with the current implementation of the method had shown that typical theorem provers working only with the unrestricted resolution calculus had been unable to refute clause sets within a reasonable time. The price paid for CERes's flexibility and generality is the large search space for unrestricted resolution refutations. The task of the refinements (defined in Chapter 6) is to allow controlled restrictions of CERes in order to make refutations easier to find.

In principle there could be many ways for refining the resolution calculus and restricting CERes. In this thesis, focus lies on defining refinements and restrictions such that the restricted CERes methods are, in a certain sense, intermediary between reductive methods and the unrestricted CERes. Consequently, Chapter 5 and most of Chapter 6 are dedicated to comparing CERes with reductive methods. The refinements could only be developed by understanding the differences and similarities between them, and hence these chapters also contribute to clarifying the essential ideas behind both methods.

For the technical reason that profile clause sets (developed in [66]) are more invariant under rank reduction than standard clause sets, its use had been preferred in this thesis, especially because the proofs of many theorems in Chapters 5 and 6 become simpler. However, the original definition of profile clause sets is highly technical and lacks an intuitive explanation. This served as a motivation to look for more natural ways of defining the CERes method, which eventually culminated in an approach that relates construction of clause sets to conjunctive normal form transformations and abandons the operations of union and merge that used to be employed before (Section 4.2.1). Even if naturality is essentially a matter of personal taste, at least the approach defined here provides a different angle to look at and understand how the CERes method works. Moreover within this approach, it is shown that the optimizations of profile clause sets correspond to exploiting the possibility of swapping inferences less redundantly in the input proof (Section 4.4). The investigation of inference swapping and normalization for clause sets led to the invention of swapped clause sets (Section 4.3), which behave more uniformly with respect to inferece swapping than profile clause sets. Furthermore, the fact that the construction of standard, swapped and profile clause sets is analogous to standard conjunctive normal form transformation makes it evident that these clause sets suffer from the same kind of exponential blow-up in size. To avoid this problem, structural conjunctive normal form transformations were investigated, leading to the development of definitional and swapped definitional clause sets (Sections 4.5 and 4.6). Other variations (Sections 4.8 and 4.9) of the CERes method were inspired by previous experience with Herbrand sequent extraction [95, 94], for which it is sufficient to eliminate quantifiers from cut-formulas.

Investigations of the behavior of CERes using the more recent variants of clause sets (i.e. profile and swapped clause sets) together with a new kind of projection, called O-projection, on proofs of different kinds led to the realization that the method could be modified to do not cut-elimination but rather cut-introduction by resolution (CIRes, Chapter 7), resulting in potential compression of proofs.

Throughout the thesis all the developed and described methods are compared mostly in a purely qualitative manner. Nevertheless, some quantitative complexity results comparing sizes of search spaces, proof normal forms and clause sets are shown in Chapter 8.

In an attempt to make this thesis reasonably self-contained, a brief chapter containing basic notions of Logic (Chapter 2) is also present. Moreover, for the non-proof-theorist reader who might wonder if this technical proof-theoretical work and, more generally, the study of the cut rule, its elimination and introduction is relevant outside proof theory, an informal chapter (Chapter 9) discusses and exemplifies the existence, introduction and elimination of cuts that are implicit in other scientific disciplines.

Chapter 2

Basics of Logic

As Modern Logic is a very broad and ever expanding field of studies, any attempt to define it precisely is unlikely to be successful. Nevertheless, one can easily notice that logicians are usually concerned with syntactically well-defined *languages*, whose sentences have meanings (frequently *truth* and *falsity*) depending on *interpretations*. Furthermore, one of the main problems in Logic is that of deciding (or trying to decide, as it is usually an undecidable problem) *entailment* ($\models$) of a sentence F from a set of sentences S, deciding whether it holds that, if all sentences of S are true under some interpretation, then F must also be true under the same interpretation. As it would be impractical to exhaustively test all possible interpretations, logicians attempt to *prove* or *derive* F from S in formal *calculi*. Such calculi must be carefully designed to be *sound* (i.e. if there is a proof of F from S, then F is entailed by S) and, if possible[a], *complete* (i.e. if F is entailed by S, then there is a proof of F from S).

This thesis is mainly concerned with a language for pure predicate logic, defined in Section 2.1, and a sequent calculus for this language, defined in Section 2.2 and known to be sound and complete. Moreover, the resolution calculus, defined in Section 2.3, is an essential auxiliary calculus necessary for the technique of transformation of sequent calculus proofs known as cut-elimination by resolution, which is explained in Chapter 4 and refined in Chapter 6.

[a]It is a consequence of Goedel's Incompleteness Theorem that a complete calculus is not always possible [56, 57]

2.1 The Language

The *language* defined in this section has *sentences* composed of expressions that can be distinguished as either *terms* or *formulas* inductively defined over a *signature*. It is a quite standard language for *predicate logic* or *first-order logic*, so called because, from a type theoretical point of view, its expressions have types of order smaller or equal to one and quantifications are allowed on variables of individual type.

Definition 2.1.1 (Signature)**.** The *signature* of the language consists of:

1. A countably infinite set of *variables* $\mathcal{V}$
2. For every $n \geq 0$, a countably infinite set of *function symbols* $\mathcal{F}_n$. ($\mathcal{F} \doteq \bigcup_{n\geq 0} \mathcal{F}_n$ is the set of all function symbols. $\mathcal{F}_0$ is the set of *constant symbols*).
3. For every $n \geq 0$, a countably infinite set of *predicate symbols* $\mathcal{P}_n$. ($\mathcal{P} \doteq \bigcup_{n\geq 0} \mathcal{P}_n$ is the set of all predicate symbols).
4. The set of propositional connectives $\{\vee, \wedge, \rightarrow, \neg\}$
5. The set of quantifiers $\{\forall, \exists\}$
6. The set of parentheses $\{(,)\}$

In this thesis, uppercase letters (e.g. A, B, C,. . . , P, Q, R) are used for predicate symbols; lowercase letters are used either for function symbols (e.g. $a, b, c, \ldots, f, g, h,\ldots$) or for variables (e.g. $x, y, z,\ldots$).

Definition 2.1.2 (Terms)**.** The *set of terms* $\mathcal{T}$ is defined as the smallest set satisfying:

1. $\mathcal{V} \subset \mathcal{T}$
2. For all $n \geq 0$: If $f \in \mathcal{F}_n$ and $t_1, \ldots, t_n \in \mathcal{T}$, then $f(t_1, \ldots, t_n) \in \mathcal{T}$.

Definition 2.1.3 (Formulas)**.** The *set of formulas* $\mathcal{B}$ is defined as the smallest set satisfying:

1. For all $n \geq 0$: If $P \in \mathcal{P}_n$ and $t_1, \ldots, t_n \in \mathcal{T}$, then $P(t_1, \ldots, t_n) \in \mathcal{B}$.
2. If $F \in \mathcal{B}$, then $F' \equiv \neg F \in \mathcal{B}$
3. If $F_1, F_2 \in \mathcal{B}$ and $\circ \in \{\wedge, \vee, \rightarrow\}$, then $F' \equiv (F_1 \circ F_2) \in \mathcal{B}$
4. If $F \in \mathcal{B}$ and $x \in \mathcal{V}$ and $Q \in \{\exists, \forall\}$, then $F' \equiv (Qx)F \in \mathcal{B}$

Formulas of the first base kind above are called *atomic*. Formulas constructed inductively according to the three last cases above are called *compound*. In such cases, the formulas F_1, F_2 and F are called *direct subformulas* of F'. Inductively, the *subformulas* of F' are its direct subformulas, all the subformulas of its direct subformulas, and F' itself.

Definition 2.1.4 (Annotated Formula)**.** An *annotated formula* is a pair (F, l) where F is a formula and l is an annotation of that formula occurrence (e.g. a label, a color, or any other structure that could store relevant information for manipulations of the formula)[b].

Definition 2.1.5 (Scope of Quantifiers and Bound and Free Variables)**.** In a formula F of the form $(Qx)F'$, with $Q \in \{\forall, \exists\}$, F' is the *scope* of (Qx). An occurrence of x in F' is *bound* by (Qx) if and only if there is no $(Q'x)$ in F' such that the occurrence is in the scope of $(Q'x)$. A variable occurrence is *free* if and only if it is not bound by any quantifier.

By a suitable renaming of the variable occurrences bound by each quantifier to new distinct variables, a formula can always be brought to an equivalent form in which all quantifiers bind different variables and all the occurrences of each variable are either all bound or all free. Subsequently, it will be assumed that all formulas are in this form.

Definition 2.1.6 (Sentence)**.** A formula is a *sentence* if and only if it has no free variables.

Definition 2.1.7 (Polarity of Formula Occurrences)**.** The *polarity* of a formula occurrence F' is defined as follows:

be a direct subformula occurrence of a formula occurrence F with an assigned polarity $\mathrm{p}(F) \in \{\text{positive}, \text{negative}\}$. Then:

- If F' is not a subformula occurrence of any other formula occurrence, then $\mathrm{p}(F') = \text{positive}$.
- Otherwise, let F be the formula occurrence of which F' is a direct subformula occurrence. Then:
  - If $F \equiv \neg F'$, then $\mathrm{p}(F') = \overline{\mathrm{p}(F)}$.
  - If $F \equiv (F' \rightarrow F'')$, then $\mathrm{p}(F') = \overline{\mathrm{p}(F)}$.
  - Otherwise, $\mathrm{p}(F') = \mathrm{p}(F)$

[b]The use of labels in proof theory is widespread. I unified treatment of labeled deductive systems can be found in [45]. Examples of labels used for resolution refinements can be found, for example, in [80, 44]

Definition 2.1.8 (Strong and Weak Quantifiers)**.**
Let $F \doteq (Qx)F'$ be a subformula occurrence. Q is called *strong* if and only if $(Q, \mathrm{p}(F) \in \{(\forall, \text{positive}), (\exists, \text{negative})\}$. Otherwise it is called *weak*.

2.2 Sequent Calculus

The first two variants of *sequent calculi* were invented by Gentzen [48, 49], and they were called **LK**, for classical predicate logic, and **LJ**, for intuitionistic predicate logic. In this thesis a variant of **LK** is used, in which no implicit weakening and no implicit contraction occur in inferences[c]. The reason for this is that contraction and weakening play an important role in cut-elimination, and the fact that they can only occur explicitly makes it easier to keep track of their effect on cut-elimination methods and, consequently, certain proofs in Chapters 4 and 6 turn out to be more elegant than if they were shown for other variants of **LK**.

A sequent calculus relies on the notion of *sequent*, and its *proofs* are composed of *inferences* that operate on sequents according to the *inference rules* of the calculus. Formal definitions of these concepts are given subsequently in this Section.

Definition 2.2.1 (Sequent)**.** A *sequent* $\Gamma \vdash \Delta$ is composed of two multisets[d] of formulas: the *antecedent*, Γ, and the *consequent* , Δ. The formulas in the consequent are assigned positive polarity, and the ones in the antecedent are assigned negative polarity.

Definition 2.2.2 (Formula Corresponding to a Sequent)**.** The *formula corresponding to a sequent* $\Gamma \vdash \Delta$ is:

[c]Such a variant sequent calculus is usually called *purely multiplicative*, following a terminology from linear logics [52, 53]

[d]It is also common to define sequents using *sets* or *lists*, instead of multisets. The use of sets would imply that contractions are implicit, because a set does not state the multiplicity of its elements. Indeed, contraction inferences would be superfluous, since their premise sequents would be the same as their conclusion sequents, and the issues of contractions would be transfered to the quantifier rules, which would have to be modified slightly to implicitly simulate contraction. A definition using sets would therefore not be convenient for the investigations of this thesis, exactly because contraction would be implicit. The use of lists would have the advantage of being conceptually closer to simple and yet efficient implementations, as lists are common data structures in many computer languages. However, it requires additional structural rules of inference like *exchange* or *permutation* to take care of the correct positioning of formulas. A sequent calculus using lists for sequents, including a description of cut-elimination by resolution in this calculus, can be seen in [102] and served as the theoretical basis for the implementation of the method in the system **CERes**. As the position of formulas is not relevant in classical predicate logic, however, the definition via multisets is clearly more convenient for the theoretical investigations of this thesis. Note, however, that some *substructural logics*, such as, for example, the logic defined by the *Lambek calculus*, do not admit the exchange rule, and hence they do not admit the multiset definition for sequents.

$$\mathcal{F}(\Gamma \vdash \Delta) \doteq ((\bigwedge \Gamma) \rightarrow (\bigvee \Delta))$$

Definition 2.2.3 (Substitution)**.** A *substitution* is a finite set of assignments[e] of terms to variables. If $\sigma \equiv \{v_1 \leftarrow t_1, \ldots, v_n \leftarrow t_n\}$ is a substitution, then $F\sigma$ is the formula obtained from F by substituting all occurrences of free-variables v_j by the corresponding terms t_j.

Definition 2.2.4 (Inference Rules (of the Sequent Calculus **LK**))**.** The *inference rules* of sequent calculus **LK** are shown below[f]:

- **The Axiom Rule**:

$$\frac{}{A \vdash A}\ axiom$$

  where A is any atomic formula[g].

- **Propositional Rules**:

$$\frac{F_1, F_2, \Gamma \vdash \Delta}{F_1 \wedge F_2, \Gamma \vdash \Delta}\ \wedge_l \qquad \frac{\Gamma_1 \vdash \Delta_1, F_1 \quad \Gamma_2 \vdash \Delta_2, F_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, F_1 \wedge F_2}\ \wedge_r$$

$$\frac{F_1, \Gamma_1 \vdash \Delta_1 \quad F_2, \Gamma_2 \vdash \Delta_2}{F_1 \vee F_2, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\ \vee_l \qquad \frac{\Gamma \vdash \Delta, F_1, F_2}{\Gamma \vdash \Delta, F_1 \vee F_2}\ \vee_r$$

$$\frac{\Gamma_1 \vdash \Delta_1, F_1 \quad F_2, \Gamma_2 \vdash \Delta_2}{F_1 \rightarrow F_2, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\ \rightarrow_l \qquad \frac{F_1, \Gamma \vdash \Delta, F_2}{\Gamma \vdash \Delta, F_1 \rightarrow F_2}\ \rightarrow_r$$

[e] Alternatively, but equivalently, a substitution can be defined as a function from variables to terms that is different from the identity function only in a finite subset of its domain.

[f] Apart from the rules shown here, there are two kinds of rules that are particularly useful in the formalization of mathematical proofs. As equality is a very common predicate in mathematics, it makes sense to have specialized rules to handle it. One approach is to have *equality rules* that resemble *paramodulation* [34], while another complementary approach is to work with *deduction modulo* [36, 37]. Cut-elimination by resolution in a sequent calculus with equality rules has been studied in [102]. Even though, for simplicity, these rules are not used in this thesis, the techniques developed here could be applied to cut-elimination in a sequent calculus having these rules as well. Cut-elimination by resolution in a calculus with deduction modulo is currently being investigated in [103].

[g] The axiom rule has many variants. Its conclusion sequent could, for example, have contexts (i.e. $\Delta, A \vdash A, \Gamma$). While this would have the advantage of rendering the weakening rules superfluous (i.e. eliminable or admissible), it is not appropriate for the goals of this thesis, since weakening would then be essentially built implicitly in the axiom rule. Another variant consists of allowing arbitrary atomic formulas in the sequent. This allows more compact and human-readable proofs but prevents the eliminability of atomic cuts in general [22]. Furthermore, the restriction that formulas should be atomic could be dropped, but this would obscure investigations of the complexity of cut-elimination and renders some transformations of proofs impossible (e.g. Skolemization of proofs would be impossible if formulas containing quantifiers were allowed in axiom sequents) [10].

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

- **Structural Rules**:

  - **Weakening Rules**[h]:

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

  - **Contraction Rules**[i]:

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

- **The Cut Rule**[j]:

$$\frac{\Gamma_1 \vdash \Delta_1, F \quad F, \Gamma_2 \vdash \Gamma_2}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\ cut$$

  The cut rule is an *analytic cut-rule* iff F is a subformula occurring in Γ_1, Γ_2, Δ_1 or Δ_2.

- **(First-order)**[k] **quantifier rules**:

$$\frac{F\{x \leftarrow t\}, \Gamma \vdash \Delta}{(\forall x)F, \Gamma \vdash \Delta}\ \forall_l \qquad \frac{\Gamma \vdash \Delta, F\{x \leftarrow \alpha\}}{\Gamma \vdash \Delta, (\forall x)F}\ \forall_r$$

$$\frac{F\{x \leftarrow \alpha\}, \Gamma \vdash \Delta}{(\exists x)F, \Gamma \vdash \Delta}\ \exists_l \qquad \frac{\Gamma \vdash \Delta, F\{x \leftarrow t\}}{\Gamma \vdash \Delta, (\exists x)F}\ \exists_r$$

[h] The acceptance of weakening rules is closely related to the fact that the logic considered here is *monotonic* [45]. In *non-monotonic* logics, such as *default logic* [100, 58] and reasoning under closed world assumption [99], circumscription [82, 41], negation-as-failure [25], answer set programming [47], the addition of more information to the knowledge base (i.e. w_l) could falsify sentences that were previously considered true.

[i] The acceptance of contraction rules is related to the fact that, in the logic considered here, sequents are judgments about the relative *truth* of the formulas that it contains. On other logics, however, they are seen as judgments about, for example, *resources* of the kinds expressed by the formulas, and hence contraction is not accepted because resources cannot be indefinitely consumed [52]; Other logics that reject contraction are investigated in [101].

[j] Although the cut rule is usually classified as a structural rule, because, as in the case of weakening and contraction, the logical forms of the formulas are not changed, this is not done in this thesis, simply because of the special role played by the cut.

[k] A much harder case for cut-elimination by resolution occurs if inference rules that go beyond predicate logic and into higher-order logics are added to the calculus, such as rules for quantifiers over expressions of higher-order types. This has been partially studied in [71, 69] and is still under investigation in [116, 81].

For the $\forall_r$ and the $\exists_l$ rules[l] the variable α must not occur in Γ nor in Δ nor in F. This is the *eigenvariable condition*.

For the $\forall_l$ and the $\exists_r$ rules the term t must not contain a variable that is bound in F.

The sequent below the line of an inference rule is its *conclusion*, while the sequents above the line are its *premises* . An inference rule is *nullary*, *unary*, *binary*, *n-ary* if and only if it has, respectively, 0, 1, 2, n premises.

In the inference rules above, the colored formulas are called *active* Γ,Δ,Γ_1,Δ_1,Γ_2,Δ_2 are multisets of formulas called *contexts*. The active formulas in the conclusion sequent of a rule, colored in red, are the *main* formula of the rule, while the active formulas in the premises, colored in blue, are the *auxiliary* formulas.

Definition 2.2.5 (Inference Rules (of the Sequent Calculus **LKD**))**.** The *inference rules* of the sequent calculus **LKD** are the rules of the sequent calculus **LK** together with the definition rules shown below[m]:

- **Definition Rules**:

$$\frac{F[x_1,\dots,x_n],\Gamma \vdash \Delta}{P(x_1,\dots,x_n),\Gamma \vdash \Delta}\ \forall_l \qquad \frac{\Gamma \vdash \Delta, F[x_1,\dots,x_n]}{\Gamma \vdash \Delta, P(x_1,\dots,x_n)}\ \forall_r$$

  where P is a predicate defined by

$$P(x_1,\dots,x_n) \leftrightarrow F[x_1,\dots,x_n]$$

Definition 2.2.6 (Inference)**.** An *inference* is an instance of an inference rule.

Definition 2.2.7 (Proof)**.** A *proof* is a tree of inferences in which every premise sequent of an inference is a conclusion sequent of another inference. The last sequent derived by a proof is its *end-sequent*.

Definition 2.2.8 (Ancestor Relation)**.** For every inference ρ of a proof φ, each auxiliary formula occurrence ω_a^i is the *immediate ancestor* of the corresponding main formula occurrence ω_m. Moreover, every formula

[l]According to Definitions 2.2.1 and 2.1.8, the $\forall_r$ and the $\exists_l$ always introduce strong quantifiers. For this reason, they are also called *strong quantifier rules*. Analogously, the other quantifier rules introduce only weak quantifiers and are called *weak quantifier rules*

[m]The main advantage of having definition rules lies in the fact that mathematical proofs frequently define new concepts as a way to structure reasoning. This corresponds to the *extension* principle in Logic and can be handled by *definition rules* or *superdeduction rules* [17] (essentially macro-inferences composed of definition rules and propositional rules).

occurrence ω_{pc} in the context of the premise sequent is also an *immediate ancestor* of the corresponding formula occurrence ω_{cc} in the context of the conclusion sequent. The *ancestor relation* is the reflexive, transitive and compatible (with respect to subformulas) closure of the immediate ancestor relation. The fact that a (sub)formula occurrence ω_i is an ancestor of a (sub)formula occurrence ω_j is denoted: $\omega_i \searrow \omega_j$.

Theorem 2.1 (Soundness of the Sequent Calculus)**.**
If $F_1, \ldots, F_n \vdash F'_1, \ldots, F'_m$ is the end-sequent of a sequent calculus proof, then $\{F_1 \wedge \ldots \wedge F_n\} \vDash F'_1 \vee \ldots \vee F'_m$.

Proof. A detailed proof can be found in [112]. A sketch follows:

By induction on the length of proofs. In the base case, consider a proof consisting of a single axiom inference. As induction hypothesis, assume that the theorem holds for the immediate subproofs (which have shorter lengths, of course) above the last inference, and show that it also holds for the whole proof no matter the rule ($\vee_r$,$\wedge^l$, $\forall_l$, *cut*,...) of which the last inference is an instance. □

Theorem 2.2 (Completeness of the Sequent Calculus)**.**
If $\{F_1 \wedge \ldots \wedge F_n\} \vDash F'_1 \vee \ldots \vee F'_m$, then $F_1, \ldots, F_n \vdash F'_1, \ldots, F'_m$ is the end-sequent of a sequent calculus proof.

Proof. Gentzen, in [48], did not prove the completeness of sequent calculus as stated above, but only the relative completeness with respect to a Hilbert deduction system [74] and natural deduction, a calculus that was also defined in [48] and further investigated in [97]. A comprehensive detailed proof can be found in [112]. □

Remark 2.2.1 (The Meanings of Sequents, Rules and Proofs)**.** By theorems 2.1 and 2.2, the meaning of a sequent is precisely fixed by the used semantics (i.e. by the entailment or logical consequence relation $\vDash$). In predicate logic, the entailment relation satisfies the deduction theorem, and, therefore, the meaning of a sequent $\Gamma \vdash \Delta$ is the same as the meaning of the formula $\bigwedge \Gamma \rightarrow \bigvee \Delta$[n]. In other words, the commas in the antecedent of a sequent can be seen as conjunctions, the commas in the consequent of a sequent can be seen as disjunctions, and the sequent symbol ($\vdash$) can be seen as implication. In this sense, the rules of sequent calculus can be seen as operating on subformulas at depth 2 of formulas having a particular structure.

[n]Or $\Gamma \vdash \Delta$ could be understood as meaning that $\bigwedge \Gamma \wedge \bigwedge \neg\Delta$ is unsatisfiable, in which case sequent calculus acquires a refutational flavor [35].

Alternatively, sequents can be seen as rules and rules of the sequent calculus can be seen as meta-rules generating new (conclusion) rules from previous (premise) rules [117]. So, the sequent $\Gamma \vdash F$ can be read as the rule that F can be inferred (or "proved", or "derived") from the assumptions Γ. The meaning of cut within this alternative is clear: the left premise states that the *lemma* F (assume Δ_1 is empty for simplicity) can be inferred (or "proved") from Γ_1; the right premise states that the lemma, together with Γ_2 can be used to infer (or "prove") something else Δ_2; finally the conclusion of the cut, which is now seen as a meta-rule, is a rule stating that Δ_2 can be inferred (or "proved") from the assumptions Γ_1 and Γ_2.

It is this richness of possible meanings that makes sequent calculus so convenient for many purposes, including the formalization of mathematical proofs.

Definition 2.2.9 (Proof Sizes)**.** Let φ be a proof. Then *length*(φ) denotes the length of φ (i.e. the total number of inferences in φ), $|\varphi|_a$ denotes the atomic size of φ (i.e. the total number of predicate sysmbols occurring in φ), and $|\varphi|$ denotes the symbolic size of φ (i.e. the total number of (constant, function, variable, predicate, connective and quantifier) symbols in φ).

2.3 Resolution Calculus

While the sequent calculus presented in the previous section is convenient for the formalization of proofs, it is inadequate for automated proof search. This inadequacy originates mainly from the fact that the instantiation (considering a bottom-up proof construction) of weakly quantified variables according to weak quantifier rules is unrestricted. This inadequacy led to the development of calculi such as the *Resolution Calculus* (**R**) [104, 80], whose power in proof search essentially stems from restricting substitutions of variables to *most general unifiers*. The use of *clauses*, instead of arbitrary sequents, further eases the search process, as the lack of logical structure (i.e. all formulas are simply atoms) allows the calculus to have just two rules of inference: *resolution* and *factoring*.

Although resolution deals with sequents in a restricted form (i.e. clauses), it can be applied in general, because there are transformations (i.e. *CNF-transformations*[2]) that transform any formula into a satisfiability-equivalent set of clauses. These algorithms will not be described in this thesis, however, as they are not necessary for cut-elimination by resolution, which extracts clause sets directly from a sequent calculus proof.

Definition 2.3.1 (Clause)**.** A *clause* is a sequent in which all formulas are atomic[o].

Definition 2.3.2 (Generality (for Expressions))**.** An expression (term, formula, clause, sequent) e_1 is *more general than (or equally general to)* an expression e_2, denoted $e_1 \leq_s e_2$, if and only if there is a substitution τ such that $(e_1)\tau = e_2$[p]. A substitution σ_1 is *more general than (or equally general to)* a substitution σ_2 (denoted $\sigma_1 \leq_s \sigma_2$) if and only if there is a substitution τ such that, for all expressions t, $(t\sigma_1)\tau = t\sigma_2$.

Definition 2.3.3 ((Most General) Unifier)**.** Let F_1 and F_2 be formulas. If there is a substitution σ such that $F_1\sigma = F_2\sigma$, then F_1 and F_2 are said to be *unifiable* and σ is called a *unifier* for F_1 and F_2. A unifier σ of F_1 and F_2 is a *most general unifier* if and only if, for any unifier σ' of F_1 and F_2, $\sigma \leq_s \sigma'$.

Definition 2.3.4 (Inference Rules (of the Resolution Calculus))**.** The *inference rules* of the Resolution Calculus **R** are shown below:

- **Initial Rule**:

$$\frac{}{\Gamma \vdash \Delta}$$

- **Resolution Rule**[q]:

$$\frac{\Gamma_1 \vdash \Delta_1, A_1 \qquad A_2, \Gamma_2 \vdash \Delta_2}{\Gamma_1\sigma\eta, \Gamma_2\sigma\eta \vdash \Delta_1\sigma\eta, \Delta_2\sigma\eta}\, r(\sigma)$$

  where $\Gamma_1 \vdash \Delta_1, A_1$ and $A_2, \Gamma_2 \vdash \Delta_2$ do not have any variable in common, σ is a most general unifier of A_1 and A_2, and η is a substitution that renames all variables to globally new fresh ones. The conclusion of a resolution rule is called a *resolvent* of its premises.

[o]Here the definition of clause is based on that of sequent. The reason for this choice is that the similarity between the cut and resolution rules and the possibility of using a resolution refutation as a skeleton for a sequent calculus proof (as described in Chapter 4) become more apparent. However, in the automated deduction community, clauses are more frequently defined as either sets or multisets or disjunctions (lists) of *literals*. A comparison of different definitions can be seen in [80].

[p]Here it is assumed that the definition of substitutions can be extended in such a way that they can be applied to any kind of expression possibly containing variables, and not just to formulas as in Definition 2.2.3.

[q]The resolution rule is very similar to the cut rule. Indeed, if σ is the identity substitution, then the resolution rule is essentially just a cut rule with an atomic cut-formula. Analogously, the factoring rules are very similar to the contraction rules. The resolution rule is more commonly defined (e.g. [80]) in a way that implicitly incorporates factoring. The resolution rule described here, on the other hand, corresponds to the simpler *binary resolution rule* described in [80], and is more convenient in the context of this thesis exactly because the correspondence between cut and resolution rules and contraction and factoring rules becomes simpler.

- **Factoring Rules**:

$$\frac{A, A', \Gamma \vdash \Delta}{A\sigma\eta, \Gamma\sigma\eta \vdash \Delta\sigma\eta}\ f_l(\sigma) \qquad\qquad \frac{\Gamma \vdash \Delta, A, A'}{\Gamma\sigma\eta \vdash \Delta\sigma\eta, A\sigma\eta}\ f_r(\sigma)$$

  where σ is a most general unifier of A and A' and η is a substitution that renames all variables to globally new fresh ones. The conclusion of factoring rule is called a *factor* of its premise.

If the rules above are restricted to having the empty substitution, then they are called *propositional* resolution and factoring rules (pr, pf_l, pf_r).

Definition 2.3.5 (Deductions and Refutations)**.** A *resolution deduction* of a clause $\Gamma \vdash \Delta$ from a set of clauses C is a tree[r] of resolution inferences (i.e. instances of the inference rules of the Resolution Calculus), in which every premise clause of an inference is the conclusion clause of another inference, and the conclusion clauses of initial inferences are instances of clauses of S. A *resolution refutation* of a set of clauses C is a resolution deduction of the empty clause (i.e. $\vdash$) from C. A resolution deduction containing only pr, pf_l and pf_r inferences is called a *propositional resolution deduction*. A *ground resolution deduction* is a propositional resolution deduction in which no variables occur and all terms are built from the function symbols occurring in C (i.e. all terms belong to Herbrand universe of the signature of C).

Definition 2.3.6 (Generality (for Resolution Deductions))**.** Let δ and δ' be resolution deductions of $\Gamma \vdash \Delta$ and $\Gamma' \vdash \Delta'$ respectively. δ is *more general than (or equally general to)* δ', denoted $\delta \leq_s \delta'$, if and only if $\Gamma \vdash \Delta \leq_s \Gamma' \vdash \Delta'$ and, if δ_1 and δ_2 are immediate subproofs of δ and δ'_1 and δ'_2 are immediate subproofs of δ', then $\delta_1 \leq_s \delta'_1$ and $\delta_2 \leq_s \delta'_2$.

Theorem 2.3 (Refutational Completeness of the Resolution Calculus)**.** If C is an unsatisfiable set of clauses (i.e. if there is no interpretation under which all clauses of C are true), then there exists a resolution refutation of C.

[r] It is also common, indeed even more common, to define resolution deductions as lists or directed acyclic graphs. It is more convenient for proof search and deductions are smaller (e.g. the size of a refutation of $\{\vdash P(x); P(x) \vdash P(f(x)); P(f^{2^n}(x)) \vdash\}$ is necessarily $O[2^n]$ if the tree format is used, while it can be $O[n]$ if lists or directed acyclic graphs are used. The reason why trees are used here is that the method of cut-elimination by resolution needs refutations in the tree format, so that they can be converted to a sequent calculus proof with atomic cuts.

Proof. A detailed proof can be found in [80]. It proves firstly a restricted form of the theorem, for sets of ground clauses and, correspondingly, ground resolution refutations. This proof relies on the notion of *semantic tree*, which is very similar to a tableau calculus [30] having analytic atomic cut as its only rule. In this sense, the proof can be seen as a relative proof of completeness with respect to the completeness of this semantic tree or tableau calculus. The completeness of this semantic tree calculus is proved semantically, resorting to Herbrand interpretations. The *finite* completeness (i.e. that an unsatisfiable set of clauses has a finite closed semantic tree) depends on *Koenig's lemma* [108, 77], a weaker form of the *axiom of choice* [108, 119], and is closely related to the property of *compactness* [32] that holds for predicate logic.

The general theorem is then obtained by *lifting* the ground case with the *lifting theorem* (Theorem 2.4).

It is also possible to prove the relative completeness of resolution with respect to the completeness of sequent calculus [39, 3]. □

Lemma 2.1 (Lifting Lemma for the Factoring Rules)**.** Let $\Gamma \vdash \Delta$ and $\Gamma' \vdash \Delta'$ be clauses such that $\Gamma \vdash \Delta \leq_s \Gamma' \vdash \Delta'$. If $\Gamma'_f \vdash \Delta'_f$ is a factor of $\Gamma' \vdash \Delta'$, then there exists a factor $\Gamma_f \vdash \Delta_f$ of $\Gamma \vdash \Delta$ such that $\Gamma_f \vdash \Delta_f \leq_s \Gamma'_f \vdash \Delta'_f$.

Proof. A detailed proof is available in [80]. It consists basically of constructing $\Gamma_f \vdash \Delta_f$ from $\Gamma'_f \vdash \Delta'_f$ by using, in a reversed way, the substitutions that are given by the assumptions that $\Gamma \vdash \Delta \leq_s \Gamma' \vdash \Delta'$ and that $\Gamma'_f \vdash \Delta'_f$ is a factor of $\Gamma' \vdash \Delta'$. It is interesting to note that factors, as defined here, are necessarily factors of degree 1, as defined in [80], and hence induction is not necessary in this case. □

Lemma 2.2 (Lifting Lemma for the Resolution Rule)**.** Let $\Gamma_1 \vdash \Delta_1$, $\Gamma_2 \vdash \Delta_2$, $\Gamma'_1 \vdash \Delta'_1$, $\Gamma'_2 \vdash \Delta'_2$ be clauses such that $\Gamma_1 \vdash \Delta_1 \leq_s \Gamma'_1 \vdash \Delta'_1$ and $\Gamma_2 \vdash \Delta_2 \leq_s \Gamma'_2 \vdash \Delta'_2$. If $\Gamma'_r \vdash \Delta'_r$ is a resolvent of $\Gamma'_1 \vdash \Delta'_1$ and $\Gamma'_2 \vdash \Delta'_2$, then there exists a resolvent $\Gamma_r \vdash \Delta_r$ of $\Gamma_1 \vdash \Delta_1$ and $\Gamma_2 \vdash \Delta_2$ such that $\Gamma_r \vdash \Delta_r \leq_s \Gamma'_r \vdash \Delta'_r$.

Proof. A detailed proof is available in [80]. It is analogous to the proof of Lemma 2.1. □

Theorem 2.4 (Lifting Theorem)**.** Let C and C' be sets of clauses such that $C \leq_s C'$[s]. Let δ' be a resolution deduction from C'. Then there exists a resolution deduction δ from C such that $\delta \leq_s \delta'$.

[s] $C \leq_s C'$ if and only if for every clause c' of C' there is a clause c of C such that $c \leq_s c'$

Proof. A detailed proof is available in [80]. It is a proof by induction on the size of the deductions, where the base case is trivial, and the inductive case can be proved with the lifting Lemmas 2.1 and 2.2. □

Definition 2.3.7 (Resolution Refinement). A *resolution refinement*[t] replaces the rules $r(\sigma)$, $f_l(\sigma)$ and $f_r(\sigma)$ by refined rules $r_x(\sigma)$, $f_{x_l}(\sigma)$ and $f_{x_r}(\sigma)$ such that the refined rules are more restricted than the original rules, i.e., if c_r is a resolvent of c_1 and c_2 according to $r_x(\sigma)$, then c_r is also a resolvent of c_1 and c_2 according to $r(\sigma)$, and if c_f is a factor of c according to $f_{x_r}(\sigma)$ or $f_{x_l}(\sigma)$, then c_f is also a factor of c according to $f_r(\sigma)$ or $f_l(\sigma)$, but the converses do not hold.

[t]Definition 2.3.7 of resolution refinement is not as general as that in [80]. It roughly coincides with the definition of resolution refinement operator given in [80]. There are many well known refinements, such as linear resolution and hyper-resolution, that are not (at least not without an unnatural use of annotations and labels) considered refinements by the definition given here. Nevertheless, Definition 2.3.7 is sufficient for the refinements that are described in Chapter 6.

Chapter 3

Cut-Elimination

Simultaneously with the invention of sequent calculus, Gentzen proved the *cut-elimination theorem*[a] [48], which states that, if a sequent is the end-sequent of a proof φ with cuts, then there is a proof φ' without cuts of the same end-sequent; or, in other words, sequent calculus without the cut rule is still complete. Moreover, his proof not only proved the existence of such a proof φ', but described a constructive method based on proof rewriting rules to eliminate the cuts from any proof φ, transforming it into a proof φ' without cuts.

This chapter starts with a general discussion of the importance of cut-elimination for logic (Section 3.1) and ends with a detailed description of proof rewriting rules for cut-elimination in the spirit of Gentzen's (Section 3.2).

3.1 The Importance of Cut-Elimination for Logic

The cut-elimination theorem plays a very significant role in logic, because it has many important corollaries. Firstly, any cut-free proof φ' has the *subformula property*: the formulas occurring in the premises of an inference of φ' are subformulas of formulas occurring in the conclusion sequent of the inference. This allows many advances in automated deduction, such as relatively easy bottom-up (backward, goal-oriented) proof search in automated deduction (as new formulas do not have to be guessed[b]), analytic tableau [43, 30], Maslov's inverse method [35] and

[a]The cut-elimination theorem is also known as Gentzen's *Hauptsatz*.

[b]Note, however, that in the case of weak quantifier rules, an instance for the variable still has to be guessed from an infinite number of potential terms. The superiority of the resolution calculus

logic programming [110].

Moreover, the subformula property also leads to the major theoretical application of the cut-elimination theorem: *consistency proofs*. For a theory is trivially consistent if all its theorems can be derived by a sequent calculus where cut is eliminable, simply because there is no cut-free proof of the empty sequent. Consistency proofs were, in fact, the main motivation for the development of sequent calculus, and cut-elimination is therefore one approach within Hilbert's program [74].

Another interesting consequence of the cut-elimination theorem is derived from the fact that cut-free proofs allow the construction of interpolants of their end-sequents via *Maehara's lemma* [112, 9], by induction on the structures of the proofs. In this sense, the cut-elimination theorem and Maehara's lemma provide a purely syntactical and constructive proof of *Craig's interpolation theorem* [27]. Moreover, a corollary, *Beth's definability theorem*, can also be given a constructive proof: if a predicate symbol is defined implicitly, then cut-elimination and the method of Maehara's lemma can be used to construct an explicit definition [9].

Not only interpolants but also *Herbrand disjunctions* can be easily constructed or extracted from cut-free proofs. This has been shown in Gentzen's *Midsequent theorem*[c][48, 49] for the prenex case and extensions to more general cases have been studied and compared in [94, 95]. Therefore, from a certain point of view, the midsequent theorem can be seen as the missing link relating the cut-elimination theorem with *Herbrand's theorem* [64, 62, 63, 21]. Indeed, Herbrand's theorem preceded the cut-elimination theorem and was hard to prove without it, but relatively easier proofs can be obtained by using cut-elimination as a lemma and then using either the mid-sequent theorem, for the prenex case, or arguments like the ones shown in [22, 85, 95], for more general cases.

If sequent calculus is seen as a meta-calculus about proofs in another calculus (e.g. natural deduction [48]), then the cut rule essentially says that the proofs denoted by the premise sequents can be composed, resulting in a proof denoted by the conclusion sequent. But it does not say how the composition has to be done. Cut-elimination methods can therefore be seen as processes that actually do the composition of the proofs. In certain areas of logic, such as that which relates proofs to functional pro-

over the standard sequent calculus for proof search lies exactly on the fact that resolution restricts instantiation to most general unifiers, while the standard sequent calculus does not. There are, however, analytic tableaux and hence sequent calculi that incorporate most general unification in proof search.

[c]The midsequent theorem is also known as Gentzen's *sharpened Hauptsatz*

grams via the *Curry-Howard isomorphism* [28, 29, 33], it may be important that this composition be deterministic and unique, because the composition (cut-elimination) of proofs corresponds to the computation of the corresponding programs. This requirement is equivalent to demanding that methods of cut-elimination should be confluent.

The desire to have minimalistic proof representations (e.g. generalized lambda terms [96], proof nets [53], short tautologies [68], Deep Inference (formalism A, atomic flows and deductive nets) [60, 18, 19], Herbrand sequents [95, 94, 70] , logical flow graphs [20, 24], . . .) that are "free of bureaucracy" is equivalent to the question of identity[d] of proofs [111, 78] ("when should two proofs be considered the same?"), for bureaucracy can be defined as all the syntactic differences between two proofs that should be considered the same. This gives rise to the approach of *semantics of proofs* [78], that attempts to assign meanings to proofs and then answer the question of the identity of proofs affirmatively if the proofs have the same meaning. This includes assigning meanings to proofs with cuts, and hence it might again be relevant to enforce confluence of cut-elimination, as it is usually desirable to have the same meaning for a proof with cuts and its cut-free normal forms and this might be harder to achieve if there are too many cut-free normal forms.

Even though cut-elimination and many other proof-theoretical techniques had been developed for purely foundational reasons and thus were intended to be used only in an abstract sense, later they came to be applied to actual mathematical proofs. Girard's informal application of cut-elimination to eliminate the lemmas of a mathematical proof by Fuerstenberg and Weiss of van der Waerden's theorem, resulting in van der Waerden's original mathematical proof, is one of the most famous examples [54]. Luckhardt's extraction of bounds for the number of solutions of certain Diophantine equations from proofs of Roth's theorem by analyzing the terms of Herbrand disjunctions extractable from these proofs is another impressive example, especially considering that the achievement

[d]By Leibniz's law of the identity of indiscernibles [79, 89] ($\forall x \forall y (\forall P(P(x) \leftrightarrow P(y)) \rightarrow x = y)$) applied to proofs, and by the adoption of Henkin semantics [61] for second-order quantification, it is clear that the question of the identity of proofs is only well-defined if the domain over which the variable P is quantified is precisely fixed, which is frequently not the case. Leibniz's law also suggests that the identity of proofs can only be assessed relative to a domain of properties of proofs that are of interest for a given application (e.g. the property of "proving the same theorem" might be sufficient to classify two proofs as the same, if the only interest is to know whether the theorem holds, while the additional interest of analyzing the mathematical content of a proof could require additional properties such as "having the same Herbrand sequent"). Searching for an absolute concept of proof identity is futile.

of such bounds by standard number-theoretic techniques occurred only later and with much more effort [16].

Cut-elimination is also related to a principle of parsimony and simplicity popularly known as *Ockham's razor* [87], according to which simpler explanations should be preferred over more complex ones. A proof with non-analytic cuts does not satisfy the subformula property, and hence contains superfluous concepts, which, as shown by the cut-elimination theorem, are not really necessary to derive the end-sequent. Therefore, in a qualitative sense of Ockham's razor, cut-free proofs are simpler and ought to be preferred over proofs with cuts. Therefore, methods of cut-elimination ought to be seen as means to achieve the simplicity demanded by Ockham's razor in the specific area of logic. On the other hand, cut-free proofs can be significantly larger than proofs with cuts [109, 93]. Therefore, according to a purely quantitative Ockham's razor principle, cut-elimination ought to be avoided. The phenomenon of cut-elimination is then almost a paradox for the Ockham's razor principle. It shows the relativity of simplicity in a very simple way, by exhibiting an intrinsic trade-off between the qualitative and quantitative kinds of simplicity. Surprisingly, though, cut-elimination seems to have not yet been a case studied by philosophers concerned with issues of this popular principle.

3.1.1 Related Approaches

While sequent calculus and cut-elimination were Gentzen's late approach to Hilbert's program, Hilbert himself had originally chosen a different approach, based on the *epsilon calculus* [72] and the *epsilon substitution method* [1, 90]. Particularly, both a cut-free proof in the sequent calculus and a proof without epsilon-terms in the epsilon calculus allow easy extraction of Herbrand disjunctions [91] . Historically interesting is also the fact that the epsilon calculus preceded Herbrand's theorem by a few years, and later enabled the first correct proof of Herbrand's theorem to be obtained as a corollary of an extended version of the first epsilon theorem [91, 73].

Yet another approach to proving consistency is Goedel's "Dialectica" method of functional interpretation [55]. Interestingly, the focus of this method (and its variants, such as the refined A-translation [15, 113]) also migrated from foundational issues to applications on actual mathematical proofs [76, 113]. Indeed, not only "programs" (functionals) can be extracted from proofs in this way, but also Herbrand disjunctions [51], since the functional interpretation essentially encodes all the terms of a

Herbrand disjunction into recursive functionals.

3.2 Reductive Cut-elimination Proof Rewriting Rules

It is evident that a cut can be easily eliminated if it occurs immediately below an axiom inference. This can be done by rewriting the proof according to the rewriting rule shown in Definition 3.2.6. This leads to the following idea for the elimination of cuts in general: swap the cut inferences upward, until they are immediately below axiom inferences, where they can easily be eliminated. This swapping can be done according to the rewriting rules of Definition 3.2.9. However, if the cut-formula occurrence is not atomic, the swapping will eventually be blocked when the cut occurs immediately below the inferences that introduce both its cut-formula occurrences. When this happens, the cut (and the inferences immediately above it) can be replaced by new cuts such that their cut-formula occurrences are proper subformulas of the cut-formula of the replaced cut. As the blocking inferences are also removed by the replacement, it is clear that the upward swapping can proceed. This replacement can be done according to Definition 3.2.14, if the cut-formula has a propositional connective in its shallowest level, or Definition 3.2.17, if the cut-formula has a quantifier in its shallowest level. Structural inferences above cuts can also block their upward swapping. In the case of weakening, the cut can be simply eliminated, analogously to the case for axiom inferences, according to Definition 3.2.19. The case for contraction is more complicated, since it requires the duplication of some subproofs, as shown in Definition 3.2.20.

Historically, cut-elimination methods based on proof rewriting methods have been known as *reductive cut-elimination methods*, because the upward swapping of cuts reduces its *rank* (Definition 3.2.9) and the replacement of a cut by others obviously (except in the case of contraction) reduces the overall complexity (*grade* (e.g. Definitions 3.2.14 and 3.2.17)) of the cut-formulas of the proof.

As usual, rewriting rules are seen here as abstract definitions of compatible closed relations over the set of proofs. Infix notation is used: if φ can be rewritten to φ', then $\varphi \rhd_x \varphi'$ (meaning $(\varphi, \varphi') \in \rhd_x$).

3.2.1 Preliminaries about Rewriting Relations

The following definitions are standard from rewriting systems. They are heavily used in all subsequent chapters.

Definition 3.2.1 (Transitive Closure)**.** Let $\blacktriangleright$ be a rewriting relation. Then:

- $\varphi \blacktriangleright^n \varphi_n$ if and only if there are proofs $\varphi_1, \ldots, \varphi_{n-1}$ such that $\varphi \blacktriangleright \varphi_1$, $\varphi_1 \blacktriangleright \varphi_2, \ldots, \varphi_{n-1} \blacktriangleright \varphi_n$.
- $\varphi \blacktriangleright^* \varphi'$ if and only if there is a natural number n such that $\varphi \blacktriangleright^n \varphi_n$.

$\blacktriangleright^*$ is the *reflexive transitive closure* of $\blacktriangleright$.

Definition 3.2.2 (Normal form)**.** Let $\blacktriangleright$ be a rewriting relation on proofs. A proof φ is a *$\blacktriangleright$-normal-form* if and only if $\varphi \blacktriangleright \varphi'$ for no proof φ'.

A proof φ' is a *$\blacktriangleright$-normal-form of* φ if and only if $\varphi \blacktriangleright^* \varphi'$ and φ' is a $\blacktriangleright$-normal-form.

Definition 3.2.3 (Normalization Relations)**.** Let $\blacktriangleright$ be a rewriting relation on proofs. Then $\blacktriangleright^{\downarrow}$ is the sub-relation of $\blacktriangleright^*$ such that $\varphi \blacktriangleright^{\downarrow} \psi$ if and only if $\varphi \blacktriangleright^* \psi$ and ψ is a $\blacktriangleright$-normal-form.

Definition 3.2.4 (Confluence)**.** A rewriting relation $\blacktriangleright$ is *confluent* if and only if every proof φ has only one $\blacktriangleright$-normal-form.

Definition 3.2.5 (Normalization)**.** A rewriting relation $\blacktriangleright$ is *weakly normalizing* if and only if, for every proof φ, there is a proof φ' such that $\varphi \blacktriangleright^* \varphi'$ and φ' is a $\blacktriangleright$-normal-form.

A rewriting relation $\blacktriangleright$ is *strongly normalizing* if and only if, for every proof φ_0, there is no infinite sequence of proofs $(\varphi_1, \varphi_2, \ldots)$ such that $\varphi_n \blacktriangleright \varphi_{n+1}$ (for all $n \geq 0$).

3.2.2 The Rewriting Rules

Definition 3.2.6 ($\triangleright_a$)**.** Cut-elimination over axiom inferences:

$$\dfrac{A \vdash A \qquad \overset{\varphi_r}{A, \Pi \vdash \Lambda}}{A, \Pi \vdash \Lambda}\,cut \qquad\qquad \dfrac{\overset{\varphi_l}{\Gamma \vdash \Delta, A} \qquad A \vdash A}{\Gamma, \vdash \Delta, A}\,cut$$

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

$$\overset{\varphi_r}{A, \Pi \vdash \Lambda} \qquad\qquad\qquad\qquad \overset{\varphi_l}{\Gamma \vdash \Delta, A}$$

Definition 3.2.7 ($\triangleright_{r_1}$)**.** Upward swapping of cuts over unary inferences (unary rank reduction):

$$\dfrac{\overset{\varphi_l}{\Gamma \vdash \Delta, A} \qquad \dfrac{\overset{\varphi_r}{A, \Pi' \vdash \Lambda'}}{A, \Pi \vdash \Lambda}\,\rho}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut \qquad\qquad \dfrac{\dfrac{\overset{\varphi_l}{\Gamma' \vdash \Delta', A}}{\Gamma \vdash \Delta, A}\,\rho \qquad \overset{\varphi_r}{A, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut$$

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

$$\dfrac{\dfrac{\overset{\varphi_l}{\Gamma \vdash \Delta, A} \qquad \overset{\varphi_r}{A, \Pi' \vdash \Lambda'}}{\Gamma, \Pi' \vdash \Delta, \Lambda'}\,cut}{\Gamma, \Pi \vdash \Delta, \Lambda}\,\rho \qquad\qquad \dfrac{\dfrac{\overset{\varphi_l}{\Gamma' \vdash \Delta', A} \qquad \overset{\varphi_r}{A, \Pi \vdash \Lambda}}{\Gamma', \Pi \vdash \Delta', \Lambda}\,cut}{\Gamma, \Pi \vdash \Delta, \Lambda}\,\rho$$

Definition 3.2.8 ($\triangleright_{r_2}$)**.** Upward swapping of cuts over binary inferences (binary rank reduction):

$$\dfrac{\overset{\varphi_l}{\Pi \vdash \Lambda, A} \qquad \dfrac{\overset{\varphi_1}{A, \Gamma_1 \vdash \Delta_1} \qquad \overset{\varphi_2}{\Gamma_2 \vdash \Delta_2}}{A, \Gamma \vdash \Delta}\,\rho}{\Pi, \Gamma \vdash \Lambda, \Delta}\,cut \qquad\qquad \dfrac{\overset{\varphi_l}{\Pi \vdash \Lambda, A} \qquad \dfrac{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1} \qquad \overset{\varphi_2}{A, \Gamma_2 \vdash \Delta_2}}{A, \Gamma \vdash \Delta}\,\rho}{\Pi, \Gamma \vdash \Lambda, \Delta}\,cut$$

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

$$\dfrac{\dfrac{\overset{\varphi_l}{\Pi \vdash \Lambda, A} \qquad \overset{\varphi_1}{A, \Gamma_1 \vdash \Delta_1}}{\Pi, \Gamma_1 \vdash \Lambda, \Delta_1}\,cut \qquad \overset{\varphi_2}{\Gamma_2 \vdash \Delta_2}}{\Pi, \Gamma \vdash \Lambda, \Delta}\,\rho \qquad\qquad \dfrac{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1} \qquad \dfrac{\overset{\varphi_l}{\Pi \vdash \Lambda, A} \qquad \overset{\varphi_2}{A, \Gamma_2 \vdash \Delta_2}}{\Pi, \Gamma_2 \vdash \Lambda, \Delta_2}\,cut}{\Pi, \Gamma \vdash \Lambda, \Delta}\,\rho$$

$$
\dfrac{\dfrac{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1, A} \qquad \overset{\varphi_2}{\Gamma_2 \vdash \Delta_2}}{\Gamma \vdash \Delta, A}\,\rho \qquad \overset{\varphi_r}{A, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut
\qquad
\dfrac{\dfrac{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1} \qquad \overset{\varphi_2}{\Gamma_2 \vdash \Delta_2, A}}{\Gamma \vdash \Delta, A}\,\rho \qquad \overset{\varphi_r}{A, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut
$$

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

$$
\dfrac{\dfrac{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1, A} \qquad \overset{\varphi_r}{A, \Pi \vdash \Lambda}}{\Gamma_1, \Pi \vdash \Delta_1, \Lambda}\,cut \qquad \overset{\varphi_2}{\Gamma_2 \vdash \Delta_2}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,\rho
\qquad
\dfrac{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1} \qquad \dfrac{\overset{\varphi_2}{\Gamma_2 \vdash \Delta_2, A} \qquad \overset{\varphi_r}{A, \Pi \vdash \Lambda}}{\Gamma_2, \Pi \vdash \Delta_2, \Lambda}\,cut}{\Gamma, \Pi \vdash \Delta, \Lambda}\,\rho
$$

Definition 3.2.9 ($\rhd_r$)**.** Upward swapping of cuts (rank reduction):

$$
\rhd_r \doteq \rhd_{r_1} \cup \rhd_{r_2}
$$

$\rhd_{r_{non-atomic}}$ is the sub-relation of $\rhd_r$ obtained by restricting it to non-atomic cut-formulas.

Definition 3.2.10 ($\rhd_{p_\wedge}$)**.** Reduction of complexity of a cut-formula having $\wedge$ as shallowest connective (grade reduction):

$$
\dfrac{\dfrac{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1, B} \qquad \overset{\varphi_2}{\Gamma_2 \vdash \Delta_2, C}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, B \wedge C}\,\wedge_r \qquad \dfrac{\overset{\varphi_r}{B, C, \Pi \vdash \Lambda}}{B \wedge C, \Pi \vdash \Lambda}\,\wedge_l}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda}\,cut
$$

$$
\Downarrow
$$

$$
\dfrac{\overset{\varphi_2}{\Gamma_2 \vdash \Delta_2, C} \qquad \dfrac{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1, B} \qquad \overset{\varphi_r}{B, C, \Pi \vdash \Lambda}}{C, \Gamma_1, \Pi \vdash \Delta_1, \Lambda}\,cut}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda}\,cut
$$

Definition 3.2.11 ($\rhd_{p_\vee}$). Reduction of complexity of a cut-formula having $\vee$ as shallowest connective (grade reduction):

$$\dfrac{\dfrac{\stackrel{\varphi_l}{\Pi \vdash \Lambda, B, C}}{\Pi \vdash \Lambda, B \vee C}\,\vee_r \qquad \dfrac{\stackrel{\varphi_1}{B, \Gamma_1 \vdash \Delta_1} \qquad \stackrel{\varphi_2}{C, \Gamma_2 \vdash \Delta_2}}{B \vee C, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\,\vee_l}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda}\,cut$$

$$\Downarrow$$

$$\dfrac{\dfrac{\stackrel{\varphi_l}{\Pi \vdash \Lambda, B, C} \qquad \stackrel{\varphi_2}{C, \Gamma_2 \vdash \Delta_2}}{\Pi, \Gamma_2 \vdash \Delta_2, \Lambda, B}\,cut \qquad \stackrel{\varphi_1}{B, \Gamma_1 \vdash \Delta_1}}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda}\,cut$$

Definition 3.2.12 ($\rhd_{p_\rightarrow}$). Reduction of complexity of a cut-formula having $\rightarrow$ as shallowest connective (grade reduction):

$$\dfrac{\dfrac{\stackrel{\varphi_l}{B, \Pi \vdash \Lambda, C}}{\Pi \vdash \Lambda, B \rightarrow C}\,\rightarrow_r \qquad \dfrac{\stackrel{\varphi_1}{\Gamma_1 \vdash \Delta_1, B} \qquad \stackrel{\varphi_2}{C, \Gamma_2 \vdash \Delta_2}}{B \rightarrow C, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\,\rightarrow_l}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda}\,cut$$

$$\Downarrow$$

$$\dfrac{\stackrel{\varphi_1}{\Gamma_1 \vdash \Delta_1, B} \qquad \dfrac{\stackrel{\varphi_l}{B, \Pi \vdash \Lambda, C} \qquad \stackrel{\varphi_2}{C, \Gamma_2 \vdash \Delta_2}}{B, \Pi, \Gamma_2 \vdash \Delta_2, \Lambda}\,cut}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda}\,cut$$

Definition 3.2.13 ($\rhd_{p_\neg}$). Reduction of complexity of a cut-formula having $\neg$ as shallowest connective (grade reduction):

$$\dfrac{\dfrac{\stackrel{\varphi_l}{B, \Gamma \vdash \Delta}}{\Gamma \vdash \Delta, \neg B}\,\neg_r \qquad \dfrac{\stackrel{\varphi_r}{\Pi \vdash \Lambda, B}}{\neg B, \Pi \vdash \Lambda}\,\neg_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut$$

$$\Downarrow$$

$$\dfrac{\stackrel{\varphi_r}{\Pi \vdash \Lambda, B} \qquad \stackrel{\varphi_l}{B, \Gamma \vdash \Delta}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut$$

Definition 3.2.14 ($\rhd_p$)**.** Propositional grade reduction:

$$\rhd_p \doteq \rhd_{p_\wedge} \cup \rhd_{p_\vee} \cup \rhd_{p_\rightarrow} \cup \rhd_{p_\neg}$$

Definition 3.2.15 ($\rhd_{q_\forall}$)**.** Reduction of complexity of a cut-formula having a universal quantifier at its shallowest level:

$$\cfrac{\cfrac{\overset{\varphi_l}{\Gamma \vdash \Delta, B\{x \leftarrow \alpha\}}}{\Gamma \vdash \Delta, \forall x B}\,\forall_r \qquad \cfrac{\overset{\varphi_r}{B\{x \leftarrow t\}, \Pi \vdash \Lambda}}{\forall x B, \Pi \vdash \Lambda}\,\forall_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut$$

$$\Downarrow$$

$$\cfrac{\overset{\varphi_l'\{\alpha \leftarrow t\}}{\Gamma \vdash \Delta, B\{x \leftarrow t\}} \qquad \overset{\varphi_r}{B\{x \leftarrow t\}, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut$$

where φ_l' is obtained from φ_l by renaming all bound variables to globally fresh new ones, so that they are not equal to any free variable in the term t.

Definition 3.2.16 ($\rhd_{q_\exists}$)**.** Reduction of complexity of a cut-formula having an existential quantifier at its shallowest level:

$$\cfrac{\cfrac{\overset{\varphi_l}{\Gamma \vdash \Delta, B\{x \leftarrow t\}}}{\Gamma \vdash \Delta, \exists x B}\,\exists_r \qquad \cfrac{\overset{\varphi_r}{B\{x \leftarrow \alpha\}, \Pi \vdash \Lambda}}{\exists x B, \Pi \vdash \Lambda}\,\exists_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut$$

$$\Downarrow$$

$$\cfrac{\overset{\varphi_l}{\Gamma \vdash \Delta, B\{x \leftarrow t\}} \qquad \overset{\varphi_r'\{\alpha \leftarrow t\}}{B\{x \leftarrow t\}, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut$$

where φ_r' is obtained from φ_r by renaming all bound variables to globally fresh new ones, so that they are not equal to any free variable in the term t.

Definition 3.2.17 ($\rhd_q$)**.** Quantificational grade reduction:

$$\rhd_q \doteq \rhd_{q_\forall} \cup \rhd_{q_\exists}$$

Definition 3.2.18 ($\rhd_d$)**.** Reduction over definition inferences:

$$\dfrac{\dfrac{\overset{\varphi_l}{\Gamma \vdash \Delta, A[x_1,\ldots,x_n]}}{\Gamma \vdash \Delta, P(x_1,\ldots,x_n)}\,d_r \qquad \dfrac{\overset{\varphi_r}{A[x_1,\ldots,x_n],\Pi \vdash \Lambda}}{P(x_1,\ldots,x_n),\Pi \vdash \Lambda}\,d_l}{\Gamma,\Pi \vdash \Delta,\Lambda}\,cut$$

$$\Downarrow$$

$$\dfrac{\overset{\varphi_l}{\Gamma \vdash \Delta, A[x_1,\ldots,x_n]} \qquad \overset{\varphi_r}{A[x_1,\ldots,x_n],\Pi \vdash \Lambda}}{\Gamma,\Pi \vdash \Delta,\Lambda}\,cut$$

Definition 3.2.19 ($\rhd_w$)**.** Cut-elimination over weakening inferences:

$$\dfrac{\dfrac{\overset{\varphi_l}{\Gamma \vdash \Delta}}{\Gamma \vdash \Delta, A}\,w_r \qquad \overset{\varphi_r}{A,\Pi \vdash \Lambda}}{\Gamma,\Pi \vdash \Delta,\Lambda}\,cut \qquad\qquad \dfrac{\overset{\varphi_l}{\Gamma \vdash \Delta, A} \qquad \dfrac{\overset{\varphi_r}{\Pi \vdash \Lambda}}{A,\Pi \vdash \Lambda}\,w_l}{\Gamma,\Pi \vdash \Delta,\Lambda}\,cut$$

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

$$\dfrac{\overset{\varphi_l}{\Gamma \vdash \Delta}}{\Gamma,\Pi \vdash \Delta,\Lambda}\,w_r^*, w_l^* \qquad\qquad \dfrac{\overset{\varphi_r}{\Pi \vdash \Lambda}}{\Gamma,\Pi \vdash \Delta,\Lambda}\,w_r^*, w_l^*$$

$\rhd_{w_{non-atomic}}$ is the sub-relation of $\rhd_w$ obtained by restricting it to non-atomic cut-formulas.

Definition 3.2.20 ($\rhd_c$)**.** Duplication of cuts over contraction inferences[e]:

$$\dfrac{\dfrac{\overset{\varphi_l}{\Gamma \vdash \Delta, A, A}}{\Gamma \vdash \Delta, A}\,c_r \qquad \overset{\varphi_r}{A,\Pi \vdash \Lambda}}{\Gamma,\Pi \vdash \Delta,\Lambda}\,cut \qquad\qquad \dfrac{\overset{\varphi_l}{\Gamma \vdash \Delta, A} \qquad \dfrac{\overset{\varphi_r}{A,A,\Pi \vdash \Lambda}}{A,\Pi \vdash \Lambda}\,c_l}{\Gamma,\Pi \vdash \Delta,\Lambda}\,cut$$

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

$$\dfrac{\dfrac{\dfrac{\overset{\varphi_l}{\Gamma \vdash \Delta, A, A} \qquad \overset{\varphi_r}{A,\Pi \vdash \Lambda}}{\Gamma,\Pi \vdash \Delta,\Lambda,A}\,cut \qquad \overset{\varphi_r'}{A,\Pi \vdash \Lambda}}{\Gamma,\Pi,\Pi \vdash \Delta,\Lambda,\Lambda,}\,cut}{\Gamma,\Pi \vdash \Delta,\Lambda}\,c_l^*, c_r^* \qquad \dfrac{\dfrac{\overset{\varphi_l'}{\Gamma \vdash \Delta, A} \qquad \dfrac{\overset{\varphi_l}{\Gamma \vdash \Delta, A} \qquad \overset{\varphi_r}{A,A,\Pi \vdash \Lambda}}{A,\Gamma,\Pi \vdash \Delta,\Lambda}\,cut}{\Gamma,\Gamma,\Pi \vdash \Delta,\Delta,\Lambda}\,cut}{\Gamma,\Pi \vdash \Delta,\Lambda}\,c_l^*, c_r^*$$

[e] The use of a purely multiplicative calculus, in which all contractions occur explicitly via contraction inferences, allows the isolation of the phenomenon of duplication of subproofs (and the need for renaming of eigenvariables) to the case of cut-reduction over contractions. Had an additive or mixed calculus been chosen, implicit contractions would occur, and its treatment would not be as transparent.

where φ'_l and φ'_r are variants of, respectively, φ_l and φ_r, in which the eigenvariables are renamed to preserve proof regularity.

$\rhd_{c_{non-atomic}}$ is the sub-relation of $\rhd_c$ obtained by restricting it to non-atomic cut-formulas.

Definition 3.2.21 ($\rhd_{\tilde{a}}$)**.** Reduction to atomic cuts (with possibly contracted or weakened cut-formula-occurrences):

$$\rhd_{\tilde{a}} \doteq \rhd_{r_{non-atomic}} \cup \rhd_p \cup \rhd_q \cup \rhd_{w_{non-atomic}} \cup \rhd_{c_{non-atomic}}$$

Definition 3.2.22 ($\rhd_{\overline{a}}$)**.** Reduction to atomic cuts (so that no cut-formula-occurrence is contracted or weakened):

$$\rhd_{\overline{a}} \doteq \rhd_r \cup \rhd_p \cup \rhd_q \cup \rhd_w \cup \rhd_c$$

Definition 3.2.23 ($\rhd$)**.** Reductive cut-elimination:

$$\rhd \doteq \rhd_{\overline{a}} \cup \rhd_a$$

3.2.3 Examples of Reductive Cut-Elimination

Two detailed simple examples of reductive cut-elimination are shown below. In each example the arrows have a subscript indicating which proof rewriting rule has been applied.

Example 3.1. *Let φ be the proof below:*

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\AxiomC{$A \vdash A$}
\RightLabel{w_l}
\UnaryInfC{$C, A \vdash A$}
\AxiomC{$B \vdash B$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash B, \neg B$}
\RightLabel{$\vee_r$}
\UnaryInfC{$\vdash B \vee \neg B$}
\AxiomC{$B \vdash B$}
\AxiomC{$B \vdash B$}
\RightLabel{$\neg_l$}
\UnaryInfC{$\neg B, B \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$B \vee \neg B, B \vdash B$}
\RightLabel{$\rightarrow_r$}
\UnaryInfC{$B \vee \neg B \vdash B \rightarrow B$}
\RightLabel{cut}
\BinaryInfC{$\vdash B \rightarrow B$}
\RightLabel{w_l}
\UnaryInfC{$D \vdash B \rightarrow B$}
\RightLabel{$\vee_l$}
\BinaryInfC{$A, C \vee D \vdash A, B \rightarrow B$}
\RightLabel{cut}
\BinaryInfC{$A, C \vee D \vdash A, B \rightarrow B$}
\end{prooftree}
$$

Its cuts can be eliminated according to the following proof rewriting sequence:

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\AxiomC{$A \vdash A$}
\RightLabel{w_l}
\UnaryInfC{$C, A \vdash A$}
\AxiomC{$B \vdash B$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash B, \neg B$}
\RightLabel{$\vee_r$}
\UnaryInfC{$\vdash B \vee \neg B$}
\AxiomC{$B \vdash B$}
\AxiomC{$B \vdash B$}
\RightLabel{$\neg_l$}
\UnaryInfC{$\neg B, B \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$B \vee \neg B, B \vdash B$}
\RightLabel{$\rightarrow_r$}
\UnaryInfC{$B \vee \neg B \vdash B \rightarrow B$}
\RightLabel{cut}
\BinaryInfC{$\vdash B \rightarrow B$}
\RightLabel{w_l}
\UnaryInfC{$D \vdash B \rightarrow B$}
\RightLabel{$\vee_l$}
\BinaryInfC{$A, C \vee D \vdash A, B \rightarrow B$}
\RightLabel{cut}
\BinaryInfC{$A, C \vee D \vdash A, B \rightarrow B$}
\end{prooftree}
$$

$$\Downarrow_{\rhd_{r_1}}$$

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\AxiomC{$A \vdash A$}
\RightLabel{w_l}
\UnaryInfC{$C, A \vdash A$}
\AxiomC{$B \vdash B$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash B, \neg B$}
\RightLabel{$\vee_r$}
\UnaryInfC{$\vdash B \vee \neg B$}
\AxiomC{$B \vdash B$}
\AxiomC{$B \vdash B$}
\RightLabel{$\neg_l$}
\UnaryInfC{$\neg B, B \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$B \vee \neg B, B \vdash B$}
\RightLabel{cut}
\BinaryInfC{$B \vdash B$}
\RightLabel{$\rightarrow_r$}
\UnaryInfC{$\vdash B \rightarrow B$}
\RightLabel{w_l}
\UnaryInfC{$D \vdash B \rightarrow B$}
\RightLabel{$\vee_l$}
\BinaryInfC{$A, C \vee D \vdash A, B \rightarrow B$}
\RightLabel{cut}
\BinaryInfC{$A, C \vee D \vdash A, B \rightarrow B$}
\end{prooftree}
$$

$$\Downarrow_{\rhd_{p_\vee}}$$

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\AxiomC{$A \vdash A$}
\RightLabel{w_l}
\UnaryInfC{$C, A \vdash A$}
\AxiomC{$B \vdash B$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash B, \neg B$}
\AxiomC{$B \vdash B$}
\RightLabel{$\neg_l$}
\UnaryInfC{$\neg B, B \vdash$}
\RightLabel{cut}
\BinaryInfC{$B \vdash B$}
\AxiomC{$B \vdash B$}
\RightLabel{cut}
\BinaryInfC{$B \vdash B$}
\RightLabel{$\rightarrow_r$}
\UnaryInfC{$\vdash B \rightarrow B$}
\RightLabel{w_l}
\UnaryInfC{$D \vdash B \rightarrow B$}
\RightLabel{$\vee_l$}
\BinaryInfC{$A, C \vee D \vdash A, B \rightarrow B$}
\RightLabel{cut}
\BinaryInfC{$A, C \vee D \vdash A, B \rightarrow B$}
\end{prooftree}
$$

$$\Downarrow_{\rhd_{p_\neg}}$$

$$\dfrac{A \vdash A \qquad \dfrac{\dfrac{A \vdash A}{C, A \vdash A}\,w_l \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{B \vdash B \qquad B \vdash B}{B \vdash B}\,cut \qquad B \vdash B}{B \vdash B}\,cut}{\vdash B \to B}\,\to_r}{D \vdash B \to B}\,w_l}{A, C \vee D \vdash A, B \to B}\,\vee_l}{A, C \vee D \vdash A, B \to B}\,cut$$

$$\Downarrow_{\rhd_{r_2}}$$

$$\dfrac{\dfrac{A \vdash A \qquad \dfrac{A \vdash A}{C, A \vdash A}\,w_l}{A, C \vdash A}\,cut \qquad \dfrac{\dfrac{\dfrac{\dfrac{B \vdash B \qquad B \vdash B}{B \vdash B}\,cut \qquad B \vdash B}{B \vdash B}\,cut}{\vdash B \to B}\,\to_r}{D \vdash B \to B}\,w_l}{A, C \vee D \vdash A, B \to B}\,\vee_l$$

$$\Downarrow_{\rhd_w}$$

$$\dfrac{\dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A \vdash A}\,cut}{C, A \vdash A}\,w_l \qquad \dfrac{\dfrac{\dfrac{\dfrac{B \vdash B \qquad B \vdash B}{B \vdash B}\,cut \qquad B \vdash B}{B \vdash B}\,cut}{\vdash B \to B}\,\to_r}{D \vdash B \to B}\,w_l}{A, C \vee D \vdash A, B \to B}\,\vee_l$$

$$\Downarrow_{\rhd_a}$$

$$\dfrac{\dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A \vdash A}\,cut}{C, A \vdash A}\,w_l \qquad \dfrac{\dfrac{\dfrac{B \vdash B \qquad B \vdash B}{B \vdash B}\,cut}{\vdash B \to B}\,\to_r}{D \vdash B \to B}\,w_l}{A, C \vee D \vdash A, B \to B}\,\vee_l$$

$$\Downarrow_{\rhd_a}$$

$$\dfrac{\dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A \vdash A}\,cut}{C, A \vdash A}\,w_l \qquad \dfrac{\dfrac{B \vdash B}{\vdash B \to B}\,\to_r}{D \vdash B \to B}\,w_l}{A, C \vee D \vdash A, B \to B}\,\vee_l$$

$$\Downarrow_{\rhd_a}$$

$$\dfrac{\dfrac{A \vdash A}{C, A \vdash A}\,w_l \qquad \dfrac{\dfrac{B \vdash B}{\vdash B \to B}\,\to_r}{D \vdash B \to B}\,w_l}{A, C \vee D \vdash A, B \to B}\,\vee_l$$

Example 3.2. *Let φ be the proof below:*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(u) \vdash P(u) \qquad Q(u) \vdash Q(u)}{P(u), P(u) \to Q(u) \vdash Q(u)}\to_l}{P(u) \to Q(u) \vdash P(u) \to Q(u)}\to_r}{P(u) \to Q(u) \vdash \exists y(P(u) \to Q(y))}\exists_r}{\forall x(P(x) \to Q(x)) \vdash \exists y(P(u) \to Q(y))}\forall_l}{\forall x(P(x) \to Q(x)) \vdash \forall x\exists y(P(x) \to Q(y))}\forall_r \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(a) \vdash P(a) \qquad Q(v) \vdash Q(v)}{P(a), P(a) \to Q(v) \vdash Q(v)}\to_l}{P(a) \to Q(v) \vdash P(a) \to Q(v)}\to_r}{P(a) \to Q(v) \vdash \exists y(P(a) \to Q(y))}\exists_r}{\exists y(P(a) \to Q(y)) \vdash \exists y(P(a) \to Q(y))}\exists_l}{\forall x\exists y(P(x) \to Q(y)) \vdash \exists y(P(a) \to Q(y))}\forall_l}{\forall x(P(x) \to Q(x)) \vdash \exists y(P(a) \to Q(y))}\mathit{cut}$$

Its cut can be eliminated according to the following proof rewriting sequence:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(u) \vdash P(u) \qquad Q(u) \vdash Q(u)}{P(u), P(u) \to Q(u) \vdash Q(u)}\to_l}{P(u) \to Q(u) \vdash P(u) \to Q(u)}\to_r}{P(u) \to Q(u) \vdash \exists y(P(u) \to Q(y))}\exists_r}{\forall x(P(x) \to Q(x)) \vdash \exists y(P(u) \to Q(y))}\forall_l}{\forall x(P(x) \to Q(x)) \vdash \forall x\exists y(P(x) \to Q(y))}\forall_r \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(a) \vdash P(a) \qquad Q(v) \vdash Q(v)}{P(a), P(a) \to Q(v) \vdash Q(v)}\to_l}{P(a) \to Q(v) \vdash P(a) \to Q(v)}\to_r}{P(a) \to Q(v) \vdash \exists y(P(a) \to Q(y))}\exists_r}{\exists y(P(a) \to Q(y)) \vdash \exists y(P(a) \to Q(y))}\exists_l}{\forall x\exists y(P(x) \to Q(y)) \vdash \exists y(P(a) \to Q(y))}\forall_l}{\forall x(P(x) \to Q(x)) \vdash \exists y(P(a) \to Q(y))}\mathit{cut}$$

$$\Downarrow_{\rhd_{q\forall}}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(a) \vdash P(a) \qquad Q(a) \vdash Q(a)}{P(a), P(a) \to Q(a) \vdash Q(a)}\to_l}{P(a) \to Q(a) \vdash P(a) \to Q(a)}\to_r}{P(a) \to Q(a) \vdash \exists y(P(a) \to Q(y))}\exists_r}{\forall x(P(x) \to Q(x)) \vdash \exists y(P(a) \to Q(y))}\forall_l \qquad \dfrac{\dfrac{\dfrac{\dfrac{P(a) \vdash P(a) \qquad Q(v) \vdash Q(v)}{P(a), P(a) \to Q(v) \vdash Q(v)}\to_l}{P(a) \to Q(v) \vdash P(a) \to Q(v)}\to_r}{P(a) \to Q(v) \vdash \exists y(P(a) \to Q(y))}\exists_r}{\exists y(P(a) \to Q(y)) \vdash \exists y(P(a) \to Q(y))}\exists_l}{\forall x(P(x) \to Q(x)) \vdash \exists y(P(a) \to Q(y))}\mathit{cut}$$

$$\Downarrow_{\rhd_{r_1}}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(a) \vdash P(a) \qquad Q(a) \vdash Q(a)}{P(a), P(a) \to Q(a) \vdash Q(a)}\to_l}{P(a) \to Q(a) \vdash P(a) \to Q(a)}\to_r}{P(a) \to Q(a) \vdash \exists y(P(a) \to Q(y))}\exists_r \qquad \dfrac{\dfrac{\dfrac{\dfrac{P(a) \vdash P(a) \qquad Q(v) \vdash Q(v)}{P(a), P(a) \to Q(v) \vdash Q(v)}\to_l}{P(a) \to Q(v) \vdash P(a) \to Q(v)}\to_r}{P(a) \to Q(v) \vdash \exists y(P(a) \to Q(y))}\exists_r}{\exists y(P(a) \to Q(y)) \vdash \exists y(P(a) \to Q(y))}\exists_l}{P(a) \to Q(a) \vdash \exists y(P(a) \to Q(y))}\mathit{cut}}{\forall x(P(x) \to Q(x)) \vdash \exists y(P(a) \to Q(y))}\forall_l$$

$$\Downarrow_{\rhd_{q\exists}}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{P(a) \vdash P(a) \qquad Q(a) \vdash Q(a)}{P(a), P(a) \to Q(a) \vdash Q(a)}\to_l}{P(a) \to Q(a) \vdash P(a) \to Q(a)}\to_r \qquad \dfrac{\dfrac{\dfrac{P(a) \vdash P(a) \qquad Q(a) \vdash Q(a)}{P(a), P(a) \to Q(a) \vdash Q(a)}\to_l}{P(a) \to Q(a) \vdash P(a) \to Q(a)}\to_r}{P(a) \to Q(a) \vdash \exists y(P(a) \to Q(y))}\exists_r}{P(a) \to Q(a) \vdash \exists y(P(a) \to Q(y))}\mathit{cut}}{\forall x(P(x) \to Q(x)) \vdash \exists y(P(a) \to Q(y))}\forall_l$$

$$\Downarrow_{\rhd_{r_1}}$$

$$
\begin{prooftree}
\AxiomC{$P(a) \vdash P(a)$}
\AxiomC{$Q(a) \vdash Q(a)$}
\RightLabel{$\rightarrow_l$}
\BinaryInfC{$P(a), P(a) \rightarrow Q(a) \vdash Q(a)$}
\RightLabel{$\rightarrow_r$}
\UnaryInfC{$P(a) \rightarrow Q(a) \vdash P(a) \rightarrow Q(a)$}
\AxiomC{$P(a) \vdash P(a)$}
\AxiomC{$Q(a) \vdash Q(a)$}
\RightLabel{$\rightarrow_l$}
\BinaryInfC{$P(a), P(a) \rightarrow Q(a) \vdash Q(a)$}
\RightLabel{$\rightarrow_r$}
\UnaryInfC{$P(a) \rightarrow Q(a) \vdash P(a) \rightarrow Q(a)$}
\RightLabel{cut}
\BinaryInfC{$P(a) \rightarrow Q(a) \vdash P(a) \rightarrow Q(a)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$P(a) \rightarrow Q(a) \vdash \exists y(P(a) \rightarrow Q(y))$}
\RightLabel{$\forall_l$}
\UnaryInfC{$\forall x(P(x) \rightarrow Q(x)) \vdash \exists y(P(a) \rightarrow Q(y))$}
\end{prooftree}
$$

$$\Downarrow_{\rhd_{r_1}}$$

$$
\begin{prooftree}
\AxiomC{$P(a) \vdash P(a)$}
\AxiomC{$Q(a) \vdash Q(a)$}
\RightLabel{$\rightarrow_l$}
\BinaryInfC{$P(a), P(a) \rightarrow Q(a) \vdash Q(a)$}
\RightLabel{$\rightarrow_r$}
\UnaryInfC{$P(a) \rightarrow Q(a) \vdash P(a) \rightarrow Q(a)$}
\AxiomC{$P(a) \vdash P(a)$}
\AxiomC{$Q(a) \vdash Q(a)$}
\RightLabel{$\rightarrow_l$}
\BinaryInfC{$P(a), P(a) \rightarrow Q(a) \vdash Q(a)$}
\RightLabel{cut}
\BinaryInfC{$P(a) \rightarrow Q(a), P(a) \vdash Q(a)$}
\RightLabel{$\rightarrow_r$}
\UnaryInfC{$P(a) \rightarrow Q(a) \vdash P(a) \rightarrow Q(a)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$P(a) \rightarrow Q(a) \vdash \exists y(P(a) \rightarrow Q(y))$}
\RightLabel{$\forall_l$}
\UnaryInfC{$\forall x(P(x) \rightarrow Q(x)) \vdash \exists y(P(a) \rightarrow Q(y))$}
\end{prooftree}
$$

$$\Downarrow_{\rhd_{p\rightarrow}}$$

$$
\begin{prooftree}
\AxiomC{$P(a) \vdash P(a)$}
\AxiomC{$P(a) \vdash P(a)$}
\AxiomC{$Q(a) \vdash Q(a)$}
\RightLabel{$\rightarrow_l$}
\BinaryInfC{$P(a), P(a) \rightarrow Q(a) \vdash Q(a)$}
\AxiomC{$Q(a) \vdash Q(a)$}
\RightLabel{cut}
\BinaryInfC{$P(a), P(a) \rightarrow Q(a) \vdash Q(a)$}
\RightLabel{cut}
\BinaryInfC{$P(a) \rightarrow Q(a), P(a) \vdash Q(a)$}
\RightLabel{$\rightarrow_r$}
\UnaryInfC{$P(a) \rightarrow Q(a) \vdash P(a) \rightarrow Q(a)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$P(a) \rightarrow Q(a) \vdash \exists y(P(a) \rightarrow Q(y))$}
\RightLabel{$\forall_l$}
\UnaryInfC{$\forall x(P(x) \rightarrow Q(x)) \vdash \exists y(P(a) \rightarrow Q(y))$}
\end{prooftree}
$$

$$\Downarrow_{\rhd_{r_2}}$$

$$
\begin{prooftree}
\AxiomC{$P(a) \vdash P(a)$}
\AxiomC{$P(a) \vdash P(a)$}
\AxiomC{$Q(a) \vdash Q(a)$}
\AxiomC{$Q(a) \vdash Q(a)$}
\RightLabel{cut}
\BinaryInfC{$Q(a) \vdash Q(a)$}
\RightLabel{$\rightarrow_l$}
\BinaryInfC{$P(a), P(a) \rightarrow Q(a) \vdash Q(a)$}
\RightLabel{cut}
\BinaryInfC{$P(a) \rightarrow Q(a), P(a) \vdash Q(a)$}
\RightLabel{$\rightarrow_r$}
\UnaryInfC{$P(a) \rightarrow Q(a) \vdash P(a) \rightarrow Q(a)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$P(a) \rightarrow Q(a) \vdash \exists y(P(a) \rightarrow Q(y))$}
\RightLabel{$\forall_l$}
\UnaryInfC{$\forall x(P(x) \rightarrow Q(x)) \vdash \exists y(P(a) \rightarrow Q(y))$}
\end{prooftree}
$$

$$\Downarrow_{\rhd_{r_2}}$$

$$
\begin{prooftree}
\AxiomC{$P(a) \vdash P(a)$}
\AxiomC{$P(a) \vdash P(a)$}
\RightLabel{cut}
\BinaryInfC{$P(a) \vdash P(a)$}
\AxiomC{$Q(a) \vdash Q(a)$}
\AxiomC{$Q(a) \vdash Q(a)$}
\RightLabel{cut}
\BinaryInfC{$Q(a) \vdash Q(a)$}
\RightLabel{$\rightarrow_l$}
\BinaryInfC{$P(a) \rightarrow Q(a), P(a) \vdash Q(a)$}
\RightLabel{$\rightarrow_r$}
\UnaryInfC{$P(a) \rightarrow Q(a) \vdash P(a) \rightarrow Q(a)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$P(a) \rightarrow Q(a) \vdash \exists y(P(a) \rightarrow Q(y))$}
\RightLabel{$\forall_l$}
\UnaryInfC{$\forall x(P(x) \rightarrow Q(x)) \vdash \exists y(P(a) \rightarrow Q(y))$}
\end{prooftree}
$$

$$\Downarrow_{\rhd_{a}}$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{P(a) \vdash P(a) \qquad \dfrac{Q(a) \vdash Q(a) \qquad Q(a) \vdash Q(a)}{Q(a) \vdash Q(a)}\,cut}{P(a) \to Q(a), P(a) \vdash Q(a)}\,\to_l}{P(a) \to Q(a) \vdash P(a) \to Q(a)}\,\to_r}{P(a) \to Q(a) \vdash \exists y(P(a) \to Q(y))}\,\exists_r}{\forall x(P(x) \to Q(x)) \vdash \exists y(P(a) \to Q(y))}\,\forall_l
$$

$$
\Downarrow_{\rhd_a}
$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{P(a) \vdash P(a) \qquad Q(a) \vdash Q(a)}{P(a) \to Q(a), P(a) \vdash Q(a)}\,\to_l}{P(a) \to Q(a) \vdash P(a) \to Q(a)}\,\to_r}{P(a) \to Q(a) \vdash \exists y(P(a) \to Q(y))}\,\exists_r}{\forall x(P(x) \to Q(x)) \vdash \exists y(P(a) \to Q(y))}\,\forall_l
$$

3.2.4 Some Properties of Reductive Cut-elimination

The following properties of reductive cut-elimination are well-known. They are included here for the sake of self-containment.

Theorem 3.1 (Lack of Strong Normalization for $\rhd$)**.** $\rhd$ is not strongly normalizing.

Proof. Consider the proof φ below:

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\AxiomC{$A \vdash A$}
\RightLabel{$\vee_l$}
\BinaryInfC{$A \vee A \vdash A, A$}
\RightLabel{c_r}
\UnaryInfC{$A \vee A \vdash A$}
\AxiomC{$A \vdash A$}
\AxiomC{$A \vdash A$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A, A \vdash A \wedge A$}
\RightLabel{c_l}
\UnaryInfC{$A \vdash A \wedge A$}
\RightLabel{cut}
\BinaryInfC{$A \vee A \vdash A \wedge A$}
\end{prooftree}
$$

As shown in [66, 67] , there is a rewriting sequence leading to φ':

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\AxiomC{$A \vdash A$}
\RightLabel{$\vee_l$}
\BinaryInfC{$A \vee A \vdash A, A$}
\AxiomC{$A \vdash A$}
\AxiomC{$A \vdash A$}
\RightLabel{$\vee_l$}
\BinaryInfC{$A \vee A \vdash A, A$}
\AxiomC{$A \vdash A$}
\AxiomC{$A \vdash A$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A, A \vdash A \wedge A$}
\RightLabel{cut}
\BinaryInfC{$A \vee A, A \vdash A, A \wedge A$}
\RightLabel{cut}
\BinaryInfC{$A \vee A, A \vee A \vdash A, A, A \wedge A$}
\RightLabel{c_l}
\UnaryInfC{$A \vee A \vdash A, A, A \wedge A$}
\RightLabel{c_r}
\UnaryInfC{$A \vee A \vdash A, A \wedge A$}
\AxiomC{$A \vdash A$}
\AxiomC{$A \vdash A$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A, A \vdash A \wedge A$}
\RightLabel{c_l}
\UnaryInfC{$A \vdash A \wedge A$}
\RightLabel{cut}
\BinaryInfC{$A \vee A \vdash A \wedge A, A \wedge A$}
\RightLabel{c_r}
\UnaryInfC{$A \vee A \vdash A \wedge A$}
\end{prooftree}
$$

Both auxiliary formula occurrences of the lowermost cut are again main formula occurrences of contractions. Essentially the same reductions can be employed indefinitely, in order to produce always larger proofs and never reaching a $\rhd$-normal-form. $\square$

Theorem 3.2 (Lack of Confluence for $\rhd$)**.** $\rhd$ is not confluent.

Proof. Consider the proof φ_w below:

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\RightLabel{$\neg_l$}
\UnaryInfC{$A, \neg A \vdash$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge \neg A \vdash$}
\RightLabel{w_r}
\UnaryInfC{$A \wedge \neg A \vdash A$}
\AxiomC{$A \vdash A$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash A, \neg A$}
\RightLabel{$\vee_r$}
\UnaryInfC{$\vdash A \vee \neg A$}
\RightLabel{w_l}
\UnaryInfC{$A \vdash A \vee \neg A$}
\BinaryInfC{$A \wedge \neg A \vdash A \vee \neg A$}
\end{prooftree}
$$

It has two distinct $\rhd$-normal-forms, depending on which side is preferred for the cut-reduction over weakening:

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\RightLabel{$\neg_l$}
\UnaryInfC{$A, \neg A \vdash$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge \neg A \vdash$}
\RightLabel{w_r}
\UnaryInfC{$A \wedge \neg A \vdash A \vee \neg A$}
\end{prooftree}
\qquad
\begin{prooftree}
\AxiomC{$A \vdash A$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash A, \neg A$}
\RightLabel{$\vee_r$}
\UnaryInfC{$\vdash A \vee \neg A$}
\RightLabel{w_l}
\UnaryInfC{$A \wedge \neg A \vdash A \vee \neg A$}
\end{prooftree}
$$

Non-confluence can also be caused by contractions on both sides of a cut, as shown in the proof φ_c below [66, 31, 46, 115]:

$$\dfrac{\dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A \vee A \vdash A, A}\,\vee_l}{A \vee A \vdash A}\,c_r \qquad \dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A, A \vdash A \wedge A}\,\wedge_r}{A \vdash A \wedge A}\,c_l}{A \vee A \vdash A \wedge A}\,cut$$

φ_c has the following two $\rhd$-normal-forms:

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A, A \vdash A \wedge A}\,\wedge_r}{A \vdash A \wedge A}\,c_l \qquad \dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A, A \vdash A \wedge A}\,\wedge_r}{A \vdash A \wedge A}\,c_l}{A \vee A \vdash A \wedge A, A \wedge A}\,\vee_l}{A \vee A \vdash A \wedge A}\,c_r$$

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A \vee A \vdash A, A}\,\vee_l}{A \vee A \vdash A}\,c_r \qquad \dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A \vee A \vdash A, A}\,\vee_l}{A \vee A \vdash A}\,c_r}{A \vee A, A \vee A \vdash A \wedge A}\,\wedge_r}{A \vee A \vdash A \wedge A}\,c_l$$

□

Theorem 3.3 (Weak Normalization for $\rhd$)**.** $\rhd$ is weakly normalizing.

Proof. Two well-known terminating strategies for $\rhd$ are Gentzen's [48, 13], which selects uppermost cuts, and Tait's [9], which selects cuts of maximal logical complexity. □

Theorem 3.4 (Cut-elimination Theorem (Gentzen's Hauptsatz))**.** If there is a proof φ with end-sequent $\Gamma \vdash \Delta$, then there exists a cut-free[f] proof φ' with end-sequent $\Gamma \vdash \Delta$.

Proof. By Theorem 3.3, there exists a proof φ' with end-sequent $\gamma \vdash \delta$ and such that $\varphi \rhd^* \varphi'$ and φ' is a $\rhd$-normal-form. Assume, for the sake of contradiction, that φ' has cuts. Then, at least one of the proof rewriting rules that define $\rhd$ can be applied, and hence φ' is not a $\rhd$-normal-form. As this contradicts the fact that φ' is a $\rhd$-normal-form, φ' must be cut-free. Therefore, there exists a cut-free proof with end-sequent $\Gamma \vdash \Delta$.

While Gentzen proved his Hauptsatz syntactically and constructively by using essentially $\rhd$. It is also possible to give a semantic proof based on the facts that sequent calculus is semantically sound and cut-free sequent calculus is semantically complete [112]. However, this proof has the disadvantage of not providing a method (except, of course, naive proof

[f]Gentzen's Hauptsatz holds for proofs with axiom sequents of the form $A \vdash A$, as assumed in this Thesis. However, for mathematical applications, it is frequently convenient to allow more flexible forms of axiom sequents. In this case, some cuts might not be completely eliminable. An in-depth discussion of this topic can be found in [22].

search) for actually constructing a cut-free proof from a proof with cuts. □

Remark 3.2.1 (The Complexity of Cut-Elimination)**.** Since [93] and [109] it is known that the complexity of cut-elimination is non-elementary. The elimination of cuts from a proof φ can result in a proof whose size and length are non-elementarily bigger than the size of φ. Detailed discussions of the complexity of cut-elimination can be found in [22, 98, 13, 106]. More refined bounds, showing the role of quantifier alternations and contractions, can be found in [50].

Chapter 4

Cut-Elimination by Resolution

Although reductive cut-elimination methods are sufficient for theoretical purposes, such as Gentzen's original aim of proving the cut-elimination theorem and its corollaries, they have a few drawbacks[a] when it comes to practical applications, such as the actual automated elimination of cuts from formalized mathematical proofs in order to obtain new and simpler mathematical proofs [6].

Firstly, reductive cut-elimination methods are not robust to changes in the sequent calculus. If new rules were added to the sequent calculus or if the existing rules were just slightly modified (e.g. if rules with implicit contractions and weakenings were used, such as in an additive or mixed calculus), then new rewriting rules would have to be formulated to cope with the modified calculus.

Secondly, the elimination of cuts by reductive methods is a long and costly process of local rewritings. This locality of the method also implies that it is not capable of exploiting the global structure of the proof in order to potentially achieve significantly shorter cut-free proofs.

Finally, a comparison [9, 11] of Gentzen's rewriting strategy to that of Tait and Schuette shows that there are two sequences of proofs $(\varphi_n)_{n\in\mathbb{N}}$ and $(\psi_n)_{n\in\mathbb{N}}$ such that:

- there is a sequence of cut-free proofs $(\varphi_n^G)_{n\in\mathbb{N}}$ obtained from $(\varphi_n)_{n\in\mathbb{N}}$ via $\rhd^G$ whose sizes of the cut-elimination sequences grow polynomially with respect to the sizes of the proofs in $(\varphi_n^G)_{n\in\mathbb{N}}$ but the sizes

[a]It is interesting to note that Goedel's alternative approach, the dialectica functional interpretation, also has some drawbacks in the actual extraction of functionals from proofs, which led to the development of the refined A-translation [113].

of the cut-free proofs of any sequence $(\varphi_n^T)_{n\in\mathbb{N}}$ obtained from $(\varphi_n)_{n\in\mathbb{N}}$ via $\triangleright^T$ grow non-elementarily.

- there is a sequence of cut-free proofs $(\psi_n^T)_{n\in\mathbb{N}}$ obtained from $(\psi_n)_{n\in\mathbb{N}}$ via $\triangleright^T$ whose sizes of the cut-elimination sequences grow polynomially with respect to the sizes of the proofs in $(\psi_n^G)_{n\in\mathbb{N}}$ but the sizes of the cut-free proofs of any sequence $(\psi_n^G)_{n\in\mathbb{N}}$ obtained from $(\psi_n)_{n\in\mathbb{N}}$ via $\triangleright^G$ grow non-elementarily.

This shows that the non-deterministic choices made in reductive cut-elimination can have a significant effect on the size of the cut-free proofs. But the right choices could only be done with a global analysis of the proof.

These drawbacks motivate the continuous development of an alternative method of *cut-elimination by resolution*, which:

- is compactly and abstractly described, ensuring robustness with respect to details of the sequent calculus.
- eliminates all cuts at once.
- is global.
- is such that there is always a sequence $(\psi'_n)_{n\in\mathbb{N}}$ of cut-free proofs obtained from $(\psi_n)_{n\in\mathbb{N}}$ by cut-elimination by resolution whose sizes grow at most elementarily with respect to the sizes of the proofs $(\psi_n^S)_{n\in\mathbb{N}}$ obtained from $(\psi_n)_{n\in\mathbb{N}}$ by reductive cut-elimination following any rewriting strategy.

Cut-elimination by resolution starts with the extraction of a *cut pertinent struct* from a proof φ with end-sequent $\Gamma \vdash \Delta$. This struct can be transformed into a *cut-pertinent clause set*, which is always unsatisfiable. By the refutational completeness of the resolution calculus (Theorem 2.3), there is a resolution refutation δ of the cut-pertinent clause set. A new proof CERes(φ, δ) of $\Gamma \vdash \Delta$ can then be constructed by using the refutation as a skeleton (with an appropriate conversion of factoring to contraction inferences and resolution inferences to cuts), on which the leaf subproofs (each consisting of an initial inference deriving a clause from the cut-pertinent clause set) are replaced by *projections*. CERes(φ, δ) is not completely cut-free, but its cuts have atomic cut-formulas only. Such cuts are inessential, because, as their cut-formulas do not have quantifiers, they do not prevent the extraction of a Herbrand sequent [94].

The first Section (Section 4.1) of this Chapter explains the most fundamental concept of cut-elimination by resolution as presented here: the cut-pertinent struct. The following Sections (4.2, 4.3, 4.4, 4.5, 4.6) explain different ways[b] to transform the cut-pertinent struct into a cut-pertinent clause set, as well as different algorithms to construct projections for the clauses of such clause sets. Finally, Section 4.7 explains how to put refutations and projections together, in order to obtain CERes-Normal-Forms.

Although the rules of the sequent calculus defined and used in previous chapters have at most two premises, sequent calculi with rules having greater arity do exist. This is the case, for example, of sequent calculi with superdeduction [36, 17] and sequent calculi for multi-valued logics [5, 4, 118]. Therefore, in contrast to usual presentations of the method[c], cut-elimination by resolution is defined here in a general way, making no assumptions about the arities of the rules present in the calculus. Nevertheless, examples and proofs of some theorems stick to the case of sequent calculi having at most binary rules.

4.1 Cut-Pertinent Struct

In order to extract from proofs information that is pertinent to the cuts in the proofs, auxiliary definitions (4.1.5, 4.1.2, 4.1.1, 4.9.2) are necessary to clarify the notion of *pertinence* in this context. Then, the concept of *struct* (Definition 4.1.4) can be used as a compact way to store the cut-pertinent information of a proof (Definition 4.1.7).

Definition 4.1.1 (Cut-Pertinent and Cut-Impertinent Occurrences)**.** A formula occurrence is *cut-pertinent* if and only if it is an ancestor of a cut formula occurrence. The *set of cut-pertinent formula occurrences* of a proof φ is denoted $\Omega_{CP}(\varphi)$.

A formula occurrence is *cut-impertinent* if and only if it is not cut-pertinent. The *set of cut-impertinent formula occurrences* of a proof φ is denoted $\Omega_{CI}(\varphi)$.

Remark 4.1.1. In **LK**, a cut-impertinent formula occurrence is an ancestor of a formula occurrence in the end-sequent.

[b] For the reader who is not yet so familiar with cut-elimination by resolution, it might be profitable to skip Sections 4.3, 4.4, 4.5 and 4.6 in a first read, focusing instead on the simplest way of constructing clause sets described in Section 4.2 only.

[c] Note, however, that a method of cut-elimination by resolution for sequent calculi for multi-valued logics containing rules with arbitrary arity has already been defined in [5]

Definition 4.1.2 (Ω-Pertinence)**.** Let ρ be an inference of a proof φ, and let Ω be a set of atomic formula occurrences of φ. Then ρ *operates within* Ω if and only if all its active atomic occurrences are in Ω; and ρ *partially operates within* Ω if and only if at least one of its active atomic subformula occurrences is in Ω. Moreover, ρ is Ω-*pertinent* if and only if ρ operates within Ω; ρ is Ω-*partially-pertinent* if and only if ρ partially operates within Ω; and ρ is Ω-*impertinent* if and only if ρ is neither Ω-*pertinent* nor Ω-*partially-pertinent*.

Definition 4.1.3 (Cut-Pertinence)**.** An inference ρ is *cut-pertinent* if and only if ρ is $\Omega_{CP}(\varphi)$-pertinent.

An inference ρ is *cut-impertinent* if and only if ρ is $\Omega_{CI}(\varphi)$-pertinent.

Definition 4.1.4 (Structs)**.** The *set of structs*[d] **S** is defined as the smallest set satisfying:

1. If A is an atomic formula occurrence, then $A \in \mathbf{S}$.
2. $\epsilon_\otimes \in \mathbf{S}$ (the empty ⊗-junction).
3. $\epsilon_\oplus \in \mathbf{S}$ (the empty ⊕-junction).
4. If $S_1, \ldots, S_n \in \mathbf{S}$, then $(S_1 \otimes \ldots \otimes S_n) \in \mathbf{S}$.
5. If $S_1, \ldots, S_n \in \mathbf{S}$, then $(S_1 \oplus \ldots \oplus S_n) \in \mathbf{S}$.
6. If $S \in \mathbf{S}$, then $\neg S \in \mathbf{S}$.

The connectives ⊗ and ⊕ are assumed to be commutative. So, the order of substructs in ⊗-junctions and ⊕-junctions does not matter. Moreover, they are also assumed to be associative. So, inner parentheses can be dropped in structs that contain either only ⊗ or only ⊕.

Definition 4.1.5 (Closure for Sets of Formula Occurrences)**.** A set Ω of atomic (sub)formula occurrences of a proof φ is *closed* if and only if, if any active atomic subformula occurrence of an inference ρ in φ is in Ω, then all active atomic subformula occurrences of ρ are in Ω.

[d] A struct is essentially the same as a clause term, as defined in [10, 13] . A new name is given here for three reasons:

- there is a subtle, although inessential, difference: in clause terms, leaves are occupied by clauses; while in structs, leaves are occupied by atomic formulas.
- the name "term" is already used in Definition 2.1.2.
- the name "struct" explicits the fact that structs extracted from proofs (e.g. cut-pertinent structs) have the same branching structure as their proofs, as shown in Example 4.2.

Remark 4.1.2. If Ω is closed, then every inference ρ is either Ω-pertinent or Ω-impertinent, but never Ω-partially-pertinent.

Definition 4.1.6 (Pertinent Struct)**.** The Ω-*pertinent struct* $\mathcal{S}^{\Omega}_{\varphi}$ of a proof φ with respect to a closed set of formula occurrences Ω is defined inductively, as follows:

- If φ consists of an axiom inference ρ only: Let $\omega^+_1, \ldots, \omega^+_n \in \Omega$ be positive formula occurrences and $\omega^-_1, \ldots, \omega^-_m \in \Omega$ be negative formula occurrences[e] in the axiom sequent of φ. Then:

  $$\mathcal{S}^{\Omega}_{\varphi} \doteq \neg\omega^-_1 \otimes_\rho \ldots \otimes_\rho \neg\omega^-_m \otimes_\rho \omega^+_1 \otimes_\rho \ldots \otimes_\rho \omega^+_n$$

  If $n = 0$ and $m = 0$, then $\mathcal{S}^{\Omega}_{\varphi}$ is the empty $\otimes$-junction, denoted $\epsilon_\otimes$.

- If φ ends with an n-ary Ω-pertinent inference ρ: Let $\varphi_1, \ldots, \varphi_n$ be the immediate subproofs of φ. Then:

  $$\mathcal{S}^{\Omega}_{\varphi} \doteq \mathcal{S}^{\Omega}_{\varphi_1} \oplus_\rho \ldots \oplus_\rho \mathcal{S}^{\Omega}_{\varphi_n}$$

- If φ ends with an n-ary Ω-impertinent inference ρ: Let $\varphi_1, \ldots, \varphi_n$ be the immediate subproofs of φ. Then:

  $$\mathcal{S}^{\Omega}_{\varphi} \doteq \mathcal{S}^{\Omega}_{\varphi_1} \otimes_\rho \ldots \otimes_\rho \mathcal{S}^{\Omega}_{\varphi_n}$$

A connective $\otimes_\rho$ or $\oplus_\rho$ is said to *correspond* to the inference ρ. When this correspondence is clear or irrelevant, the subscript ρ can be simply omitted.

Definition 4.1.7 (Cut-Pertinent Struct)**.** The *cut-pertinent struct* of a proof φ is defined as:

$$\mathcal{S}_{\varphi} \doteq \mathcal{S}^{\Omega_{CP}(\varphi)}_{\varphi}$$

Example 4.1 (Cut-Pertinent Struct)**.** *Let φ be the proof below:*

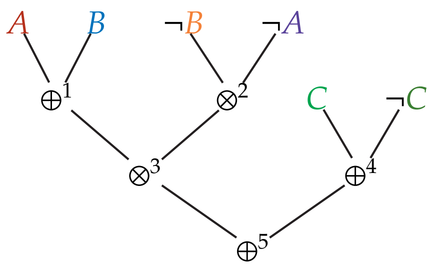
$$
\begin{prooftree}
\AxiomC{$A \vdash {\color{red}A}$}
\AxiomC{$B \vdash {\color{blue}B}$}
\RightLabel{$\wedge^1_r$}
\BinaryInfC{$A, B \vdash A \wedge B$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash A \wedge B$}
\AxiomC{${\color{orange}B} \vdash B$}
\AxiomC{${\color{violet}A} \vdash A$}
\RightLabel{$\wedge^2_r$}
\BinaryInfC{$A, B \vdash B \wedge A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash B \wedge A$}
\RightLabel{cut^3}
\BinaryInfC{$A \wedge B \vdash B \wedge A$}
\AxiomC{$C \vdash {\color{green}C}$}
\AxiomC{${\color{green}C} \vdash C$}
\RightLabel{cut^4}
\BinaryInfC{$C \vdash C$}
\RightLabel{$\vee^5_l$}
\BinaryInfC{$(A \wedge B) \vee C \vdash B \wedge A, C$}
\end{prooftree}
$$

[e] In the standard sequent calculus **LK**, only axiom sequents of the form $A \vdash A$ are allowed. Therefore, $n \leq 1$ and $m \leq 1$.

Its cut-pertinent struct is:

$$\mathcal{S}_\varphi \equiv ((A \oplus^1 B) \oplus^3 (\neg B \otimes^2 \neg A)) \otimes^5 (C \oplus^4 \neg C)$$

Example 4.2 (Struct Displayed as a Tree)**.** *Let φ be the proof of Example 4.1. Its cut-pertinent struct $\mathcal{S}_\varphi$ can be displayed as a tree, making it clearly visible that $\mathcal{S}_\varphi$ has the same branching structure as φ.*



4.2 CERes$_S$

4.2.1 Cut-Pertinent Standard Clause Set

Cut-pertinent structs can be transformed into clause sets (Definition 4.2.3) that are unsatisfiable (Theorem 4.1). The transformation is analogous to the standard clause form transformation for formulas in negation normal form [2], with $\oplus$ playing the role of $\wedge$ and $\otimes$ playing the role of $\vee$. Firstly, $\oplus\otimes$-Normalization (Definition 4.2.1) distributes $\otimes$ connectives over the $\oplus$ connectives until a struct in normal form is reached, which is a $\oplus$-junction of $\otimes$-junctions. Then each $\otimes$-junction can be interpreted as a clause (Definition 4.2.2).

Definition 4.2.1 ($\rightsquigarrow_{\oplus\otimes}$)**.** The standard struct normalization is defined by the following struct rewriting rules:

$$S \otimes (S_1 \oplus \ldots \oplus S_n) \rightsquigarrow_{\oplus\otimes} (S \otimes S_1) \oplus \ldots \oplus (S \otimes S_n)$$

$$(S_1 \oplus \ldots \oplus S_n) \otimes S \rightsquigarrow_{\oplus\otimes} (S_1 \otimes S) \oplus \ldots \oplus (S_n \otimes S)$$

Definition 4.2.2 (Clausification of Structs)**.** Let $S \equiv \bigoplus_{i \in I}(\bigotimes_{1 \leq j' \leq j_i} \neg\omega^-_{ij'} \otimes \bigotimes_{1 \leq h' \leq h_i} \omega^+_{ih'})$ be a struct in $\oplus\otimes$-normal-form. Then:

$$\mathrm{cl}(S) \doteq \{\omega^-_{i1}, \ldots, \omega^-_{ij_i} \vdash \omega^+_{i1}, \ldots, \omega^+_{ih_i} | i \in I\}$$

Definition 4.2.3 (Cut-pertinent Standard Clause Set)**.** The *cut-pertinent standard clause set*[f,g] of a proof φ is:

$$C^S_\varphi \doteq \mathrm{cl}(\mathcal{S}^*_\varphi)$$

where $\mathcal{S}^*_\varphi$ is the $\rightsquigarrow_{\oplus\otimes}$-normal-form of $\mathcal{S}_\varphi$.

Theorem 4.1 (Unsatisfiability of the Cut-Pertinent Clause Set)**.** C^S_φ is unsatisfiable, for any proof φ.

Proof. Detailed proofs for sequent calculi with at most binary rules are available in [9, 10]. A proof for certain calculi for multi-valued logics with rules of arbitrary arity is available in [5]. The essential idea of the proof is to construct a sequent calculus refutation φ' (i.e. a proof of the empty-sequent $\vdash$) of the clause set by modifying φ. In [9, 10], this is done by induction, removing cut-impertinent formula occurrences (this guarantees that the end-sequent of φ' is the empty sequent), skipping unary cut-impertinent inferences and performing a technically intricate proof merging procedure in the case of binary cut-impertinent inferences. This merging procedure guarantees that the axiom sequents of φ' are clauses of C^S_φ. φ' is therefore a sequent calculus refutation of C^S_φ, and by the soundness of the sequent calculus inference rules, C^S_φ is unsatisfiable.

It is interesting to remark that, by using proof transformation techniques that are developed in later sections and chapters of this thesis, the construction of φ' can be described in a simpler. Namely, φ' is construct by removing cut-impertinent formula occurrences, replacing cut-impertinent inferences by Y-inferences and then performing merging Y-elimination 5.1.2. $\square$

[f] Clause sets can be defined more generally with respect to any φ-closed set of formula occurrences, and not just with respect to the set of cut-pertinent formula occurrences. For example, *cut-impertinent* clause sets are used in [68].

[g] The cut-pertinent (standard) clause set is exactly the same as the *characteristic clause set* [13], which is the original clause set developed together with and for the method of cut-elimination by resolution [9, 10, 12]. It is interesting to note, however, that they are constructed in substantially different ways. While the *characteristic clause set* is obtained by interpreting $\oplus$ and $\otimes$ in the characteristic clause term as, respectively, a set union and a clause set merge operation (although it is also possible to construct the characteristic clause term directly from the proof, without prior extraction of the characteristic clause term [6]), the cut-pertinent (standard) clause set is constructed via $\oplus\otimes$-normalization of the cut-pertinent struct. Here, this alternative approach via $\oplus\otimes$-normalization is chosen not only to show that it is possible, but also for two other reasons. Firstly, the analogy between transformations of structs into clause sets and transformations of formulas into clause forms becomes clearer. Secondly, this approach provides a good framework in which improvements of the standard clause set (e.g. the profile clause set (Section 4.4) and the definitional clause set (Section 4.5) can all be seen as improved ways of $\oplus\otimes$-normalizing the cut-pertinent struct.

Example 4.3 (Cut-Pertinent Standard Clause Set)**.** *Let φ be the proof shown in Example 4.1. Its cut-pertinent struct $\mathcal{S}_\varphi$ can be $\oplus\otimes$-normalized as follows:*

$$
\begin{array}{rcl}
\mathcal{S}_\varphi & \equiv & ((A \oplus B) \oplus (\neg B \otimes \neg A)) \otimes (C \oplus \neg C) \\
& \rightsquigarrow_{\oplus\otimes} & ((A \oplus B) \otimes (C \oplus \neg C)) \oplus ((\neg B \otimes \neg A) \otimes (C \oplus \neg C)) \\
& \rightsquigarrow_{\oplus\otimes} & ((A \otimes (C \oplus \neg C)) \oplus (B \otimes (C \oplus \neg C))) \oplus ((\neg B \otimes \neg A) \otimes (C \oplus \neg C)) \\
& \rightsquigarrow_{\oplus\otimes} & (((A \otimes C) \oplus (A \otimes \neg C)) \oplus (B \otimes (C \oplus \neg C))) \oplus ((\neg B \otimes \neg A) \otimes (C \oplus \neg C)) \\
& \rightsquigarrow_{\oplus\otimes} & (((A \otimes C) \oplus (A \otimes \neg C)) \oplus ((B \otimes C) \oplus (B \otimes \neg C))) \oplus ((\neg B \otimes \neg A) \otimes (C \oplus \neg C)) \\
& \rightsquigarrow_{\oplus\otimes} & (((A \otimes C) \oplus (A \otimes \neg C)) \oplus ((B \otimes C) \oplus (B \otimes \neg C))) \oplus (((\neg B \otimes \neg A) \otimes C) \oplus ((\neg B \otimes \neg A) \otimes \neg C)) \\
& \equiv & (A \otimes C) \oplus (A \otimes \neg C) \oplus (B \otimes C) \oplus (B \otimes \neg C) \oplus (\neg B \otimes \neg A \otimes C) \oplus (\neg B \otimes \neg A \otimes \neg C)
\end{array}
$$

And the cut-pertinent clause set of φ is:

$$
C_\varphi \equiv \{\vdash A, C \ ; \ C \vdash A \ ; \ \vdash B, C \ ; \ C \vdash B \ ; \ B, A \vdash C \ ; \ B, A, C \vdash\}
$$

4.2.2 Projections

Since a projection's purpose is to replace a leaf in a refutation of a clause set, its end-sequent must contain the leaf's clause as a subsequent. Moreover, if its end-sequent contains any other formula occurrence, then this formula must appear in the end-sequent of the original proof with cuts, because this formula occurrence is propagated downward after the replacement and thus necessarily appears in the end-sequent of the CERes-normal-form. So, if the formula were not in the end-sequent of the original proof, the CERes-normal-form's end-sequent would be necessarily different from that of the original proof with cuts, and this is not intended. Finally, a projection must, of course, be cut-free, otherwise the CERes-normal-form would contain more (and potentially essential) cuts in addition to those inessential atomic cuts originating from the refutation. These three conditions are formally expressed in Definition 4.2.4.

Two alternative ways of constructing projections are defined in this Subsection:
S-projections (first Subsubsection below) and O-projections (second Subsubsection below).

Definition 4.2.4 (Projection)**.** Let φ be a proof with end-sequent $\Gamma \vdash \Delta$ and $c \equiv \Gamma_c \vdash \Delta_c \in C_\varphi$. Any cut-free proof of $(\Gamma', \Gamma_c \vdash \Delta', \Delta_c)\sigma$, where $\Gamma' \subseteq \Gamma$, $\Delta' \subseteq \Delta$ and σ is a substitution, is a *projection* of φ with respect to c.

S-Projections

In this subsection the standard method for the construction of projections is described. It has already been extensively described [13, 7, 10, 9, 102], and it is frequently implicitly defined by an inductive proof of the existence of projections. Here an explicit definition (Definition 4.2.5) is given, which is essentially the proof-recursive algorithm that can be directly extracted from the usual inductive proof of existence of projections. The main advantage of this algorithm is that it is very easy to prove its correctness, exactly because of its closeness to the proof of existence of projections.

Definition 4.2.5 (S-Projection)**.** Let φ be a proof and $c \in C_\varphi$. Let $\Lambda^{\Omega_{CP}(\varphi)}$ denote the $\Omega_{CP}(\varphi)$-pertinent subset of Λ and $\Lambda^{\Omega_{CI}(\varphi)}$ denote the $\Omega_{CP}(\varphi)$-impertinent subset of Λ. Then $\lfloor \varphi' \rfloor_{c'}$ is constructed for every subproof φ' of φ with c' the subclause of c containing occurrences from φ' only:

- If φ' is of the form

$$\frac{}{\Gamma \vdash \Delta}\ axiom$$

then $\lfloor \varphi' \rfloor_{c'}$ is:

$$\frac{}{\Gamma \vdash \Delta}\ axiom$$

and, clearly:

$$\begin{aligned} \Gamma \vdash \Delta &= \Gamma^{\Omega_{CI}(\varphi)}, \Gamma^{\Omega_{CP}(\varphi)} \vdash \Delta^{\Omega_{CI}(\varphi)}, \Delta^{\Omega_{CP}(\varphi)} \\ &= (\Gamma^{\Omega_{CI}(\varphi)} \vdash \Delta^{\Omega_{CI}(\varphi)}) \circ c' \end{aligned}$$

- If φ' is of the form

$$\frac{\begin{matrix}\varphi'_1 \\ \Gamma_1 \vdash \Delta_1\end{matrix} \quad \dots \quad \begin{matrix}\varphi'_n \\ \Gamma_n \vdash \Delta_n\end{matrix}}{\Gamma \vdash \Delta}\ \rho$$

where ρ is a $\Omega_{CP}(\varphi)$-impertinent inference, then $\lfloor \varphi' \rfloor_{c'}$ is:

$$\frac{\begin{matrix}\lfloor \varphi'_1 \rfloor_{c'_1} \\ (\Gamma_1^{\Omega_{CI}(\varphi)} \vdash \Delta_1^{\Omega_{CI}(\varphi)}) \circ c'_1\end{matrix} \quad \dots \quad \begin{matrix}\lfloor \varphi'_n \rfloor_{c'_n} \\ (\Gamma_n^{\Omega_{CI}(\varphi)} \vdash \Delta_n^{\Omega_{CI}(\varphi)}) \circ c'_n\end{matrix}}{(\Gamma^{\Omega_{CI}(\varphi)} \vdash \Delta^{\Omega_{CI}(\varphi)}) \circ c'_1 \circ \ldots \circ c'_n}\ \rho$$

and, by definition of cut-pertinent clause set, $c'_1 \circ \ldots \circ c'_n = c'$

- If φ' is of the form

$$\frac{\begin{matrix}\varphi'_1 \\ \Gamma_1 \vdash \Delta_1\end{matrix} \quad \dots \quad \begin{matrix}\varphi'_n \\ \Gamma_n \vdash \Delta_n\end{matrix}}{\Gamma \vdash \Delta}\ \rho$$

where ρ is a $\Omega_{CP}(\varphi)$-pertinent inference, then, by definition of cut-pertinent clause set, $c' = c'_j$ for some j such that $1 \leq j \leq n$ and hence $\lfloor \varphi' \rfloor_{c'}$ can be constructed as:

$$\begin{array}{c}\lfloor \varphi'_j \rfloor_{c'_j} \\ (\Gamma_j^{\Omega_{CI}(\varphi)} \vdash \Delta_j^{\Omega_{CI}(\varphi)}) \circ c'_j \\ \hline\hline (\Gamma^{\Omega_{CI}(\varphi)} \vdash \Delta^{\Omega_{CI}(\varphi)}) \circ c' \end{array}\ w^*$$

Finally, the *S-projection*[h] of φ with respect to c is:

$$\lfloor \varphi \rfloor_c^S \doteq \lfloor \varphi \rfloor_c$$

Definition 4.2.6 (Skolemized Proof)**.** A proof is *Skolemized* if and only if φ has no cut-impertinent strong quantifier inferences.

Lemma 4.1 (S-Projections are Proofs)**.** Let φ be a skolemized proof and $c \equiv \omega_1^-, \ldots, \omega_n^- \vdash \omega_1^+, \ldots, \omega_m^+ \in C_\varphi$. Then $\lfloor \varphi \rfloor^S_{\omega_1^-,\ldots,\omega_n^- \vdash \omega_1^+,\ldots,\omega_m^+}$ is a proof.

Proof. The only critical case in the inductive construction of S-projections according to Definition 4.2.5 is the case when ρ is a $\Omega_{CP}(\varphi)$-impertinent strong quantifier inference. As c'_j have been merged to the premise sequents, a violation of eigen-variable condition for ρ could occur if c'_j contained the eigen-variable of ρ. However, as φ is assumed to be skolemized, this critical case cannot occur. □

Example 4.4 (Violation of Eigen-Variable Condition in S-Projection of a Non-Skolemized Proof)**.** *Let φ be the non-skolemized proof below:*

$$\dfrac{\dfrac{\dfrac{P(\alpha) \vdash P(\alpha)}{(\forall x)P(x) \vdash P(\alpha)}\ \forall_l}{(\forall x)P(x) \vdash (\forall x)P(x)}\ \forall_r \qquad \dfrac{\dfrac{P(\beta) \vdash P(\beta)}{(\forall x)P(x) \vdash P(\beta)}\ \forall_l}{(\forall x)P(x) \vdash (\forall x)P(x)}\ \forall_r}{(\forall x)P(x) \vdash (\forall x)P(x)}\ cut$$

It cut-pertinent clause set is:

$$C_\varphi \equiv \{\vdash P(\alpha) \ ; \ P(\beta) \vdash\}$$

And $\lfloor \varphi \rfloor^S_{P(\beta)\vdash}$ is:

$$\dfrac{\dfrac{P(\beta) \vdash P(\beta)}{P(\beta) \vdash (\forall x)P(x)}\ \forall_r}{P(\beta), (\forall x)P(x) \vdash (\forall x)P(x)}\ w_l$$

$\lfloor \varphi \rfloor^S_{P(\beta)\vdash}$ *is clearly not a proof, because the eigenvariable condition for the $\forall_r$ inference is violated.*

Lemma 4.2 (S-Projections are Cut-free)**.**
Let φ be a proof and $c \equiv \omega_1^-, \ldots, \omega_n^- \vdash \omega_1^+, \ldots, \omega_m^+ \in C_\varphi$.
Then $\lfloor \varphi \rfloor^S_{\omega_1^-,\ldots,\omega_n^- \vdash \omega_1^+,\ldots,\omega_m^+}$ is cut-free.

[h] S-projections are simply called projections in other publications about cut-elimination by resolution, because they used to be the only projections considered. Here they have a distinct name to distinguish them from projections constructed by other methods, such as O-projections.

Proof. In the inductive construction of S-projections, no cuts are introduced and all $\Omega_{CP}(\varphi)$-inferences (including, of course, all cuts of φ) are skipped or replaced by sequences of weakening inferences. Therefore, S-projections are cut-free. $\square$

Lemma 4.3 (S-Projections have Correct End-Sequents)**.** Let φ be a proof with endsequent $\Gamma \vdash \Delta$ and $c \equiv \omega_1^-, \ldots, \omega_n^- \vdash \omega_1^+, \ldots, \omega_m^+ \in C_\varphi$. Then $\lfloor\varphi\rfloor^S_{\omega_1^-,\ldots,\omega_n^- \vdash \omega_1^+,\ldots,\omega_m^+}$ has end-sequent $\Gamma, \omega_1^-, \ldots, \omega_n^- \vdash \Delta, \omega_1^+, \ldots, \omega_m^+$.

Proof. In the inductive construction of S-Projections, for every subproof φ' of φ, $\lfloor\varphi'\rfloor_{c'}$ has an end-sequent that is the composition of the $\Omega_{CP}(\varphi)$-impertinent subsequent of the end-sequent of φ' with the subclause c' of c containing only atoms that occur in φ'. For $\varphi' = \varphi$, the $\Omega_{CP}(\varphi)$-impertinent subsequent of $\Gamma \vdash \Delta$ is $\Gamma \vdash \Delta$ itself and c' is equal to c. Therefore, the end-sequent of $\lfloor\varphi\rfloor^S_{\omega_1^-,\ldots,\omega_n^- \vdash \omega_1^+,\ldots,\omega_m^+}$ is $\Gamma, \omega_1^-, \ldots, \omega_n^- \vdash \Delta, \omega_1^+, \ldots, \omega_m^+$. $\square$

Theorem 4.2 (Correctness of S-Projections)**.** A S-projection of a skolemized proof φ with respect to a clause c is a projection of φ with respect to c.

Proof. This is an immediate consequence of Lemmas 4.1, 4.2 and 4.3. $\square$

Example 4.5 (S-Projection)**.** *Consider the proof φ of Example 4.1:*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad B \vdash B}{A, B \vdash A \wedge B}\wedge_r}{A \wedge B \vdash A \wedge B}\wedge_l \qquad \dfrac{\dfrac{B \vdash B \qquad A \vdash A}{A, B \vdash B \wedge A}\wedge_r}{A \wedge B \vdash B \wedge A}\wedge_l}{A \wedge B \vdash B \wedge A}\,cut \qquad \dfrac{C \vdash C \qquad C \vdash C}{C \vdash C}\,cut}{(A \wedge B) \vee C \vdash B \wedge A, C}\vee_l}{}$$

Then, the projections $\lfloor\varphi\rfloor^S_{\vdash A,C}$ and $\lfloor\varphi\rfloor^S_{\vdash B,C}$ are:

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{A, B \vdash A}w_l}{A \wedge B \vdash A}\wedge_l}{A \wedge B \vdash A, B \wedge A}w_r \qquad \dfrac{C \vdash C}{C \vdash C, C}w_r}{(A \wedge B) \vee C \vdash A, C, B \wedge A, C}\vee_l \qquad\qquad \dfrac{\dfrac{\dfrac{\dfrac{B \vdash B}{A, B \vdash B}w_l}{A \wedge B \vdash B}\wedge_l}{A \wedge B \vdash B, B \wedge A}w_r \qquad \dfrac{C \vdash C}{C \vdash C, C}w_r}{(A \wedge B) \vee C \vdash B, C, B \wedge A, C}\vee_l$$

The projections $\lfloor\varphi\rfloor^S_{C \vdash A}$ and $\lfloor\varphi\rfloor^S_{C \vdash B}$ are:

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{A, B \vdash A}w_l}{A \wedge B \vdash A}\wedge_l}{A \wedge B \vdash A, B \wedge A}w_r \qquad \dfrac{C \vdash C}{C, C \vdash C}w_l}{(A \wedge B) \vee C, C \vdash A, B \wedge A, C}\vee_l \qquad\qquad \dfrac{\dfrac{\dfrac{\dfrac{B \vdash B}{A, B \vdash B}w_l}{A \wedge B \vdash B}\wedge_l}{A \wedge B \vdash B, B \wedge A}w_r \qquad \dfrac{C \vdash C}{C, C \vdash C}w_l}{(A \wedge B) \vee C, C \vdash B, B \wedge A, C}\vee_l$$

The projections $\lfloor\varphi\rfloor^S_{B,A\vdash C}$ *and* $\lfloor\varphi\rfloor^S_{B,A,C\vdash}$ *are:*

$$\dfrac{\dfrac{\dfrac{B \vdash B \qquad A \vdash A}{B, A, \vdash B \wedge A}\,\wedge_r}{A \wedge B, B, A \vdash B \wedge A}\,w_l \qquad \dfrac{C \vdash C}{C \vdash C, C}\,w_r}{(A \wedge B) \vee C, B, A \vdash C, B \wedge A, C}\,\vee_l \qquad \dfrac{\dfrac{\dfrac{B \vdash B \qquad A \vdash A}{B, A \vdash B \wedge A}\,\wedge_r}{A \wedge B, B, A \vdash B \wedge A}\,w_l \qquad \dfrac{C \vdash C}{C, C \vdash C}\,w_l}{(A \wedge B) \vee C, B, A, C \vdash B \wedge A, C}\,\vee_l$$

O-Projections

Construction algorithms directly extracted from proofs of existence are usually not optimal[i], and this is the case for the algorithm that constructs S-projections. S-projections are very redundant, simply because their end-sequents must always contain the whole end-sequent of the original proof with cuts.

This Subsubsection describes a method to construct less redundant projections, which are called O-projections. Many auxiliary definitions are needed to formalize precisely certain proof transformations. While these auxiliary definitions might make the construction of O-projections seem to be a quite technical procedure, the intuitive idea behind this construction is quite simple and perhaps even more intuitive than the proof-recursive construction of S-projections. The method follows roughly three steps: firstly, cut-pertinent inferences are "deleted" (so that cut-pertinent occurrences are propagated down to the end-sequent, and thus the end-sequent now contains every clause of the clause set); then, occurrences and inferences that are not in a certain sense relevant to the occurrences of the specific clause under consideration are also "deleted" (this guarantees that only that clause will occur in the end-sequent of the projection); and finally, the problems introduced by the previous two steps are fixed.

Most of the auxiliary transformations make use of Y-inferences. This is because inferences cannot be simply "deleted", especially if they have arity greater than one, since then it would not be clear how to merge the subproofs of all branches. That is why inferences are actually not deleted, but rather replaced by Y-inferences of arbitrary arity. Later, the Y-inferences can be eliminated (Definition 4.2.19).

Definition 4.2.7 (Y Rule)**.** The Y rule of inference is shown below:

$$\dfrac{\begin{matrix}\varphi_1 \\ \Gamma_1 \vdash \Delta_1\end{matrix} \qquad \ldots \qquad \begin{matrix}\varphi_n \\ \Gamma_n \vdash \Delta_n\end{matrix}}{\Gamma_1, \ldots, \Gamma_n \vdash \Delta_1 \ldots, \Delta_n}\,Y$$

[i] For another example of non-optimal algorithms extracted from proofs, note that reductive methods of cut-elimination can be seen as algorithms directly extracted from Gentzen's proof of the cut-elimination theorem, which is essentially a proof of existence of cut-free proofs.

Definition 4.2.8 (Inference Replacement)**.** Let φ be the proof below:

$$\dfrac{\vdots \qquad \dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1,\Gamma_1^{\rho}\vdash\Delta_1,\Delta_1^{\rho}\end{array} \qquad \cdots \qquad \begin{array}{c}\varphi_n\\ \Gamma_n,\Gamma_n^{\rho}\vdash\Delta_n,\Delta_n^{\rho}\end{array}}{\Gamma_1,\ldots,\Gamma_n,\Gamma^{\rho_1}\vdash\Delta_1,\ldots,\Delta_n,\Delta^{\rho_1}}\,\rho_1 \qquad \vdots}{\Gamma\vdash\Delta}\,\psi$$

Then the proof φ' shown below is the result of replacing the inference ρ_1 by the inference sequence ρ_2^*.

$$\dfrac{\vdots \qquad \dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1,\Gamma_1^{\rho}\vdash\Delta_1,\Delta_1^{\rho}\end{array} \qquad \cdots \qquad \begin{array}{c}\varphi_n\\ \Gamma_n,\Gamma_n^{\rho}\vdash\Delta_n,\Delta_n^{\rho}\end{array}}{\Gamma_1,\ldots,\Gamma_n,\Gamma^{\rho_2^*},\Pi\vdash\Delta_1,\ldots,\Delta_n,\Delta^{\rho_2^*},\Lambda}\,\rho_2^* \qquad \vdots}{\Gamma',\Pi\vdash\Delta',\Lambda}\,\psi'$$

where ψ' is the proof thread ψ with all descendant occurrences of Γ^{ρ_1} and Δ^{ρ_1} replaced by the corresponding occurrences of $\Gamma^{\rho_2^*}$ and $\Delta^{\rho_2^*}$ and having the extra conclusion occurrences of ρ_2^*, Π and Λ, added to each of its sequents (i.e. the extra occurrences are propagated downward to the end-sequent).

Example 4.6 (Inference Replacement)**.** *Let φ be the proof below:*

$$\dfrac{A\vdash A \qquad A\vdash A}{A\vdash A}\,cut$$

Then φ_Y below is the result of replacing cut by a Y-inference:

$$\dfrac{A\vdash A \qquad A\vdash A}{A,A\vdash A,A}\,Y$$

And $\varphi_{\neg\wedge}$ below is the result of replacing cut by a $\neg_r$ in the right branch and a $\wedge_r$:

$$\dfrac{A\vdash A \qquad \dfrac{A\vdash A}{\vdash A,\neg A}\,\neg_r}{A\vdash A,(A\wedge\neg A)}\,\wedge_r$$

Definition 4.2.9 (Pertinent Y-Replacement)**.** $Y_{\oplus}^{\Omega}(\varphi)$ denotes the result of replacing all Ω-pertinent inferences in φ by Y inferences, called the Ω-*pertinent Y-replacement* of φ.

Example 4.7 (Pertinent Υ-Replacement)**.** *Let φ be the proof below, where the cut-pertinent atomic occurrences and inferences have been highlighted:*

$$
\begin{prooftree}
\AxiomC{$A \vdash {\color{red}A}$}
\AxiomC{$B \vdash {\color{red}B}$}
\RightLabel{${\color{red}\wedge_r}$}
\BinaryInfC{$A, B \vdash {\color{red}A} \wedge {\color{red}B}$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash {\color{red}A} \wedge {\color{red}B}$}
\AxiomC{${\color{red}B} \vdash B$}
\AxiomC{${\color{red}A} \vdash A$}
\RightLabel{$\wedge_r$}
\BinaryInfC{${\color{red}A}, {\color{red}B} \vdash B \wedge A$}
\RightLabel{${\color{red}\wedge_l}$}
\UnaryInfC{${\color{red}A} \wedge {\color{red}B} \vdash B \wedge A$}
\RightLabel{${\color{red}cut}$}
\BinaryInfC{$A \wedge B \vdash B \wedge A$}
\AxiomC{$C \vdash {\color{red}C}$}
\AxiomC{${\color{red}C} \vdash C$}
\RightLabel{${\color{red}cut}$}
\BinaryInfC{$C \vdash C$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(A \wedge B) \vee C \vdash B \wedge A, C$}
\end{prooftree}
$$

Then $\Upsilon_{\oplus}^{\Omega_{CP}(\varphi)}(\varphi)$ is:

$$
\begin{prooftree}
\AxiomC{$A \vdash {\color{red}A}$}
\AxiomC{$B \vdash {\color{red}B}$}
\RightLabel{${\color{red}\Upsilon_{\oplus}}$}
\BinaryInfC{$A, B \vdash {\color{red}A}, {\color{red}B}$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash {\color{red}A}, {\color{red}B}$}
\AxiomC{${\color{red}B} \vdash B$}
\AxiomC{${\color{red}A} \vdash A$}
\RightLabel{$\wedge_r$}
\BinaryInfC{${\color{red}A}, {\color{red}B} \vdash B \wedge A$}
\RightLabel{${\color{red}\Upsilon_{\oplus}}$}
\UnaryInfC{${\color{red}A}, {\color{red}B} \vdash B \wedge A$}
\RightLabel{${\color{red}\Upsilon_{\oplus}}$}
\BinaryInfC{$A \wedge B, {\color{red}A}, {\color{red}B} \vdash {\color{red}A}, {\color{red}B}, B \wedge A$}
\AxiomC{$C \vdash {\color{red}C}$}
\AxiomC{${\color{red}C} \vdash C$}
\RightLabel{${\color{red}\Upsilon_{\oplus}}$}
\BinaryInfC{$C, C \vdash C, C$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(A \wedge B) \vee C, {\color{red}A}, {\color{red}B}, {\color{red}C} \vdash {\color{red}A}, {\color{red}B}, B \wedge A, C, {\color{red}C}$}
\end{prooftree}
$$

Definition 4.2.10 (Impertinent Υ-Replacement)**.** $\Upsilon_{\otimes}^{\Omega}(\varphi)$ denotes the result of replacing all Ω-impertinent inferences in φ by Υ inferences, called the Ω-*impertinent Υ-replacement* of φ.

Definition 4.2.11 (Eigen-Υ-Replacement)**.** A Ω-pertinent Υ-replacement of φ is an *eigen-Υ-replacement* if and only if $\Upsilon_{\oplus}^{\Omega}(\varphi)$ violates no eigen-variable conditions. In this case, φ is called Ω-*pertinent eigen-Υ-replaceable*.

A Ω-impertinent Υ-replacement of φ is an *eigen-Υ-replacement* if and only if $\Upsilon_{\otimes}^{\Omega}(\varphi)$ violates no eigen-variable conditions. In this case, φ is called Ω-*impertinent eigen-Υ-replaceable*.

Example 4.8 (Υ-replacement with Eigen-Variable Violations)**.** *Let φ be the proof below:*

$$
\begin{prooftree}
\AxiomC{$P(\alpha) \vdash P(\alpha)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$(\forall x)P(x) \vdash P(\alpha)$}
\RightLabel{$\forall_r$}
\UnaryInfC{$(\forall x)P(x) \vdash (\forall x)P(x)$}
\AxiomC{$P(\alpha) \vdash P(\alpha)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$(\forall x)P(x) \vdash P(\alpha)$}
\RightLabel{$\forall_r$}
\UnaryInfC{$(\forall x)P(x) \vdash (\forall x)P(x)$}
\RightLabel{cut}
\BinaryInfC{$(\forall x)P(x) \vdash (\forall x)P(x)$}
\end{prooftree}
$$

Then, $\Upsilon_{\oplus}^{\Omega_{CP}(\varphi)}(\varphi)$, shown below, has a violation of the eigenvariable condition for the $\forall_r$ inference:

$$
\begin{prooftree}
\AxiomC{$P(\alpha) \vdash P(\alpha)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$(\forall x)P(x) \vdash P(\alpha)$}
\RightLabel{Υ}
\UnaryInfC{$(\forall x)P(x) \vdash P(\alpha)$}
\AxiomC{$P(\alpha) \vdash P(\alpha)$}
\RightLabel{Υ}
\UnaryInfC{$P(\alpha) \vdash P(\alpha)$}
\RightLabel{$\forall_r$}
\UnaryInfC{$P(\alpha) \vdash (\forall x)P(x)$}
\RightLabel{Υ}
\BinaryInfC{$(\forall x)P(x), P(\alpha) \vdash P(\alpha), (\forall x)P(x)$}
\end{prooftree}
$$

Lemma 4.4 ($\Omega_{CP}(\varphi)$-pert. Eigen-Y-replaceability of Skolemized Proofs)**.** If φ is a Skolemized proof, then φ is $\Omega_{CP}(\varphi)$-pertinent eigen-Y-replaceable.

Proof. By definition of Skolemized proof (Definition 4.2.6), φ has no cut-impertinent strong quantifier inferences. By definition of pertinent Y-replacement, any cut-pertinent strong quantifier inference in φ are replaced by Y inferences in the $\Omega_{CP}(\varphi)$-pertinent Y-replacement $Y_{\oplus}^{\Omega_{CP}(\varphi)}(\varphi)$. Therefore, $Y_{\oplus}^{\Omega_{CP}(\varphi)}(\varphi)$ has no strong quantifier inference, and hence violates no eigen-variable condition. By definition of eigen-replaceability (Definition 4.2.11), φ is therefore $\Omega_{CP}(\varphi)$-pertinent eigen-Y-replaceable. □

Example 4.9 (Non-Skolemized but $\Omega_{CP}(\varphi)$-pertinent Eigen-Y-Replaceable Proof)**.** *Let φ be the proof below:*

$$
\dfrac{\dfrac{P \vdash P \qquad P \vdash P}{P \vdash P}\,cut \qquad \dfrac{\dfrac{Q(\alpha) \vdash Q(\alpha)}{(\forall x)Q(x) \vdash Q(\alpha)}\,\forall_l}{(\forall x)Q(x) \vdash (\forall x)Q(x)}\,\forall_r}{P, (\forall x)Q(x) \vdash P \wedge (\forall x)Q(x)}\,\wedge_r
$$

Then, even though φ is not skolemized, $Y_{\oplus}^{\Omega_{CP}(\varphi)}(\varphi)$, shown below, has no violation of eigenvariable conditions:

$$
\dfrac{\dfrac{P \vdash P \qquad P \vdash P}{P, P \vdash P, P}\,Y \qquad \dfrac{\dfrac{Q(\alpha) \vdash Q(\alpha)}{(\forall x)Q(x) \vdash Q(\alpha)}\,\forall_l}{(\forall x)Q(x) \vdash (\forall x)Q(x)}\,\forall_r}{P, P, (\forall x)Q(x) \vdash P, P \wedge (\forall x)Q(x)}\,\wedge_r
$$

Definition 4.2.12 (Axiom Linkage)**.** Two atomic (sub)formula occurrences ω_1 and ω_2 in a proof φ are *axiom-linked*, denoted $\omega_1 \multimap\!\!\bullet\; \omega_2$, if and only if there are formula occurrences ω_1' and ω_2' in an axiom sequent of φ such that $\omega_1' \searrow \omega_1$ and $\omega_2' \searrow \omega_2$.

Definition 4.2.13 (Axiom-Linked Set of Occurrences)**.** Let ω be an atomic (sub)formula occurrence in a proof φ. The *set of occurrences axiom-linked to ω in φ* is:

$$\Omega_{\omega}(\varphi) \doteq \{\omega_i | \omega_i \multimap\!\!\bullet\; \omega\}$$

Example 4.10 (Axiom-Linked Sets of Occurrences). *Let φ and $Y_{\oplus}^{\Omega_{CP}(\varphi)}(\varphi)$ be the proofs shown in Example 4.7. These proofs are shown again below, with each color representing a set of mutually axiom-linked atomic occurrences.*

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad B \vdash B}{A, B \vdash A \wedge B}\,\wedge_r}{A \wedge B \vdash A \wedge B}\,\wedge_l \qquad \dfrac{\dfrac{B \vdash B \qquad A \vdash A}{A, B \vdash B \wedge A}\,\wedge_r}{A \wedge B \vdash B \wedge A}\,\wedge_l}{A \wedge B \vdash B \wedge A}\,cut \qquad \dfrac{C \vdash C \qquad C \vdash C}{C \vdash C}\,cut}{(A \wedge B) \vee C \vdash B \wedge A, C}\,\vee_l$$

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad B \vdash B}{A, B \vdash A, B}\,Y_{\oplus}}{A \wedge B \vdash A, B}\,\wedge_l \qquad \dfrac{\dfrac{B \vdash B \qquad A \vdash A}{A, B \vdash B \wedge A}\,\wedge_r}{A, B \vdash B \wedge A}\,Y_{\oplus}}{A \wedge B, A, B \vdash A, B, B \wedge A}\,Y_{\oplus} \qquad \dfrac{C \vdash C \qquad C \vdash C}{C, C \vdash C, C}\,Y_{\oplus}}{(A \wedge B) \vee C, A, B, C \vdash A, B, B \wedge A, C, C}\,\vee_l$$

Definition 4.2.14 (Proofoid). The *proofoid* $(\!|\varphi|\!)_\Omega$ of a proof φ with respect to a set of occurrences Ω (an Ω-*proofoid*) is obtained by:

1. performing Ω-impertinent Y-replacement.
2. removing all Ω-impertinent formula occurrences.

Definition 4.2.15 (Proof-Slice). The *slice* of a proof φ with respect to a set of atomic occurrences $\{\omega_1, \ldots, \omega_n\}$ is:

$$\wr \varphi \int_{\{\omega_1,\ldots,\omega_n\}} \doteq (\!|\varphi|\!)_{\bigcup_{i=1}^{n} \Omega_{\omega_i}(\varphi)}$$

Remark 4.2.1. A slice of φ is an Ω-proofoid of φ with the additional restriction that Ω should be the union of sets of axiom-linked atomic occurrences.

Example 4.11 (Slice). *Consider $Y_{\oplus}^{\Omega_{CP}(\varphi)}(\varphi)$ shown in Example 4.7. Then $\wr Y_{\oplus}^{\Omega_{CP}(\varphi)}(\varphi) \int_{\{A,C\}}$ is:*

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad \vdash}{A \;\; \vdash A}\,Y_{\oplus}}{A \wedge B \vdash A}\,\wedge_l \qquad \dfrac{\dfrac{\vdash \qquad \vdash}{\vdash}\,Y_{\otimes}}{\vdash}\,Y_{\otimes}}{A \wedge B \quad \vdash A}\,Y_{\oplus} \qquad \dfrac{C \vdash C \qquad \vdash}{C \;\; \vdash C}\,Y_{\oplus}}{(A \wedge B) \vee C \quad \vdash A, \quad C}\,\vee_l$$

The $\wedge_l$ inference in the proofoid above is "broken", since one of its auxiliary formula occurrences is missing. This motivates Definitions 4.2.16 and 4.2.17, which aim at fixing such problems.

Definition 4.2.16 (Broken Inference)**.** An inference ρ is *broken* in a proofoid ψ if and only if some of its auxiliary occurrences are missing.

Definition 4.2.17 (W-Fixing)**.** Fixing a broken inference ρ that is not a contraction can be done by adding weakening inferences according to the proof rewriting rule shown below:

$$\dfrac{\begin{matrix}\varphi_1 \\ \Gamma_1 \vdash \Delta_1\end{matrix} \quad \ldots \quad \begin{matrix}\varphi_n \\ \Gamma_n \vdash \Delta_n\end{matrix}}{\Gamma \vdash \Delta}\,\rho$$

$$\Downarrow$$

$$\dfrac{\dfrac{\begin{matrix}\varphi_1 \\ \Gamma_1 \vdash \Delta_1\end{matrix}}{\Gamma_1, \Gamma'_1 \vdash \Delta_1, \Delta'_1}\,w^* \quad \ldots \quad \dfrac{\begin{matrix}\varphi_n \\ \Gamma_n \vdash \Delta_n\end{matrix}}{\Gamma_n, \Gamma'_n \vdash \Delta_n, \Delta'_n}\,w^*}{\Gamma \vdash \Delta}\,\rho$$

where Γ'_j and Δ'_j $(1 \leq j \leq n)$ are the missing auxiliary occurrences of ρ.

If ρ is a broken contraction, simply skipping it is better than adding weakening inferences to fix it. This is done with the following proof rewriting rules:

$$\dfrac{\begin{matrix}\varphi' \\ \Gamma \vdash \Delta, F\end{matrix}}{\Gamma \vdash \Delta, F}\,c_r$$

$$\Downarrow$$

$$\begin{matrix}\varphi' \\ \Gamma \vdash \Delta, F\end{matrix}$$

$$\dfrac{\begin{matrix}\varphi' \\ F, \Gamma \vdash \Delta\end{matrix}}{F, \Gamma \vdash \Delta}\,c_l$$

$$\Downarrow$$

$$\begin{matrix}\varphi' \\ F, \Gamma \vdash \Delta\end{matrix}$$

Example 4.12 (W-Fixing)**.** *Consider* $\wr Y_{\oplus}^{\Omega_{CP}(\varphi)}(\varphi)\wr_{\{A,C\}}$ *shown in Example 4.11. Then* $W_{\text{fix}}(\wr Y_{\oplus}^{\Omega_{CP}(\varphi)}(\varphi)\wr_{\{A,C\}})$ *is:*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad \vdash}{A, \;\vdash A}\,Y_\oplus}{A, B \vdash A}\,w_l}{A \wedge B \vdash A}\,\wedge_l \qquad \dfrac{\dfrac{\vdash \qquad \vdash}{\vdash}\,Y_\otimes}{\vdash}\,Y_\otimes}{A \wedge B \quad \vdash A}\,Y_\oplus \qquad \dfrac{C \vdash C \qquad \vdash}{C \quad \vdash C}\,Y_\oplus}{(A \wedge B) \vee C \quad \vdash A, \quad C}\,\vee_l$$

Definition 4.2.18 (Argmin)**.** Let S be a set and $f : S \to \mathbb{N}$.
Then $\underset{x \in S}{\text{argmin}}\, f(x)$ denotes an $x \in S$ such that for all $y \in S$, $f(x) \leq f(y)$.

Definition 4.2.19 (Y-Elimination)**.** The elimination of Y inferences follows the proof rewriting rule shown below:

$$\dfrac{\begin{matrix}\varphi_1 \\ \Gamma_1 \vdash \Delta_1\end{matrix} \qquad \ldots \qquad \begin{matrix}\varphi_n \\ \Gamma_n \vdash \Delta_n\end{matrix}}{\Gamma_1, \ldots, \Gamma_n \vdash \Delta_1 \ldots, \Delta_n}\,Y$$

$$\Downarrow$$

$$\dfrac{\begin{matrix}\varphi_j \\ \Gamma_j \vdash \Delta_j\end{matrix}}{\Gamma_1, \ldots, \Gamma_n \vdash \Delta_1 \ldots, \Delta_n}\,w^*$$

as long as the following contraints are satisfied:

- The proofoids $\varphi_1, \ldots, \varphi_n$ should not contain Y-inferences (this enforces an uppermost Y-elimination strategy and guarantees confluence).

- Let $P \doteq \{\psi | \psi = \varphi_i \text{ and } 1 \leq i \leq n \text{ and } \varphi_i \text{ is a proof}\}$. If $P \neq \emptyset$, then $\varphi_j = \underset{\psi \in P}{\text{argmin}}\, |\psi|$.

$[\![\varphi]\!]$ denotes the normal-form of φ according to the Y-elimination proof rewriting rules.

Example 4.13 (Y-Elimination). *Consider $W_{\text{fix}}(\wr Y_{\oplus}^{\Omega_{CP}(\varphi)}(\varphi)\wr_{\{A,C\}})$ shown in Example 4.12. A Y-elimination rewriting sequence for it is shown below:*

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\AxiomC{$\vdash$}
\RightLabel{$Y_\oplus$}
\BinaryInfC{$A, \vdash A$}
\RightLabel{w_l}
\UnaryInfC{$A, B \vdash A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash A$}
\AxiomC{$\vdash$}
\AxiomC{$\vdash$}
\RightLabel{$Y_\otimes$}
\BinaryInfC{$\vdash$}
\RightLabel{$Y_\otimes$}
\UnaryInfC{$\vdash$}
\RightLabel{$Y_\oplus$}
\BinaryInfC{$A \wedge B \vdash A$}
\AxiomC{$C \vdash C$}
\AxiomC{$\vdash$}
\RightLabel{$Y_\oplus$}
\BinaryInfC{$C \vdash C$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(A \wedge B) \vee C \vdash A, C$}
\end{prooftree}
$$

$$\Downarrow$$

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\RightLabel{w_l}
\UnaryInfC{$A, B \vdash A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash A$}
\AxiomC{$\vdash$}
\AxiomC{$\vdash$}
\RightLabel{$Y_\otimes$}
\BinaryInfC{$\vdash$}
\RightLabel{$Y_\otimes$}
\UnaryInfC{$\vdash$}
\RightLabel{$Y_\oplus$}
\BinaryInfC{$A \wedge B \vdash A$}
\AxiomC{$C \vdash C$}
\AxiomC{$\vdash$}
\RightLabel{$Y_\oplus$}
\BinaryInfC{$C \vdash C$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(A \wedge B) \vee C \vdash A, C$}
\end{prooftree}
$$

$$\Downarrow$$

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\RightLabel{w_l}
\UnaryInfC{$A, B \vdash A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash A$}
\AxiomC{$\vdash$}
\RightLabel{$Y_\otimes$}
\UnaryInfC{$\vdash$}
\RightLabel{$Y_\oplus$}
\BinaryInfC{$A \wedge B \vdash A$}
\AxiomC{$C \vdash C$}
\AxiomC{$\vdash$}
\RightLabel{$Y_\oplus$}
\BinaryInfC{$C \vdash C$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(A \wedge B) \vee C \vdash A, C$}
\end{prooftree}
$$

$$\Downarrow$$

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\RightLabel{w_l}
\UnaryInfC{$A, B \vdash A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash A$}
\AxiomC{$\vdash$}
\RightLabel{$Y_\oplus$}
\BinaryInfC{$A \wedge B \vdash A$}
\AxiomC{$C \vdash C$}
\AxiomC{$\vdash$}
\RightLabel{$Y_\oplus$}
\BinaryInfC{$C \vdash C$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(A \wedge B) \vee C \vdash A, C$}
\end{prooftree}
$$

$$\Downarrow$$

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\RightLabel{w_l}
\UnaryInfC{$A, B \vdash A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash A$}
\AxiomC{$C \vdash C$}
\AxiomC{$\vdash$}
\RightLabel{$Y_\oplus$}
\BinaryInfC{$C \vdash C$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(A \wedge B) \vee C \vdash A, C$}
\end{prooftree}
$$

$$\Downarrow$$

$$\dfrac{\dfrac{\dfrac{A \ \vdash A}{A, B \vdash A}\,w_l}{A \wedge B \vdash A}\,\wedge_l \qquad C \vdash C}{(A \wedge B) \vee C \qquad \vdash A, \qquad\qquad C}\,\vee_l$$

Therefore $[\![W_{\mathrm{fix}}(\wr Y_{\oplus}^{\Omega_{CP}(\varphi)}(\varphi)\wr_{\{A,C\}})]\!]$ *is:*

$$\dfrac{\dfrac{\dfrac{A \vdash A}{A, B \vdash A}\,w_l}{A \wedge B \vdash A}\,\wedge_l \qquad C \vdash C}{(A \wedge B) \vee C \vdash A, C}\,\vee_l$$

Definition 4.2.20 (Necessary Empty $\otimes$-juncts)**.** Let S be a $\otimes$-junct in a $\oplus\otimes$-normal-form of the cut-pertinent struct $\mathcal{S}_\varphi$ of a proof φ. If S has an empty $\otimes$-junct occurrence $\epsilon^*_\otimes$ as a substruct, $\epsilon^*_\otimes$ is called *necessary* if and only if there is an inference ρ in φ and a formula occurrence ω in S such that ρ is simultaneously $\Omega_\omega(\varphi)$-pertinent and $\Omega_{\epsilon^*_\otimes}(\varphi)$-pertinent.

Definition 4.2.21 (O-Projection (With Respect to a $\otimes$-junct))**.** Let φ be a proof and S be a $\otimes$-junct in a $\oplus\otimes$-normal-form of the cut-pertinent struct $\mathcal{S}_\varphi$. Let S' be S after removal of all unnecessary empty $\otimes$-juncts. S' is of the form $S_1 \otimes \ldots \otimes S_n$ where S_i is either a formula occurrence ω_i, its dual $\neg\omega_i$ or a necessary empty $\otimes$-junct $\epsilon^i_\otimes$. The *O-projection* of φ with respect to $S_1 \otimes \ldots \otimes S_n$ is:

$$\lfloor \varphi \rfloor^O_S \doteq [\![W_{\mathrm{fix}}(\wr Y_{\oplus}^{\Omega_{CP}(\varphi)}(\varphi)\wr_{\{\varpi_1,\ldots,\varpi_n\}})]\!]$$

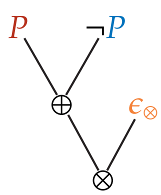
where ϖ_i is the corresponding ω_i or $\epsilon^i_\otimes$.

Remark 4.2.2. The reason why O-projections must take the empty $\otimes$-juncts $\epsilon^i_\otimes$ into account is that otherwise $\wr Y_{\oplus}^{\Omega_{CP}(\varphi)}(\varphi)\wr_{\{\varpi_1,\ldots,\varpi_n\}}$ would contain non-fixable broken binary inferences.

Example 4.14 (O-Projection (With Respect to a $\otimes$-junct))**.** *Let φ be the proof below:*

$$\dfrac{\dfrac{P \vdash P \qquad P \vdash P}{P \vdash P}\,cut \qquad Q \vdash Q}{P, Q \vdash P \wedge Q}\,\wedge_r$$

Its cut-pertinent struct $\mathcal{S}_\varphi$ is:



It can be $\rightsquigarrow_{\oplus\otimes}$*-normalized to:*

$$(P \otimes \epsilon_{\otimes}) \oplus (\neg P \otimes \epsilon_{\otimes})$$

Then $\lfloor\varphi\rfloor^{O}_{P\otimes\epsilon_{\otimes}}$ *is shown below (note that* $\epsilon_{\otimes}$ *is unnecessary in* $P \otimes \epsilon_{\otimes}$*):*

$$P \vdash P$$

And $\lfloor\varphi\rfloor^{O}_{\neg P\otimes\epsilon_{\otimes}}$ *is shown below (note that* $\epsilon_{\otimes}$ *is necessary in* $\neg P \otimes \epsilon_{\otimes}$*):*

$$\dfrac{P \vdash P \qquad Q \vdash Q}{P, Q \vdash P \wedge Q}\ \wedge_r$$

Remark 4.2.3. Interestingly, the distinction between necessary and unnecessary empty $\otimes$-juncts is only required in the case of $\rightsquigarrow_{\oplus\otimes}$-normalization. If $\rightsquigarrow_{\oplus\otimes_P}$-normalization (Definition 4.4.1) and $\rightsquigarrow_{\oplus\otimes_W}$-normalization (Definition 4.3.12) are used, all empty $\otimes$-juncts are necessary.

Definition 4.2.22 (O-Projection)**.**
Let φ be a proof and c be clause from C_{φ} and S be the $\otimes$-junct whose clausification resulted in c. The *O-projection* of φ with respect to the clause c is:

$$\lfloor\varphi\rfloor^{O}_{c} \doteq \lfloor\varphi\rfloor^{O}_{S}$$

Remark 4.2.4. The reason why the O-projection with respect to a clause is defined as the O-projection with respect to its corresponding $\otimes$-junct is that the $\otimes$-junct still contains information about the empty $\otimes$-juncts (which originate from axiom sequents containing no cut-pertinent formula occurrences), which is lost during clausification.

Lemma 4.5 (O-Projections are Proofs)**.**
Let φ be a $\Omega_{CP}(\varphi)$-pertinent eigen-Y-replaceable proof and $c \equiv \omega_1^-, \ldots, \omega_n^- \vdash \omega_1^+, \ldots, \omega_m^+ \in C_{\varphi}$. Then $\lfloor\varphi\rfloor^{O}_{\omega_1^-,\ldots,\omega_n^- \vdash \omega_1^+,\ldots,\omega_m^+}$ is a proof.

Proof. All Y inferences are eliminated. No eigen-variable conditions are violated, since φ is $\Omega_{CP}(\varphi)$-pertinent eigen-Y-replaceable. All other inferences are correct due to W-fixing. $\square$

Lemma 4.6 (O-Projections are Cut-free)**.**
Let φ be a proof and $c \equiv \omega_1^-, \ldots, \omega_n^- \vdash \omega_1^+, \ldots, \omega_m^+ \in C_{\varphi}$.
Then $\lfloor\varphi\rfloor^{O}_{\omega_1^-,\ldots,\omega_n^- \vdash \omega_1^+,\ldots,\omega_m^+}$ is cut-free.

Proof. $\Upsilon_{\oplus}^{\Omega_{CP}(\varphi)}(\varphi)$ is the result of replacing all cut-pertinent inferences in φ by Υ inferences. Since cuts are cut-pertinent, they are also replaced. Therefore $\lfloor \varphi \rfloor^{O}_{\omega_1^-,\ldots,\omega_n^- \vdash \omega_1^+,\ldots,\omega_m^+}$ must be cut-free, because no procedure in the construction of O-projections adds cuts. □

Lemma 4.7 (O-Projections have Correct End-Sequents)**.** Let φ be a proof with endsequent $\Gamma \vdash \Delta$ and $c \equiv \omega_1^-, \ldots, \omega_n^- \vdash \omega_1^+, \ldots, \omega_m^+ \in C_\varphi$. Then $\lfloor \varphi \rfloor^{O}_{\omega_1^-,\ldots,\omega_n^- \vdash \omega_1^+,\ldots,\omega_m^+}$ has end-sequent $\Gamma', \omega_1^-, \ldots, \omega_n^- \vdash \Delta', \omega_1^+, \ldots, \omega_m^+$, with $\Gamma' \subseteq \Gamma$ and $\Delta' \subseteq \Delta$.

Proof. $\omega_1^-, \ldots, \omega_n^-, \omega_1^+, \ldots, \omega_m^+$ are ancestors of cuts and, after cut-pertinent Υ-replacement, they will appear in the end-sequent of $\Upsilon_{\oplus}^{\Omega_{CP}(\varphi)}(\varphi)$. Moreover, $\omega_1^-, \ldots, \omega_n^-, \omega_1^+, \ldots, \omega_m^+$ are $((\bigcup_{i=1}^{n} \Omega_{\omega_i^-}(\varphi)) \cup (\bigcup_{i=1}^{m} \Omega_{\omega_i^+}(\varphi)))$-pertinent, and hence they are not removed during slicing. On the other hand, slicing removes other cut-pertinent occurrences (i.e. it removes occurrences from other clauses from the end-sequent), since these occurrences are necessarily $(\bigcup_{\omega' \in \{\omega_1^-,\ldots,\omega_n^-,\omega_1^+,\ldots,\omega_m^+\}} \Omega_{\omega'}(\varphi))$-impertinent.

Γ' and Δ' are the multisets having the formula occurrences of Γ and Δ which are $(\bigcup_{\omega' \in \{\omega_1^-,\ldots,\omega_n^-,\omega_1^+,\ldots,\omega_m^+\}} \Omega_{\omega'}(\varphi))$-partially-pertinent. The formula occurrences that are $(\bigcup_{\omega' \in \{\omega_1^-,\ldots,\omega_n^-,\omega_1^+,\ldots,\omega_m^+\}} \Omega_{\omega'}(\varphi))$-impertinent are removed by slicing. □

Theorem 4.3 (Correctness of O-Projections)**.** An O-projection of a $\Omega_{CP}(\varphi)$-pertinent eigen-Υ-replaceable proof φ with respect to a clause c is a projection of φ with respect to c.

Proof. This is an immediate consequence of Lemmas 4.5, 4.6 and 4.7. □

Example 4.15 (O-Projections)**.** *Consider the proof φ of Example 4.1:*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \quad B \vdash B}{A, B \vdash A \wedge B}\wedge_r}{A \wedge B \vdash A \wedge B}\wedge_l \quad \dfrac{\dfrac{B \vdash B \quad A \vdash A}{A, B \vdash B \wedge A}\wedge_r}{A \wedge B \vdash B \wedge A}\wedge_l}{A \wedge B \vdash B \wedge A}cut \quad \dfrac{C \vdash C \quad C \vdash C}{C \vdash C}cut}{(A \wedge B) \vee C \vdash B \wedge A, C}\vee_l$$

Then, the projections $\lfloor \varphi \rfloor^{O}_{\vdash A, C}$ and $\lfloor \varphi \rfloor^{O}_{\vdash B, C}$ are:

$$\dfrac{\dfrac{\dfrac{A \vdash A}{A, B \vdash A}w_l}{A \wedge B \vdash A}\wedge_l \quad C \vdash C}{(A \wedge B) \vee C \vdash A, C}\vee_l \qquad \dfrac{\dfrac{\dfrac{B \vdash B}{A, B \vdash B}w_l}{A \wedge B \vdash B}\wedge_l \quad C \vdash C}{(A \wedge B) \vee C \vdash B, C}\vee_l$$

The projections $\lfloor \varphi \rfloor^{O}_{C \vdash A}$ and $\lfloor \varphi \rfloor^{O}_{C \vdash B}$ are:

$$\dfrac{\dfrac{\dfrac{A \vdash A}{A, B \vdash A}\,w_l}{A \wedge B \vdash A}\,\wedge_l \qquad \dfrac{C \vdash C}{C, C \vdash C}\,w_l}{(A \wedge B) \vee C, C \vdash A, C}\,\vee_l \qquad\qquad \dfrac{\dfrac{\dfrac{B \vdash B}{A, B \vdash B}\,w_l}{A \wedge B \vdash B}\,\wedge_l \qquad \dfrac{C \vdash C}{C, C \vdash C}\,w_l}{(A \wedge B) \vee C, C \vdash B, C}\,\vee_l$$

The projections $\lfloor \varphi \rfloor^{O}_{B,A \vdash C}$ *and* $\lfloor \varphi \rfloor^{O}_{B,A,C \vdash}$ *are:*

$$\dfrac{\dfrac{\dfrac{B \vdash B \qquad A \vdash A}{A, B \vdash B \wedge A}\,\wedge_r}{A \wedge B, A, B \vdash B \wedge A}\,w_l \qquad C \vdash C}{(A \wedge B) \vee C, A, B \vdash B \wedge A, C}\,\vee_l \qquad\qquad \dfrac{\dfrac{\dfrac{B \vdash B \qquad A \vdash A}{A, B \vdash B \wedge A}\,\wedge_r}{A \wedge B, A, B \vdash B \wedge A}\,w_l \qquad \dfrac{C \vdash C}{C, C \vdash C}\,w_l}{(A \wedge B) \vee C, A, B \vdash B \wedge A, C}\,\vee_l$$

4.3 CERes$_W$

The construction of a clause set from the cut-pertinent struct requires that the struct be transformed to a $\oplus\otimes$-normal-form. In the case of standard clause sets, this transformation is done via $\rightsquigarrow_{\oplus\otimes}$, which basically distributes $\otimes$ over $\oplus$, causing many duplications. Remembering that cut-impertinent inferences correspond to $\otimes$ and cut-pertinent inferences correspond to $\oplus$, a preprocessing that swapped cut-impertinent inferences upward above cut-pertinent inferences would lead to proofs having cut-pertinent structs where $\otimes$ (corresponding to the cut-impertinent inferences that are swapped upward) connectives occur above $\oplus$ connectives (corresponding to cut-pertinent inferences over which cut-impertinent inferences are swapped). Therefore, fewer distributions and duplications would be necessary, and thus simpler clause sets would result.

Moreover, it is sometimes also the case that, by swapping weakening inferences downward, whole subproofs could be deleted, resulting in proofs with even simpler cut-pertinent structs and clause sets.

However, performing local proof rewritings like inference swapping goes against the philosophy of cut-elimination by resolution, that strives to abstract away from the proof and work only with compact representations (e.g. structs and clause sets) of the information relevant for cut-elimination. With this philosophy in mind, it turns out that it is possible to employ an improved struct rewriting system ($\rightsquigarrow_{\oplus\otimes_W}$) that transforms structs into $\oplus\otimes$-normal-forms taking the possibility of inference swapping into account, without actually performing the swapping in the proof. Indeed, it can be shown that $\rightsquigarrow_{\oplus\otimes_W}$ actually corresponds to inference swapping, in a sense that is made precise in Lemma 4.9. The improved cut-pertinent clause set that results from using $\rightsquigarrow_{\oplus\otimes_W}$ instead of $\rightsquigarrow_{\oplus\otimes}$ is known as the *cut-pertinent swapped clause set*.

4.3.1 Inference Swapping

In this Subsection, a proof rewriting system (Definition 4.3.10) for inference swapping[j] is described. It is subdivided according to the kind of dependence (Definition 4.3.1) between the inferences that are being swapped. If the lower inference is independent of the upper inference, then they can easily be swapped (Definition 4.3.2), with no increase of proof size. However, if the lower inference is indirectly dependent on the

[j]The study of inference swapping in Gentzen's sequent calculi goes back to [75]. Another modern and more abstract study of inference swapping can be found in [59].

upper inference, then swapping requires a duplication of the lower inference, as well as the introduction of weakening and contraction inferences (Definition 4.3.3). The case of eigen-variable dependence can be avoided by considering skolemized proofs only. Even though two inferences cannot generally be swapped if there is direct dependence between them, swapping is possible in the particular case when the upper inference is a contraction (Definition 4.3.5) or a weakening (downward swapping of weakening inferences, Definition 4.3.5).

Definition 4.3.1 (Inference Dependence)**.** An inference ρ_1 is *directly dependent* on another inference ρ_2, denoted $\rho_1 \prec_D \rho_2$, if and only if a main occurrence of ρ_2 is an ancestor of an auxiliary occurrence of ρ_1.

A strong quantifier inference ρ_1 is *eigenvariable-dependent* on another inference ρ_2 occurring above ρ_1, denoted $\rho_1 \prec_Q \rho_2$, if and only if the substitution term of ρ_2 contains an occurrence of the eigenvariable of ρ_1.

An inference ρ_1 is *indirectly dependent* on another inference ρ_2 occurring above ρ_1, denoted $\rho_1 \prec_I \rho_2$, if and only if it is not directly dependent on ρ_2 and the auxiliary occurrences of ρ_1 have ancestors in more than one premise sequent of ρ_2.

An inference ρ_1 is *independent* of another inference ρ_2 if and only if ρ_1 is neither directly dependent nor eigenvariable-dependent nor indirectly dependent on ρ_2.

Example 4.16 (Directly Dependent Inferences)**.** *In the proof φ below, $\vee_l \prec_D \wedge_l$ (i.e. $\vee_l$ is directly dependent on $\wedge_l$):*

$$\cfrac{\cfrac{\cfrac{\cfrac{A \vdash A \quad B \vdash B}{A, B \vdash A \wedge B}\wedge_r}{A \wedge B \vdash A \wedge B}\wedge_l \quad \cfrac{\cfrac{B \vdash B \quad A \vdash A}{A, B \vdash B \wedge A}\wedge_r}{A \wedge B \vdash B \wedge A}\wedge_l}{A \wedge B \vdash B \wedge A}cut_1 \quad \cfrac{C \vdash C \quad C \vdash C}{C \vdash C}cut_2}{(A \wedge B) \vee C \vdash B \wedge A, C}\vee_l$$

Example 4.17 (Eigenvariable-Dependent Inferences)**.** *In the proof φ below, $\forall_r \prec_Q \forall_l$.*

$$\cfrac{\cfrac{P(f(\alpha)) \vdash P(f(\alpha))}{(\forall x)P(x) \vdash P(f(\alpha))}\forall_l}{(\forall x)P(x) \vdash (\forall x)P(f(x))}\forall_r$$

Example 4.18 (Indirectly Dependent Inferences). *In the proof φ below, the ancestors or descendants of active occurrences of $\wedge_l$ are highlighted in green:*

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\AxiomC{$B \vdash B$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A, B \vdash A \wedge B$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash A \wedge B$}
\AxiomC{$B \vdash B$}
\AxiomC{$A \vdash A$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A, B \vdash B \wedge A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash B \wedge A$}
\RightLabel{cut_1}
\BinaryInfC{$A \wedge B \vdash B \wedge A$}
\AxiomC{$C \vdash C$}
\AxiomC{$C \vdash C$}
\RightLabel{cut_2}
\BinaryInfC{$C \vdash C$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(A \wedge B) \vee C \vdash B \wedge A, C$}
\end{prooftree}
$$

Since occurrences highlighted in green occur in two premise sequents of $\wedge_r$, $\wedge_l \prec_I \wedge_r$ (i.e. $\wedge_l$ is indirectly dependent on $\wedge_r$).

Moreover, the green occurrences are also ancestors or descendents of $\vee_l$. Therefore, it is also the case that $\vee_r \prec_I \wedge_r$.

Example 4.19 (Independent Inferences). *In the proof φ below, ancestors and descendants of active occurrences of $\vee_l$ are highlighted in red:*

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\AxiomC{$B \vdash B$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A, B \vdash A \wedge B$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash A \wedge B$}
\AxiomC{$B \vdash B$}
\AxiomC{$A \vdash A$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A, B \vdash B \wedge A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash B \wedge A$}
\RightLabel{cut_1}
\BinaryInfC{$A \wedge B \vdash B \wedge A$}
\AxiomC{$C \vdash C$}
\AxiomC{$C \vdash C$}
\RightLabel{cut_2}
\BinaryInfC{$C \vdash C$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(A \wedge B) \vee C \vdash B \wedge A, C$}
\end{prooftree}
$$

Since occurrences highlighted in red occur in only one premise sequent of cut_1 and $\vee_l$, $\vee_l$ is not indirectly dependent on cut_1. Moreover, $\vee_l$ is also clearly neither directly dependent nor eigenvariable-dependent on cut_1. Therefore, $\vee_l$ is independent of cut_1.

Definition 4.3.2 ($\gg_I$). Swapping of Independent Inferences:

$$
\begin{prooftree}
\AxiomC{φ_1}
\noLine
\UnaryInfC{$\Gamma_1^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1$}
\RightLabel{ρ_1}
\UnaryInfC{$\Gamma^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1 \vdash \Delta^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1$}
\RightLabel{ρ_2}
\UnaryInfC{$\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1$}
\end{prooftree}
$$

$$\Downarrow$$

$$
\begin{prooftree}
\AxiomC{φ_1}
\noLine
\UnaryInfC{$\Gamma_1^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1$}
\RightLabel{ρ_2}
\UnaryInfC{$\Gamma_1^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta^{\rho_2}, \Delta_1$}
\RightLabel{ρ_1}
\UnaryInfC{$\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1$}
\end{prooftree}
$$

$$
\dfrac{\dfrac{\stackrel{\varphi_1}{\Gamma_1^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1} \qquad \stackrel{\varphi_2}{\Gamma_2^{\rho_1}, \Gamma_2 \vdash \Delta_2^{\rho_1}, \Delta_2}}{\Gamma^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1, \Gamma_2 \vdash \Delta^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1, \Delta_2}\,\rho_1}{\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2}\,\rho_2
$$

$$\Downarrow$$

$$
\dfrac{\dfrac{\stackrel{\varphi_1}{\Gamma_1^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1}}{\Gamma_1^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta^{\rho_2}, \Delta_1}\,\rho_2 \qquad \stackrel{\varphi_2}{\Gamma_2^{\rho_1}, \Gamma_2 \vdash \Delta_2^{\rho_1}, \Delta_2}}{\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2}\,\rho_1
$$

$$
\dfrac{\dfrac{\stackrel{\varphi_2}{\Gamma_2^{\rho_1}, \Gamma_2 \vdash \Delta_2^{\rho_1}, \Delta_2} \qquad \stackrel{\varphi_1}{\Gamma_1^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1}}{\Gamma^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1, \Gamma_2 \vdash \Delta^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1, \Delta_2}\,\rho_1}{\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2}\,\rho_2
$$

$$\Downarrow$$

$$
\dfrac{\stackrel{\varphi_2}{\Gamma_2^{\rho_1}, \Gamma_2 \vdash \Delta_2^{\rho_1}, \Delta_2} \qquad \dfrac{\stackrel{\varphi_1}{\Gamma_1^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1}}{\Gamma_1^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta^{\rho_2}, \Delta_1}\,\rho_2}{\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2}\,\rho_1
$$

$$
\dfrac{\dfrac{\stackrel{\varphi_1}{\Gamma_1^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1}}{\Gamma^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1 \vdash \Delta^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1}\,\rho_1 \qquad \stackrel{\varphi_2}{\Gamma_2^{\rho_2}, \Gamma_2 \vdash \Delta_2^{\rho_2}, \Delta_2}}{\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2}\,\rho_2
$$

$$\Downarrow$$

$$
\dfrac{\dfrac{\stackrel{\varphi_1}{\Gamma_1^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1} \qquad \stackrel{\varphi_2}{\Gamma_2^{\rho_2}, \Gamma_2 \vdash \Delta_2^{\rho_2}, \Delta_2}}{\Gamma_1^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2 \vdash \Delta_1^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2}\,\rho_2}{\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2}\,\rho_1
$$

$$
\dfrac{\stackrel{\varphi_2}{\Gamma_2^{\rho_2}, \Gamma_2 \vdash \Delta_2^{\rho_2}, \Delta_2} \qquad \dfrac{\stackrel{\varphi_1}{\Gamma_1^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1}}{\Gamma^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1 \vdash \Delta^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1}\,\rho_1}{\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2}\,\rho_2
$$

$$\Downarrow$$

$$\dfrac{\dfrac{\overset{\varphi_2}{\Gamma_2^{\rho_2}, \Gamma_2 \vdash \Delta_2^{\rho_2}, \Delta_2} \qquad \overset{\varphi_1}{\Gamma_1^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1}}{\Gamma_1^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2 \vdash \Delta_1^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2}\,\rho_2}{\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2}\,\rho_1$$

$$\dfrac{\dfrac{\overset{\varphi_1}{\Gamma_1^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1} \qquad \overset{\varphi_2}{\Gamma_2^{\rho_1}, \Gamma_2 \vdash \Delta_2^{\rho_1}, \Delta_2}}{\Gamma^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1, \Gamma_2 \vdash \Delta^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1, \Delta_2}\,\rho_1 \qquad \overset{\varphi_3}{\Gamma_3^{\rho_2}, \Gamma_3 \vdash \Delta_3^{\rho_2}, \Delta_3}}{\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2, \Gamma_3 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2, \Delta_3}\,\rho_2$$

$$\Downarrow$$

$$\dfrac{\dfrac{\overset{\varphi_1}{\Gamma_1^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1} \qquad \overset{\varphi_3}{\Gamma_3^{\rho_2}, \Gamma_3 \vdash \Delta_3^{\rho_2}, \Delta_3}}{\Gamma_1^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_3 \vdash \Delta_1^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_3}\,\rho_2 \qquad \overset{\varphi_2}{\Gamma_2^{\rho_1}, \Gamma_2 \vdash \Delta_2^{\rho_1}, \Delta_2}}{\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2, \Gamma_3 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2, \Delta_3}\,\rho_1$$

$$\dfrac{\dfrac{\overset{\varphi_1}{\Gamma_1^{\rho_1}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1} \qquad \overset{\varphi_2}{\Gamma_2^{\rho_1}, \Gamma_2^{\rho_2}, \Gamma_2 \vdash \Delta_2^{\rho_1}, \Delta_2^{\rho_2}, \Delta_2}}{\Gamma^{\rho_1}, \Gamma_2^{\rho_2}, \Gamma_1, \Gamma_2 \vdash \Delta^{\rho_1}, \Delta_2^{\rho_2}, \Delta_1, \Delta_2}\,\rho_1 \qquad \overset{\varphi_3}{\Gamma_3^{\rho_2}, \Gamma_3 \vdash \Delta_3^{\rho_2}, \Delta_3}}{\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2, \Gamma_3 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2, \Delta_3}\,\rho_2$$

$$\Downarrow$$

$$\dfrac{\overset{\varphi_1}{\Gamma_1^{\rho_1}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1} \qquad \dfrac{\overset{\varphi_2}{\Gamma_2^{\rho_1}, \Gamma_2^{\rho_2}, \Gamma_2 \vdash \Delta_2^{\rho_1}, \Delta_2^{\rho_2}, \Delta_2} \qquad \overset{\varphi_3}{\Gamma_3^{\rho_2}, \Gamma_3 \vdash \Delta_3^{\rho_2}, \Delta_3}}{\Gamma_2^{\rho_1}, \Gamma^{\rho_2}, \Gamma_2, \Gamma_3 \vdash \Delta_2^{\rho_1}, \Delta^{\rho_2}, \Delta_2, \Delta_3}\,\rho_2}{\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2, \Gamma_3 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2, \Delta_3}\,\rho_1$$

$$\dfrac{\overset{\varphi_3}{\Gamma_3^{\rho_2}, \Gamma_3 \vdash \Delta_3^{\rho_2}, \Delta_3} \qquad \dfrac{\overset{\varphi_1}{\Gamma_1^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1} \qquad \overset{\varphi_2}{\Gamma_2^{\rho_1}, \Gamma_2 \vdash \Delta_2^{\rho_1}, \Delta_2}}{\Gamma^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1, \Gamma_2 \vdash \Delta^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1, \Delta_2}\,\rho_1}{\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2, \Gamma_3 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2, \Delta_3}\,\rho_2$$

$$\Downarrow$$

$$\dfrac{\dfrac{\begin{array}{c}\varphi_3\\ \Gamma_3^{\rho_2}, \Gamma_3 \vdash \Delta_3^{\rho_2}, \Delta_3\end{array} \qquad \begin{array}{c}\varphi_1\\ \Gamma_1^{\rho_1}, \Gamma_1^{\rho_2}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1^{\rho_2}, \Delta_1\end{array}}{\Gamma_1^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_3 \vdash \Delta_1^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_3}\,\rho_2 \qquad \begin{array}{c}\varphi_2\\ \Gamma_2^{\rho_1}, \Gamma_2 \vdash \Delta_2^{\rho_1}, \Delta_2\end{array}}{\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2, \Gamma_3 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2, \Delta_3}\,\rho_1$$

$$\dfrac{\begin{array}{c}\varphi_3\\ \Gamma_3^{\rho_2}, \Gamma_3 \vdash \Delta_3^{\rho_2}, \Delta_3\end{array} \qquad \dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1^{\rho_1}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1\end{array} \qquad \begin{array}{c}\varphi_2\\ \Gamma_2^{\rho_1}, \Gamma_2^{\rho_2}, \Gamma_2 \vdash \Delta_2^{\rho_1}, \Delta_2^{\rho_2}, \Delta_2\end{array}}{\Gamma^{\rho_1}, \Gamma_2^{\rho_2}, \Gamma_1, \Gamma_2 \vdash \Delta^{\rho_1}, \Delta_2^{\rho_2}, \Delta_1, \Delta_2}\,\rho_1}{\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2, \Gamma_3 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2, \Delta_3}\,\rho_2$$

$$\Downarrow$$

$$\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1^{\rho_1}, \Gamma_1 \vdash \Delta_1^{\rho_1}, \Delta_1\end{array} \qquad \dfrac{\begin{array}{c}\varphi_3\\ \Gamma_3^{\rho_2}, \Gamma_3 \vdash \Delta_3^{\rho_2}, \Delta_3\end{array} \qquad \begin{array}{c}\varphi_2\\ \Gamma_2^{\rho_1}, \Gamma_2^{\rho_2}, \Gamma_2 \vdash \Delta_2^{\rho_1}, \Delta_2^{\rho_2}, \Delta_2\end{array}}{\Gamma_2^{\rho_1}, \Gamma^{\rho_2}, \Gamma_2, \Gamma_3 \vdash \Delta_2^{\rho_1}, \Delta^{\rho_2}, \Delta_2, \Delta_3}\,\rho_2}{\Gamma^{\rho_1}, \Gamma^{\rho_2}, \Gamma_1, \Gamma_2, \Gamma_3 \vdash \Delta^{\rho_1}, \Delta^{\rho_2}, \Delta_1, \Delta_2, \Delta_3}\,\rho_1$$

Remark 4.3.1. For the sequent calculus **LK**, the proof rewriting rules defined in Definition 4.3.2 are sufficient. For sequent calculi in general, analogous rules can be defined for the cases where ρ_1 or ρ_2 have arity greater than two.

Example 4.20 (Swapping of Independent Inferences)**.** *As shown in Example 4.19 $\vee_l$ is independent of cut_1 in the proof φ below:*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \quad B \vdash B}{A, B \vdash A \wedge B}\,\wedge_r}{A \wedge B \vdash A \wedge B}\,\wedge_l \qquad \dfrac{\dfrac{B \vdash B \quad A \vdash A}{A, B \vdash B \wedge A}\,\wedge_r}{A \wedge B \vdash B \wedge A}\,\wedge_l}{A \wedge B \vdash B \wedge A}\,cut_1 \qquad \dfrac{C \vdash C \quad C \vdash C}{C \vdash C}\,cut_2}{(A \wedge B) \vee C \vdash B \wedge A, C}\,\vee_l}{}$$

Therefore, $\vee_l$ can be swapped above cut_1, resulting in the following proof ψ with $\varphi \gg_I \psi$:

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \quad B \vdash B}{A, B \vdash A \wedge B}\,\wedge_r}{A \wedge B \vdash A \wedge B}\,\wedge_l \qquad \dfrac{C \vdash C \quad C \vdash C}{C \vdash C}\,cut_2}{(A \wedge B) \vee C \vdash A \wedge B, C}\,\vee_l \qquad \dfrac{\dfrac{B \vdash B \quad A \vdash A}{A, B \vdash B \wedge A}\,\wedge_r}{A \wedge B \vdash B \wedge A}\,\wedge_l}{(A \wedge B) \vee C \vdash B \wedge A, C}\,cut_1$$

The cut-pertinent struct of ψ is:

$$\mathcal{S}_\psi \equiv ((A \oplus B) \otimes (C \oplus \neg C)) \oplus (\neg B \otimes \neg A)$$

While the cut-pertinent struct of φ, as shown in Example 4.1, is:

$$\mathcal{S}_\varphi \equiv ((A \oplus B) \oplus (\neg B \otimes \neg A)) \otimes (C \oplus \neg C)$$

By comparing the cut-pertinent structs, it is possible to see that the effect of swapping a cut-impertinent inference ($\vee_l$) above a cut-pertinent inference (cut_1) was that the $\otimes$-junction corresponding to $\vee_l$ was swapped within (above, if the struct is displayed as a tree) the $\oplus$-junction corresponding to cut_1.

$\vee_l$ can be further swapped above cut_2, resulting in the following proof ψ' with $\psi \gg_I \psi'$:

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\AxiomC{$B \vdash B$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A, B \vdash A \wedge B$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash A \wedge B$}
\AxiomC{$C \vdash C$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(A \wedge B) \vee C \vdash A \wedge B, C$}
\AxiomC{$C \vdash C$}
\RightLabel{cut_2}
\BinaryInfC{$(A \wedge B) \vee C \vdash A \wedge B, C$}
\AxiomC{$B \vdash B$}
\AxiomC{$A \vdash A$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A, B \vdash B \wedge A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash B \wedge A$}
\RightLabel{cut_1}
\BinaryInfC{$(A \wedge B) \vee C \vdash B \wedge A, C$}
\end{prooftree}
$$

Its cut-pertinent struct is:

$$\mathcal{S}_{\psi'} \equiv (((A \oplus B) \otimes C) \oplus \neg C) \oplus (\neg B \otimes \neg A)$$

And its cut-pertinent clause set is:

$$\mathcal{C}_{\psi'} \equiv \{\vdash A, C \ ; \ \vdash B, C \ ; \ C \vdash \ ; \ B, A \vdash \ \}$$

Comparing it with the cut-pertinent clause set of φ shown below, it is clear that effect of swapping cut-impertinent inferences upward is significant.

$$\mathcal{C}_\varphi \equiv \{\vdash A, C \ ; \ C \vdash A \ ; \ \vdash B, C \ ; \ C \vdash B \ ; \ B, A \vdash C \ ; \ B, A, C \vdash\}$$

The reason for this beneficial effect is that $\oplus\otimes$-normalization distributes $\otimes$ over all $\oplus$-juncts of a $\oplus$-junction, causing many duplications. Swapping, on the other hand, distributes only over one $\oplus$-junct, causing no duplications.

Definition 4.3.3 ($\gg_{ID}$)**.** Distributional Swapping of Indirectly Dependent Inferences:

$$\dfrac{\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1^{\rho_1},\Gamma_1^{\rho_2},\Gamma_1 \vdash \Delta_1^{\rho_1},\Delta_1^{\rho_2},\Delta_1\end{array} \qquad \begin{array}{c}\varphi_2\\ \Gamma_2^{\rho_1},\Gamma_2^{\rho_2},\Gamma_2 \vdash \Delta_2^{\rho_1},\Delta_2^{\rho_2},\Delta_2\end{array}}{\Gamma^{\rho_1},\Gamma_1^{\rho_2},\Gamma_2^{\rho_2},\Gamma_1,\Gamma_2 \vdash \Delta^{\rho_1},\Delta_1^{\rho_2},\Gamma_2^{\rho_2},\Delta_1,\Delta_2}\,\rho_1}{\Gamma^{\rho_1},\Gamma^{\rho_2},\Gamma_1,\Gamma_2 \vdash \Delta^{\rho_1},\Delta^{\rho_2},\Delta_1,\Delta_2}\,\rho_2$$

$$\Downarrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1^{\rho_1},\Gamma_1^{\rho_2},\Gamma_1 \vdash \Delta_1^{\rho_1},\Delta_1^{\rho_2},\Delta_1\end{array}}{\Gamma_1^{\rho_1},\Gamma_1^{\rho_2},\Gamma_2^{\rho_2},\Gamma_1 \vdash \Delta_1^{\rho_1},\Delta_1^{\rho_2},\Delta_2^{\rho_2}\Delta_1}\,w^*}{\Gamma_1^{\rho_1},\Gamma^{\rho_2},\Gamma_1 \vdash \Delta_1^{\rho_1},\Delta^{\rho_2},\Delta_1}\,\rho_2 \qquad \dfrac{\dfrac{\begin{array}{c}\varphi_2\\ \Gamma_2^{\rho_1},\Gamma_2^{\rho_2},\Gamma_2 \vdash \Delta_2^{\rho_1},\Delta_2^{\rho_2},\Delta_2\end{array}}{\Gamma_2^{\rho_1},\Gamma_1^{\rho_2},\Gamma_2^{\rho_2},\Gamma_2 \vdash \Delta_2^{\rho_1},\Delta_1^{\rho_2},\Delta_2^{\rho_2},\Delta_2}\,w^*}{\Gamma_2^{\rho_1},\Gamma^{\rho_2},\Gamma_2 \vdash \Delta_2^{\rho_1},\Delta^{\rho_2},\Delta_2}\,\rho_2}{\Gamma^{\rho_1},\Gamma^{\rho_2},\Gamma^{\rho_2},\Gamma_1,\Gamma_2 \vdash \Delta^{\rho_1},\Delta^{\rho_2},\Delta^{\rho_2},\Delta_1,\Delta_2}\,\rho_1}{\Gamma^{\rho_1},\Gamma^{\rho_2},\Gamma_1,\Gamma_2 \vdash \Delta^{\rho_1},\Delta^{\rho_2},\Delta_1,\Delta_2}\,c^*$$

Remark 4.3.2. For the sequent calculus **LK**, the proof rewriting rule defined in Definition 4.3.3 is sufficient. However, for sequent calculi in general, the rule must be generalized to cases where, for example, ρ_2 has arity greater than one. Nevertheless, such more general cases are analogous to the case shown here.

Remark 4.3.3. While the inference ρ_2 in the proof rewriting rules of Definition 4.3.3 can be a contraction, there are cases in which contractions can be swapped upward in a smarter way, as shown in Definition 4.3.4

Example 4.21 (Distributional Swapping of Indirectly Dependent Inferences)**.** *In the proof ψ' below, $\wedge_l <_I \wedge_r$.*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad B \vdash B}{A, B \vdash A \wedge B}\,\wedge_r}{A \wedge B \vdash A \wedge B}\,\wedge_l \qquad C \vdash C}{(A \wedge B) \vee C \vdash A \wedge B, C}\,\vee_l \qquad C \vdash C}{(A \wedge B) \vee C \vdash A \wedge B, C}\,cut_2 \qquad \dfrac{\dfrac{B \vdash B \qquad A \vdash A}{A, B \vdash B \wedge A}\,\wedge_r}{A \wedge B \vdash B \wedge A}\,\wedge_l}{(A \wedge B) \vee C \vdash B \wedge A, C}\,cut_1$$

$\wedge_l$ can be swapped above $\wedge_r$, resulting in the following proof ψ'' with $\psi' \gg_{ID} \psi''$:

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\RightLabel{w_l}
\UnaryInfC{$A, B \vdash A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash A$}
\AxiomC{$B \vdash B$}
\RightLabel{w_l}
\UnaryInfC{$A, B \vdash B$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash B$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A \wedge B, A \wedge B \vdash A \wedge B$}
\RightLabel{c_l}
\UnaryInfC{$A \wedge B \vdash A \wedge B$}
\AxiomC{$C \vdash C$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(A \wedge B) \vee C \vdash A \wedge B, C$}
\AxiomC{$C \vdash C$}
\RightLabel{cut_2}
\BinaryInfC{$(A \wedge B) \vee C \vdash A \wedge B, C$}
\AxiomC{$B \vdash B$}
\AxiomC{$A \vdash A$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A, B \vdash B \wedge A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash B \wedge A$}
\RightLabel{cut_1}
\BinaryInfC{$(A \wedge B) \vee C \vdash B \wedge A, C$}
\end{prooftree}
$$

Definition 4.3.4 ($\gg_{IDC}$)**.** Swapping of indirectly dependent contractions:

$$
\begin{prooftree}
\AxiomC{φ_1}
\noLine
\UnaryInfC{$\Gamma_1, \Gamma_\rho, \Gamma_\rho \vdash \Delta_1, \Delta_\rho, \Delta_\rho$}
\RightLabel{ρ}
\UnaryInfC{$\Gamma_1, \Gamma_\rho, \Pi_\rho \vdash \Delta_1, \Delta_\rho, \Lambda_\rho$}
\RightLabel{ρ}
\UnaryInfC{$\Gamma_1, \Pi_\rho, \Pi_\rho \vdash \Delta_1, \Lambda_\rho, \Lambda_\rho$}
\doubleLine
\RightLabel{c^*}
\UnaryInfC{$\Gamma_1, \Pi_\rho \vdash \Delta_1, \Lambda_\rho$}
\end{prooftree}
$$

$$\Downarrow$$

$$
\begin{prooftree}
\AxiomC{φ_1}
\noLine
\UnaryInfC{$\Gamma_1, \Gamma_\rho, \Gamma_\rho \vdash \Delta_1, \Delta_\rho, \Delta_\rho$}
\doubleLine
\RightLabel{c^*}
\UnaryInfC{$\Gamma_1, \Gamma_\rho \vdash \Delta_1, \Delta_\rho$}
\RightLabel{ρ}
\UnaryInfC{$\Gamma_1, \Pi_\rho \vdash \Delta_1, \Lambda_\rho$}
\end{prooftree}
$$

$$
\begin{prooftree}
\AxiomC{φ_1}
\noLine
\UnaryInfC{$\Gamma_1, \Gamma_1^\rho, \Gamma_1^\rho \vdash \Delta_1, \Delta_\rho, \Delta_\rho$}
\AxiomC{φ_2}
\noLine
\UnaryInfC{$\Gamma_2, \Gamma_2^\rho \vdash \Delta_2, \Delta_2^\rho$}
\RightLabel{ρ}
\BinaryInfC{$\Gamma_1, \Gamma_\rho, \Pi_\rho \vdash \Delta_1, \Delta_\rho, \Lambda_\rho$}
\AxiomC{φ_2}
\noLine
\UnaryInfC{$\Gamma_2, \Gamma_2^\rho \vdash \Delta_2, \Delta_2^\rho$}
\RightLabel{ρ}
\BinaryInfC{$\Gamma_1, \Pi_\rho, \Pi_\rho \vdash \Delta_1, \Lambda_\rho, \Lambda_\rho$}
\doubleLine
\RightLabel{c^*}
\UnaryInfC{$\Gamma_1, \Pi_\rho \vdash \Delta_1, \Lambda_\rho$}
\end{prooftree}
$$

$$\Downarrow$$

$$
\begin{prooftree}
\AxiomC{φ_1}
\noLine
\UnaryInfC{$\Gamma_1, \Gamma_1^\rho, \Gamma_1^\rho \vdash \Delta_1, \Delta_1^\rho, \Delta_1^\rho$}
\doubleLine
\RightLabel{c^*}
\UnaryInfC{$\Gamma_1, \Gamma_\rho \vdash \Delta_1, \Delta_\rho$}
\AxiomC{φ_2}
\noLine
\UnaryInfC{$\Gamma_2, \Gamma_2^\rho \vdash \Delta_2, \Delta_2^\rho$}
\RightLabel{ρ}
\BinaryInfC{$\Gamma_1, \Pi_\rho \vdash \Delta_1, \Lambda_\rho$}
\end{prooftree}
$$

$$
\begin{prooftree}
\AxiomC{φ_2}
\noLine
\UnaryInfC{$\Gamma_2, \Gamma_2^\rho \vdash \Delta_2, \Delta_2^\rho$}
\AxiomC{φ_2}
\noLine
\UnaryInfC{$\Gamma_2, \Gamma_2^\rho \vdash \Delta_2, \Delta_2^\rho$}
\AxiomC{φ_1}
\noLine
\UnaryInfC{$\Gamma_1, \Gamma_1^\rho, \Gamma_1^\rho \vdash \Delta_1, \Delta_\rho, \Delta_\rho$}
\RightLabel{ρ}
\BinaryInfC{$\Gamma_1, \Gamma_\rho, \Pi_\rho \vdash \Delta_1, \Delta_\rho, \Lambda_\rho$}
\RightLabel{ρ}
\BinaryInfC{$\Gamma_1, \Pi_\rho, \Pi_\rho \vdash \Delta_1, \Lambda_\rho, \Lambda_\rho$}
\doubleLine
\RightLabel{c^*}
\UnaryInfC{$\Gamma_1, \Pi_\rho \vdash \Delta_1, \Lambda_\rho$}
\end{prooftree}
$$

$$\Downarrow$$

$$\dfrac{\begin{array}{c}\varphi_2\\ \Gamma_2,\Gamma_2^{\rho} \vdash \Delta_2,\Delta_2^{\rho}\end{array} \qquad \dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1,\Gamma_1^{\rho},\Gamma_1^{\rho} \vdash \Delta_1,\Delta_1^{\rho},\Delta_1^{\rho}\end{array}}{\Gamma_1,\Gamma_{\rho} \vdash \Delta_1,\Delta_{\rho}}\, c^*}{\Gamma_1,\Pi_{\rho} \vdash \Delta_1,\Lambda_{\rho}}\, \rho$$

Definition 4.3.5 ($\gg_C$). Distributional Swapping over contractions:

$$\dfrac{\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1,\Gamma_{\rho},\Gamma'_{\rho} \vdash \Delta_1,\Delta_{\rho},\Delta'_{\rho}\end{array}}{\Gamma_1,\Gamma_{\rho} \vdash \Delta_1,\Delta_{\rho}}\, c^*}{\Gamma_1,\Pi_{\rho} \vdash \Delta_1,\Lambda_{\rho}}\, \rho$$

$$\Downarrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1,\Gamma_{\rho},\Gamma'_{\rho} \vdash \Delta_1,\Delta_{\rho},\Delta'_{\rho}\end{array}}{\Gamma_1,\Gamma_{\rho},\Gamma_{\rho} \vdash \Delta_1,\Delta_{\rho},\Delta_{\rho}}\, w^*}{\Gamma_1,\Gamma_{\rho},\Pi_{\rho} \vdash \Delta_1,\Delta_{\rho},\Lambda_{\rho}}\, \rho}{\Gamma_1,\Pi_{\rho},\Pi_{\rho} \vdash \Delta_1,\Lambda_{\rho},\Lambda_{\rho}}\, \rho}{\Gamma_1,\Pi_{\rho} \vdash \Delta_1,\Lambda_{\rho}}\, c^*$$

$$\dfrac{\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1,\Gamma_1^{\rho},\Gamma_1\rho' \vdash \Delta_1,\Delta_1^{\rho},\Delta_1^{\rho'}\end{array}}{\Gamma_1,\Gamma_{\rho} \vdash \Delta_1,\Delta_{\rho}}\, c^* \qquad \begin{array}{c}\varphi_2\\ \Gamma_2,\Gamma_2^{\rho} \vdash \Delta_2,\Delta_2^{\rho}\end{array}}{\Gamma_1,\Pi_{\rho} \vdash \Delta_1,\Lambda_{\rho}}\, \rho$$

$$\Downarrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1,\Gamma_1^{\rho},\Gamma_1^{\rho'} \vdash \Delta_1,\Delta_1^{\rho},\Delta_1^{\rho'}\end{array}}{\Gamma_1,\Gamma_1^{\rho},\Gamma_1^{\rho} \vdash \Delta_1,\Delta_{\rho},\Delta_{\rho}}\, w^* \qquad \begin{array}{c}\varphi_2\\ \Gamma_2,\Gamma_2^{\rho} \vdash \Delta_2,\Delta_2^{\rho}\end{array}}{\Gamma_1,\Gamma_{\rho},\Pi_{\rho} \vdash \Delta_1,\Delta_{\rho},\Lambda_{\rho}}\, \rho \qquad \begin{array}{c}\varphi_2\\ \Gamma_2,\Gamma_2^{\rho} \vdash \Delta_2,\Delta_2^{\rho}\end{array}}{\Gamma_1,\Pi_{\rho},\Pi_{\rho} \vdash \Delta_1,\Lambda_{\rho},\Lambda_{\rho}}\, \rho}{\Gamma_1,\Pi_{\rho} \vdash \Delta_1,\Lambda_{\rho}}\, c^*$$

$$
\begin{prooftree}
\AxiomC{φ_2}
\noLine
\UnaryInfC{$\Gamma_2, \Gamma_2^{\rho} \vdash \Delta_2, \Delta_2^{\rho}$}
\AxiomC{φ_1}
\noLine
\UnaryInfC{$\Gamma_1, \Gamma_1^{\rho}, \Gamma_1^{\rho'} \vdash \Delta_1, \Delta_1^{\rho}, \Delta_1^{\rho'}$}
\doubleLine
\RightLabel{c^*}
\UnaryInfC{$\Gamma_1, \Gamma_{\rho} \vdash \Delta_1, \Delta_{\rho}$}
\RightLabel{ρ}
\BinaryInfC{$\Gamma_1, \Pi_{\rho} \vdash \Delta_1, \Lambda_{\rho}$}
\end{prooftree}
$$

$$\Downarrow$$

$$
\begin{prooftree}
\AxiomC{φ_2}
\noLine
\UnaryInfC{$\Gamma_2, \Gamma_2^{\rho} \vdash \Delta_2, \Delta_2^{\rho}$}
\AxiomC{φ_2}
\noLine
\UnaryInfC{$\Gamma_2, \Gamma_2^{\rho} \vdash \Delta_2, \Delta_2^{\rho}$}
\AxiomC{φ_1}
\noLine
\UnaryInfC{$\Gamma_1, \Gamma_1^{\rho}, \Gamma_1^{\rho'} \vdash \Delta_1, \Delta_1^{\rho}, \Delta_1^{\rho'}$}
\doubleLine
\RightLabel{w^*}
\UnaryInfC{$\Gamma_1, \Gamma_1^{\rho}, \Gamma_1^{\rho} \vdash \Delta_1, \Delta_{\rho}, \Delta_{\rho}$}
\RightLabel{ρ}
\BinaryInfC{$\Gamma_1, \Gamma_{\rho}, \Pi_{\rho} \vdash \Delta_1, \Delta_{\rho}, \Lambda_{\rho}$}
\RightLabel{ρ}
\BinaryInfC{$\Gamma_1, \Pi_{\rho}, \Pi_{\rho} \vdash \Delta_1, \Lambda_{\rho}, \Lambda_{\rho}$}
\doubleLine
\RightLabel{c^*}
\UnaryInfC{$\Gamma_1, \Pi_{\rho} \vdash \Delta_1, \Lambda_{\rho}$}
\end{prooftree}
$$

Example 4.22 (Distributional Swapping over Contractions). *In the proof ψ''' (with $\psi'' \gg_C \psi'''$) below, $\vee_l$ has been swapped above the contraction and duplicated into two copies: $\vee_l$ and $\vee_l$.*

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\RightLabel{w_l}
\UnaryInfC{$A, B \vdash A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash A$}
\AxiomC{$B \vdash B$}
\RightLabel{w_l}
\UnaryInfC{$A, B \vdash B$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash B$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A \wedge B, A \wedge B \vdash A \wedge B$}
\AxiomC{$C \vdash C$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(A \wedge B) \vee C, A \wedge B \vdash A \wedge B, C$}
\AxiomC{$C \vdash C$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(A \wedge B) \vee C, (A \wedge B) \vee C \vdash A \wedge B, C, C$}
\RightLabel{c_l}
\UnaryInfC{$(A \wedge B) \vee C \vdash A \wedge B, C, C$}
\RightLabel{c_r}
\UnaryInfC{$(A \wedge B) \vee C \vdash A \wedge B, C$}
\AxiomC{$C \vdash C$}
\RightLabel{cut_2}
\BinaryInfC{$(A \wedge B) \vee C \vdash A \wedge B, C$}
\AxiomC{$B \vdash B$}
\AxiomC{$A \vdash A$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A, B \vdash B \wedge A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash B \wedge A$}
\RightLabel{cut_1}
\BinaryInfC{$(A \wedge B) \vee C \vdash B \wedge A, C$}
\end{prooftree}
$$

Example 4.23 (More Swapping). *Now that $\vee_l$ has been duplicated, $\vee_l$ and $\vee_l$ can be swapped above $\wedge_r$, since each copy is now independent of $\wedge_r$. The resulting proof ψ^* is:*

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\RightLabel{w_l}
\UnaryInfC{$A, B \vdash A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash A$}
\AxiomC{$C \vdash C$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(A \wedge B) \vee C, \vdash A, C$}
\AxiomC{$B \vdash B$}
\RightLabel{w_l}
\UnaryInfC{$A, B \vdash B$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash B$}
\AxiomC{$C \vdash C$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(A \wedge B) \vee C \vdash B, C$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$(A \wedge B) \vee C, (A \wedge B) \vee C \vdash A \wedge B, C, C$}
\RightLabel{c_l}
\UnaryInfC{$(A \wedge B) \vee C \vdash A \wedge B, C, C$}
\RightLabel{c_r}
\UnaryInfC{$(A \wedge B) \vee C \vdash A \wedge B, C$}
\AxiomC{$C \vdash C$}
\RightLabel{cut_2}
\BinaryInfC{$(A \wedge B) \vee C \vdash A \wedge B, C$}
\AxiomC{$B \vdash B$}
\AxiomC{$A \vdash A$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A, B \vdash B \wedge A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash B \wedge A$}
\RightLabel{cut_1}
\BinaryInfC{$(A \wedge B) \vee C \vdash B \wedge A, C$}
\end{prooftree}
$$

Its cut-pertinent struct is:

$$\mathcal{S}_{\psi^*} \equiv (((A \otimes C) \oplus (B \otimes C)) \oplus \neg C) \oplus (\neg B \otimes \neg A) \equiv (A \otimes C) \oplus (B \otimes C) \oplus \neg C \oplus (\neg B \otimes \neg A)$$

Note that, due to all the swapping $\mathcal{S}_{\psi^*}$ is already in $\oplus\otimes$-normal-form, and hence the cut-pertinent clause set of ψ^* can be obtained simply by clausification of the struct:

$$C_{\psi^*} \equiv \{\vdash A, C \ ; \ \vdash B, C \ ; \ C \vdash \ ; \ B, A \vdash \ \}$$

It is interesting to note that $C_{\psi^*} = C_{\psi'}$ (ψ' is shown in Example 4.21), even though $\mathcal{S}_{\psi^*} \neq \mathcal{S}_{\psi'}$. This is because the $\oplus\otimes$-normalization of $\mathcal{S}_{\psi'}$ has the same effect of distributional duplications as the distributional swapping over contractions.

Definition 4.3.6 ($\gg_{WI}$)**.** Downward swapping of weakening inferences over independent inferences.

$$\dfrac{\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1^\rho, \Gamma \vdash \Delta_1^\rho, \Delta\end{array}}{F, \Gamma_1^\rho, \Gamma \vdash \Delta_1^\rho, \Delta}\, w_l}{F, \Gamma^\rho, \Gamma \vdash \Delta^\rho, \Delta}\, \rho \qquad\qquad \dfrac{\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1^\rho, \Gamma \vdash \Delta_1^\rho, \Delta\end{array}}{\Gamma_1^\rho, \Gamma \vdash \Delta_1^\rho, \Delta, F}\, w_r}{F, \Gamma^\rho, \Gamma \vdash \Delta^\rho, \Delta, F}\, \rho$$

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

$$\dfrac{\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1^\rho, \Gamma \vdash \Delta_1^\rho, \Delta\end{array}}{\Gamma^\rho, \Gamma \vdash \Delta^\rho, \Delta}\, \rho}{F, \Gamma^\rho, \Gamma \vdash \Delta^\rho, \Delta}\, w_l \qquad\qquad \dfrac{\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1^\rho, \Gamma \vdash \Delta_1^\rho, \Delta\end{array}}{\Gamma^\rho, \Gamma \vdash \Delta^\rho, \Delta}\, \rho}{\Gamma^\rho, \Gamma \vdash \Delta^\rho, \Delta, F}\, w_l$$

$$\dfrac{\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1^\rho, \Gamma_1 \vdash \Delta_1^\rho, \Delta_1\end{array}}{F, \Gamma_1^\rho, \Gamma_1 \vdash \Delta_1^\rho, \Delta_1}\, w_l \quad \begin{array}{c}\varphi_2\\ \Gamma_2^\rho, \Gamma_2 \vdash \Delta_2^\rho, \Delta_2\end{array}}{F, \Gamma^\rho, \Gamma_1, \Gamma_2 \vdash \Delta^\rho, \Delta_1, \Delta_2}\, \rho \qquad \dfrac{\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1^\rho, \Gamma_1 \vdash \Delta_1^\rho, \Delta_1\end{array}}{\Gamma_1^\rho, \Gamma_1 \vdash \Delta_1^\rho, \Delta_1, F}\, w_r \quad \begin{array}{c}\varphi_2\\ \Gamma_2^\rho, \Gamma_2 \vdash \Delta_2^\rho, \Delta_2\end{array}}{\Gamma^\rho, \Gamma_1, \Gamma_2 \vdash \Delta^\rho, \Delta_1, F, \Delta_2}\, \rho$$

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

$$\dfrac{\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1^\rho, \Gamma_1 \vdash \Delta_1^\rho, \Delta_1\end{array} \quad \begin{array}{c}\varphi_2\\ \Gamma_2^\rho, \Gamma_2 \vdash \Delta_2^\rho, \Delta_2\end{array}}{\Gamma^\rho, \Gamma_1, \Gamma_2 \vdash \Delta^\rho, \Delta_1, \Delta_2}\, \rho}{F, \Gamma^\rho, \Gamma_1, \Gamma_2 \vdash \Delta^\rho, \Delta_1, \Delta_2}\, w_l \qquad \dfrac{\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1^\rho, \Gamma_1 \vdash \Delta_1^\rho, \Delta_1\end{array} \quad \begin{array}{c}\varphi_2\\ \Gamma_2^\rho, \Gamma_2 \vdash \Delta_2^\rho, \Delta_2\end{array}}{\Gamma^\rho, \Gamma_1, \Gamma_2 \vdash \Delta^\rho, \Delta_1, \Delta_2}\, \rho}{\Gamma^\rho, \Gamma_1, \Gamma_2 \vdash \Delta^\rho, \Delta_1, F, \Delta_2}\, w_r$$

$$\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1^\rho, \Gamma_1 \vdash \Delta_1^\rho, \Delta_1\end{array} \quad \dfrac{\begin{array}{c}\varphi_2\\ \Gamma_2^\rho, \Gamma_2 \vdash \Delta_2^\rho, \Delta_2\end{array}}{F, \Gamma_2^\rho, \Gamma_2 \vdash \Delta_2^\rho, \Delta_2}\, w_l}{F, \Gamma^\rho, \Gamma_1, \Gamma_2 \vdash \Delta^\rho, \Delta_1, \Delta_2}\, \rho \qquad \dfrac{\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1^\rho, \Gamma_1 \vdash \Delta_1^\rho, \Delta_1\end{array} \quad \begin{array}{c}\varphi_2\\ \Gamma_2^\rho, \Gamma_2 \vdash \Delta_2^\rho, \Delta_2\end{array}}{\Gamma^\rho, \Gamma_1, \Gamma_2 \vdash \Delta^\rho, \Delta_1, \Delta_2}\, \rho}{F, \Gamma^\rho, \Gamma_1, \Gamma_2 \vdash \Delta^\rho, \Delta_1, \Delta_2}\, w_l$$

$$\Downarrow$$

$$\dfrac{\begin{matrix}\varphi_1\\ \Gamma_1^\rho,\Gamma_1 \vdash \Delta_1^\rho,\Delta_1\end{matrix} \qquad \dfrac{\begin{matrix}\varphi_2\\ \Gamma_2^\rho,\Gamma_2\vdash\Delta_2^\rho,\Delta_2\end{matrix}}{\Gamma_2^\rho,\Gamma_2\vdash\Delta_2^\rho,\Delta_2,F}\,w_r}{\Gamma^\rho,\Gamma_1,\Gamma_2\vdash\Delta^\rho,\Delta_1,F,\Delta_2}\,\rho \qquad \dfrac{\dfrac{\begin{matrix}\varphi_1\\ \Gamma_1^\rho,\Gamma_1 \vdash \Delta_1^\rho,\Delta_1\end{matrix} \qquad \begin{matrix}\varphi_2\\ \Gamma_2^\rho,\Gamma_2\vdash\Delta_2^\rho,\Delta_2\end{matrix}}{\Gamma^\rho,\Gamma_1,\Gamma_2\vdash\Delta^\rho,\Delta_1,\Delta_2}\,\rho}{\Gamma^\rho,\Gamma_1,\Gamma_2\vdash\Delta^\rho,\Delta_1,F,\Delta_2}\,w_r$$

$$\Downarrow$$

Definition 4.3.7 (Degenerate Inferences)**.** An inference ρ in a proof φ is *degenerate* when all its auxiliary formula occurrences are descendants of main formula occurrences of weakening inferences. When only some auxiliary (sub)-formula occurrences of ρ are descendants of main formula occurrences of weakening inferences, ρ is *partially degenerate*.

Definition 4.3.8 ($\gg_{WD}$)**.** Downward swapping of weakening inferences over directly dependent inferences.

$$\dfrac{\dfrac{\begin{matrix}\varphi_1\\ \Gamma\vdash\Delta\end{matrix}}{\Gamma_1^\rho,\Gamma\vdash\Delta_1^\rho,\Delta}\,w^*}{\Gamma^\rho,\Gamma\vdash\Delta^\rho,\Delta}\,\rho$$

$$\Downarrow$$

$$\dfrac{\begin{matrix}\varphi_1\\ \Gamma\vdash\Delta\end{matrix}}{\Gamma^\rho,\Gamma\vdash\Delta^\rho,\Delta}\,w^*$$

$$\dfrac{\dfrac{\begin{matrix}\varphi_1\\ \Gamma_1\vdash\Delta_1\end{matrix}}{\Gamma_1^\rho,\Gamma_1\vdash\Delta_1^\rho,\Delta_1}\,w^* \qquad \begin{matrix}\varphi_2\\ \Gamma_2^\rho,\Gamma_2\vdash\Delta_2^\rho,\Delta_2\end{matrix}}{\Gamma^\rho,\Gamma_1,\Gamma_2\vdash\Delta^\rho,\Delta_1,\Delta_2}\,\rho \qquad \dfrac{\begin{matrix}\varphi_2\\ \Gamma_2^\rho,\Gamma_2\vdash\Delta_2^\rho,\Delta_2\end{matrix} \qquad \dfrac{\begin{matrix}\varphi_1\\ \Gamma_1\vdash\Delta_1\end{matrix}}{\Gamma_1^\rho,\Gamma_1\vdash\Delta_1^\rho,\Delta_1}\,w^*}{\Gamma^\rho,\Gamma_2,\Gamma_1\vdash\Delta^\rho,\Delta_2,\Delta_1}\,\rho$$

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

$$\dfrac{\begin{matrix}\varphi_1\\ \Gamma_1\vdash\Delta_1\end{matrix}}{\Gamma^\rho,\Gamma_1,\Gamma_2\vdash\Delta^\rho,\Delta_1,\Delta_2}\,w^* \qquad \dfrac{\begin{matrix}\varphi_1\\ \Gamma_1\vdash\Delta_1\end{matrix}}{\Gamma^\rho,\Gamma_2,\Gamma_1\vdash\Delta^\rho,\Delta_2,\Delta_1}\,w^*$$

Definition 4.3.9 ($\gg_W$)**.** The proof rewriting relation for *downward swapping of weakening* is:

$$\gg_W \doteq (\gg_{WI} \cup \gg_{WD})$$

Definition 4.3.10 ($\gg$)**.** The proof rewriting relation for *inference swapping* is:

$$\gg \doteq (\gg_I \cup \gg_{ID} \cup \gg_{IDC} \cup \gg_C \cup \gg_W)$$

4.3.2 Cut-Pertinent Swapped Clause Sets

In this section another struct rewriting system ($\rightsquigarrow_{\oplus\otimes_W}$) to transform structs into $\oplus\otimes$-normal-forms is described. It takes the possibility of inference swapping into account and distributes $\otimes$ not among all $\oplus$-juncts, but only among some of them. Indeed, it is shown in Lemma 4.8 that every rewriting of the struct according to $\rightsquigarrow_{\oplus\otimes_W}$ corresponds to an inference swapping sequence in the corresponding proof according to $\gg$. In summary, while $\rightsquigarrow_{\oplus\otimes}$ does full distribution of $\otimes$ over $\oplus$, as if cut-impertinent inferences were always indirectly dependent on the cut-pertinent inferences above them, $\rightsquigarrow_{\oplus\otimes_W}$ does partial distribution when the corresponding inferences are independent. Moreover, $\rightsquigarrow_{\oplus\otimes_W}$ is capable of exploiting the presence of weakening in the proof.

In order for the partial distribution and the correspondence to inference swapping to be possible, the struct must encode not only the branching structure of the proof, in the form of $\oplus$ and $\otimes$ connectives, but also enough extra information to allow the retrieval of the dependencies between the branching inferences. Although there could be various ways to extend structs to encode dependency information directly, an indirect and quite minimalistic extension is given here. In the proof of Lemma 4.8 it becomes clear that it is possible to retrieve dependency information from structs containing additional information regarding the pertinence of its formula occurrences in the sets as described in Definition 4.3.11.

Cut-pertinent swapped clause sets (Definition 4.3.13) of proofs are simply defined as the clausification of $\rightsquigarrow_{\oplus\otimes_W}$-normal-forms of the cut-pertinent structs of the proofs. While swapped clause sets are always smaller than or of equal size to standard clause sets, they have the disadvantage of being non-unique, because $\rightsquigarrow_{\oplus\otimes_W}$ is non-confluent. Therefore, in general more than one swapped clause set of a proof is necessary to fully characterize the set of all possible CERes_W-normal-forms for the proof. Nevertheless, it is important to note that each swapped clause set is unsatisfiable (Theorem 4.5).

Swapped clause sets are very similar to profile clause sets, which have been defined in [66]. In fact, the concept of swapped clause set evolved from attempts to find an intuitively simpler definition for profile clause sets. In Subsection 4.4, profile clause sets are defined and the slight difference with respect to swapped clause sets is discussed.

Definition 4.3.11 (Inference Occurrences)**.** Let $\omega_1, \ldots, \omega_n$ be all the occurrences of atomic subformulas of auxiliary occurrences of an inference ρ

in a proof φ. Then:

$$\Omega_{\rho}(\varphi) \doteq \bigcup_{1 \leq i \leq n} \Omega_{\omega_i}(\varphi)$$

Example 4.24 (Inference Occurrences). *In the proof φ below, the occurrences belonging to $\Omega_{\vee_l}(\varphi)$ have been highlighted in red:*

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \quad B \vdash B}{A, B \vdash A \wedge B}\wedge_r}{A \wedge B \vdash A \wedge B}\wedge_l \quad \dfrac{\dfrac{B \vdash B \quad A \vdash A}{A, B \vdash B \wedge A}\wedge_r}{A \wedge B \vdash B \wedge A}\wedge_l}{A \wedge B \vdash B \wedge A}cut_1 \quad \dfrac{C \vdash C \quad C \vdash C}{C \vdash C}cut_2}{(A \wedge B) \vee C \vdash B \wedge A, C}\vee_l$$

And below, the occurrences belonging to $\Omega_{cut_1}(\varphi)$ have been highlighted in blue:

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \quad B \vdash B}{A, B \vdash A \wedge B}\wedge_r}{A \wedge B \vdash A \wedge B}\wedge_l \quad \dfrac{\dfrac{B \vdash B \quad A \vdash A}{A, B \vdash B \wedge A}\wedge_r}{A \wedge B \vdash B \wedge A}\wedge_l}{A \wedge B \vdash B \wedge A}cut_1 \quad \dfrac{C \vdash C \quad C \vdash C}{C \vdash C}cut_2}{(A \wedge B) \vee C \vdash B \wedge A, C}\vee_l$$

Definition 4.3.12 ($\rightsquigarrow_{\oplus\otimes W}$). In the struct rewriting rules below, let ρ be the inference in φ corresponding to $\otimes_\rho$. For the rewriting rules to be applicable[k], $S_{n+1}, \ldots, S_{n+m}$ and S must contain at least one occurrence from $\Omega_\rho(\varphi)$ each (i.e. there is an atomic substruct S'_{n+k} of S_{n+k} such that $S'_{n+k} \in \Omega_\rho(\varphi)$), and $S_1, \ldots, S_n$ and S_l and S_r should not contain any occurrence from $\Omega_\rho(\varphi)$. Moreover, an innermost rewriting strategy is enforced: only minimal reducible substructs (i.e. structs having no reducible proper substruct) can be rewritten.

$$S \otimes_\rho (S_1 \oplus \ldots \oplus S_n \oplus S_{n+1} \oplus \ldots \oplus S_{n+m}) \rightsquigarrow_{\oplus\otimes W} S_1 \oplus \ldots \oplus S_n \oplus (S \otimes_\rho S_{n+1}) \oplus \ldots \oplus (S \otimes_\rho S_{n+m})$$

$$(S_1 \oplus \ldots \oplus S_n \oplus S_{n+1} \oplus \ldots \oplus S_{n+m}) \otimes_\rho S \rightsquigarrow_{\oplus\otimes W} S_1 \oplus \ldots \oplus S_n \oplus (S_{n+1} \otimes_\rho S) \oplus \ldots \oplus (S_{n+m} \otimes_\rho S)$$

$$S_l \otimes_\rho S_r \rightsquigarrow_{\oplus\otimes W} S_l \qquad S_l \otimes_\rho S_r \rightsquigarrow_{\oplus\otimes W} S_r \qquad S_l \oplus_\rho S_r \rightsquigarrow_{\oplus\otimes W} S_l \qquad S_l \oplus_\rho S_r \rightsquigarrow_{\oplus\otimes W} S_r$$

$$S \oplus_\rho S_r \rightsquigarrow_{\oplus\otimes W} S_r \qquad\qquad S_l \oplus_\rho S \rightsquigarrow_{\oplus\otimes W} S_l$$

Remark 4.3.4. The struct rewriting rules of $\rightsquigarrow_{\oplus\otimes W}$ include not only rules to distribute $\otimes$ over $\oplus$ in a more clever way, but also rules to handle

[k] Note that m can be equal to zero, in which case the first two rewriting rules simply degenerate to:

$$S \otimes_\rho (S_1 \oplus \ldots \oplus S_n) \rightsquigarrow_{\oplus\otimes W} S_1 \oplus \ldots \oplus S_n$$
$$(S_1 \oplus \ldots \oplus S_n) \otimes_\rho S \rightsquigarrow_{\oplus\otimes W} S_1 \oplus \ldots \oplus S_n$$

struct connectives that correspond to degenerate and partially degenerate inferences. These rules are related to downward swapping of weakening inferences, as shown in Lemma 4.8.

Example 4.25 ($\oplus\otimes_W$-Normalization)**.** *Let φ be the proof below:*

$$
\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad B \vdash B}{A, B \vdash A \wedge B}\wedge_r^1}{A \wedge B \vdash A \wedge B}\wedge_l \qquad \dfrac{\dfrac{B \vdash B \qquad A \vdash A}{A, B \vdash B \wedge A}\wedge_r^2}{A \wedge B \vdash B \wedge A}\wedge_l}{A \wedge B \vdash B \wedge A}cut^3 \qquad \dfrac{C \vdash C \qquad C \vdash C}{C \vdash C}cut^4}{(A \wedge B) \vee C \vdash B \wedge A, C}\vee_l^5
$$

Its cut-pertinent struct is:

$$\mathcal{S}_\varphi \equiv ((A \oplus^1 B) \oplus^3 (\neg B \otimes^2 \neg A)) \otimes^5 (C \oplus^4 \neg C)$$

Considering that $\{A, B, C\} \subset \Omega_{\vee_l^5}(\varphi)$ and $\{A, B, C\} \cap \Omega_{\vee_l^5}(\varphi) = \emptyset$, the struct can be normalized in the two ways shown below:

$$
\begin{aligned}
\mathcal{S}_\varphi &\equiv ((A \oplus^1 B) \oplus^3 (\neg B \otimes^2 \neg A)) \otimes^5 (C \oplus^4 \neg C) \\
&\rightsquigarrow_{\oplus\otimes_W} ((A \oplus^1 B) \otimes^5 (C \oplus^4 \neg C)) \oplus^3 (\neg B \otimes^2 \neg A) \\
&\rightsquigarrow_{\oplus\otimes_W} ((A \otimes^5 (C \oplus^4 \neg C)) \oplus^1 (B \otimes^5 (C \oplus^4 \neg C))) \oplus^3 (\neg B \otimes^2 \neg A) \\
&\rightsquigarrow_{\oplus\otimes_W} (((A \otimes^5 C) \oplus^4 \neg C) \oplus^1 ((B \otimes^5 C) \oplus^4 \neg C)) \oplus^3 (\neg B \otimes^2 \neg A) \\
&\equiv (A \otimes^5 C) \oplus \neg C \oplus (B \otimes^5 C) \oplus \neg C \oplus (\neg B \otimes^2 \neg A) \\
&\equiv S_1
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{S}_\varphi &\equiv ((A \oplus^1 B) \oplus^3 (\neg B \otimes^2 \neg A)) \otimes^5 (C \oplus^4 \neg C) \\
&\rightsquigarrow_{\oplus\otimes_W} ((A \oplus^1 B) \otimes^5 (C \oplus^4 \neg C)) \oplus^3 (\neg B \otimes^2 \neg A) \\
&\rightsquigarrow_{\oplus\otimes_W} ((((A \oplus^1 B) \otimes^5 C) \oplus^4 \neg C)) \oplus^3 (\neg B \otimes^2 \neg A) \\
&\rightsquigarrow_{\oplus\otimes_W} ((((A \otimes^5 C) \oplus^1 (B \otimes^5 C)) \oplus^4 \neg C)) \oplus^3 (\neg B \otimes^2 \neg A) \\
&\equiv (A \otimes^5 C) \oplus (B \otimes^5 C) \oplus \neg C \oplus (\neg B \otimes^2 \neg A) \\
&\equiv S_2
\end{aligned}
$$

Theorem 4.4 (Non-Confluence of $\rightsquigarrow_{\oplus\otimes_W}$)**.** $\rightsquigarrow_{\oplus\otimes_W}$ is non-confluent.

Proof. Example 4.25 shows a struct having two different $\rightsquigarrow_{\oplus\otimes_W}$-normal-forms. Therefore $\rightsquigarrow_{\oplus\otimes_W}$ is non-confluent. Example 4.25 also shows that non-confluence can be caused when there is a reducible substruct on which both of the two first rewriting rules shown in Definition 4.3.12 can be applied. Yet another and more obvious source of non-confluence, however, are cases in which any of the four last rewriting rules shown in Definition 4.3.12 can be applied, because then another of those rules can always be applied. It is necessary to choose which $\oplus$-junct (or $\otimes$-junct) to delete; and, unless both juncts are equal, each choice will clearly lead to a distinct normal form. □

Definition 4.3.13 (Cut-pertinent Swapped Clause Set)**.** A *cut-pertinent swapped clause set* of a proof φ with respect to a $\rightsquigarrow_{\oplus\otimes_W}$-normal-form S of $\mathcal{S}_\varphi$ is:

$$C^W_{\varphi|S} \doteq \mathrm{cl}(S)$$

In cases where a proof φ has only one cut-pertinent swapped clause set, it can be denoted simply as C^W_φ.

Example 4.26 (Cut-Pertinent Swapped Clause Set)**.** *Let φ be the proof considered in Example 4.25 and S_1 and S_2 the two $\rightsquigarrow_{\oplus\otimes_W}$-normal-forms of $\mathcal{S}_\varphi$ shown there. Then:*

$$C^W_{\varphi|S_1} = \{ \vdash A, C \ ; \ \vdash B, C \ ; \ C \vdash \ ; \ C \vdash \ ; \ B, A \vdash \}$$

$$C^W_{\varphi|S_2} = \{ \vdash A, C \ ; \ \vdash B, C \ ; \ C \vdash \ ; \ B, A \vdash \}$$

It is interesting to note that $C^W_{\varphi|S_1} = C^W_{\varphi|S_2}$ (since they are sets). This is no coincidence. It always occurs when the non-confluence in the struct level is due to non-degenerated applications of the first two rewriting rules.

Lemma 4.8 (Correspondence between $\rightsquigarrow_{\oplus\otimes_W}$ and $\gg$)**.** If φ is skolemized and $\mathcal{S}_\varphi \rightsquigarrow_{\oplus\otimes_W} S$, then there exists a proof ψ such that $\varphi \gg^* \psi$ and $\mathcal{S}_\psi = S$.

Proof. The proof can be subdivided into the possible cases of $\rightsquigarrow_{\oplus\otimes_W}$-rewriting:

- Case 1: the selected reducible substruct has the form $S \otimes_\rho (S_1 \oplus \ldots \oplus S_n)$ (and is rewritten to $S_1 \oplus \ldots \oplus S_n$):

  Let ρ be the inference in φ corresponding to $\otimes_\rho$. Let φ' be the subproof having the conclusion sequent of ρ as end-sequent, φ'_1 be the subproof of φ having the left premise sequent of ρ as its end-sequent and φ'_2 be the subproof of φ having the right premise sequent of ρ as its end-sequent. Clearly, $\mathcal{S}^{\Omega_{CP}(\varphi)}_{\varphi'_1} = S$ and $\mathcal{S}^{\Omega_{CP}(\varphi)}_{\varphi'_2} = (S_1 \oplus \ldots \oplus S_n)$.

  Since $\mathcal{S}^{\Omega_{CP}(\varphi)}_{\varphi'_2}$ contains no occurrence of $\Omega_\rho(\varphi)$, it must be the case that all auxiliary occurrences ω_i of ρ occurring in its right premise sequent are descendants of main occurrences of weakening inferences. Hence, there is a proof $\gg_W$-normal-form φ_2'' of φ_2'.

Moreover, since $S\otimes(S_1 \oplus \ldots \oplus S_n)$ is a minimal reducible substruct, the occurrences ω_i are not ancestors of any cut-impertinent binary inference (for if they were, there would be a reducible substruct of $S\otimes(S_1 \oplus \ldots \oplus S_n)$, contradicting the fact that $S\otimes(S_1 \oplus \ldots \oplus S_n)$ is a minimal reducible substruct of $\mathcal{S}_\varphi$). And, clearly, ω_i are also not ancestors of any cut-pertinent binary inference, because ρ is cut-impertinent. Therefore, in the sequence rewriting φ_2' into φ_2'', none of the rewriting rules of $\gg_{WD}$ that delete binary inferences is used. Consequently, $\mathcal{S}_{\varphi_2''}^{\Omega_{CP}(\varphi)} = \mathcal{S}_{\varphi_2'}^{\Omega_{CP}(\varphi)}$.

Let φ'' be the result of replacing φ_2' by φ_2'' in φ'. Clearly, φ'' is of the form:

$$\dfrac{\dfrac{\varphi_1'}{\Gamma_1^\rho, \Gamma_1 \vdash \Delta_1^\rho, \Delta_1} \qquad \dfrac{\dfrac{\varphi_2'''}{\Gamma_2 \vdash \Delta_2}}{\Gamma_2^\rho, \Gamma_2 \vdash \Delta_2^\rho, \Delta_2}\, w^*}{\Gamma^\rho, \Gamma_2, \Gamma_1 \vdash \Delta^\rho, \Delta_2, \Delta_1}\, \rho$$

where φ_2'' is:

$$\dfrac{\dfrac{\varphi_2'''}{\Gamma_2 \vdash \Delta_2}}{\Gamma_2^\rho, \Gamma_2 \vdash \Delta_2^\rho, \Delta_2}\, w^*$$

By using one of the rewriting rules of $\gg_{WD}$, φ'' can be rewritten to the proof ψ' below:

$$\dfrac{\dfrac{\varphi_2'''}{\Gamma_2 \vdash \Delta_2}}{\Gamma^\rho, \Gamma_2, \Gamma_1 \vdash \Delta^\rho, \Delta_2, \Delta_1}\, w^*$$

Clearly, $\mathcal{S}_{\psi'}^{\Omega_{CP}(\varphi)} = (S_1 \oplus \ldots \oplus S_n)$

Therefore, there exists a proof ψ (namely, the proof obtainable from φ by replacing its subproof φ' by ψ') such that $\varphi \gg_W \psi$ and $\mathcal{S}_\psi$ is $\mathcal{S}_\varphi$ with $S\otimes(S_1 \oplus \ldots \oplus S_n)$ rewritten to $S_1 \oplus \ldots \oplus S_n$.

- Case 2: the selected reducible substruct has the form $(S_1 \oplus \ldots \oplus S_n)\otimes S$ (and is rewritten to $S_1 \oplus \ldots \oplus S_n$): Symmetric to case 1.

- Case 3: the selected reducible substruct has the form $S_l \otimes S_r$ (and is rewritten to S_r): Analogous to case 1.
- Case 4: the selected reducible substruct has the form $S_l \otimes S_r$ (and is rewritten to S_l): Analogous to case 2.
- Case 5: the selected reducible substruct has the form $S_l \oplus S_r$ (and is rewritten to S_r): Analogous to case 1.
- Case 6: the selected reducible substruct has the form $S_l \oplus S_r$ (and is rewritten to S_l): Analogous to case 2.
- Case 7: the selected reducible substruct has the form $S_l \oplus S_r$ (and is rewritten to S_r): Analogous to case 1.
- Case 8: the selected reducible substruct has the form $S_l \oplus S_r$ (and is rewritten to S_l): Analogous to case 2.
- Case 9: the selected reducible substruct has the form $S \otimes_{\rho_2} (S' \oplus_{\rho_1} S'')$ (and is rewritten to $S' \oplus_{\rho_1} (S \otimes_{\rho_2} S'')$):

  Let ρ_2 be the inference corresponding to $\otimes_{\rho_2}$ and ρ_1 be the inference corresponding to $\oplus_{\rho_1}$. Since ρ_1 is cut-pertinent and ρ_2 is cut-impertinent, ρ_2 is not directly dependent on ρ_1. Moreover, φ is skolemized and thus ρ_2 is also not eigen-variable dependent on ρ_1. As only S'' has occurrences of $\Omega_{\rho_2}(\varphi)$, $\Omega_{\rho_2}(\varphi)$ has occurrences in at most one premise sequent of ρ_1, and hence ancestors of auxiliary occurrences of ρ_2 occur in at most one premise sequent of ρ_1. Therefore ρ_2 is independent of ρ_1. Moreover, any inference ρ_i (on the path between ρ_2 and ρ_1) on which ρ_2 directly depends is also independent of ρ_1.

  Consequently, there exists a proof ψ with $\varphi \gg_I^* \psi$ where ρ_2 and all inferences ρ_i on which it depends have been swapped above ρ_1, so that $\mathcal{S}_\psi$ is $\mathcal{S}_\varphi$ with $S \otimes (S' \oplus S'')$ rewritten to $S' \oplus (S \otimes S'')$.
- Case 10: the selected reducible substruct has the form $S \otimes (S' \oplus S'')$ (and is rewritten to $(S \otimes S') \oplus S''$): Analogous to case 7.
- Case 11: the selected reducible substruct has the form $(S' \oplus S'') \otimes S$ (and is rewritten to $(S' \otimes S) \oplus S''$): Symmetric to case 7.
- Case 12: the selected reducible substruct has the form $(S' \oplus S'') \otimes S$ (and is rewritten to $S' \oplus (S'' \otimes S)$): Analogous to case 9.

- Case 13: the selected reducible substruct has the form $(S' \oplus_{\rho_1} S'')\otimes_{\rho_2} S$ (and is rewritten to $(S'\otimes_{\rho_2} S) \oplus_{\rho_1} (S''\otimes_{\rho_2} S)$):

  Since ρ_1 is cut-pertinent and ρ_2 is cut-impertinent, ρ_2 is not directly dependent on ρ_1. Moreover, φ is skolemized and thus ρ_2 is also not eigen-variable dependent on ρ_1. However, as both S' and S'' have occurrences of $\Omega_{\rho_2}(\varphi)$, $\rho_2 \prec_I \rho_1$. In the sequent calculus **LK**, this can only happen if there exists a sequence of unary[1] inferences $\rho_D^* \equiv (\rho_{D_1}, \ldots, \rho_{D_n})$ on the path between ρ_2 and ρ_1 such that ρ_{D_i} is indirectly dependent on ρ_1 and ρ_2 depends on ρ_{D_i}, for any i such that $1 \leq i \leq n$. Moreover, any inference ρ_{D_i} (on the path between ρ_2 and ρ_1) on which ρ_2 directly depends is also independent of ρ_1.

  Let φ' be the subproof of φ having the conclusion sequent of ρ_2 as its end-sequent. Then, there exists a proof φ'' with $\varphi' \gg^* \varphi''$ (swapping ρ_D^* and ρ_2 above all inferences on which they do not depend) such that φ'' has the following form (or a form symmetric to it):

$$\dfrac{\dfrac{\dfrac{\overset{\varphi_1}{\Gamma_1,\Gamma_1^{\rho_2},\Gamma_1^{\rho_1} \vdash \Delta_1,\Delta_1^{\rho_2},\Delta_1^{\rho_1}} \qquad \overset{\varphi_2}{\Gamma_2,\Gamma_2^{\rho_2},\Gamma_2^{\rho_1} \vdash \Delta_2,\Delta_2^{\rho_2},\Delta_2^{\rho_1}}}{\Gamma_1,\Gamma_2,\Gamma_1^{\rho_2},\Gamma_2^{\rho_2},\Gamma^{\rho_1} \vdash \Delta_1,\Delta_2,\Delta_1^{\rho_2},\Delta_2^{\rho_2},\Delta^{\rho_1}}\,\rho_1}{\Gamma_1,\Gamma_2,\Gamma^{\rho_1} \vdash \Delta_1,\Delta_2,F_{12}^{\rho_2},\Delta^{\rho_1}}\,\rho_D^* \qquad \overset{\varphi_3}{\Gamma_3 \vdash \Delta_3,F_3^{\rho_2}}}{\Gamma_1,\Gamma_2,\Gamma_3,\Gamma^{\rho_1} \vdash \Delta_1,\Delta_2,\Delta_3,F^{\rho_2},\Delta^{\rho_1}}\,\rho_2$$

  Then there exists a proof φ''' with $\varphi'' \gg^* \varphi'''$ (swapping ρ_D^* above ρ_1) such that φ''' has the following form:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overset{\varphi_1}{\Gamma_1,\Gamma_1^{\rho_2},\Gamma_1^{\rho_1} \vdash \Delta_1,\Delta_1^{\rho_2},\Delta_1^{\rho_1}}}{\Gamma_1,\Gamma_1^{\rho_2},\Gamma_2^{\rho_2},\Gamma_1^{\rho_1} \vdash \Delta_1,\Delta_1^{\rho_2},\Delta_2^{\rho_2},\Delta_1^{\rho_1}}\,w^*}{\Gamma_1,\Gamma_1^{\rho_1} \vdash \Delta_1,F_{12}^{\rho_2},\Delta_1^{\rho_1}}\,\rho_D^* \qquad \dfrac{\dfrac{\overset{\varphi_2}{\Gamma_2,\Gamma_2^{\rho_2},\Gamma_2^{\rho_1} \vdash \Delta_2,\Delta_2^{\rho_2},\Delta_2^{\rho_1}}}{\Gamma_2,\Gamma_1^{\rho_2},\Gamma_2^{\rho_2},\Gamma_2^{\rho_1} \vdash \Delta_2,\Delta_1^{\rho_2},\Delta_2^{\rho_2},\Delta_2^{\rho_1}}\,w^*}{\Gamma_2,\Gamma_2^{\rho_1} \vdash \Delta_2,F_{12}^{\rho_2},\Delta_2^{\rho_1}}\,\rho_D^*}{\Gamma_1,\Gamma_2,\Gamma^{\rho_1} \vdash \Delta_1,\Delta_2,F_{12}^{\rho_2},F_{12}^{\rho_2},\Delta^{\rho_1}}\,\rho_1}{\Gamma_1,\Gamma_2,\Gamma^{\rho_1} \vdash \Delta_1,\Delta_2,F_{12}^{\rho_2},\Delta^{\rho_1}}\,c_r \qquad \overset{\varphi_3}{\Gamma_3 \vdash \Delta_3,F_3^{\rho_2}}}{\Gamma_1,\Gamma_2,\Gamma_3,\Gamma^{\rho_1} \vdash \Delta_1,\Delta_2,\Delta_3,F^{\rho_2},\Delta^{\rho_1}}\,\rho_2$$

  Then there exists a proof $\varphi^{(4)}$ with $\varphi''' \gg_C \varphi^{(4)}$ (swapping ρ_2 above the contraction) such that $\varphi^{(4)}$ has the following form:

[1]If the inferences were not unary, the struct would simply not be of the form $(S' \oplus_{\rho_1} S'')\otimes_{\rho_2} S$.

$$
\begin{prooftree}
\AxiomC{φ_1}
\noLine
\UnaryInfC{$\Gamma_1, \Gamma_1^{\rho_2}, \Gamma_1^{\rho_1} \vdash \Delta_1, \Delta_1^{\rho_2}, \Delta_1^{\rho_1}$}
\doubleLine
\RightLabel{w^*}
\UnaryInfC{$\Gamma_1, \Gamma_1^{\rho_2}, \Gamma_2^{\rho_2}, \Gamma_1^{\rho_1} \vdash \Delta_1, \Delta_1^{\rho_2}, \Delta_2^{\rho_2}, \Delta_1^{\rho_1}$}
\doubleLine
\RightLabel{ρ_D^*}
\UnaryInfC{$\Gamma_1, \Gamma_1^{\rho_1} \vdash \Delta_1, F_{12}^{\rho_2}, \Delta_1^{\rho_1}$}
\AxiomC{φ_2}
\noLine
\UnaryInfC{$\Gamma_2, \Gamma_2^{\rho_2}, \Gamma_2^{\rho_1} \vdash \Delta_2, \Delta_2^{\rho_2}, \Delta_2^{\rho_1}$}
\doubleLine
\RightLabel{w^*}
\UnaryInfC{$\Gamma_2, \Gamma_1^{\rho_2}, \Gamma_2^{\rho_2}, \Gamma_2^{\rho_1} \vdash \Delta_2, \Delta_1^{\rho_2}, \Delta_2^{\rho_2}, \Delta_2^{\rho_1}$}
\doubleLine
\RightLabel{ρ_D^*}
\UnaryInfC{$\Gamma_2, \Gamma_2^{\rho_1} \vdash \Delta_2, F_{12}^{\rho_2}, \Delta_2^{\rho_1}$}
\RightLabel{ρ_1}
\BinaryInfC{$\Gamma_1, \Gamma_2, \Gamma^{\rho_1} \vdash \Delta_1, \Delta_2, F_{12}^{\rho_2}, F_{12}^{\rho_2}, \Delta^{\rho_1}$}
\AxiomC{φ_3}
\noLine
\UnaryInfC{$\Gamma_3 \vdash \Delta_3, F_3^{\rho_2}$}
\RightLabel{ρ_2}
\BinaryInfC{$\Gamma_1, \Gamma_2, \Gamma_3, \Gamma^{\rho_1} \vdash \Delta_1, \Delta_2, \Delta_3, F_{12}^{\rho_2}, F^{\rho_2}, \Delta^{\rho_1}$}
\AxiomC{φ_3}
\noLine
\UnaryInfC{$\Gamma_3 \vdash \Delta_3, F_3^{\rho_2}$}
\RightLabel{ρ_2}
\BinaryInfC{$\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_3, \Gamma^{\rho_1} \vdash \Delta_1, \Delta_2, \Delta_3, \Delta_3, F^{\rho_2}, F^{\rho_2}, \Delta^{\rho_1}$}
\RightLabel{c_r}
\UnaryInfC{$\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_3, \Gamma^{\rho_1} \vdash \Delta_1, \Delta_2, \Delta_3, \Delta_3, F^{\rho_2}, \Delta^{\rho_1}$}
\doubleLine
\RightLabel{c^*}
\UnaryInfC{$\Gamma_1, \Gamma_2, \Gamma_3, \Gamma^{\rho_1} \vdash \Delta_1, \Delta_2, \Delta_3, F^{\rho_2}, \Delta^{\rho_1}$}
\end{prooftree}
$$

Finally, there exists a proof ψ' with $\varphi^{(4)} \gg_I^* \psi'$ (swapping each copy of ρ_2 above ρ_1) such that ψ' has the following form:

$$
\begin{prooftree}
\AxiomC{φ_1}
\noLine
\UnaryInfC{$\Gamma_1, \Gamma_1^{\rho_2}, \Gamma_1^{\rho_1} \vdash \Delta_1, \Delta_1^{\rho_2}, \Delta_1^{\rho_1}$}
\doubleLine
\RightLabel{w^*}
\UnaryInfC{$\Gamma_1, \Gamma_1^{\rho_2}, \Gamma_2^{\rho_2}, \Gamma_1^{\rho_1} \vdash \Delta_1, \Delta_1^{\rho_2}, \Delta_2^{\rho_2}, \Delta_1^{\rho_1}$}
\doubleLine
\RightLabel{ρ_D^*}
\UnaryInfC{$\Gamma_1, \Gamma_1^{\rho_1} \vdash \Delta_1, F_{12}^{\rho_2}, \Delta_1^{\rho_1}$}
\AxiomC{φ_3}
\noLine
\UnaryInfC{$\Gamma_3 \vdash \Delta_3, F_3^{\rho_2}$}
\RightLabel{ρ_2}
\BinaryInfC{$\Gamma_1, \Gamma_3, \Gamma_1^{\rho_1} \vdash \Delta_1, \Delta_3, F^{\rho_2}, \Delta_1^{\rho_1}$}
\AxiomC{φ_2}
\noLine
\UnaryInfC{$\Gamma_2, \Gamma_2^{\rho_2}, \Gamma_2^{\rho_1} \vdash \Delta_2, \Delta_2^{\rho_2}, \Delta_2^{\rho_1}$}
\doubleLine
\RightLabel{w^*}
\UnaryInfC{$\Gamma_2, \Gamma_1^{\rho_2}, \Gamma_2^{\rho_2}, \Gamma_2^{\rho_1} \vdash \Delta_2, \Delta_1^{\rho_2}, \Delta_2^{\rho_2}, \Delta_2^{\rho_1}$}
\doubleLine
\RightLabel{ρ_D^*}
\UnaryInfC{$\Gamma_2, \Gamma_2^{\rho_1} \vdash \Delta_2, F_{12}^{\rho_2}, \Delta_2^{\rho_1}$}
\AxiomC{φ_3}
\noLine
\UnaryInfC{$\Gamma_3 \vdash \Delta_3, F_3^{\rho_2}$}
\RightLabel{ρ_2}
\BinaryInfC{$\Gamma_2, \Gamma_3, \Gamma_2^{\rho_1} \vdash \Delta_2, \Delta_3, F^{\rho_2}, \Delta_2^{\rho_1}$}
\RightLabel{ρ_1}
\BinaryInfC{$\Gamma_1, \Gamma_2, \Gamma^{\rho_1} \vdash \Delta_1, \Delta_2, F_{12}^{\rho_2}, F_{12}^{\rho_2}, \Delta^{\rho_1}$}
\RightLabel{c_r}
\UnaryInfC{$\Gamma_1, \Gamma_2, \Gamma_3, \Gamma_3, \Gamma^{\rho_1} \vdash \Delta_1, \Delta_2, \Delta_3, \Delta_3, F^{\rho_2}, \Delta^{\rho_1}$}
\doubleLine
\RightLabel{c^*}
\UnaryInfC{$\Gamma_1, \Gamma_2, \Gamma_3, \Gamma^{\rho_1} \vdash \Delta_1, \Delta_2, \Delta_3, F^{\rho_2}, \Delta^{\rho_1}$}
\end{prooftree}
$$

Consequently, there exists a proof ψ with $\varphi \gg^* \psi$ (namely, the proof obtained from φ by rewriting its subproof φ' to ψ' as shown above) where ρ_2 and all unary inferences ρ_{D_i} on which it depends have been swapped above ρ_1, so that $\mathcal{S}_\psi$ is $\mathcal{S}_\varphi$ with $(S' \oplus S'') \otimes S$ rewritten to $(S' \otimes S) \oplus (S'' \otimes S)$.

- Case 14: the selected reducible substruct has the form $S \otimes (S' \oplus S'')$ (and is rewritten to $(S \otimes S') \oplus (S \otimes S'')$): Symmetric to case 11.

$\square$

Lemma 4.9 (Iterated Correspondence between $\rightsquigarrow_{\oplus\otimes W}$ and $\gg$)**.** If φ is skolemized and $\mathcal{S}_\varphi \rightsquigarrow_{\oplus\otimes W}^* S$, then there exists a proof ψ such that $\varphi \gg^* \psi$ and $\mathcal{S}_\psi = S$.

Proof. This Lemma is just the iterated version of Lemma 4.8 and can be easily proved by induction on the number of rewriting steps to rewrite $\mathcal{S}_\varphi$ to S_ψ. $\square$

Theorem 4.5 (Unsatisfiability of the Cut-Pertinent Swapped Clause Set)**.** For any skolemized proof φ and any $\rightsquigarrow_{\oplus\otimes W}$-normal-form S of $\mathcal{S}_\varphi$, $C^W_{\varphi|S}$ is unsatisfiable.

Proof. By Lemma 4.9, there exists a proof ψ such that $\mathcal{S}_\psi = S$. Clearly, $C^W_{\varphi|S} = C^W_\psi$. But $C^W_\psi = C^S_\psi$, since S is also a $\rightsquigarrow_{\oplus\otimes}$-normal-form. Therefore, $C^W_{\varphi|S} = C^S_\psi$ and, by Theorem 4.1, it is unsatisfiable. $\square$

Proof. Yet another approach involves an actual construction of a resolution refutation for $C^W_{\varphi|S}$, as shown in Chapter 6. By the soundness of the resolution calculus, $C^W_{\varphi|S}$ must then be unsatisfiable. □

Remark 4.3.5. It is not possible to prove Theorem 4.5 analogously to the proof of the unsatisfiability of the profile shown in [66, 67], which is essentially based on the fact that the profile clause set of φ subsumes C^S_φ. Unfortunately, $C^W_{\varphi|S}$ does not subsume C^S_φ in general. In particular, the subsumption fails when φ contains degenerate cut-impertinent inferences, in which case $\rightsquigarrow_{\oplus\otimes_W}$ prunes too much of the struct and hence some clauses of C^S_φ are not subsumed by any clause of $C^W_{\varphi|S}$.

4.3.3 Projections

The method for constructing S-projections does not work with swapped clause sets. Specifically the inductive case for cut-impertinent inferences is problematic. In the case of the standard clause set, in which $\otimes$ is fully distributed over $\oplus$, it must be the case that such cut-impertinent inferences must always appear in the S-projection. In the case of the swapped clause set, on the other hand, the existence of degenerate cases with no distribution imply that such cut-impertinent inferences should not always appear in the S-projection. While it is definitely possible to define a modified S-projection with a more fine-grained analysis of the inductive case for cut-impertinent inferences, as has been done for the profile in [66] for example, this will not be done here, because it is much simpler to use O-projections instead.

The construction of O-projections assumes only that the signature of the struct remains constant[m] during the struct normalization process. As this is the case for both $\rightsquigarrow_{\oplus\otimes}$ and $\rightsquigarrow_{\oplus\otimes_W}$, O-projections work equally well for standard and swapped clause sets. In fact, O-projections were originally developed here as projections that would work for swapped and profile clause sets, but they turned out to be quite independent of and hence robust to changes in the normalization process, allowing them to be used together with standard clause sets too.

[m] In Section 4.5, a struct normalizing process that extends the signature is defined. Therefore O-projections cannot be used there.

Example 4.27 (O-Projections for Swapped Clause Sets). *Consider the proof φ of Example 4.1 and its cut-pertinent swapped clause set shown in Example 4.26.*

Then, the projections $\lfloor\varphi\rfloor^{O}_{\vdash A,C}$ and $\lfloor\varphi\rfloor^{O}_{\vdash B,C}$ are:

$$\dfrac{\dfrac{\dfrac{A \vdash A}{A, B \vdash A}\, w_l}{A \wedge B \vdash A}\,\wedge_l \qquad C \vdash C}{(A \wedge B) \vee C \vdash A, C}\,\vee_l \qquad\qquad \dfrac{\dfrac{\dfrac{B \vdash B}{A, B \vdash B}\, w_l}{A \wedge B \vdash B}\,\wedge_l \qquad C \vdash C}{(A \wedge B) \vee C \vdash B, C}\,\vee_l$$

The projections $\lfloor\varphi\rfloor^{O}_{C\vdash}$ and $\lfloor\varphi\rfloor^{O}_{B,A\vdash}$ are:

$$C \vdash C \qquad\qquad \dfrac{B \vdash B \qquad A \vdash A}{A, B \vdash B \wedge A}\,\wedge_r$$

Comparing the projections above with the projections with respect to the clauses of C^{S}_{φ} shown in Example 4.15, it is clear that not only the number of projections is reduced (from six to four), since C^{W}_{φ} has fewer clauses than C^{S}_{φ}, but also the projections are less redundant as a whole, since each initial cut-pertinent occurrence is duplicated less often and thus the clauses have fewer occurrences.

4.4 CERes$_P$

Another cut-pertinent clause set, called profile, was developed in [66]. While there the profile is extracted directly from proofs where formula occurrences are annotated with labels, it is possible to define the profile in terms of an alternative struct rewriting system for the $\oplus\otimes$-normalization of cut-pertinent structs: $\rightsquigarrow_{\oplus\otimes_P}$ (Definition 4.4.1).

The rewriting system $\rightsquigarrow_{\oplus\otimes_P}$ is very similar to $\rightsquigarrow_{\oplus\otimes_W}$, with two differences: It avoids the first source of non-confluence[n], i.e. cases when the first two rules of $\rightsquigarrow_{\oplus\otimes_W}$ are applicable, because the only rewriting rule of $\rightsquigarrow_{\oplus\otimes_P}$ is essentially equivalent to a unique and minimal choice of application order of the two applicable rules of $\rightsquigarrow_{\oplus\otimes_W}$ in such cases. And it avoids the second source of non-confluence, i.e. the "degenerate" cases in which a whole substruct is deleted by one of the last four rules of $\rightsquigarrow_{\oplus\otimes_W}$, simply by keeping both substructs connected by a $\oplus$ connective (i.e. "degenerate" $\oplus$ remain $\oplus$ while "degenerate" $\otimes$ are converted to $\oplus$). Hence there is no choice of what substruct to delete in such cases anymore.

[n]Nevertheless, note that the first source of non-confluence for $\rightsquigarrow_{\oplus\otimes_W}$ is not so important, because it only creates copies that are eventually disregarded during clausification, since clause sets are sets. In other words, all swapped clause sets obtained from $\rightsquigarrow_{\oplus\otimes_W}$-normal-forms that differ only because of the first kind of non-confluence are equal.

Since the last four rules of $\rightsquigarrow_{\oplus\otimes_W}$ correspond to the downward swapping of weakening inferences ($\gg_{WD}$), it is clear that swapped clause sets exploit the presence of weakening in the proofs better than profile clause sets. This is the price paid by the profile clause set for its confluence. Related to this way of handling weakening is the fact that the profile clause set of a proof is the union of all the swapped clause sets of the proof.

It is interesting to note that it is possible to define an additional proof rewriting relation, $\gg_{cut}$, that corresponds to the conversion of "degenerate" $\otimes$ into $\oplus$. Therefore, while $\rightsquigarrow_{\oplus\otimes_W}$ corresponds to $\gg$ (Lemma 4.9), $\rightsquigarrow_{\oplus\otimes_P}$ corresponds to $\gg \cup \gg_{cut}$ (Lemma 4.11).

4.4.1 Cut-Pertinent Profile Clause Set

Definition 4.4.1 ($\rightsquigarrow_{\oplus\otimes_P}$). In the struct rewriting rule below, let ρ be the inference corresponding to $\otimes$ in the proof φ from which the struct was extracted. For the rewriting rule to be applicable $S_{n+1}, \ldots, S_{n+m}$ and $T_{k+1}, \ldots, T_{k+l}$ must contain at least one occurrence from $\Omega_\rho(\varphi)$ each, and $S_1, \ldots, S_n$ and $T_1, \ldots, T_k$ should not contain any occurrence from $\Omega_\rho(\varphi)$. Moreover, an innermost rewriting strategy is enforced: only minimal reducible substructs (i.e. structs having no reducible proper substruct) can be rewritten.

$$(T_1 \oplus \ldots \oplus T_k \oplus T_{k+1} \oplus \ldots \oplus T_{k+l}) \otimes (S_1 \oplus \ldots \oplus S_n \oplus S_{n+1} \oplus \ldots \oplus S_{n+m})$$

$$\rightsquigarrow_{\oplus\otimes_P}$$

$$(T_1 \oplus \ldots \oplus T_k \oplus S_1 \oplus \ldots \oplus S_n) \oplus \Big(\bigoplus_{\substack{1 \le i \le l \\ 1 \le j \le m}} T_{k+i} \otimes S_{n+j} \Big)$$

Example 4.28 ($\oplus\otimes_P$-Normalization). *Let φ be the proof below:*

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad B \vdash B}{A, B \vdash A \wedge B}\wedge_r^1}{A \wedge B \vdash A \wedge B}\wedge_l \qquad \dfrac{\dfrac{B \vdash B \qquad A \vdash A}{A, B \vdash B \wedge A}\wedge_r^2}{A \wedge B \vdash B \wedge A}\wedge_l}{A \wedge B \vdash B \wedge A}cut^3 \qquad \dfrac{C \vdash C \qquad C \vdash C}{C \vdash C}cut^4}{(A \wedge B) \vee C \vdash B \wedge A, C}\vee_l^5$$

Its cut-pertinent struct is:

$$\mathcal{S}_\varphi \equiv ((A \oplus^1 B) \oplus^3 (\neg B \otimes^2 \neg A)) \otimes^5 (C \oplus^4 \neg C)$$

And it can be normalized as follows:

$$\begin{array}{rcl}
\mathcal{S}_\varphi & \equiv & ((A \oplus^1 B) \oplus^3 (\neg B \otimes^2 \neg A)) \otimes^5 (C \oplus^4 \neg C) \\
& \rightsquigarrow_{\oplus\otimes_P} & ((A \oplus^1 B) \otimes^5 (C \oplus^4 \neg C)) \oplus^3 (\neg B \otimes^2 \neg A) \\
& \rightsquigarrow_{\oplus\otimes_P} & ((((A \otimes^5 C) \oplus^1 (B \otimes^5 C)) \oplus^4 \neg C)) \oplus^3 (\neg B \otimes^2 \neg A) \\
& \equiv & (A \otimes^5 C) \oplus (B \otimes^5 C) \oplus \neg C \oplus (\neg B \otimes^2 \neg A) \\
& \equiv & S
\end{array}$$

Note that $S = S_2$, for S_2 shown in Example 4.25. One application of a rewriting rule of $\rightsquigarrow_{\oplus\otimes_P}$ corresponded to two applications of rewriting rules of $\rightsquigarrow_{\oplus\otimes_W}$ in an optimal order.

Theorem 4.6 (Confluence of $\rightsquigarrow_{\oplus\otimes_P}$)**.** $\rightsquigarrow_{\oplus\otimes_P}$ is confluent.

Proof. This follows from the non-existence of critical pairs, which is easy to see, as $\rightsquigarrow_{\oplus\otimes_P}$ has only one rewriting rule and an innermost rewriting strategy is enforced. □

Definition 4.4.2 (Cut-pertinent Profile Clause Set)**.** The *cut-pertinent profile clause set* of a proof φ is:

$$C^P_\varphi \doteq \mathrm{cl}(S)$$

where S is the $\rightsquigarrow_{\oplus\otimes_P}$-normal-form of $\mathcal{S}_\varphi$.

Example 4.29 (Cut-Pertinent Profile Clause Set)**.** *Let φ be the proof considered in Example 4.28. Then:*

$$C^P_\varphi = \{ \vdash A, C \ ; \ \vdash B, C \ ; \ C \vdash \ ; \ B, A \vdash \}$$

Comparing C^P_φ with $C^W_{\varphi|S_1}$ and $C^W_{\varphi|S_2}$ shown in Example 4.26, it is interesting to note that $C^P_\varphi = C^W_{\varphi|S_1} = C^W_{\varphi|S_2}$. This is so because there are no degenerate cut-impertinent inferences in φ.

Example 4.30 (Cut-Pertinent Profile Clause Set)**.**

Definition 4.4.3 ($\gg_{cut}$)**.** Conversion of degenerate inferences to cuts:

$$\dfrac{\dfrac{\dfrac{\varphi_1}{\Gamma_1 \vdash \Delta_1}}{\Gamma_1^\rho, \Gamma_1 \vdash \Delta_1^\rho, \Delta_1}\, w^* \qquad \dfrac{\dfrac{\varphi_2}{\Gamma_2 \vdash \Delta_2}}{\Gamma_2^\rho, \Gamma_2 \vdash \Delta_2^\rho, \Delta_2}\, w^*}{\Gamma^\rho, \Gamma_1, \Gamma_2 \vdash \Delta^\rho, \Delta_1, \Delta_2}\, \rho$$

$$\Downarrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\varphi_1}{\Gamma_1 \vdash \Delta_1}}{\Gamma_1 \vdash \Delta_1, A}\, w_r \qquad \dfrac{\dfrac{\varphi_2}{\Gamma_2 \vdash \Delta_2}}{A, \Gamma_2 \vdash \Delta_2}\, w_l}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\, cut}{\Gamma^\rho, \Gamma_1, \Gamma_2 \vdash \Delta^\rho, \Delta_1, \Delta_2}\, w^*$$

Remark 4.4.1. A degenerate inference ρ, as shown in Definition 4.4.3, could also be removed by using $\gg_W$. However, in this case only one of the subproofs could be kept and the end-sequent of the other subproof together with the main formula of ρ would be introduced by weakening. Therefore, it would be necessary to non-deterministically choose which subproof to keep.

It is precisely in the presence of degenerate inferences that profile clause sets and swapped clause sets differ. While profile clause sets avoid non-determinism by using a struct normalization that corresponds to $\gg_{cut}$ in such cases, swapped clause sets do not care about non-determinism and prefer to prune whole subproofs by using a struct normalization that corresponds to $\gg_W$ in such degenerate cases.

Lemma 4.10 (Correspondence between $\rightsquigarrow_{\oplus\otimes_P}$ and $\gg \cup \gg_{cut}$)**.** If φ is skolemized and $\mathcal{S}_\varphi \rightsquigarrow_{\oplus\otimes_P} S_\psi$, then there exists a proof ψ such that $\varphi(\gg \cup \gg_{cut})^*\psi$ and $\mathcal{S}_\psi = S_\psi$.

Proof. The proof is analogous to the proof of Lemma 4.8, except that the case for degenerate cut-impertinent inferences corresponds to $\gg_{cut}$. □

Lemma 4.11 (Iterated Correspondence between $\rightsquigarrow_{\oplus\otimes_P}$ and $\gg \cup \gg_{cut}$)**.** If φ is skolemized and $\mathcal{S}_\varphi \rightsquigarrow^*_{\oplus\otimes_P} S_\psi$, then there exists a proof ψ such that $\varphi(\gg \cup \gg_{cut})^*\psi$ and $\mathcal{S}_\psi = S_\psi$.

Proof. This Lemma is just the iterated version of Lemma 4.10 and can be easily proved by induction on the number of rewriting steps to rewrite $\mathcal{S}_\varphi$ to S_ψ. □

Theorem 4.7 (Unsatisfiability of the Cut-Pertinent Profile Clause Set)**.** For any skolemized proof φ, C^P_φ is unsatisfiable.

Proof. The original proof of the unsatisfiability of the profile can be found in [66, 67]. The idea of the proof is to note that C^P_φ always subsumes C^S_φ, which is, by Theorem 4.1, unsatisfiable. Therefore, C^P_φ is also unsatisfiable. □

Proof. Alternatively, this theorem can be proved analogously to the proof of Theorem 4.5. By Lemma 4.11, there exists a proof ψ such that $\mathcal{S}_\psi = S$. Clearly, $C^P_\psi = C^S_\psi$, since S is also a $\rightsquigarrow_{\oplus\otimes}$-normal-form. Therefore, $C^P_\varphi = C^S_\psi$ and, by Theorem 4.1, it is unsatisfiable. □

4.4.2 Projections

A modified inductive method for constructing S-projections for profile clause sets is shown in [66]. The method for constructing O-projections, on the other hand works without any modification for profile clause sets.

Example 4.31 (O-Projections for Profile Clause Sets)**.** *Consider the proof φ of Example 4.1. Since its profile clause set C^P_φ is equal to its swapped clause set C^W_φ shown in Example 4.26, the O-projections with respect to the clauses of C^P_φ are exactly the same as the O-projections with respect to the clauses of C^W_φ, which are shown in Example 4.27.*

4.5 CERes$_D$

The construction of standard clause sets from structs is analogous to the standard transformation of formulas to conjunctive normal forms. Consequently, it has the same well-known disadvantage of increasing the size significantly in the worst case. Indeed, the size of a standard clause set can be exponential with respect to the size of the struct from which it is constructed, in the same way that the size of a clause normal form of a formula can be exponential with respect to the size of the formula itself. There exists, however, an improved technique known as *structural clause form transformation* [2], based on the extension principle. By using this technique, it can be shown that the atomic size of the clause normal form of a formula is in the worst case only linearly bigger[o] than the size of the formula itself. The price paid is that the structural conjunctive normal form of a formula is not logically equivalent to the formula anymore,

[o]However, the symbolic size can be quadratic due to Skolem terms produced by skolemization.

because new defined predicate symbols are added, thus extending the signature. Nevertheless, satisfiability-equivalence is preserved: the formula is unsatisfiable if and only if its structural clause form is unsatisfiable.

Although profile clause sets and swapped clause sets are great improvements of the standard clause set, it is not so hard to see that, in the worst case, the size of these clause sets is still exponential with respect to the size of the struct, because distributive duplications still occur (in cases corresponding to swapping of indirectly dependent inferences). It is therefore only natural to investigate the possibility of adapting the idea of structural clause form transformation to the construction of clause sets from structs, in order to avoid the exponential blow-up in size in the worst case[p]. The purpose of this section is to show how this can be done.

4.5.1 Cut-Pertinent Definitional Clause Set

Definition 4.5.1 adapts to structs the idea of structural conjunctive normal form transformation. Every substruct is given a new name, a new predicate symbol defined to be equivalent to the substruct. The defining formulas are very shallow formulas and can be easily transformed to $\oplus$-junctions of $\otimes$-junctions.

Definition 4.5.1 ($\rightsquigarrow_{\oplus\otimes_D}$)**.** Let S be a struct. For every non-literal substruct $S' \equiv S'_1 \otimes \ldots \otimes S'_n$ or $S' \equiv S'_1 \oplus \ldots \oplus S'_n$ of S, a new predicate symbol can be created together with a corresponding defining formula:

$$Def_{S'} \doteq N_{S'}(x_1, \ldots, x_m) \leftrightarrow N_{n(S'_1)\otimes \ldots \otimes n(S'_n)}(x_1, \ldots, x_m) \leftrightarrow n(S'_1) \otimes \ldots \otimes n(S'_n)$$

or

$$Def_{S'} \doteq N_{S'}(x_1, \ldots, x_m) \leftrightarrow N_{n(S'_1)\oplus \ldots \oplus n(S'_n)}(x_1, \ldots, x_m) \leftrightarrow n(S'_1) \oplus \ldots \oplus n(S'_n)$$

where $x_1, \ldots, x_m$ are the free variables of S', $n(S'_k)$ is S'_k if S'_k is a literal struct and $N_{S'_k}(y_1, \ldots, y_j)$ if S'_k is a non-literal struct with free variables $y_1, \ldots, y_j$. The connective $\leftrightarrow$ is considered to be just an abbreviation[q]:

$$A \leftrightarrow B_1 \otimes \ldots \otimes B_n \doteq (\overline{A} \otimes B_1 \otimes \ldots \otimes B_n) \oplus (\overline{B_1} \otimes A) \oplus \ldots \oplus (\overline{B_n} \otimes A)$$

$$A \leftrightarrow B_1 \oplus \ldots \oplus B_n \doteq (\overline{B_1} \otimes \ldots \otimes \overline{B_n} \otimes A) \oplus (\overline{A} \otimes B_1) \oplus \ldots \oplus (\overline{A} \otimes B_n)$$

[p]Since structs do not contain quantifiers, no skolemization is necessary. Therefore, by adapting the structural clause form transformation technique to structs, not only the atomic size of the clause set remains linearly bounded with respect to the atomic size of the struct but also its symbolic size remains linearly bounded with respect to the symbolic size of the struct.

[q]The abbreviation can be intuitively understood due to the analogy of $\otimes$ with $\vee$ and $\oplus$ with $\wedge$.

where $\overline{C}$ is $\neg D$, if $C = D$, and D, if $C = \neg D$.

Then:

$$S \rightsquigarrow_{\oplus\otimes_D} S^*$$

where:

$$S^* \doteq n(S) \oplus \bigoplus_{\text{non-literal substructs } S' \text{ of } S} Def_{S'}$$

Each defining formula $Def_{S'}$ originates so-called *definitional* $\otimes$*-junctions*. All other $\otimes$-junctions (e.g. $n(S)$) are called *proper* $\otimes$*-junctions*.

Example 4.32 ($\oplus\otimes_D$-Normalization). *Let φ be the proof below:*

$$\begin{prooftree}
\AxiomC{$A \vdash A$}
\AxiomC{$B \vdash B$}
\RightLabel{$\wedge_r^1$}
\BinaryInfC{$A, B \vdash A \wedge B$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash A \wedge B$}
\AxiomC{$B \vdash B$}
\AxiomC{$A \vdash A$}
\RightLabel{$\wedge_r^2$}
\BinaryInfC{$A, B \vdash B \wedge A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash B \wedge A$}
\RightLabel{cut^3}
\BinaryInfC{$A \wedge B \vdash B \wedge A$}
\AxiomC{$C \vdash C$}
\AxiomC{$C \vdash C$}
\RightLabel{cut^4}
\BinaryInfC{$C \vdash C$}
\RightLabel{$\vee_l^5$}
\BinaryInfC{$(A \wedge B) \vee C \vdash B \wedge A, C$}
\end{prooftree}$$

Its cut-pertinent struct is:

$$\mathcal{S}_\varphi \equiv ((A \oplus B) \oplus (\neg B \otimes \neg A)) \otimes (C \oplus \neg C)$$

New predicate symbols can be created and defined by the following formulas:

- $D \leftrightarrow C \oplus \neg C$
- $E \leftrightarrow \neg B \otimes \neg A$
- $F \leftrightarrow A \oplus B$
- $G \leftrightarrow F \oplus E$
- $H \leftrightarrow G \otimes D$

Finally, the $\rightsquigarrow_{\oplus\otimes_D}$-normal-form of $\mathcal{S}_\varphi$ is:

$$\begin{aligned}
S^* \quad \doteq \quad & H \oplus \\
& (\neg D \otimes C) \oplus (\neg D \otimes \neg C) \oplus (\neg C \otimes D \otimes C) \oplus \\
& (\neg E \otimes \neg B \otimes \neg A) \oplus (E \otimes B) \oplus (E \otimes A) \oplus \\
& (\neg F \otimes A) \oplus (\neg F \otimes B) \oplus (\neg A \otimes \neg B \otimes F) \oplus \\
& (\neg G \otimes F) \oplus (\neg G \otimes E) \oplus (\neg E \otimes \neg F \otimes G) \oplus \\
& (\neg H \otimes G \otimes D) \oplus (\neg G \otimes H) \oplus (\neg D \otimes H)
\end{aligned}$$

Definition 4.5.2 (Cut-pertinent Definitional Clause Set)**.** The *cut-pertinent definitional clause set* of a proof φ is:

$$C^D_\varphi \doteq \mathrm{cl}(S)$$

where S is the $\rightsquigarrow_{\oplus\otimes_D}$-normal-form of $\mathcal{S}_\varphi$.

The clauses corresponding to definitional $\otimes$-junctions are called *definitional clauses*. The clauses corresponding to proper $\otimes$-junctions are called *proper clauses*.

Example 4.33 (Definitional Clause Set)**.** *Let φ be the proof in Example 4.32. Then, its definitional clause set C^D_φ consists of the following clauses. The proper clause is $\vdash H$. All other clauses are definitional clauses.*

$D \vdash C$	$D, C \vdash$	$C \vdash D, C$
$E, B, A \vdash$	$\vdash E, B$	$\vdash E, A$
$F \vdash A$	$F \vdash B$	$A, B \vdash F$
$G \vdash F$	$G \vdash E$	$E, F \vdash G$
$H \vdash G, D$	$G \vdash H$	$D \vdash H$
$\vdash H$		

4.5.2 Projections

The construction of projections requires special care when definitional clause sets are used. The reason is that the clauses now contain many new predicate symbols which do not occur in the proof. Since S-projections and O-projections contain only symbols that occur in the proof, it is clear that they cannot be used with definitional clause sets. New kinds of projections, called *D-projections* have to be developed.

D-Projections

For all definitional clauses of a definitional clause set, projections can be constructed very easily by using definition rules, even without any dependence on the proof. These projections are the *definitional D-projections* explained in Definition 4.5.3. However, in every definitional clause set there is exactly one clause, namely the proper clause, for which definitional D-projections do not work. Then a *proper D-projection* (Definition 4.5.4) is necessary. It is called proper, because it actually depends on the proof.

Definition 4.5.3 (Definitional D-Projection)**.** Let φ be a proof and c a definitional clause in C^D_φ. The *Definitional D-projection* $\lfloor\varphi\rfloor_c^{D_D}$ with respect to the clause c can be easily constructed by using definition rules, as exemplified below:

Assume c is one of the definitional clauses originating from the following defining formula:

$$Def_{S'} \doteq N_{n(S'_1)\otimes\ldots\otimes n(S'_n)}(x_1,\ldots,x_m) \leftrightarrow n(S'_1)\otimes\ldots\otimes n(S'_n)$$

Then c is one of the following clauses:

- $N_{n(S'_1)\otimes\ldots\otimes n(S'_n)}(x_1,\ldots,x_m) \vdash n(S'_1),\ldots,n(S'_n)$
- $n(S'_1) \vdash N_{n(S'_1)\otimes\ldots\otimes n(S'_n)}(x_1,\ldots,x_m)$
- ...
- $n(S'_n) \vdash N_{n(S'_1)\otimes\ldots\otimes n(S'_n)}(x_1,\ldots,x_m)$

And the definitional D-projections are:

$\lfloor\varphi\rfloor^{D_D}_{N_{S'_1\otimes\ldots\otimes S'_n}(x_1,\ldots,x_m)\vdash n(S'_1),\ldots,n(S'_n)}$:

$$\cfrac{\cfrac{n(S'_1)\vdash n(S'_1) \quad \ldots \quad n(S'_n)\vdash n(S'_n)}{n(S'_1)\vee\ldots\vee n(S'_n)\vdash n(S'_1),\ldots,n(S'_n)}\ \vee_l^*}{N_{n(S'_1)\otimes\ldots\otimes n(S'_n)}(x_1,\ldots,x_m)\vdash n(S'_1),\ldots,n(S'_n)}\ d_l$$

$\lfloor\varphi\rfloor^{D_D}_{n(S'_k)\vdash N_{S'_1\otimes\ldots\otimes S'_n}(x_1,\ldots,x_m)}$:

$$\cfrac{\cfrac{\cfrac{n(S'_k)\vdash n(S'_k)}{n(S'_k)\vdash n(S'_1),\ldots,n(S'_n)}\ w_r^*}{n(S'_k)\vdash n(S'_1)\vee\ldots\vee n(S'_n)}\ \vee_r^*}{n(S'_k)\vdash N_{n(S'_1)\otimes\ldots\otimes n(S'_n)}(x_1,\ldots,x_m)}\ d_r$$

If c, on the other hand is one of the definitional clauses originating from the following defining formula:

$$Def_{S'} \doteq N_{n(S'_1)\oplus\ldots\oplus n(S'_n)}(x_1,\ldots,x_m) \leftrightarrow n(S'_1)\oplus\ldots\oplus n(S'_n)$$

Then c is one of the following clauses:

- $n(S'_1),\ldots,n(S'_n) \vdash N_{n(S'_1)\oplus\ldots\oplus n(S'_n)}(x_1,\ldots,x_m)$
- $N_{n(S'_1)\oplus\ldots\oplus n(S'_n)}(x_1,\ldots,x_m) \vdash n(S'_1)$
- ...
- $N_{n(S'_1)\oplus\ldots\oplus n(S'_n)}(x_1,\ldots,x_m) \vdash n(S'_n)$

And the definitional D-projections are:

$\lfloor\varphi\rfloor^{D_D}_{N_{S'_1\otimes\ldots\otimes S'_n}(x_1,\ldots,x_m)\vdash n(S'_1),\ldots,n(S'_n)}$:

$$\dfrac{\dfrac{n(S'_1)\vdash n(S'_1) \qquad \ldots \qquad n(S'_n)\vdash n(S'_n)}{n(S'_1),\ldots,n(S'_n)\vdash n(S'_1)\wedge\ldots\wedge n(S'_n)}\,\wedge_r^*}{n(S'_1),\ldots,n(S'_n)\vdash N_{n(S'_1)\oplus\ldots\oplus n(S'_n)}(x_1,\ldots,x_m)}\,d_r$$

$\lfloor\varphi\rfloor^{D_D}_{n(S'_k)\vdash N_{S'_1\otimes\ldots\otimes S'_n}(x_1,\ldots,x_m)}$:

$$\dfrac{\dfrac{\dfrac{n(S'_k)\vdash n(S'_k)}{n(S'_1),\ldots,n(S'_n)\vdash n(S'_k)}\,w_l^*}{n(S'_1)\wedge\ldots\wedge n(S'_n)\vdash n(S'_k)}\,\wedge_l^*}{N_{n(S'_1)\oplus\ldots\oplus n(S'_n)}(x_1,\ldots,x_m)\vdash n(S'_k)}\,d_l$$

If S'_k is a negative literal, it is necessary to add negation inferences to the definitional D-projections above.

Example 4.34 (Definitional D-Projection)**.** *The simple D-projections are:*

$\lfloor\varphi\rfloor^{D_D}_{D\vdash C}$:

$$\dfrac{\dfrac{\dfrac{C\vdash C}{C,\neg C\vdash C}\,w_l}{C\wedge\neg C\vdash C}\,\wedge_l}{D\vdash C}\,d_l$$

$\lfloor\varphi\rfloor^{D_D}_{D,C\vdash}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{C\vdash C}{C,C\vdash C}\,w_l}{C,\neg C,C\vdash}\,\neg_l}{C\wedge\neg C,C\vdash}\,\wedge_l}{D,C\vdash}\,d_l$$

$\lfloor\varphi\rfloor^{D_D}_{C\vdash D,C}$:

$$\dfrac{\dfrac{C\vdash C \qquad \dfrac{C\vdash C}{\vdash C,\neg C}\,\neg_r}{C\vdash C,C\wedge\neg C}\,\wedge_r}{C\vdash C,D}\,d_r$$

$\lfloor\varphi\rfloor^{D_D}_{\vdash E,A}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{A\vdash A}{\vdash\neg A,A}\,\neg_r}{\vdash\neg B,\neg A,A}\,w_r}{\vdash\neg B\vee\neg A,A}\,\vee_r}{\vdash E,A}\,d_r$$

$\lfloor\varphi\rfloor^{D_D}_{\vdash E,B}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{B\vdash B}{\vdash\neg B,B}\,\neg_r}{\vdash\neg B,\neg A,B}\,w_r}{\vdash\neg B\vee\neg A,B}\,\vee_r}{\vdash E,B}\,d_r$$

$\lfloor\varphi\rfloor^{D_D}_{E,A,B\vdash}$:

$$\dfrac{\dfrac{\dfrac{B\vdash B}{\neg B,B\vdash}\,\neg_l \qquad \dfrac{A\vdash A}{\neg A,A\vdash}\,\neg_l}{\neg B\vee\neg A,B,A\vdash}\,\vee_l}{E,B,A\vdash}\,d_l$$

$\lfloor\varphi\rfloor^{D_D}_{F\vdash A}$:

$$\dfrac{\dfrac{\dfrac{A\vdash A}{A,B\vdash A}\,w_l}{A\wedge B\vdash A}\,\wedge_l}{F\vdash A}\,d_l$$

$\lfloor\varphi\rfloor^{D_D}_{F\vdash B}$:

$$\dfrac{\dfrac{\dfrac{B\vdash B}{A,B\vdash B}\,w_l}{A\wedge B\vdash B}\,\wedge_l}{F\vdash B}\,d_l$$

$\lfloor\varphi\rfloor^{D_D}_{A,B\vdash F}$:

$$\dfrac{\dfrac{A\vdash A \qquad B\vdash B}{A,B\vdash A\wedge B}\,\wedge_r}{A,B\vdash F}\,d_r$$

$\lfloor\varphi\rfloor^{D_D}_{G\vdash F}$:

$$\dfrac{\dfrac{\dfrac{F\vdash F}{F,E\vdash F}\,w_l}{F\wedge E\vdash F}\,\wedge_l}{G\vdash F}\,d_l$$

$\lfloor\varphi\rfloor^{D_D}_{G\vdash E}$:

$$\dfrac{\dfrac{\dfrac{E\vdash E}{F,E\vdash E}\,w_l}{F\wedge E\vdash E}\,\wedge_l}{G\vdash E}\,d_l$$

$\lfloor\varphi\rfloor^{D_D}_{F,E\vdash G}$:

$$\dfrac{\dfrac{F\vdash F \qquad E\vdash E}{F,E\vdash F\wedge E}\,\wedge_r}{F,E\vdash G}\,d_r$$

$\lfloor\varphi\rfloor^{D_D}_{D\vdash H}$:

$$\dfrac{\dfrac{\dfrac{D\vdash D}{D\vdash G,D}\,w_r}{D\vdash G\vee D}\,\vee_r}{D\vdash H}\,d_r$$

$\lfloor\varphi\rfloor^{D_D}_{G\vdash H}$:

$$\dfrac{\dfrac{\dfrac{G \vdash G}{G \vdash G, D}\, w_r}{G \vdash G \vee D}\, \vee_r}{G \vdash H}\, d_r \qquad \lfloor\varphi\rfloor^{D_D}_{H \vdash G,D}: \qquad \dfrac{\dfrac{G \vdash G \quad D \vdash D}{G \vee D \vdash G, D}\, \vee_l}{H \vdash G, D}\, d_l$$

Definition 4.5.4 (Proper D-Projection)**.** Let φ be a proof and $\mathcal{S}_\varphi$ its cut-pertinent struct. Then, the *proper D-projection* $\lfloor\varphi\rfloor^{D_P}_{\vdash n(\mathcal{S}_\varphi)}$ can be constructed inductively. Let φ' be a subproof of φ having ρ as its last inference and let S' be the corresponding substruct of $\mathcal{S}_\varphi$. The following cases can be distinguished:

- ρ is an axiom inference: Then φ' is of the form:

$$\dfrac{}{A \vdash A}\, \rho$$

  - If both occurrences of A are in $\Omega_{CP}(\varphi)$ (i.e. they are ancestors of cut-formulas), then φ'' is defined as:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash A}\, \rho}{\vdash \neg A, A}\, \neg_r}{\vdash \neg A \vee A}\, \vee_r}{\vdash n(S')}\, d_r$$

  - If only the occurrence of A in the antecedent is in $\Omega_{CP}(\varphi)$ (i.e. an ancestor of a cut-formula), then φ'' is defined as:

$$\dfrac{\dfrac{}{A \vdash A}\, \rho}{\vdash \neg A, A}\, \neg_r$$

  - Otherwise, $\varphi'' \doteq \varphi'$

- ρ is a n-ary inference (with $n \geq 2$): Then φ' is of the form:

$$\dfrac{\begin{array}{c}\psi'_1\\ \Gamma'_1 \vdash \Delta'_1\end{array} \quad \dots \quad \begin{array}{c}\psi'_n\\ \Gamma'_n \vdash \Delta'_n\end{array}}{\Gamma' \vdash \Delta'}\, \rho$$

  By induction, ψ''_k is of the form:

$$\begin{array}{c}\psi''_k\\ \Gamma''_1 \vdash \Delta''_1, n(S'_{\psi'_k})\end{array}$$

  where $S'_{\psi'_k}$ is the substruct of S' corresponding to ψ'_k.

– ρ is cut-impertinent: Then φ'' is defined as:

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\psi_1'' \\ \Gamma_1'' \vdash \Delta_1'', n(S'_{\psi_1'})\end{array} \quad \ldots \quad \begin{array}{c}\psi_n'' \\ \Gamma_n'' \vdash \Delta_n'', n(S'_{\psi_n'})\end{array}}{\Gamma'' \vdash \Delta'', n(S'_{\psi_1'}), \ldots, n(S'_{\psi_n'})}\,\rho}{\Gamma'' \vdash \Delta'', n(S'_{\psi_1'}) \vee \ldots \vee n(S'_{\psi_n'})}\,\vee_r}{\Gamma'' \vdash \Delta'', n(S')}\,d_r$$

More informally, after the cut-impertinent inference ρ, the defining components of $n(S')$ are available to be combined disjunctively. By the defining formula of $n(S')$, a d_r inference can be used to encapsulate the disjunction in the single defined predicate symbol $n(S')$.

– ρ is cut-pertinent: Then φ'' is defined as:

$$\dfrac{\dfrac{\begin{array}{c}\psi_1'' \\ \Gamma_1'' \vdash \Delta_1'', n(S'_{\psi_1'})\end{array} \quad \ldots \quad \begin{array}{c}\psi_n'' \\ \Gamma_n'' \vdash \Delta_n'', n(S'_{\psi_n'})\end{array}}{\Gamma'' \vdash \Delta'', n(S'_{\psi_1'}) \wedge \ldots \wedge n(S'_{\psi_n'})}\,\wedge_r}{\Gamma'' \vdash \Delta'', n(S')}\,d_r$$

More informally, the cut-pertinent inference ρ can be replaced by a $\wedge_r$ inference, which combines the defining components of $n(S')$ conjunctively. By the defining formula of $n(S')$, a d_r inference can be used to encapsulate the conjunction in the single defined predicate symbol $n(S')$.

- ρ is a unary inference: Then φ' is of the form:

$$\dfrac{\psi'}{\Gamma' \vdash \Delta'}\,\rho$$

– ρ is cut-pertinent: then φ'' is defined as:

$$\psi''$$

More informally, ρ is simply skipped.

– ρ is cut-impertinent: then φ'' is defined as:

$$\dfrac{\psi''}{\Gamma'' \vdash \Delta''}\,\rho$$

> More informally, ρ is simply kept and nothing changes, except for the downward propagation of changes that occurred in transforming the proof ψ' above to ψ''.

The *proper D-projection* $\lfloor \varphi \rfloor^{D_P}_{\vdash n(\mathcal{S}_\varphi)}$ is the final result of this inductive construction, i.e. it is φ'' when the subproof φ' coincides with the whole proof φ.

Example 4.35 (Proper D-Projection)**.** *Consider again the proof φ from previous examples:*

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}\AxiomC{$B \vdash B$}\RightLabel{$\wedge_r^1$}\BinaryInfC{$A, B \vdash A \wedge B$}\RightLabel{$\wedge_l$}\UnaryInfC{$A \wedge B \vdash A \wedge B$}
\AxiomC{$B \vdash B$}\AxiomC{$A \vdash A$}\RightLabel{$\wedge_r^2$}\BinaryInfC{$A, B \vdash B \wedge A$}\RightLabel{$\wedge_l$}\UnaryInfC{$A \wedge B \vdash B \wedge A$}
\RightLabel{cut^3}\BinaryInfC{$A \wedge B \vdash B \wedge A$}
\AxiomC{$C \vdash C$}\AxiomC{$C \vdash C$}\RightLabel{cut^4}\BinaryInfC{$C \vdash C$}
\RightLabel{$\vee_l^5$}\BinaryInfC{$(A \wedge B) \vee C \vdash B \wedge A, C$}
\end{prooftree}
$$

Below the inductive construction of the proper D-projection $\lfloor \varphi \rfloor^{D_D}_{\vdash H}$ is shown step-by-step. An informal skeleton of the original proof is shown in every step, just to emphasize that the construction follows the structure of the original proof.

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}\AxiomC{$B \vdash B$}\RightLabel{$\wedge_r^1$}\BinaryInfC{}\RightLabel{$\wedge_l$}\UnaryInfC{}
\AxiomC{$B \vdash B$}\AxiomC{$A \vdash A$}\RightLabel{$\wedge_r^2$}\BinaryInfC{}\RightLabel{$\wedge_l$}\UnaryInfC{}
\RightLabel{cut^3}\BinaryInfC{}
\AxiomC{$C \vdash C$}\AxiomC{$C \vdash C$}\RightLabel{cut^4}\BinaryInfC{}
\RightLabel{$\vee_l^5$}\BinaryInfC{}
\end{prooftree}
$$

Some of the axiom sequents contain cut-pertinent formula occurrences in the antecedents. It is necessary, therefore, to add $\neg_r$ inferences to move these formula occurrences to the consequents:

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}\AxiomC{$B \vdash B$}\RightLabel{$\wedge_r^1$}\BinaryInfC{}\RightLabel{$\wedge_l$}\UnaryInfC{}
\AxiomC{$B \vdash B$}\RightLabel{$\neg_r$}\UnaryInfC{$\vdash \neg B, B$}\AxiomC{$A \vdash A$}\RightLabel{$\neg_r$}\UnaryInfC{$\vdash \neg A, A$}\RightLabel{$\wedge_r^2$}\BinaryInfC{}\RightLabel{$\wedge_l$}\UnaryInfC{}
\RightLabel{cut^3}\BinaryInfC{}
\AxiomC{$C \vdash C$}\AxiomC{$C \vdash C$}\RightLabel{$\neg_r$}\UnaryInfC{$\vdash \neg C, C$}\RightLabel{cut^4}\BinaryInfC{}
\RightLabel{$\vee_l^5$}\BinaryInfC{}
\end{prooftree}
$$

$\wedge_r^1$ and cut^4 are cut-pertinent inferences, and hence they must be replaced by $\wedge_r$ inferences followed by appropriate d_r inferences. $\wedge_r^2$, on the other hand, is a cut-impertinent inference. Therefore, a $\vee_r$ inference and a d_r inference must be added after $\wedge_r^2$:

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\AxiomC{$B \vdash B$}
\RightLabel{$\wedge$}
\BinaryInfC{$A, B \vdash A \wedge B$}
\RightLabel{d_r}
\UnaryInfC{$A, B \vdash F$}
\RightLabel{$\wedge_l$}
\UnaryInfC{}
\AxiomC{$B \vdash B$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash \neg B, B$}
\AxiomC{$A \vdash A$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash \neg A, A$}
\RightLabel{$\wedge_r^2$}
\BinaryInfC{$\vdash \neg B, \neg A, B \wedge A$}
\RightLabel{$\vee_r$}
\UnaryInfC{$\vdash \neg B \vee \neg A, B \wedge A$}
\RightLabel{d_r}
\UnaryInfC{$\vdash E, B \wedge A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{}
\RightLabel{cut^3}
\BinaryInfC{}
\AxiomC{$C \vdash C$}
\AxiomC{$C \vdash C$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash \neg C, C$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$C \vdash C \wedge \neg C, C$}
\RightLabel{d_r}
\UnaryInfC{$C \vdash D, C$}
\RightLabel{$\vee_l^5$}
\BinaryInfC{}
\end{prooftree}
$$

The leftmost $\wedge_l$ unary inference is cut-impertinent, and hence must be kept. The rightmost $\wedge_l$ unary inference, on the other hand, is cut-impertinent, and hence must be skipped.

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\AxiomC{$B \vdash B$}
\RightLabel{$\wedge$}
\BinaryInfC{$A, B \vdash A \wedge B$}
\RightLabel{d_r}
\UnaryInfC{$A, B \vdash F$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash F$}
\AxiomC{$B \vdash B$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash \neg B, B$}
\AxiomC{$A \vdash A$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash \neg A, A$}
\RightLabel{$\wedge_r^2$}
\BinaryInfC{$\vdash \neg B, \neg A, B \wedge A$}
\RightLabel{$\vee_r$}
\UnaryInfC{$\vdash \neg B \vee \neg A, B \wedge A$}
\RightLabel{d_r}
\UnaryInfC{$\vdash E, B \wedge A$}
\RightLabel{cut^3}
\BinaryInfC{}
\AxiomC{$C \vdash C$}
\AxiomC{$C \vdash C$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash \neg C, C$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$C \vdash C \wedge \neg C, C$}
\RightLabel{d_r}
\UnaryInfC{$C \vdash D, C$}
\RightLabel{$\vee_l^5$}
\BinaryInfC{}
\end{prooftree}
$$

The procedure for cut^3 is analogous. It must be replaced by $\wedge^r$ and d^r:

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\AxiomC{$B \vdash B$}
\RightLabel{$\wedge$}
\BinaryInfC{$A, B \vdash A \wedge B$}
\RightLabel{d_r}
\UnaryInfC{$A, B \vdash F$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash F$}
\AxiomC{$B \vdash B$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash \neg B, B$}
\AxiomC{$A \vdash A$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash \neg A, A$}
\RightLabel{$\wedge_r^2$}
\BinaryInfC{$\vdash \neg B, \neg A, B \wedge A$}
\RightLabel{$\vee_r$}
\UnaryInfC{$\vdash \neg B \vee \neg A, B \wedge A$}
\RightLabel{d_r}
\UnaryInfC{$\vdash E, B \wedge A$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A \wedge B \vdash F \wedge E, B \wedge A$}
\RightLabel{d_r}
\UnaryInfC{$A \wedge B \vdash G, B \wedge A$}
\AxiomC{$C \vdash C$}
\AxiomC{$C \vdash C$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash \neg C, C$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$C \vdash C \wedge \neg C, C$}
\RightLabel{d_r}
\UnaryInfC{$C \vdash D, C$}
\RightLabel{$\vee_l^5$}
\BinaryInfC{}
\end{prooftree}
$$

Finally, $\vee_r$ and d_r are added after the cut-impertinent $\vee_l^5$ inference, thus resulting in the following proper D-projection:

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\AxiomC{$B \vdash B$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A, B \vdash A \wedge B$}
\RightLabel{d_r}
\UnaryInfC{$A, B \vdash F$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash F$}
\AxiomC{$B \vdash B$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash \neg B, B$}
\AxiomC{$A \vdash A$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash \neg A, A$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$\vdash \neg B, \neg A, B \wedge A$}
\RightLabel{$\vee_r$}
\UnaryInfC{$\vdash \neg B \vee \neg A, B \wedge A$}
\RightLabel{d_r}
\UnaryInfC{$\vdash E, B \wedge A$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A \wedge B \vdash F \wedge E, B \wedge A$}
\RightLabel{d_r}
\UnaryInfC{$A \wedge B \vdash G, B \wedge A$}
\AxiomC{$C \vdash C$}
\AxiomC{$C \vdash C$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash \neg C, C$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$C \vdash C \wedge \neg C, C$}
\RightLabel{d_r}
\UnaryInfC{$C \vdash D, C$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(A \wedge B) \vee C \vdash G, D, B \wedge A, C$}
\RightLabel{$\vee_r$}
\UnaryInfC{$(A \wedge B) \vee C \vdash G \vee D, B \wedge A, C$}
\RightLabel{d_r}
\UnaryInfC{$(A \wedge B) \vee C \vdash H, B \wedge A, C$}
\end{prooftree}
$$

4.6 CERes_{DW}

Although the number of defined symbols introduced by the construction of definitional clause sets is only linearly bounded with respect to the size of the structs, it is still far from optimal. A technique that combines ideas from swapped clause sets and from definitional clause sets can be used to significantly reduce this number. Once again, the difficulty lies in the projections. As in the case of definitional clause sets, a new notion of projection has to be developed.

4.6.1 Cut-Pertinent Swapped Definitional Clause Set

Swapped definitional clause sets are obtained by a straightforward combination of the normalizations used for swapped clause sets and for definitional clause sets. Initially, a restricted form of $\rightsquigarrow_{\oplus\otimes_W}$-normalization (namely $\rightsquigarrow_{\oplus\otimes_{DW_W}}$) can be applied as long as no duplications of substructs occur. Subsequently, a limited form of $\rightsquigarrow_{\oplus\otimes_D}$ (namely $\rightsquigarrow_{\oplus\otimes_{DW_D}}$) can be applied with the restriction that only substructs that are $\oplus$-junctions nested within $\otimes$-junctions are replaced by new defined predicates.

Definition 4.6.1 ($\rightsquigarrow_{\oplus\otimes_{DW}}$)**.** In the struct rewriting rules below, let ρ be the inference in φ corresponding to $\otimes_\rho$. For the rewriting rules to be applicable, S_k and S must contain at least one occurrence from $\Omega_\rho(\varphi)$ each (i.e. there is an atomic substruct S'_k of S_k such that $S'_k \in \Omega_\rho(\varphi)$)[r], and $S_1, \ldots, S_n$ and S_l and S_r should not contain any occurrence from $\Omega_\rho(\varphi)$. Moreover, an innermost rewriting strategy is enforced: only minimal reducible substructs (i.e. structs having no reducible proper substruct) can be rewritten.

$$S \otimes (S_1 \oplus \ldots \oplus S_k \oplus \ldots \oplus S_n) \rightsquigarrow_{\oplus\otimes_{DW_W}} S_1 \oplus \ldots \oplus (S \otimes S_k) \oplus \ldots \oplus S_n$$

$$(S_1 \oplus \ldots \oplus S_k \oplus \ldots \oplus S_n) \otimes S \rightsquigarrow_{\oplus\otimes_{DW_W}} S_1 \oplus \ldots \oplus S_n \oplus (S_k \otimes S) \oplus \ldots \oplus S_n$$

$$S \otimes S_r \rightsquigarrow_{\oplus\otimes_{DW_W}} S_r \qquad S_l \otimes S \rightsquigarrow_{\oplus\otimes_{DW_W}} S_l \qquad S \oplus S_r \rightsquigarrow_{\oplus\otimes_{DW_W}} S_r \qquad S_l \oplus S \rightsquigarrow_{\oplus\otimes_{DW_W}} S_l$$

[r] An atomic substruct is a formula occurrence. Therefore it makes sense to talk about pertinence of atomic substructs in $\Omega_\rho(\varphi)$, even though it might look strange at first.

$$S_l \otimes S_r \rightsquigarrow_{\oplus\otimes DW_W} S_l \quad S_l \otimes S_r \rightsquigarrow_{\oplus\otimes DW_W} S_r \quad S_l \oplus S_r \rightsquigarrow_{\oplus\otimes DW_W} S_l \quad S_l \oplus S_r \rightsquigarrow_{\oplus\otimes DW_W} S_r$$

In the struct rewriting rule below, $C[\;]$ is a struct context (i.e. $C[S]$ indicates that the struct S is a substruct of a struct $C[S]$). For the rewriting rule to be applicable, $S \equiv S_1 \oplus \ldots \oplus S_n$ must be a $\otimes$-junct in $C[S]$. Moreover, an innermost rewriting strategy is enforced: $S \equiv S_1 \oplus \ldots \oplus S_n$ can be replaced by $N_S(x_1, \ldots, x_m)$ only if S has no substruct S' that is a $\otimes$-junction of $\oplus$-junctions (if this were the case, then S' must be replaced before).

$$\begin{aligned} C[S] \quad &\equiv \quad C[S_1 \oplus \ldots \oplus S_n] \\ \rightsquigarrow_{\oplus\otimes DW_D} \quad & C[N_S(x_1, \ldots, x_m)] \oplus (N_S(x_1, \ldots, x_m) \leftrightarrow S'_1 \oplus \ldots \oplus S'_n) \end{aligned}$$

The relation $\rightsquigarrow_{\oplus\otimes DW}$ is the composition of $\rightsquigarrow^*_{\oplus\otimes DW_W}$ and $\rightsquigarrow^*_{\oplus\otimes DW_D}$ (i.e. $S \rightsquigarrow_{\oplus\otimes DW} S^*$ if and only if there exists S' such that $S \rightsquigarrow^*_{\oplus\otimes DW_W} S'$ and $S' \rightsquigarrow^*_{\oplus\otimes DW_D} S^*$).

The $\otimes$-junctions of a struct in $\rightsquigarrow_{\oplus\otimes DW}$ can be classified in the following way:

- If the $\otimes$-junction originates from a defining equation, it is called a *definitional $\otimes$-junction*.
- Otherwise:
  - If the $\otimes$-junction does not contain new defined predicate symbols, it is called *pure*.
  - Otherwise, it is called *mixed*.

Example 4.36 ($\oplus\otimes_{DW}$-Normalization)**.** *Let φ be the proof below:*

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad B \vdash B}{A, B \vdash A \wedge B}\wedge_r^1}{A \wedge B \vdash A \wedge B}\wedge_l \qquad \dfrac{\dfrac{B \vdash B \qquad A \vdash A}{A, B \vdash B \wedge A}\wedge_r^2}{A \wedge B \vdash B \wedge A}\wedge_l}{A \wedge B \vdash B \wedge A}cut^3 \qquad \dfrac{C \vdash C \qquad C \vdash C}{C \vdash C}cut^4}{(A \wedge B) \vee C \vdash B \wedge A, C}\vee_l^5$$

Its cut-pertinent struct is:

$$\mathcal{S}_\varphi \equiv ((A \oplus^1 B) \oplus^3 (\neg B \otimes^2 \neg A)) \otimes^5 (C \oplus^4 \neg C)$$

Considering that $\{A, B, C\} \subset \Omega_{\vee_l^5}(\varphi)$ and $\{A, B, C\} \cap \Omega_{\vee_l^5}(\varphi) = \emptyset$, the struct can be normalized in the way shown below:

$$
\begin{array}{rcl}
\mathcal{S}_\varphi & \equiv & ((A \oplus^1 B) \oplus^3 (\neg B \otimes^2 \neg A)) \otimes^5 (C \oplus^4 \neg C) \\
& \rightsquigarrow_{\oplus\otimes DW_W} & ((A \oplus^1 B) \otimes^5 (C \oplus^4 \neg C)) \oplus^3 (\neg B \otimes^2 \neg A) \\
& \rightsquigarrow_{\oplus\otimes DW_W} & ((((A \oplus^1 B) \otimes^5 C) \oplus^4 \neg C)) \oplus^3 (\neg B \otimes^2 \neg A) \\
& \rightsquigarrow_{\oplus\otimes DW_D} & (((D_{A\oplus B} \otimes^5 C) \oplus^4 \neg C)) \oplus^3 (\neg B \otimes^2 \neg A) \oplus (D_{A\oplus B} \leftrightarrow (A \oplus^1 B)) \\
& \equiv & (D_{A\oplus B} \otimes^5 C) \oplus^4 \neg C \oplus^3 (\neg B \otimes^2 \neg A) \oplus \\
& & (\neg D_{A\oplus B} \otimes A) \oplus (\neg D_{A\oplus B} \otimes B) \oplus (D_{A\oplus B} \otimes (\neg A \otimes \neg B))
\end{array}
$$

$(\neg D_{A\oplus B} \otimes A)$, $(\neg D_{A\oplus B} \otimes B)$ and $(D_{A\oplus B} \otimes (\neg A \otimes \neg B))$ are definitional $\otimes$-junctions. $\neg C$ and $(\neg B \otimes^2 \neg A)$ are pure $\otimes$-junctions. And, finally, $(D_{A\oplus B} \otimes^5 C)$ is a mixed $\otimes$-junction.

Definition 4.6.2 (Cut-pertinent Swapped Definitional Clause Set)**.** A *cut-pertinent definitional clause set* of a proof φ is:

$$C^D_{\varphi|S'} \doteq \mathrm{cl}(S)$$

where S' is a $\rightsquigarrow_{\oplus\otimes DW_W}$-normal-form of $\mathcal{S}_\varphi$ and $S' \rightsquigarrow^*_{\oplus\otimes DW_D} S$.

The clauses corresponding to definitional $\otimes$-junctions are called *definitional clauses*. The clauses corresponding to pure $\otimes$-junctions are called *pure clauses*. The clauses corresponding to mixed $\otimes$-junctions are called *mixed clauses*.

If the $\rightsquigarrow_{\oplus\otimes DW_W}$-normal-form S' is unique or clear from the context, then it can be omitted. The swapped definitional clause set is then denoted simply as C^D_φ.

Example 4.37 (Swapped Definitional Clause Set)**.** *Let φ be the proof in Example 4.25. Then one of its swapped definitional clause sets is:*

$$
C^D_\varphi \equiv \left\{
\begin{array}{l}
\vdash D_{A\oplus B}, C \quad ; \\
C \vdash \quad ; \\
B, A \vdash \quad ; \\
D_{A\oplus B} \vdash A \quad ; \\
D_{A\oplus B} \vdash B \quad ; \\
A, B \vdash D_{A\oplus B}
\end{array}
\right\}
$$

The clauses $D_{A\oplus B} \vdash A$, $D_{A\oplus B} \vdash B$ and $A, B \vdash D_{A\oplus B}$ are definitional clauses. $C \vdash$ and $B, \neg A \vdash$ are pure clauses. And $\vdash D_{A\oplus B}, C$ is a mixed clause.

4.6.2 Projections

While construction of swapped clause sets is reasonably straightforward, the construction of projections for some of the clauses presents some difficulties. As in the case of definitional clause sets, some clauses of swapped definitional clause sets are merely definitional, and hence corresponding definitional D-projections can be easily constructed. Other clauses are pure in the sense that they do not contain any defined predicate symbol, and hence O-Projections can be constructed for such clauses. However, there are mixed clauses for which none of the previously defined notions of projection work, because these clauses contain a mix of defined and undefined predicate symbols.

DW-Projections

The new notion of projection required by mixed clauses is called mixed DW-Projection (Definition 4.6.7) and it is essentially a combination of O-projection and proper D-projection. It requires the auxiliary concepts of encapsulated formula occurrences (Definition 4.6.3) and encapsulated inferences (Definition 4.6.4). Roughly, constructing a mixed DW-projection is initially similar to constructing an O-projection, taking care to include encapsulated formula occurrences in the slice. Later cut-pertinent inferences are replaced by $\wedge_r$ and d_r inferences, similarly to what is done during the construction of proper D-projections, in order to re-encapsulate the encapsulated formula occurrences into the defined predicate symbol.

Definition 4.6.3 (Encapsulated Formula Occurrences)**.** Let S be a struct and S' be a substruct of S. Let $N_{S'}$ be the defined predicate for S'. Then, the *encapsulated occurrences* of $N_{S'}$ are all the atomic occurrences of S'.

Example 4.38 (Encapsulated Formula Occurrences)**.** *The encapsulated formula occurrences of the defined predicate $D_{A \oplus B}$ of the $\rightsquigarrow_{\oplus\otimes_{DW}}$-normal-form of the struct $\mathcal{S}_\varphi$ shown in Example 4.36 are: A and B.*

Definition 4.6.4 (Encapsulated Inferences)**.** Let S be a cut-pertinent struct of a proof φ and S' be a substruct of S. Let $N_{S'}$ be the defined predicate for S'. Then, every inference ρ of φ which corresponds to a connective $\oplus_\rho$ or $\otimes_\rho$ in S' or that is an axiom inference having a formula occurrence of S' in its conclusion sequent is an *encapsulated inference* of $N_{S'}$.

Example 4.39 (Encapsulated Inferences)**.** *The encapsulated inferences of the defined predicate $D_{A \oplus B}$ of the $\rightsquigarrow_{\oplus\otimes_{DW}}$-normal-form of the struct $\mathcal{S}_\varphi$ shown in*

Example 4.36 are: $\wedge_r^1$ *and the axiom inferences having* $A \vdash A$ *and* $B \vdash B$ *as conclusion sequents.*

Definition 4.6.5 (Definitional DW-Projection)**.** Let φ be a proof and c a definitional clause in $C^D_{\varphi|S}$. Then the *definitional DW-projection* of φ with respect to c is constructed in the same way as a definitional D-projection and thus simply defined as:

$$\lfloor \varphi \rfloor_c^{DW_D} \doteq \lfloor \varphi \rfloor_c^{D_D}$$

Definition 4.6.6 (Pure DW-Projection)**.** Let φ be a proof and c a pure clause in $C^D_{\varphi|S}$. Then the *pure DW-projection* of φ with respect to c is constructed in the same way as a O-projection and thus simply defined as:

$$\lfloor \varphi \rfloor_c^{DW_D} \doteq \lfloor \varphi \rfloor_c^{O}$$

Definition 4.6.7 (Mixed DW-Projection)**.** Let φ be a proof and c a mixed clause in $C^D_{\varphi|S}$. Let Ω_E and Υ_E be the sets of, respectively, encapsulated formula occurrences and encapsulated inferences of defined predicates occurring in c. Let Ω_c be the set of undefined formula occurrences in c. Then the *mixed DW-projection* of φ with respect to c can be computed according to the following steps:

1. Construct $\varphi^1 \doteq \wr\varphi\int_{\{\Omega_E \cup \Omega_c\}}$.
2. Replace the inferences of Υ_E in φ^1 by $\neg_r$, $\wedge_r$, $\vee_r$ and d_r (analogously to what is done in the construction of proper D-projections). Let φ^2 be the resulting proofoid.
3. Construct $\varphi^3 \doteq Y_{\oplus}^{\Omega_{CP}(\varphi^2)}(\varphi^2)$ by replacing the cut-pertinent inferences of φ^2 by Y-inferences.
4. Construct $\varphi^4 \doteq W_{\text{fix}}(\varphi^3)$ by fixing broken inferences with weakening.
5. Finally, construct the mixed DW-projection $\lfloor \varphi \rfloor_c^{DW_M} \doteq [\![\varphi^4]\!]$ by eliminating the Y-inferences from φ^4.

Example 4.40 (Mixed DW-Projection)**.** *Let φ be the proof shown in Example 4.36, which is displayed again for convenience below:*

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \quad B \vdash B}{A, B \vdash A \wedge B}\wedge_r^1}{A \wedge B \vdash A \wedge B}\wedge_l \quad \dfrac{\dfrac{B \vdash B \quad A \vdash A}{A, B \vdash B \wedge A}\wedge_r^2}{A \wedge B \vdash B \wedge A}\wedge_l}{A \wedge B \vdash B \wedge A}cut^3 \quad \dfrac{C \vdash C \quad C \vdash C}{C \vdash C}cut^4}{(A \wedge B) \vee C \vdash B \wedge A, C}\vee_l^5$$

The first step in the construction of the mixed DW-projection $\lfloor\varphi\rfloor^{DW_M}_{\vdash D_{A\oplus B}, C}$ is the slicing with respect to $\Omega_E \cup \Omega_c$ where $\Omega_E = \{A, B\}$ and $\Omega_c = \{C\}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad B \vdash B}{A, B \vdash A \wedge B}\wedge_r^1}{A \wedge B \vdash A \wedge B}\wedge_l \qquad \dfrac{\dfrac{\vdash \qquad \vdash}{\vdash}Y}{\vdash}Y}{A \wedge B \vdash A \wedge B}Y \qquad \dfrac{C \vdash C \qquad \vdash}{C \vdash C}Y}{(A \wedge B) \vee C \vdash A \wedge B, \quad C}\vee_l^5$$

The second step is the introduction of definition inferences, resulting in the proofoid φ^2 below:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad B \vdash B}{A, B \vdash A \wedge B}\wedge_r}{A, B \vdash D_{A\oplus B}}d_r}{A \wedge B \vdash D_{A\oplus B}}\wedge_l \qquad \dfrac{\dfrac{\vdash \qquad \vdash}{\vdash}Y}{\vdash}Y}{A \wedge B \vdash D_{A\oplus B}}Y \qquad \dfrac{C \vdash C \qquad \vdash}{C \vdash C}Y}{(A \wedge B) \vee C \vdash D_{A\oplus B}, \quad C}\vee_l^5$$

Subsequently, cut-pertinent inferences of φ^2 should be replaced by Y-inferences. However, since φ^2 has no cuts, there is nothing to be replaced, and hence $\varphi^3 = \varphi^2$. Subsequently, broken inferences of φ^3 should be W-fixed. However, there are no broken inferences in φ^3. Therefore, only the last step of eliminating Y-inferences remains and its result is the mixed DW-projection $\lfloor\varphi\rfloor^{DW_M}_{\vdash D_{A\oplus B}, C}$ shown below:

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad B \vdash B}{A, B \vdash A \wedge B}\wedge_r}{A, B \vdash D_{A\oplus B}}d_r}{A \wedge B \vdash D_{A\oplus B}}\wedge_l \qquad C \vdash C}{(A \wedge B) \vee C \vdash D_{A\oplus B}, C}\vee_l^5$$

4.7 CERes-Normal-Forms

The refutation of a clause set and the projections of a (skolemization of a) proof with arbitrary cuts can be combined to produce a new proof whose cuts are atomic and thus inessential. This combination procedure is described in detail in Definition 4.7.1.

Definition 4.7.1 (CERes-Normal-Form)**.**
The CERes-*normal-form* CERes(φ, δ) of a proof φ with respect to a refu-

tation δ of a cut-pertinent clause set $C_{\varphi'}$ of the skolemization[s] φ' of φ is constructed according to the following steps:

1. $\hat{\delta}$ is obtained from δ by replacing each initial inference by the corresponding projection.

2. Let $\sigma_1, \ldots, \sigma_n$ be the unifiers of each resolution inference in δ and let $\sigma \doteq \sigma_1 \ldots \sigma_n$. Then $\tilde{\varphi}'$ is obtained from $\hat{\delta}\sigma$ by replacing all resolution inferences by atomic cuts and all factoring inferences by contraction inferences.

3. In general, the end-sequent of $\tilde{\varphi}'$ can have a different number of copies (even zero) of each formula of the end-sequent of φ'. Therefore, φ'' is constructed by appending weakening and contraction inferences to the end of $\tilde{\varphi}'$, so that φ'' has the same end-sequent as φ'.

4. Finally, CERes(φ, δ) is constructed by deskolemizing φ'' so that the end-sequent of CERes(φ, δ) is equal to the end-sequent of φ.

Moreover, subscripts and superscripts are used to explicitly indicate the kind of clause set and the kind of projections that have been used:

- $\text{CERes}_S^S(\varphi, \delta)$ uses the standard clause set and S-projections.
- $\text{CERes}_S^O(\varphi, \delta)$ uses the standard clause set and O-projections.
- $\text{CERes}_W^S(\varphi, \delta)$ uses a swapped clause set and S-projections.
- $\text{CERes}_W^O(\varphi, \delta)$ uses a swapped clause set and O-projections.
- $\text{CERes}_P^S(\varphi, \delta)$ uses the profile clause set and S-projections.
- $\text{CERes}_P^O(\varphi, \delta)$ uses the profile clause set and O-projections.
- $\text{CERes}_D^D(\varphi, \delta)$ uses the definitional clause set and D-projections.
- $\text{CERes}_{DW}^{DW}(\varphi, \delta)$ uses the swapped definitional clause set and DW-projections.

[s] In general, skolemization and deskolemization of proofs are not unique, since there are various alternative algorithms [8]. Nevertheless, for simplicity, it is assumed here that a fixed deterministic algorithm is used, with the respect to which every proof has a unique skolemization and a unique deskolemization. Consequently, the CERes-normal-form of a proof φ is indeed determined by the proof φ itself and by a refutation δ of its clause set.

Example 4.41 (CERes^S_S-Normal-Form)**.** *Let φ be the proof shown in Example 4.1. Its standard cut-pertinent clause set, as shown in Example 4.3 is:*

$$C^S_\varphi \equiv \{\vdash A, C \;;\; C \vdash A \;;\; \vdash B, C \;;\; C \vdash B \;;\; B, A \vdash C \;;\; B, A, C \vdash\}$$

Let δ be the following refutation of C^S_φ:

$$
\begin{prooftree}
\AxiomC{$\vdash A, C$}
\AxiomC{$C \vdash A$}
\RightLabel{r}
\BinaryInfC{$\vdash A, A$}
\RightLabel{f_r}
\UnaryInfC{$\vdash A$}
\AxiomC{$\vdash B, C$}
\AxiomC{$C \vdash B$}
\RightLabel{r}
\BinaryInfC{$\vdash B, B$}
\RightLabel{f_r}
\UnaryInfC{$\vdash B$}
\AxiomC{$B, A \vdash C$}
\AxiomC{$C, B, A \vdash$}
\RightLabel{r}
\BinaryInfC{$B, A, B, A \vdash$}
\RightLabel{f_r}
\UnaryInfC{$B, A, B \vdash$}
\RightLabel{f_r}
\UnaryInfC{$A, B \vdash$}
\RightLabel{r}
\BinaryInfC{$A \vdash$}
\RightLabel{r}
\BinaryInfC{$\vdash$}
\end{prooftree}
$$

Consider the S-projections $\lfloor\varphi\rfloor^S_{\vdash A,C}$, $\lfloor\varphi\rfloor^S_{\vdash B,C}$, $\lfloor\varphi\rfloor^S_{C\vdash A}$, $\lfloor\varphi\rfloor^S_{C\vdash B}$, $\lfloor\varphi\rfloor^S_{B,A\vdash C}$ and $\lfloor\varphi\rfloor^S_{B,A,C\vdash}$ shown in Example 4.5. Then, the result of replacing all initial inferences of δ by the corresponding projections and replacing resolution inferences by atomic cuts and factoring inferences by contraction inferences is $\tilde{\varphi}$ shown below:

$$
\begin{prooftree}
\AxiomC{$\lfloor\varphi\rfloor^S_{\vdash A,C}$}
\noLine
\UnaryInfC{$\Gamma \vdash \Delta, A, C$}
\AxiomC{$\lfloor\varphi\rfloor^S_{C\vdash A}$}
\noLine
\UnaryInfC{$C, \Gamma \vdash \Delta, A$}
\RightLabel{cut}
\BinaryInfC{$\Gamma, \Gamma \vdash \Delta, \Delta, A, A$}
\RightLabel{c_r}
\UnaryInfC{$\Gamma, \Gamma \vdash \Delta, \Delta, A$}
\AxiomC{$\lfloor\varphi\rfloor^S_{\vdash B,C}$}
\noLine
\UnaryInfC{$\Gamma \vdash \Delta, B, C$}
\AxiomC{$\lfloor\varphi\rfloor^S_{C\vdash B}$}
\noLine
\UnaryInfC{$C, \Gamma \vdash \Delta, B$}
\RightLabel{cut}
\BinaryInfC{$\Gamma, \Gamma \vdash \Delta, \Delta, B, B$}
\RightLabel{c_r}
\UnaryInfC{$\Gamma, \Gamma \vdash \Delta, \Delta, B$}
\AxiomC{$\lfloor\varphi\rfloor^S_{B,A\vdash C}$}
\noLine
\UnaryInfC{$B, A, \Gamma \vdash \Delta, C$}
\AxiomC{$\lfloor\varphi\rfloor^S_{B,A,C\vdash}$}
\noLine
\UnaryInfC{$C, B, A, \Gamma \vdash \Delta$}
\RightLabel{cut}
\BinaryInfC{$B, A, B, A, \Gamma, \Gamma \vdash \Delta, \Delta$}
\RightLabel{c_r}
\UnaryInfC{$B, A, B, \Gamma, \Gamma \vdash \Delta, \Delta$}
\RightLabel{c_r}
\UnaryInfC{$A, B, \Gamma, \Gamma \vdash \Delta, \Delta$}
\RightLabel{cut}
\BinaryInfC{$A, \Gamma, \Gamma, \Gamma, \Gamma \vdash \Delta, \Delta, \Delta, \Delta$}
\RightLabel{cut}
\BinaryInfC{$\Gamma, \Gamma, \Gamma, \Gamma, \Gamma, \Gamma \vdash \Delta, \Delta, \Delta, \Delta, \Delta, \Delta$}
\end{prooftree}
$$

Finally, CERes$^S_S(\varphi, \delta)$ *shown below is obtained by adding contraction inferences to the bottom of the proof:*

$$
\begin{prooftree}
\AxiomC{$\lfloor\varphi\rfloor^S_{\vdash A,C}$}
\noLine
\UnaryInfC{$\Gamma \vdash \Delta, A, C$}
\AxiomC{$\lfloor\varphi\rfloor^S_{C\vdash A}$}
\noLine
\UnaryInfC{$C, \Gamma \vdash \Delta, A$}
\RightLabel{cut}
\BinaryInfC{$\Gamma, \Gamma \vdash \Delta, \Delta, A, A$}
\RightLabel{c_r}
\UnaryInfC{$\Gamma, \Gamma \vdash \Delta, \Delta, A$}
\AxiomC{$\lfloor\varphi\rfloor^S_{\vdash B,C}$}
\noLine
\UnaryInfC{$\Gamma \vdash \Delta, B, C$}
\AxiomC{$\lfloor\varphi\rfloor^S_{C\vdash B}$}
\noLine
\UnaryInfC{$C, \Gamma \vdash \Delta, B$}
\RightLabel{cut}
\BinaryInfC{$\Gamma, \Gamma \vdash \Delta, \Delta, B, B$}
\RightLabel{c_r}
\UnaryInfC{$\Gamma, \Gamma \vdash \Delta, \Delta, B$}
\AxiomC{$\lfloor\varphi\rfloor^S_{B,A\vdash C}$}
\noLine
\UnaryInfC{$B, A, \Gamma \vdash \Delta, C$}
\AxiomC{$\lfloor\varphi\rfloor^S_{B,A,C\vdash}$}
\noLine
\UnaryInfC{$C, B, A, \Gamma \vdash \Delta$}
\RightLabel{cut}
\BinaryInfC{$B, A, B, A, \Gamma, \Gamma \vdash \Delta, \Delta$}
\RightLabel{c_r}
\UnaryInfC{$B, A, B, \Gamma, \Gamma \vdash \Delta, \Delta$}
\RightLabel{c_r}
\UnaryInfC{$A, B, \Gamma, \Gamma \vdash \Delta, \Delta$}
\RightLabel{cut}
\BinaryInfC{$A, \Gamma, \Gamma, \Gamma, \Gamma \vdash \Delta, \Delta, \Delta, \Delta$}
\RightLabel{cut}
\BinaryInfC{$\Gamma, \Gamma, \Gamma, \Gamma, \Gamma, \Gamma \vdash \Delta, \Delta, \Delta, \Delta, \Delta, \Delta$}
\doubleLine
\RightLabel{c^*}
\UnaryInfC{$\Gamma \vdash \Delta$}
\end{prooftree}
$$

Example 4.42 (CERes^O_S-Normal-Form)**.** *Consider again the proof φ shown in Example 4.1 and its standard cut-pertinent clause set, shown in Example 4.3. Let δ be the refutation shown in Example 4.41.*

Consider the O-projections $\lfloor\varphi\rfloor^O_{\vdash A,C}$, $\lfloor\varphi\rfloor^O_{\vdash B,C}$, $\lfloor\varphi\rfloor^O_{C\vdash A}$, $\lfloor\varphi\rfloor^O_{C\vdash B}$, $\lfloor\varphi\rfloor^O_{B,A\vdash C}$ and $\lfloor\varphi\rfloor^O_{B,A,C\vdash}$ shown in Example 4.15. Then, the result of replacing all initial inferences of δ by the corresponding projections is $\hat{\delta}$ shown below:

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{\lfloor\varphi\rfloor^O_{\vdash A,C}}{(A\wedge B)\vee C\vdash A,C}
\quad
\dfrac{\lfloor\varphi\rfloor^O_{C\vdash A}}{(A\wedge B)\vee C,C\vdash A,C}
}{(A\wedge B)\vee C,(A\wedge B)\vee C\vdash A,A,C}\,r
}{(A\wedge B)\vee C,(A\wedge B)\vee C\vdash A,C}\,f_r
\quad
\dfrac{
\dfrac{
\dfrac{
\dfrac{\lfloor\varphi\rfloor^O_{\vdash B,C}}{(A\wedge B)\vee C\vdash B,C}
\quad
\dfrac{\lfloor\varphi\rfloor^O_{C\vdash B}}{(A\wedge B)\vee C,C\vdash B,C}
}{(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B,B,C}\,r
}{(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B,C}\,f_r
\quad
\dfrac{
\dfrac{
\dfrac{
\dfrac{\lfloor\varphi\rfloor^O_{B,A\vdash C}}{(A\wedge B)\vee C,A,B\vdash B\wedge A,C}
\quad
\dfrac{\lfloor\varphi\rfloor^O_{B,A,C\vdash}}{C,(A\wedge B)\vee C,A,B\vdash B\wedge A,C}
}{A,B,A,B,(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B\wedge A,B\wedge A,C}\,r
}{A,B,B,(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B\wedge A,B\wedge A,C}\,f_r
}{A,B,(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B\wedge A,B\wedge A,C}\,f_r
}{A,(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B\wedge A,B\wedge A,C}\,r
}{(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B\wedge A,B\wedge A,C,C}\,r
$$

Then $\tilde{\varphi}$ shown below is the result of replacing resolution inferences by atomic cuts and factoring inferences by contraction inferences.

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{\lfloor\varphi\rfloor^O_{\vdash A,C}}{(A\wedge B)\vee C\vdash A,C}
\quad
\dfrac{\lfloor\varphi\rfloor^O_{C\vdash A}}{(A\wedge B)\vee C,C\vdash A,C}
}{(A\wedge B)\vee C,(A\wedge B)\vee C\vdash A,A,C}\,cut
}{(A\wedge B)\vee C,(A\wedge B)\vee C\vdash A,C}\,c_r
\quad
\dfrac{
\dfrac{
\dfrac{
\dfrac{\lfloor\varphi\rfloor^O_{\vdash B,C}}{(A\wedge B)\vee C\vdash B,C}
\quad
\dfrac{\lfloor\varphi\rfloor^O_{C\vdash B}}{(A\wedge B)\vee C,C\vdash B,C}
}{(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B,B,C}\,cut
}{(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B,C}\,c_r
\quad
\dfrac{
\dfrac{
\dfrac{
\dfrac{\lfloor\varphi\rfloor^O_{B,A\vdash C}}{(A\wedge B)\vee C,A,B\vdash B\wedge A,C}
\quad
\dfrac{\lfloor\varphi\rfloor^O_{B,A,C\vdash}}{C,(A\wedge B)\vee C,A,B\vdash B\wedge A,C}
}{A,B,A,B,(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B\wedge A,B\wedge A,C}\,cut
}{A,B,B,(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B\wedge A,B\wedge A,C}\,c_r
}{A,B,(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B\wedge A,B\wedge A,C}\,c_r
}{A,(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B\wedge A,B\wedge A,C}\,cut
}{(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B\wedge A,B\wedge A,C,C}\,cut
$$

Finally $\mathrm{CERes}^O_S(\varphi,\delta)$ *shown below is obtained by adding contraction inferences to the bottom of the proof:*

$$
\begin{prooftree}
\AxiomC{$\lfloor\varphi\rfloor^O_{\vdash A,C}$}
\noLine
\UnaryInfC{$(A\wedge B)\vee C\vdash A,C$}
\AxiomC{$\lfloor\varphi\rfloor^O_{C\vdash A}$}
\noLine
\UnaryInfC{$(A\wedge B)\vee C, C\vdash A,C$}
\RightLabel{cut}
\BinaryInfC{$(A\wedge B)\vee C,(A\wedge B)\vee C\vdash A,A,C$}
\RightLabel{c_r}
\UnaryInfC{$(A\wedge B)\vee C,(A\wedge B)\vee C\vdash A,C$}
\AxiomC{$\lfloor\varphi\rfloor^O_{\vdash B,C}$}
\noLine
\UnaryInfC{$(A\wedge B)\vee C\vdash B,C$}
\AxiomC{$\lfloor\varphi\rfloor^O_{C\vdash B}$}
\noLine
\UnaryInfC{$(A\wedge B)\vee C, C\vdash B,C$}
\RightLabel{cut}
\BinaryInfC{$(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B,B,C$}
\RightLabel{c_r}
\UnaryInfC{$(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B,C$}
\AxiomC{$\lfloor\varphi\rfloor^O_{B,A\vdash C}$}
\noLine
\UnaryInfC{$(A\wedge B)\vee C, A, B\vdash B\wedge A, C$}
\AxiomC{$\lfloor\varphi\rfloor^O_{B,A,C\vdash}$}
\noLine
\UnaryInfC{$C,(A\wedge B)\vee C, A, B\vdash B\wedge A, C$}
\RightLabel{cut}
\BinaryInfC{$A,B,A,B,(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B\wedge A,B\wedge A,C$}
\RightLabel{c_r}
\UnaryInfC{$A,B,B,(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B\wedge A,B\wedge A,C$}
\RightLabel{c_r}
\UnaryInfC{$A,B,(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B\wedge A,B\wedge A,C$}
\RightLabel{cut}
\BinaryInfC{$A,(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B\wedge A,B\wedge A,C$}
\RightLabel{cut}
\BinaryInfC{$(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C,(A\wedge B)\vee C\vdash B\wedge A,B\wedge A,C,C$}
\doubleLine
\RightLabel{c^*}
\UnaryInfC{$(A\wedge B)\vee C\vdash B\wedge A,C$}
\end{prooftree}
$$

Comparing $\mathrm{CERes}^O_S(\varphi,\delta)$ *shown above with* $\mathrm{CERes}^S_S(\varphi,\delta)$ *shown in Example 4.41, it is clear that the use of O-projections results in smaller proofs, mainly because there are fewer redundant weakening and contraction inferences.*

Example 4.43 (CERes^O_W-Normal-Form)**.** *Consider now the swapped clause set of the proof φ of Example 4.1:*

$$C^W_\varphi = \{\ \vdash A, C\ ;\ \vdash B, C\ ;\ C \vdash\ ;\ B, A \vdash\ \}$$

A refutation δ of C^W_φ is shown below:

$$\dfrac{\dfrac{\dfrac{\vdash A, C \qquad \dfrac{\vdash B, C \qquad B, A \vdash}{A \vdash C}\,r}{\vdash C, C}\,r}{\vdash C}\,f_r \qquad C \vdash}{\vdash}\,r$$

Consider the O-projections $\lfloor\varphi\rfloor^O_{\vdash A,C}$, $\lfloor\varphi\rfloor^O_{\vdash B,C}$, $\lfloor\varphi\rfloor^O_{C\vdash}$, $\lfloor\varphi\rfloor^O_{B,A\vdash}$ shown in Example 4.27. Then, the result of replacing all initial inferences of δ by the corresponding projections is $\hat{\delta}$ shown below:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{A, B \vdash A}\,w_l}{A \wedge B \vdash A}\,\wedge_l \qquad C \vdash C}{(A \wedge B) \vee C \vdash A, C}\,\vee_l \qquad \dfrac{\dfrac{\dfrac{\dfrac{B \vdash B}{A, B \vdash B}\,w_l}{A \wedge B \vdash B}\,\wedge_l \qquad C \vdash C}{(A \wedge B) \vee C \vdash B, C}\,\vee_l \qquad \dfrac{B \vdash B \qquad A \vdash A}{A, B \vdash B \wedge A}\,\wedge_r}{(A \wedge B) \vee C, A \vdash B \wedge A, C}\,r}{(A \wedge B) \vee C, (A \wedge B) \vee C \vdash B \wedge A, C, C}\,r}{(A \wedge B) \vee C, (A \wedge B) \vee C \vdash B \wedge A, C}\,f_r \qquad C \vdash C}{(A \wedge B) \vee C, (A \wedge B) \vee C \vdash B \wedge A, C}\,r$$

Then $\tilde{\varphi}$ shown below is the result of replacing resolution inferences by atomic cuts and factoring inferences by contraction inferences.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{A, B \vdash A}\,w_l}{A \wedge B \vdash A}\,\wedge_l \qquad C \vdash C}{(A \wedge B) \vee C \vdash A, C}\,\vee_l \qquad \dfrac{\dfrac{\dfrac{\dfrac{B \vdash B}{A, B \vdash B}\,w_l}{A \wedge B \vdash B}\,\wedge_l \qquad C \vdash C}{(A \wedge B) \vee C \vdash B, C}\,\vee_l \qquad \dfrac{B \vdash B \qquad A \vdash A}{A, B \vdash B \wedge A}\,\wedge_r}{(A \wedge B) \vee C, A \vdash B \wedge A, C}\,cut}{(A \wedge B) \vee C, (A \wedge B) \vee C \vdash B \wedge A, C, C}\,cut}{(A \wedge B) \vee C, (A \wedge B) \vee C \vdash B \wedge A, C}\,c_r \qquad C \vdash C}{(A \wedge B) \vee C, (A \wedge B) \vee C \vdash B \wedge A, C}\,cut$$

Finally CERes$^O_W(\varphi, \delta)$ shown below is obtained by adding contraction inferences to the bottom of the proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{A, B \vdash A}\,w_l}{A \wedge B \vdash A}\,\wedge_l \qquad C \vdash C}{(A \wedge B) \vee C \vdash A, C}\,\vee_l \qquad \dfrac{\dfrac{\dfrac{\dfrac{B \vdash B}{A, B \vdash B}\,w_l}{A \wedge B \vdash B}\,\wedge_l \qquad C \vdash C}{(A \wedge B) \vee C \vdash B, C}\,\vee_l \qquad \dfrac{B \vdash B \qquad A \vdash A}{A, B \vdash B \wedge A}\,\wedge_r}{(A \wedge B) \vee C, A \vdash B \wedge A, C}\,cut}{(A \wedge B) \vee C, (A \wedge B) \vee C \vdash B \wedge A, C, C}\,cut}{(A \wedge B) \vee C, (A \wedge B) \vee C \vdash B \wedge A, C}\,c_r \qquad C \vdash C}{(A \wedge B) \vee C, (A \wedge B) \vee C \vdash B \wedge A, C}\,cut}{(A \wedge B) \vee C \vdash B \wedge A, C}\,c_l$$

Comparing $\mathrm{CERes}_W^O(\varphi, \delta)$ *shown above with* $\mathrm{CERes}_S^S(\varphi, \delta)$ *shown in Example 4.41 and* $\mathrm{CERes}_S^O(\varphi, \delta)$*, it is clear that the use of swapped clause sets and O-projections results in smaller proofs.*

Example 4.44 (CERes_P^O-Normal-Form)**.** *For the proof* φ *of Example 4.1,* $C_\varphi^P = C_\varphi^W$ *and* $\mathrm{CERes}_P^O(\varphi, \delta) = \mathrm{CERes}_W^O(\varphi, \delta)$*, since there are no degenerate inferences.*

Example 4.45 (CERes_D^D-Normal-Form)**.** *Consider the definitional clause set* C_φ^D *of the proof* φ *shown in Example 4.33:*

$D \vdash C$	$D, C \vdash$	$C \vdash D, C$
$E, B, A \vdash$	$\vdash E, B$	$\vdash E, A$
$F \vdash A$	$F \vdash B$	$A, B \vdash F$
$G \vdash F$	$G \vdash E$	$E, F \vdash G$
$H \vdash G, D$	$G \vdash H$	$D \vdash H$
$\vdash H$		

The shortest refutation δ *of* C^{D}_{φ} *is shown below:*

$$
\begin{prooftree}
\AxiomC{$\vdash H$}
\AxiomC{$H \vdash G, D$}
\AxiomC{$G \vdash F$}
\AxiomC{$G \vdash E$}
\AxiomC{$F \vdash A$}
\AxiomC{$F \vdash B$}
\AxiomC{$E, B, A \vdash$}
\RightLabel{r}
\BinaryInfC{$F, E, A \vdash$}
\RightLabel{r}
\BinaryInfC{$F, F, E \vdash$}
\RightLabel{f_l}
\UnaryInfC{$F, E \vdash$}
\RightLabel{r}
\BinaryInfC{$G, F \vdash$}
\RightLabel{r}
\BinaryInfC{$G, G \vdash$}
\RightLabel{f_l}
\UnaryInfC{$G \vdash$}
\RightLabel{r}
\BinaryInfC{$H \vdash D$}
\AxiomC{$D \vdash C$}
\AxiomC{$D, C \vdash$}
\RightLabel{r}
\BinaryInfC{$D, D \vdash$}
\RightLabel{f_l}
\UnaryInfC{$D \vdash$}
\RightLabel{r}
\BinaryInfC{$H \vdash$}
\RightLabel{r}
\BinaryInfC{$\vdash$}
\end{prooftree}
$$

By using the proper D-projection shown in Example 4.35 and definitional D-projections shown in Example 4.34, $\mathrm{CERes}^{D}_{D}(\varphi, \delta)$ *is:*

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\AxiomC{$B \vdash B$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A, B \vdash A \wedge B$}
\RightLabel{d_r}
\UnaryInfC{$A, B \vdash F$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash F$}
\AxiomC{$B \vdash B$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash \neg B, B$}
\AxiomC{$A \vdash A$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash \neg A, A$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$\vdash \neg B, \neg A, B \wedge A$}
\RightLabel{$\vee_r$}
\UnaryInfC{$\vdash \neg B \vee \neg A, B \wedge A$}
\RightLabel{d_r}
\UnaryInfC{$\vdash E, B \wedge A$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A \wedge B \vdash F \wedge E, B \wedge A$}
\RightLabel{d_r}
\UnaryInfC{$A \wedge B \vdash G, B \wedge A$}
\AxiomC{$C \vdash C$}
\AxiomC{$C \vdash C$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash \neg C, C$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$C \vdash C \wedge \neg C, C$}
\RightLabel{d_r}
\UnaryInfC{$C \vdash D, C$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(A \wedge B) \vee C \vdash G, D, B \wedge A, C$}
\RightLabel{$\vee_r$}
\UnaryInfC{$(A \wedge B) \vee C \vdash G \vee D, B \wedge A, C$}
\RightLabel{d_r}
\UnaryInfC{$(A \wedge B) \vee C \vdash H, B \wedge A, C$}
\AxiomC{$\lfloor \varphi \rfloor^{D_D}_{H \vdash G, D}$}
\AxiomC{$\lfloor \varphi \rfloor^{D_D}_{G \vdash F}$}
\AxiomC{$\lfloor \varphi \rfloor^{D_D}_{G \vdash E}$}
\AxiomC{$\lfloor \varphi \rfloor^{D_D}_{F \vdash A}$}
\AxiomC{$\lfloor \varphi \rfloor^{D_D}_{F \vdash B}$}
\AxiomC{$\lfloor \varphi \rfloor^{D_D}_{E, B, A \vdash}$}
\RightLabel{cut}
\BinaryInfC{$F, E, A \vdash$}
\RightLabel{cut}
\BinaryInfC{$F, F, E \vdash$}
\RightLabel{c_l}
\UnaryInfC{$F, E \vdash$}
\RightLabel{cut}
\BinaryInfC{$G, F \vdash$}
\RightLabel{cut}
\BinaryInfC{$G, G \vdash$}
\RightLabel{c_l}
\UnaryInfC{$G \vdash$}
\RightLabel{cut}
\BinaryInfC{$H \vdash D$}
\AxiomC{$\lfloor \varphi \rfloor^{D_D}_{D \vdash C}$}
\AxiomC{$\lfloor \varphi \rfloor^{D_D}_{D, C \vdash}$}
\RightLabel{cut}
\BinaryInfC{$D, D \vdash$}
\RightLabel{c_l}
\UnaryInfC{$D \vdash$}
\RightLabel{cut}
\BinaryInfC{$H \vdash$}
\RightLabel{cut}
\BinaryInfC{$(A \wedge B) \vee C \vdash B \wedge A, C$}
\end{prooftree}
$$

Example 4.46 (CERes^{DW}_{DW}-Normal-Form)**.** *Consider again the swapped definitional clause set of the proof φ shown in Example 4.37:*

$$C^D_\varphi \equiv \{\vdash D_{A\oplus B}, C \;\; ; \;\; C \vdash \;\; ; \;\; B, A \vdash \;\; ; \;\; D_{A\oplus B} \vdash A \;\; ; \;\; D_{A\oplus B} \vdash B \;\; ; \;\; A, B \vdash D_{A\oplus B}\}$$

The shortest refutation δ of C^D_φ is shown below:

$$
\begin{prooftree}
\AxiomC{$\vdash D_{A\oplus B}, C$}
\AxiomC{$C \vdash$}
\RightLabel{r}
\BinaryInfC{$\vdash D_{A\oplus B}$}
\AxiomC{$D_{A\oplus B} \vdash A$}
\AxiomC{$D_{A\oplus B} \vdash B$}
\AxiomC{$B, A \vdash$}
\RightLabel{r}
\BinaryInfC{$D_{A\oplus B}, A \vdash$}
\RightLabel{r}
\BinaryInfC{$D_{A\oplus B}, D_{A\oplus B} \vdash$}
\RightLabel{f_l}
\UnaryInfC{$D_{A\oplus B} \vdash$}
\RightLabel{r}
\BinaryInfC{$\vdash$}
\end{prooftree}
$$

By using the mixed DW-projection shown in Example 4.40, pure DW-projections shown in Example 4.15 and definitional DW-projections shown in Example 4.34, $\mathrm{CERes}^{DW}_{DW}(\varphi, \delta)$ is:

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\AxiomC{$B \vdash B$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$A, B \vdash A \wedge B$}
\RightLabel{d_r}
\UnaryInfC{$A, B \vdash D_{A\oplus B}$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash D_{A\oplus B}$}
\AxiomC{$C \vdash C$}
\RightLabel{$\vee_l^5$}
\BinaryInfC{$(A \wedge B) \vee C \vdash D_{A\oplus B}, C$}
\AxiomC{$C \vdash C$}
\RightLabel{cut}
\BinaryInfC{$(A \wedge B) \vee C \vdash D_{A\oplus B}, C$}
\AxiomC{$A \vdash A$}
\RightLabel{w_l}
\UnaryInfC{$A, B \vdash A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash A$}
\RightLabel{d_l}
\UnaryInfC{$D_{A\oplus B} \vdash A$}
\AxiomC{$B \vdash B$}
\RightLabel{w_l}
\UnaryInfC{$A, B \vdash B$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash B$}
\RightLabel{d_l}
\UnaryInfC{$D_{A\oplus B} \vdash B$}
\AxiomC{$B \vdash B$}
\AxiomC{$A \vdash A$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$B, A \vdash B \wedge A$}
\RightLabel{cut}
\BinaryInfC{$D_{A\oplus B}, A \vdash B \wedge A$}
\RightLabel{cut}
\BinaryInfC{$D_{A\oplus B}, D_{A\oplus B} \vdash B \wedge A$}
\RightLabel{c_l}
\UnaryInfC{$D_{A\oplus B} \vdash B \wedge A$}
\RightLabel{cut}
\BinaryInfC{$(A \wedge B) \vee C \vdash B \wedge A, C$}
\end{prooftree}
$$

4.8 CERes Ignoring Atomic Cuts

If CERes is applied to a proof containing only atomic cuts, CERes still transforms the proof into a new proof containing only atomic cuts, but with additional structural inferences and with the atomic cuts located in the bottom of the proof. This is clearly non-ideal, because the proof could be simply left unchanged. More generally, if CERes is applied to a proof containing complex cuts and atomic cuts, CERes unnecessarily includes the atomic cuts in the process of reduction, even though atomic cuts cannot be reduced further. The inclusion of atomic cuts results in larger clause sets that are more costly to refute, and in normal forms with possibly additional structural inferences. This indicates that there is a very simple and evident improvement of the CERes method that has been thoroughly overlooked so far: instead of distinguishing between cut-pertinent and cut-impertinent formula occurrences (i.e. between ancestors and non-ancestors of *all* cut formula occurrences) and cut-pertinent and cut-impertinent inferences (i.e inferences that operate on the ancestors and on the non-ancestors of cut formula occurrences), it suffices to distinguish between ancestors of *complex* cut formula occurrences and ancestors of either occurrences in the end-sequent or of atomic cut-formula occurrences.

Definition 4.8.1 (Complex-Cut-Pertinent and Complex-Cut-Impertinent Occurrences)**.** A formula occurrence is *complex-cut-pertinent* if and only if it is an ancestor of a non-atomic cut formula occurrence. The *set of complex-cut-pertinent formula occurrences* of a proof φ is denoted $\Omega_{CCP}(\varphi)$.

A formula occurrence is *complex-cut-impertinent* if and only if it is not complex-cut-pertinent. The *set of complex-cut-impertinent formula occurrences* of a proof φ is denoted $\Omega_{CCI}(\varphi)$.

Definition 4.8.2 (Complex-Cut-Pertinence)**.** An inference ρ is *complex-cut-pertinent* if and only if ρ is $\Omega_{CCP}(\varphi)$-pertinent.

An inference ρ is *complex-cut-impertinent* if and only if ρ is $\Omega_{CCI}(\varphi)$-pertinent.

Definition 4.8.3 (CERes-Normal-Form Ignoring Atomic Cuts)**.**
The CERes-*normal-form ignoring atomic cuts* CCERes(φ, δ) of a proof φ is obtained in the same way as CERes(φ, δ) except that, in all manipulations and constructions of structs, clause sets and projections, $\Omega_{CCP}(\varphi)$ is used instead of $\Omega_{CP}(\varphi)$, $\Omega_{CCI}(\varphi)$ is used instead of $\Omega_{CI}(\varphi)$ and complex-cut-pertinence of inferences is used instead of cut-pertinence of inferences.

Example 4.47 (CERes-Normal-Form Ignoring Atomic Cuts)**.** *Let φ be the proof below:*

$$
\dfrac{
\dfrac{
\dfrac{\dfrac{A \vdash A \qquad B \vdash B}{A, B \vdash A \wedge B}\,\wedge_r^1}{A \wedge B \vdash A \wedge B}\,\wedge_l
\qquad
\dfrac{\dfrac{B \vdash B \qquad A \vdash A}{A, B \vdash B \wedge A}\,\wedge_r^2}{A \wedge B \vdash B \wedge A}\,\wedge_l
}{A \wedge B \vdash B \wedge A}\,cut^3
\qquad
\dfrac{C \vdash C \qquad C \vdash C}{C \vdash C}\,cut^4
}{(A \wedge B) \vee C \vdash B \wedge A, C}\,\vee_l^5
$$

Its complex-cut-pertinent struct is shown below. It is interesting to note that cut^4 now corresponds to a $\otimes$ connective, because cut^4 is complex-cut-impertinent.

$$\mathcal{S}_\varphi^C \equiv ((A \oplus^1 B) \oplus^3 (\neg B \otimes^2 \neg A)) \otimes^5 (\epsilon_\otimes \otimes^4 \epsilon_\otimes)$$

The struct can be $\rightsquigarrow_{\oplus\otimes W}$-normalized to:

$$S \equiv (A \otimes^5 \epsilon_\otimes \otimes^4 \epsilon_\otimes) \oplus^1 (B \otimes^5 \epsilon_\otimes \otimes^4 \epsilon_\otimes) \oplus^3 (\neg B \otimes^2 \neg A)$$

And the corresponding clause set is:

$$C_\varphi \equiv \{\ \vdash A\ \ ;\ \ \vdash B\ \ ;\ \ B, A \vdash\ \}$$

It can be refuted by the refutation δ shown below:

$$
\dfrac{\vdash A \qquad \dfrac{\vdash B \qquad B, A \vdash}{A \vdash}\,r}{\vdash}\,r
$$

The O-projection $\lfloor \varphi \rfloor^O_{\vdash A}$ is shown below. Interestingly, projections can now contain atomic cuts because they are complex-cut-impertinent inferences.

$$
\dfrac{
\dfrac{\dfrac{A \vdash A}{A, B \vdash A}\,w_l}{A \wedge B \vdash A}\,\wedge_l
\qquad
\dfrac{C \vdash C \qquad C \vdash C}{C \vdash C}\,cut^4
}{(A \wedge B) \vee C \vdash A, C}\,\vee_l^5
$$

Analogously, the O-projection $\lfloor \varphi \rfloor^O_{\vdash B}$ is:

$$
\dfrac{
\dfrac{\dfrac{B \vdash B}{A, B \vdash B}\,w_l}{A \wedge B \vdash B}\,\wedge_l
\qquad
\dfrac{C \vdash C \qquad C \vdash C}{C \vdash C}\,cut^4
}{(A \wedge B) \vee C \vdash B, C}\,\vee_l^5
$$

And the O-projection $\lfloor \varphi \rfloor^O_{B, A \vdash}$ is:

$$
\dfrac{B \vdash B \qquad A \vdash A}{A, B \vdash B \wedge A}\,\wedge_r^2
$$

Combining the refutation and the projections as usual, $\mathrm{CCERes}_W^O(\varphi, \delta)$ *is obtained:*

$$
\begin{prooftree}
\AxiomC{$A \vdash A$}
\RightLabel{w_l}
\UnaryInfC{$A, B \vdash A$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash A$}
\AxiomC{$C \vdash C$}
\AxiomC{$C \vdash C$}
\RightLabel{cut^4}
\BinaryInfC{$C \vdash C$}
\RightLabel{$\vee_l^5$}
\BinaryInfC{$(A \wedge B) \vee C \vdash A, C$}
\AxiomC{$B \vdash B$}
\RightLabel{w_l}
\UnaryInfC{$A, B \vdash B$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$A \wedge B \vdash B$}
\AxiomC{$C \vdash C$}
\AxiomC{$C \vdash C$}
\RightLabel{cut^4}
\BinaryInfC{$C \vdash C$}
\RightLabel{$\vee_l^5$}
\BinaryInfC{$(A \wedge B) \vee C \vdash B, C$}
\AxiomC{$B \vdash B$}
\AxiomC{$A \vdash A$}
\RightLabel{$\wedge_r^2$}
\BinaryInfC{$A, B \vdash B \wedge A$}
\RightLabel{cut}
\BinaryInfC{$(A \wedge B) \vee C, A \vdash B \wedge A, C$}
\RightLabel{cut}
\BinaryInfC{$(A \wedge B) \vee C, (A \wedge B) \vee C \vdash B \wedge A, C, C$}
\doubleLine
\RightLabel{c^*}
\UnaryInfC{$(A \wedge B) \vee C \vdash B \wedge A, C$}
\end{prooftree}
$$

4.9 CERes Ignoring Quantifier-Free Cuts

In fact, for some applications, such as Herbrand sequent extraction [94, 95], it suffices to eliminate only cuts that have quantifiers[t] in their cut formulas.

Definition 4.9.1 (Quantified-Cut-Pertinent and Quantified-Cut-Impertinent Occurrences)**.** A formula occurrence is *quantified-cut-pertinent* if and only if it is an ancestor of a cut formula occurrence that contains quantifiers. The *set of quantified-cut-pertinent formula occurrences* of a proof φ is denoted $\Omega_{QCP}(\varphi)$.

A formula occurrence is *quantified-cut-impertinent* if and only if it is not quantified-cut-pertinent. The *set of quantified-cut-impertinent formula occurrences* of a proof φ is denoted $\Omega_{QCI}(\varphi)$.

Definition 4.9.2 (Quantified-Cut-Pertinence)**.** An inference ρ is *quantified-cut-pertinent* if and only if ρ is $\Omega_{QCP}(\varphi)$-pertinent.

An inference ρ is *quantified-cut-impertinent* if and only if ρ is $\Omega_{QCI}(\varphi)$-pertinent.

Definition 4.9.3 (CERes-Normal-Form Ignoring Quantifier-Free Cuts)**.** The CERes-*normal-form ignoring quantifier-free cuts* QCERes(φ, δ) of a proof φ is obtained in the same way as CERes(φ, δ) except that, in all manipulations and constructions of structs, clause sets and projections, $\Omega_{QCP}(\varphi)$ is used instead of $\Omega_{CP}(\varphi)$, $\Omega_{QCI}(\varphi)$ is used instead of $\Omega_{CI}(\varphi)$ and quantified-cut-pertinence of inferences is used instead of cut-pertinence of inferences.

[t]In fact, even if the cut formula occurrences of a cut ρ in a proof φ do contain quantifiers, if these quantifiers are dummy in the sense that they were introduced by weakening inferences instead of being properly introduced by quantifier inferences, then ρ could also be considered "quantifier-free" and therefore be ignored. Nevertheless, for simplicity, this additional improvement is not considered in detail here.

Chapter 5

Comparison of Cut-Elimination Methods

The purpose of this chapter is to compare cut-elimination by resolution, as described in Chapter 4, and reductive cut-elimination, as described in Chapter 3, with respect to the normalized proofs that they produce.

A naive comparison that would check for exact syntactic equality of the normalized proofs would not be convenient, since occasional syntactical differences might occur not because of essential differences in the proofs but actually because of the bureaucracy required by sequent calculi. Indeed, such a naive comparison would tell nothing beyond the quite non-informative fact that cut-elimination by resolution is different from reductive cut-elimination by resolution, simply because the normalized proofs produced by the former always have atomic cuts in the bottom, while the normalized proofs produced by the latter usually have atomic cuts on the top.

What is needed is a comparison based on a proof equivalence relation that disregards bureaucratic matters such as the position of atomic cuts. To this aim, the notion of *canonic refutation* (Definition 5.1.4) of a normalized proof is used. Roughly speaking, two proofs are then considered *CR-equivalent* (Definition 5.1.6) when they have the same canonic refutations.

Comparisons based on CR-equivalence are much more informative. In particular, it is possible to show (Theorem 5.4) that cut-elimination by resolution *CR-simulates*[a] (Definition 5.2.2) reductive methods, i.e. for

[a] Essentially, the notion of CR-simulation allows the precise formalization of informal claims

any normalized proof φ' produced from φ by a reductive method, cut-elimination by resolution is capable of producing a proof φ^* from φ such that φ^* is CR-equivalent to φ'. On the other hand, reductive methods do not CR-simulate cut-elimination by resolution (Theorem 5.6).

5.1 Canonic Refutations

Informally and rather sketchy, the canonic refutation from a proof φ with atomic cuts consists simply of removing all the cut-impertinent formula occurrences and inferences of φ and transforming the remaining cut inferences into resolution inferences. Even more informally, the intuitive idea is that a canonic refutation from a proof is the "resolution skeleton" that lies within a proof whose cuts are atomic, if these cuts are seen as resolution inferences.

However, the concept of canonic refutation is formally not so simple, because there are many ways in which the cut-impertinent inferences could be removed. In order to avoid this complication, canonic refutations are defined only for proofs that are also in $\gg_{\oplus\otimes}$-normal-form (Definition 5.1.4), in which case the removal can be made precise and simple by using auxiliary Y inferences and a method to eliminate them (Definition 5.1.2). For proofs φ that are not in $\gg_{\oplus\otimes}$-normal-form, a set of canonic refutations can be defined as the set having the canonic refutations of all $\gg_{\oplus\otimes}$-normal-forms of φ (Definition 5.1.5).

As shown in Theorem 5.1, the canonic refutation from a proof φ is indeed a refutation of the swapped clause set of φ. This is not just a coincidence, but a fundamental goal that had to be achieved when conceiving the notion of canonic refutation, since it is essential to the proof that cut-elimination by resolution CR-simulates reductive cut-elimination.

Definition 5.1.1 ($\gg_{\oplus\otimes}$)**.** $\gg_{\oplus\otimes}$ is the sub-relation of $\gg$ with the additional restriction that n-ary (for $n > 1$) cut-pertinent inferences cannot be swapped above m-ary (for $m > 1$) cut-impertinent inferences. Moreover, in order to ensure that $\gg_{\oplus\otimes}$ is weakly-normalizing, any swapping is forbidden if the proof is already in a form such that no n-ary (for $n > 1$) cut-pertinent inference occurs above a m-ary (for $m > 1$) cut-impertinent inference.

Definition 5.1.2 (Merging Y-Elimination)**.** The merging-elimination of Y inferences follows the proof rewriting rules shown below:

---

that cut-elimination by resolution "subsumes" or is "more general than" reductive methods.

$$\dfrac{\Gamma_1 \vdash \Delta_1 \qquad \ldots \qquad \Gamma_n \vdash \Delta_n}{\Gamma_1, \ldots, \Gamma_n \vdash \Delta_1 \ldots, \Delta_n}\, Y$$

$$\Downarrow$$

$$\Gamma_1, \ldots, \Gamma_n \vdash \Delta_1 \ldots, \Delta_n$$

$$\dfrac{\dfrac{\varphi}{\Gamma \vdash \Delta}}{\Gamma \vdash \Delta}\, Y$$

$$\Downarrow$$

$$\dfrac{\varphi}{\Gamma \vdash \Delta}$$

Definition 5.1.3 (*cut*-*r*-Replacement)**.** $r^r_{cut}(\varphi)$ denotes the result of replacing all atomic cuts in φ by resolution inferences.

Definition 5.1.4 (Canonic Refutation)**.** Let φ be a proof in $\rhd_{\bar{a}}$-normal-form and in $\gg_{\oplus\otimes}$-normal-form. Then the *canonic refutation*[b] from φ is:

$$CR(\varphi) \doteq r^r_{cut}([\![Y_{\otimes}^{\Omega_{CP}(\varphi)}(\varphi)]\!]_M)$$

Example 5.1 (Canonic Refutation)**.** *Let φ be the proof below with the cut-pertinent occurrences and inferences highlighted in red:*

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{A, C \vdash A}\, w_l}{A \wedge C \vdash A}\, \wedge_l \qquad \dfrac{A \vdash A \qquad \dfrac{B \vdash B}{\vdash B, \neg B}\, \neg_r}{A \vdash B, A \wedge \neg B}\, \wedge_r}{A \wedge C \vdash B, A \wedge \neg B}\, cut \qquad B \vdash B}{A \wedge C \vdash B, A \wedge \neg B}\, cut$$

[b] This definition of canonic refutation for swapped clause sets is inspired by the definition of canonic refutation for characteristic clause sets given in [12]. However, while in [12] canonic refutations are defined for any proof containing atomic cuts only (i.e. any proof in $\rhd_{\bar{a}}$-normal-form), here proofs are also required to be in $\gg_{\oplus\otimes}$-normal-form. This is convenient, because it allows a very simple and direct extraction of canonic refutations from proofs. In [12], where this additional requirement is not made, canonic refutations are constructed inductively. While it would be in principle possible to construct canonic refutations for swapped clause sets in a similar inductive fashion and without the restriction to $\gg_{\oplus\otimes}$-normal-form, the description of the construction would be much more complex than in the case for characteristic clause sets, for two reasons. Firstly, the inductive case for binary cut-impertinent inferences would have to take their dependencies into account when merging refutations; and secondly, if a proof is not in $\gg_{\oplus\otimes}$-normal-form, its swapped clause set might not even be unique, due to degenerate inferences, and hence, such degenerate inferences would also have to be handled appropriately during an inductive construction of canonic refutations for swapped clause sets of proofs with atomic cuts in general.

φ is in $\rhd_{\tilde{a}}$-normal-form and in $\gg_{\oplus\otimes}$-normal-form. Therefore it is possible to extract a canonic refutation from it. Firstly, cut-impertinent inferences are replaced by Y-inferences and cut-impertinent formula occurrences are removed:

$$
\begin{prooftree}
\AxiomC{$\vdash A$}
\RightLabel{Y}\UnaryInfC{$\vdash A$}
\RightLabel{Y}\UnaryInfC{$\vdash A$}
\AxiomC{$A \vdash$}
\AxiomC{$\vdash B$}
\RightLabel{Y}\UnaryInfC{$\vdash B$}
\RightLabel{Y}\BinaryInfC{$A \vdash B$}
\RightLabel{cut}\BinaryInfC{$\vdash B$}
\AxiomC{$B \vdash$}
\RightLabel{cut}\BinaryInfC{$\vdash$}
\end{prooftree}
$$

Then the Y-inferences are eliminated according to the proof rewriting rules of Definition 5.1.2:

$$
\begin{prooftree}
\AxiomC{$\vdash A$}
\AxiomC{$A \vdash B$}
\RightLabel{cut}\BinaryInfC{$\vdash B$}
\AxiomC{$B \vdash$}
\RightLabel{cut}\BinaryInfC{$\vdash$}
\end{prooftree}
$$

Finally, $CR(\varphi)$ is obtained by replacing cuts by resolution inferences:

$$
\begin{prooftree}
\AxiomC{$\vdash A$}
\AxiomC{$A \vdash B$}
\RightLabel{r}\BinaryInfC{$\vdash B$}
\AxiomC{$B \vdash$}
\RightLabel{r}\BinaryInfC{$\vdash$}
\end{prooftree}
$$

Theorem 5.1 (Correctness of Canonic Refutations)**.** $CR(\varphi)$ is a refutation of C^W_φ.

Proof. The uniqueness of the swapped clause set C^W_φ is guaranteed by the fact that φ is in $\gg_{\oplus\otimes}$-normal-form.

As φ is in $\rhd_{\tilde{a}}$-normal-form, the only cut-pertinent inferences are cuts (cut-pertinent weakenings and contractions do not occur, because φ is in $\gg_{\oplus\otimes}$-normal-form), and all other inferences are replaced by Y-inferences in $Y_\otimes^{\Omega_{CP}(\varphi)}(\varphi)$. So, $Y_\otimes^{\Omega_{CP}(\varphi)}(\varphi)$ contains only atomic cuts and Y-inferences.

Since φ is in $\gg_{\oplus\otimes}$-normal-form, all the Y inferences with arity greater than one occur above cuts. Hence, merging Y-elimination can be applied and all Y inferences can be eliminated by the proof rewriting rules of Definition 5.1.2. Then the only remaining inferences are atomic cuts, and these are replaced by resolution inferences. Thus $CR(\varphi)$ is a resolution deduction.

$CR(\varphi)$ is a resolution deduction of the empty clause (i.e. a refutation), simply because the end-sequent of φ contains no cut-pertinent formula occurrences. And $CR(\varphi)$ is a refutation of C^W_φ, because, since $\mathcal{S}_\varphi$ is in $\rightsquigarrow_{\oplus\otimes_W}$-normal form, the initial clauses of $[\![Y_\otimes^{\Omega_{CP}(\varphi)}(\varphi)]\!]_M$ are in C^W_φ. □

Definition 5.1.5 (Set of Canonic Refutations)**.** Let φ be a proof in $\rhd_{\tilde{a}}$-normal-form. Then the *set of canonic refutations* of φ is:

$$S_{CR}(\varphi) \doteq \{\ CR(\psi) \mid \psi \text{ is a } \gg_{\oplus\otimes}\text{-normal-form of } \varphi\ \}$$

Remark 5.1.1. Note that, if φ has no degenerate inferences, the cardinality of $S_{CR}(\varphi)$ is equal to one. Moreover, the cardinality of $S_{CR}(\varphi)$ is equal to the number of swapped clause sets of φ.

Example 5.2 (Set of Canonic Refutations)**.** *Let φ be the proof below:*

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A \vdash A}\,cut}{A \vdash A, C}\,w_r \qquad \dfrac{\dfrac{B \vdash B \qquad B \vdash B}{B \vdash B}\,cut}{B \vdash B, D}\,w_r}{A, B \vdash A, B, C \wedge D}\,\wedge_r}{}$$

Even though φ is in $\rhd_{\tilde{a}}$-normal-form, it is not in $\gg_{\oplus\otimes}$-normal-form. Hence a canonic refutation cannot be directly extracted from φ.

However φ has two $\gg_{\oplus\otimes}$-normal-forms. One of them is the proof φ_1 shown below:

$$\dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A \vdash A}\,cut}{A, B \vdash A, B, C \wedge D}\,w^*$$

The other one is the proof φ_2 shown below:

$$\dfrac{\dfrac{B \vdash B \qquad B \vdash B}{B \vdash B}\,cut}{A, B \vdash A, B, C \wedge D}\,w^*$$

Canonic refutations can be extracted from φ_1 and φ_2. $CR(\varphi_1)$ is:

$$\dfrac{A \vdash A \qquad A \vdash A}{\vdash}\,r$$

And $CR(\varphi_2)$ is:

$$\dfrac{B \vdash B \qquad B \vdash B}{\vdash}\,cut$$

Then, by the definition of set of canonic refutations:

$$S_{CR}(\varphi) = \{CR(\varphi_1), CR(\varphi_2)\}$$

Definition 5.1.6 (CR-Equivalence)**.**
Two proofs φ_1 and φ_2 in $\rhd_{\tilde{a}}$-normal-form are *strongly CR-equivalent*, denoted $\varphi_1 =_{sCR} \varphi_2$, if and only if $S_{CR}(\varphi_1) = S_{CR}(\varphi_2)$. They are *weakly CR-equivalent*, denoted $\varphi_1 =_{wCR} \varphi_2$, if and only if there exists δ such that $\delta \in S_{CR}(\varphi_1)$ and $\delta \in S_{CR}(\varphi_2)$.

5.2 CR-Simulation

CR-simulation (Definition 5.2.2) allows the comparison of different cut-elimination methods based on whether the normalized proofs produced by them are CR-equivalent.

Definition 5.2.1 ($\rhd_{\tilde{a}}$-Normalization Method)**.**
A relation $\blacktriangleright$ is a $\rhd_{\tilde{a}}$*-normalization method* if and only if $\varphi \blacktriangleright \psi$ implies that ψ is in $\rhd_{\tilde{a}}$-normal-form.

Example 5.3 ($\rhd_{\tilde{a}}$-Normalization Method)**.** *Clearly,* $\rhd^{\downarrow}_{\tilde{a}}$ *is a* $\rhd_{\tilde{a}}$*-normalization method. Moreover,* $\rhd^{\downarrow}_{\bar{a}}$ *and* CERes *(with any kinds of clause set and projections) are also* $\rhd_{\tilde{a}}$*-normalization methods.*

Definition 5.2.2 (CR-Simulation)**.** Let $\blacktriangleright_1$ and $\blacktriangleright_2$ be two $\rhd_{\tilde{a}}$-normalization methods. $\blacktriangleright_1$ *CR-simulates* $\blacktriangleright_2$ if and only if, for any proofs φ and φ_2 with $\varphi \blacktriangleright_2 \varphi_2$, there exists φ_1 such that $\varphi \blacktriangleright_1 \varphi_1$ and $\varphi_1 =_{wCR} \varphi_2$.

5.2.1 CR-Simulation between Two Reductive Methods

Theorems 5.2 and 5.3 show simple CR-simulation results for two slightly different reductive cut-elimination methods. $\rhd^{\downarrow}_{\tilde{a}}$ (Definition 3.2.21) and $\rhd^{\downarrow}_{\bar{a}}$ (Definition 3.2.22) differ essentially only in the fact that $\rhd^{\downarrow}_{\bar{a}}$ is more eager to shift atomic cuts upward; their normal forms differ essentially only in the position of atomic cuts. Hence it is natural to assume that these methods are equivalent in an informal sense. The fact that they CR-simulate each other supports the claim that the formal notion of CR-simulation captures the informal idea of equivalence of cut-elimination methods.

Theorem 5.2. $\rhd^{\downarrow}_{\tilde{a}}$ CR-simulates $\rhd^{\downarrow}_{\bar{a}}$.

Proof. Assume $\varphi \rhd^{\downarrow}_{\bar{a}} \varphi_2$. Then, by definition of $\rhd^{\downarrow}_{\bar{a}}$, $\varphi \rhd^{*}_{\bar{a}} \varphi_2$. It is easy to see that there exists φ_1 such that $\varphi \rhd^{\downarrow}_{\tilde{a}} \varphi_1$ and $\varphi_1 (\rhd_c \cup \rhd_w \cup \rhd_r)^{\downarrow} \varphi_2$ (this is because, for any rewriting sequence $\varphi \rhd^{*}_{\bar{a}} \varphi_2$, a rewriting sequence $\varphi \rhd^{*}_{\tilde{a}} \varphi_1 (\rhd_c \cup \rhd_w \cup \rhd_r)^{*} \varphi_2$ can be obtained by postponing the rewritings that shift atomic cuts upward and possibly adding a few more rank reduction rewritings to shift more complex cuts above the atomic cuts that have been left in place). Let $\delta \in S_{CR}(\varphi_2)$. Then, by definition of set of canonic refutations, there is a proof ψ such that $CR(\psi) = \delta$ and ψ is a $\gg_{\oplus\otimes}$-normal-form of φ_2. Moreover, it must be the case that $\delta \in S_{CR}(\varphi_1)$, because ψ is a $\gg_{\oplus\otimes}$-normal-form of φ_1. Therefore, $\varphi_1 =_{wCR} \varphi_2$ and thus $\rhd^{\downarrow}_{\tilde{a}}$ CR-simulates $\rhd^{\downarrow}_{\bar{a}}$. □

Theorem 5.3. $\rhd^{\downarrow}_{\bar{a}}$ CR-simulates $\rhd^{\downarrow}_{\tilde{a}}$.

Proof. Assume $\varphi \rhd^{\downarrow}_{\tilde{a}} \varphi_2$. Then, by definition of $\rhd^{\downarrow}_{\tilde{a}}$, $\varphi \rhd^{*}_{\tilde{a}} \varphi_2$. Since $\rhd_{\tilde{a}}$ is a sub-relation of $\rhd_{\bar{a}}$, it is also the case that $\varphi \rhd^{*}_{\bar{a}} \varphi_2$. Let $\delta \in S_{CR}(\varphi_2)$. Then, by definition of set of canonic refutations, there is a proof ψ such that $CR(\psi) = \delta$ and ψ is a $\gg_{\oplus\otimes}$-normal-form of φ_2. Consider a particular $\gg_{\oplus\otimes}$ rewriting sequence from φ_2 to ψ passing through a proof φ_1, i.e. such that $\varphi_2 \gg^{*}_{\oplus\otimes} \varphi_1$ and $\varphi_1 \gg^{\downarrow}_{\oplus\otimes} \psi$, with the restriction that $\varphi_2 \gg^{*}_{\oplus\otimes} \varphi_1$ only by downward shifting of cut-pertinent contractions and weakenings. Then it is also the case that $\varphi_2(\rhd_c \cup \rhd_w \cup \rhd_r)^{\downarrow}\varphi_1$, and hence $\varphi \rhd^{\downarrow}_{\bar{a}} \varphi_1$. Since $\varphi_1 \gg^{\downarrow}_{\oplus\otimes} \psi$, it is the case that $\delta \in S_{CR}(\varphi_1)$. Therefore, $\varphi_1 =_{wCR} \varphi_2$) and thus $\rhd^{\downarrow}_{\bar{a}}$ CR-simulates $\rhd^{\downarrow}_{\tilde{a}}$. □

5.2.2 CR-Simulation by CERes

This Subsection is devoted to proving that cut-elimination by resolution CR-simulates reductive cut-elimination. The proof is based on several technical lemmas[c] collectively stating that, when a reductive cut-

[c] Similar invariance lemmas for standard clause sets can already be found in [12]. These results were improved in [66] with the invention of profile clause sets. Namely, profile clause sets are invariant under $\rhd_r$, while standard clause sets are not. The invariance lemmas presented in this section differ from the ones in the cited works mainly in two ways:

- The lemmas are stated for structs, and not for clause sets, as in [66], or for "characteristic clause terms", as in [12]. The reason for this, is that, as shown in Chapter 4, structs can be transformed into different kinds of clause sets. Therefore, proving invariance lemmas for structs is a way of proving invariance lemmas for all kinds of clause sets at once. Moreover, since structs have a good correspondence to the actual structure of proofs, as evidenced by Example 4.2 and Theorem 4.8, they allow a cleaner presentation of the invariance results.
- However, for certain cut-reductions, the struct itself does not remain invariant. Nevertheless, some of its normal-forms do remain invariant. This is the case, for example, of invariance under $\rhd_r$. The $\rightsquigarrow_{\oplus\otimes_P}$-normal-forms and the $\rightsquigarrow_{\oplus\otimes_W}$-normal-forms remain invariant. Since the invariance for $\rightsquigarrow_{\oplus\otimes_P}$-normal-forms correspond to the invariance for profile clause sets already shown in [66], only the invariance for $\rightsquigarrow_{\oplus\otimes_W}$-normal-forms is shown here. This invariance implies that swapped clause sets are also invariant under $\rhd_r$.
- Lemma 5.3 shows that $\rightsquigarrow_{\oplus\otimes_W}$-normal-forms of structs (and hence also swapped clause sets) are invariant under $\rhd_w$. Profile clause sets do not enjoy this invariance. This is

elimination step is performed on a proof, its struct and clause sets either do not change or change only in inessential ways.

Lemma 5.1 (Invariance of the Struct under $\rhd_p$).

$$\psi[\varphi] \rhd_p \psi[\varphi'] \qquad \text{implies} \qquad \mathcal{S}_{\psi[\varphi]} = \mathcal{S}_{\psi[\varphi']}$$

Proof. The case analysis below shows that the cut-pertinent struct does not change (modulo commutativity and associativity of $\oplus$) when the subproof φ of ψ is rewritten according to any of the four propositional cut reduction rules, $\rhd_{p_\wedge}$, $\rhd_{p_\vee}$, $\rhd_{p_\rightarrow}$ and $\rhd_{p_\neg}$: $\mathcal{S}_\varphi = \mathcal{S}_{\varphi'}$, and hence $\mathcal{S}_{\psi[\varphi]} = \mathcal{S}_{\psi[\varphi']}$.

- $\rhd_{p_\wedge}$: As shown below, $\mathcal{S}_\varphi = (\mathcal{S}_{\varphi_1} \oplus \mathcal{S}_{\varphi_2}) \oplus \mathcal{S}_{\varphi_r}$ and $\mathcal{S}_{\varphi'} = \mathcal{S}_{\varphi_2} \oplus (\mathcal{S}_{\varphi_1} \oplus \mathcal{S}_{\varphi_r})$. Therefore, by associativity and commutativity of $\oplus$, $\mathcal{S}_\varphi = \mathcal{S}_{\varphi'}$.

$$\dfrac{\dfrac{\dfrac{\mathcal{S}_{\varphi_1}}{\varphi_1}\atop{\Gamma_1 \vdash \Delta_1, B} \quad \dfrac{\mathcal{S}_{\varphi_2}}{\varphi_2}\atop{\Gamma_2 \vdash \Delta_2, C}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, B \wedge C}\wedge_r \quad \dfrac{\dfrac{\mathcal{S}_{\varphi_r}}{\varphi_r}\atop{B, C, \Pi \vdash \Lambda}}{B \wedge C, \Pi \vdash \Lambda}\wedge_l}{\underset{(\mathcal{S}_{\varphi_1} \oplus \mathcal{S}_{\varphi_2}) \oplus \mathcal{S}_{\varphi_r}}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda}}cut$$

$$\Downarrow$$

$$\dfrac{\dfrac{\mathcal{S}_{\varphi_2}}{\varphi_2}\atop{\Gamma_2 \vdash \Delta_2, C} \quad \dfrac{\dfrac{\mathcal{S}_{\varphi_1}}{\varphi_1}\atop{\Gamma_1 \vdash \Delta_1, B} \quad \dfrac{\mathcal{S}_{\varphi_r}}{\varphi_r}\atop{B, C, \Pi \vdash \Lambda}}{C, \Gamma_1, \Pi \vdash \Delta_1, \Lambda}cut}{\underset{\mathcal{S}_{\varphi_2} \oplus (\mathcal{S}_{\varphi_1} \oplus \mathcal{S}_{\varphi_r})}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda}}cut$$

---

another evidence of the superiority of swapped clause sets with respect to the presence of weakening in proofs.

- $\rhd_{p_\vee}$: As shown below, $\mathcal{S}_\varphi = \mathcal{S}_{\varphi_l} \oplus (\mathcal{S}_{\varphi_1} \oplus \mathcal{S}_{\varphi_2})$ and $\mathcal{S}_{\varphi'} = (\mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_2}) \oplus \mathcal{S}_{\varphi_1}$. Therefore, by associativity and commutativity of $\oplus$, $\mathcal{S}_\varphi = \mathcal{S}_{\varphi'}$.

$$\dfrac{\dfrac{\overset{\mathcal{S}_{\varphi_l}}{\overset{\varphi_l}{\Pi \vdash \Lambda, B, C}}}{\Pi \vdash \Lambda, B \vee C}\vee_r \qquad \dfrac{\overset{\mathcal{S}_{\varphi_1}}{\overset{\varphi_1}{B, \Gamma_1 \vdash \Delta_1}} \qquad \overset{\mathcal{S}_{\varphi_2}}{\overset{\varphi_2}{C, \Gamma_2 \vdash \Delta_2}}}{B \vee C, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\vee_l}{\underset{\mathcal{S}_{\varphi_l} \oplus (\mathcal{S}_{\varphi_1} \oplus \mathcal{S}_{\varphi_2})}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda}}\, cut$$

$$\Downarrow$$

$$\dfrac{\dfrac{\overset{\mathcal{S}_{\varphi_l}}{\overset{\varphi_l}{\Pi \vdash \Lambda, B, C}} \qquad \overset{\mathcal{S}_{\varphi_2}}{\overset{\varphi_2}{C, \Gamma_2 \vdash \Delta_2}}}{\Pi, \Gamma_2 \vdash \Delta_2, \Lambda, B}\, cut \qquad \overset{\mathcal{S}_{\varphi_1}}{\overset{\varphi_1}{B, \Gamma_1 \vdash \Delta_1}}}{\underset{(\mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_2}) \oplus \mathcal{S}_{\varphi_1}}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda}}\, cut$$

- $\rhd_{p_\rightarrow}$: As shown below, $\mathcal{S}_\varphi = \mathcal{S}_{\varphi_l} \oplus (\mathcal{S}_{\varphi_1} \oplus \mathcal{S}_{\varphi_2})$ and $\mathcal{S}_{\varphi'} = \mathcal{S}_{\varphi_1} \oplus (\mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_2})$. Therefore, by associativity and commutativity of $\oplus$, $\mathcal{S}_\varphi = \mathcal{S}_{\varphi'}$.

$$\dfrac{\dfrac{\overset{\mathcal{S}_{\varphi_l}}{\overset{\varphi_l}{B, \Pi \vdash \Lambda, C}}}{\Pi \vdash \Lambda, B \rightarrow C}\rightarrow_r \qquad \dfrac{\overset{\mathcal{S}_{\varphi_1}}{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1, B}} \qquad \overset{\mathcal{S}_{\varphi_2}}{\overset{\varphi_2}{C, \Gamma_2 \vdash \Delta_2}}}{B \rightarrow C, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\rightarrow_l}{\underset{\mathcal{S}_{\varphi_l} \oplus (\mathcal{S}_{\varphi_1} \oplus \mathcal{S}_{\varphi_2})}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda}}\, cut$$

$$\Downarrow$$

$$\dfrac{\overset{\mathcal{S}_{\varphi_1}}{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1, B}} \qquad \dfrac{\overset{\mathcal{S}_{\varphi_l}}{\overset{\varphi_l}{B, \Pi \vdash \Lambda, C}} \qquad \overset{\mathcal{S}_{\varphi_2}}{\overset{\varphi_2}{C, \Gamma_2 \vdash \Delta_2}}}{B, \Pi, \Gamma_2 \vdash \Delta_2, \Lambda}\, cut}{\underset{\mathcal{S}_{\varphi_1} \oplus (\mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_2})}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda}}\, cut$$

- $\rhd_{p_\neg}$: As shown below, $\mathcal{S}_\varphi = \mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}$ and $\mathcal{S}_{\varphi'} = \mathcal{S}_{\varphi_r} \oplus \mathcal{S}_{\varphi_l}$. Therefore, by commutativity of $\oplus$, $\mathcal{S}_\varphi = \mathcal{S}_{\varphi'}$.

$$\dfrac{\dfrac{\overset{\mathcal{S}_{\varphi_l}}{\overset{\varphi_l}{B,\Gamma \vdash \Delta}}}{\Gamma \vdash \Delta, \neg B}\,\neg_r \qquad \dfrac{\overset{\mathcal{S}_{\varphi_r}}{\overset{\varphi_r}{\Pi \vdash \Lambda, B}}}{\neg B, \Pi \vdash \Lambda}\,\neg_l}{\underset{\mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}}{\Gamma, \Pi \vdash \Delta, \Lambda}}\,cut$$

$$\Downarrow$$

$$\dfrac{\overset{\mathcal{S}_{\varphi_r}}{\overset{\varphi_r}{\Pi \vdash \Lambda, B}} \qquad \overset{\mathcal{S}_{\varphi_l}}{\overset{\varphi_l}{B, \Gamma \vdash \Delta}}}{\underset{\mathcal{S}_{\varphi_r} \oplus \mathcal{S}_{\varphi_l}}{\Gamma, \Pi \vdash \Delta, \Lambda}}\,cut$$

$\square$

Definition 5.2.3 (Instantiation of Structs)**.** Let S_1 and S_2 be structs. Then $S_1 \leq_s S_2$ if and only if there is a variable substitution σ such that $S_1\sigma = S_2$.

Lemma 5.2 (Instantiation of the Struct under $\rhd_q$)**.**

$$\psi[\varphi] \rhd_q \psi[\varphi'] \qquad \text{implies} \qquad \mathcal{S}_{\psi[\varphi]} \leq_s \mathcal{S}_{\psi[\varphi']}$$

Proof. The case analysis below shows that the cut-pertinent struct is only instantiated when the subproof φ of ψ is rewritten according to any of the two quantifier cut reduction rules, $\rhd_{q_\forall}$ and $\rhd_{q_\exists}$: $\mathcal{S}_\varphi \leq_s \mathcal{S}_{\varphi'}$, and hence $\mathcal{S}_{\psi[\varphi]} \leq_s \mathcal{S}_{\psi[\varphi']}$.

- $\rhd_{q_\forall}$: As shown below, $\mathcal{S}_\varphi = \mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}$ and $\mathcal{S}_{\varphi'} = \mathcal{S}_{\varphi_l}\{\alpha \leftarrow t\} \oplus \mathcal{S}_{\varphi_r}$. Since $\mathcal{S}_{\varphi_r}$ does not contain α, $\mathcal{S}_{\varphi'} = (\mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r})\{\alpha \leftarrow t\}$. Therefore, by the definition of $\leq_s$, $\mathcal{S}_\varphi \leq_s \mathcal{S}_{\varphi'}$.

$$\dfrac{\dfrac{\overset{\mathcal{S}_{\varphi_l}}{\overset{\varphi_l}{\Gamma \vdash \Delta, B\{x \leftarrow \alpha\}}}}{\Gamma \vdash \Delta, \forall x B}\,\forall_r \qquad \dfrac{\overset{\mathcal{S}_{\varphi_r}}{\overset{\varphi_r}{B\{x \leftarrow t\}, \Pi \vdash \Lambda}}}{\forall x B, \Pi \vdash \Lambda}\,\forall_l}{\underset{\mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}}{\Gamma, \Pi \vdash \Delta, \Lambda}}\,cut$$

$$\Downarrow$$

$$\dfrac{\begin{array}{c}\mathcal{S}_{\varphi_l}\{\alpha\leftarrow t\}\\ \varphi_l\{\alpha \leftarrow t\}\\ \Gamma \vdash \Delta, B\{x \leftarrow t\}\end{array} \qquad \begin{array}{c}\mathcal{S}_{\varphi_r}\\ \varphi_r\\ B\{x \leftarrow t\}, \Pi \vdash \Lambda\end{array}}{\begin{array}{c}\Gamma, \Pi \vdash \Delta, \Lambda\\ \mathcal{S}_{\varphi_l}\{\alpha\leftarrow t\} \oplus \mathcal{S}_{\varphi_r}\end{array}}\, cut$$

- $\rhd_{q_\exists}$: As shown below, $\mathcal{S}_\varphi = \mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}$ and $\mathcal{S}_{\varphi'} = \mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}\{\alpha \leftarrow t\}$. Since $\mathcal{S}_{\varphi_l}$ does not contain α, $\mathcal{S}_{\varphi'} = (\mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r})\{\alpha \leftarrow t\}$. Therefore, by the definition of $\leq_s$, $\mathcal{S}_\varphi \leq_s \mathcal{S}_{\varphi'}$.

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{S}_{\varphi_l}\\ \varphi_l\\ \Gamma \vdash \Delta, B\{x \leftarrow t\}\end{array}}{\Gamma \vdash \Delta, \exists x B}\,\exists_r \qquad \dfrac{\begin{array}{c}\mathcal{S}_{\varphi_r}\\ \varphi_r\\ B\{x \leftarrow \alpha\}, \Pi \vdash \Lambda\end{array}}{\exists x B, \Pi \vdash \Lambda}\,\exists_l}{\begin{array}{c}\Gamma, \Pi \vdash \Delta, \Lambda\\ \mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}\end{array}}\, cut$$

$$\Downarrow$$

$$\dfrac{\begin{array}{c}\mathcal{S}_{\varphi_l}\\ \varphi_l\\ \Gamma \vdash \Delta, B\{x \leftarrow t\}\end{array} \qquad \begin{array}{c}\mathcal{S}_{\varphi_r}\{\alpha\leftarrow t\}\\ \varphi_r\{\alpha \leftarrow t\}\\ B\{x \leftarrow t\}, \Pi \vdash \Lambda\end{array}}{\begin{array}{c}\Gamma, \Pi \vdash \Delta, \Lambda\\ \mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}\{\alpha\leftarrow t\}\end{array}}\, cut$$

$\square$

Lemma 5.3 (Normalized Invariance under $\rhd_w$).

$$\psi[\varphi] \rhd_w \psi[\varphi'] \qquad \text{implies} \qquad \mathcal{S}_{\psi[\varphi']} = S_\psi[S']$$

such that $\mathcal{S}_{\psi[\varphi]} = S_\psi[S] \rightsquigarrow_{\oplus\otimes_W} S_\psi[S']$

Proof. The case analysis below shows that even though the cut-pertinent struct changes because a substruct is deleted when the subproof φ of ψ is rewritten according to the weakening cut reduction rules $\rhd_w$, this change is inessential with respect to $\rightsquigarrow_{\oplus\otimes_W}$-normalization, because the deleted substruct would be deleted by $\rightsquigarrow_{\oplus\otimes_W}$-normalization anyway. In other words, the $\rightsquigarrow_{\oplus\otimes_W}$-normalized struct does not change[d].

- $\rhd_w$ with weakening in the left branch: As shown below, $\mathcal{S}_\varphi = \mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}$ and $\mathcal{S}_{\varphi'} = \mathcal{S}_{\varphi_l}$. Nevertheless, $\mathcal{S}_\varphi \rightsquigarrow_{\oplus\otimes_W} \mathcal{S}_{\varphi'}$, via the struct rewriting rule $S_l \oplus S_r \rightsquigarrow_{\oplus\otimes_W} S_l$.

$$\dfrac{\dfrac{\overset{\mathcal{S}_{\varphi_l}}{\overset{\varphi_l}{\Gamma \vdash \Delta}}}{\Gamma \vdash \Delta, A}\, w_r \qquad \overset{\mathcal{S}_{\varphi_r}}{\overset{\varphi_r}{A, \Pi \vdash \Lambda}}}{\underset{\mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r} \rightsquigarrow_{\oplus\otimes_W} \mathcal{S}_{\varphi_l}}{\Gamma, \Pi \vdash \Delta, \Lambda}}\, cut$$

$$\Downarrow$$

$$\dfrac{\overset{\mathcal{S}_{\varphi_l}}{\overset{\varphi_l}{\Gamma \vdash \Delta}}}{\underset{\mathcal{S}_{\varphi_l}}{\Gamma, \Pi \vdash \Delta, \Lambda}}\, w_r^*, w_l^*$$

- $\rhd_w$ with weakening in the right branch: As shown below, $\mathcal{S}_\varphi = \mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}$ and $\mathcal{S}_{\varphi'} = \mathcal{S}_{\varphi_r}$. Nevertheless, $\mathcal{S}_\varphi \rightsquigarrow_{\oplus\otimes_W} \mathcal{S}_{\varphi'}$, via the struct rewriting rule $S_l \oplus S_r \rightsquigarrow_{\oplus\otimes_W} S_r$.

$$\dfrac{\overset{\mathcal{S}_{\varphi_l}}{\overset{\varphi_l}{\Gamma \vdash \Delta, A}} \qquad \dfrac{\overset{\mathcal{S}_{\varphi_r}}{\overset{\varphi_r}{\Pi \vdash \Lambda}}}{A, \Pi \vdash \Lambda}\, w_l}{\underset{\mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r} \rightsquigarrow_{\oplus\otimes_W} \mathcal{S}_{\varphi_r}}{\Gamma, \Pi \vdash \Delta, \Lambda}}\, cut$$

$$\Downarrow$$

[d] Note that only $\rightsquigarrow_{\oplus\otimes_W}$-normalized structs are invariant under $\rhd_w$. $\rightsquigarrow_{\oplus\otimes_P}$-normalized structs are not invariant. This is because $\rightsquigarrow_{\oplus\otimes_W}$ is better than $\rightsquigarrow_{\oplus\otimes_P}$ when it comes to exploiting redundancies due to cut-pertinent weakening inferences. Consequently, the swapped clause sets are also invariant under $\rhd_w$, while profile clause sets are not. Indeed, the behaviour of profile clause sets under $\rhd_w$ is quite complex to describe [67, 66] .

$$\dfrac{\dfrac{\begin{array}{c} S_{\varphi_r} \\ \varphi_r \end{array}}{\Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda} \; w_r^*, w_l^*$$
$$S_{\varphi_r}$$

$\square$

Definition 5.2.4 (Equivalence of Structs Modulo Renaming and Multiplicity of $\otimes$-junctions)**.** Let S_1 and S_2 be $\otimes$-junctions (i.e. they are structs not containing the $\oplus$ connective). Then $S_1 =_s S_2$ if and only if there is a variable renaming σ such that $S_1\sigma = S_2$.

Let $S_1 \doteq \bigoplus_{1 \le i \le n} S_{1i}$ and $S_2 \doteq \bigoplus_{1 \le j \le m} S_{2j}$ be two structs where S_{1i} and S_{2j} are $\otimes$-junctions. Then $S_1 =_s S_2$ if and only if for all S_{1k} there is a S_{2l} such that $S_{2l} =_s S_{1k}$ and for all $S_{2l'}$ there is a $S_{1k'}$ such that $S_{2l'} =_s S_{1k'}$.

Let S_1 and S_2 be arbitrary structs. Then $S_1 =_s S_2$ if and only if for all S_1' such that $S_1 \rightsquigarrow_{\oplus\otimes_W} S_1'$ there exists S_2' such that $S_2 \rightsquigarrow_{\oplus\otimes_W} S_2'$ and $S_1' =_s S_2'$ and for all S_2' such that $S_2 \rightsquigarrow_{\oplus\otimes_W} S_2'$ there exists S_1' such that $S_1 \rightsquigarrow_{\oplus\otimes_W} S_1'$ and $S_1' =_s S_2'$.

Lemma 5.4 (Duplication of Substructs)**.** Let S_1 be a substruct of S. Let S' be the struct obtained from S by replacing S_1 by $S_1 \oplus S_2$ where $S_2 = S_1\sigma$ for a variable renaming σ. Let $S \rightsquigarrow_{\oplus\otimes} S_{norm}$ and $S' \rightsquigarrow_{\oplus\otimes} S'_{norm}$. Then $S_{norm} =_s S'_{norm}$.

Proof. By induction on the structure of S or on the length of the normalization sequence. $\square$

Lemma 5.5 (Duplication of Substructs)**.** Let S_1 be a substruct of S. Let S' be the struct obtained from S by replacing S_1 by $S_1 \oplus S_2$ where $S_2 = S_1\sigma$ for a variable renaming σ. Let $S \rightsquigarrow_{\oplus\otimes_P} S_{norm}$ and $S' \rightsquigarrow_{\oplus\otimes_P} S'_{norm}$. Then $S_{norm} =_s S'_{norm}$.

Proof. By induction on the structure of S or on the length of the normalization sequence. $\square$

Lemma 5.6 (Duplication of Substructs)**.** Let S_1 be a substruct of S. Let S' be the struct obtained from S by replacing S_1 by $S_1 \oplus S_2$ where $S_2 = S_1\sigma$ for a variable renaming σ. Then, for any S_{norm} such that $S \rightsquigarrow_{\oplus\otimes_W} S_{norm}$ there is a S'_{norm} such that $S' \rightsquigarrow_{\oplus\otimes_W} S'_{norm}$ and $S_{norm} =_s S'_{norm}$.

Proof. In contrast to the previous cases for $\rightsquigarrow_{\oplus\otimes}$ and $\rightsquigarrow_{\oplus\otimes_P}$, for $\rightsquigarrow_{\oplus\otimes_W}$ it was necessary to state the lemma in a slightly modified way, due to the fact that $\rightsquigarrow_{\oplus\otimes_W}$ is not confluent. The proof, however, can also be easily made by induction on the structure of S or on the length of the normalization sequence. □

Lemma 5.7 (Invariance Modulo Renaming and Multiplicity under $\rhd_c$).

$$\psi[\varphi] \rhd_c \psi[\varphi'] \qquad \text{implies} \qquad \mathcal{S}_{\psi[\varphi]} =_s \mathcal{S}_{\psi[\varphi']}$$

Proof. The case analysis below shows that applying $\rhd_c$ in a subproof φ of ψ adds copies of some clauses to the characteristic clause set. However, the new clauses are just variants obtained by renaming eigenvariables. Therefore, $\mathcal{S}_\varphi =_s \mathcal{S}_{\varphi'}$, and hence $\mathcal{S}_{\psi[\varphi]} =_s \mathcal{S}_{\psi[\varphi']}$.

- $\rhd_c$ with a contraction in the left branch: As shown below, $\mathcal{S}_\varphi = \mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}$ and $\mathcal{S}_{\varphi'} = (\mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}) \oplus \mathcal{S}_{\varphi'_r}$. Since φ'_r is just a copy of φ_r with the eigenvariables renamed for the sake of proof regularity, $\mathcal{S}_{\varphi'_r} =_s \mathcal{S}_{\varphi_r}$. Therefore, $\mathcal{S}_\varphi =_s \mathcal{S}_{\varphi'}$.

$$\dfrac{\dfrac{\begin{matrix}\mathcal{S}_{\varphi_l}\\ \varphi_l\\ \Gamma \vdash \Delta, A, A\end{matrix}}{\Gamma \vdash \Delta, A}\,c_r \qquad \begin{matrix}\mathcal{S}_{\varphi_r}\\ \varphi_r\\ A, \Pi \vdash \Lambda\end{matrix}}{\begin{matrix}\Gamma, \Pi \vdash \Delta, \Lambda\\ \mathcal{S}_{\varphi_l}\oplus\mathcal{S}_{\varphi_r}\end{matrix}}\,cut$$

$$\Downarrow$$

$$\begin{array}{c}\dfrac{\dfrac{\begin{matrix}\mathcal{S}_{\varphi_l}\\ \varphi_l\\ \Gamma \vdash \Delta, A, A\end{matrix} \qquad \begin{matrix}\mathcal{S}_{\varphi_r}\\ \varphi_r\\ A, \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi \vdash \Delta, \Lambda, A}\,cut \qquad \begin{matrix}\mathcal{S}_{\varphi'_r}\\ \varphi'_r\\ A, \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi, \Pi \vdash \Delta, \Lambda, \Lambda,}\,cut \\ \hline\hline \Gamma, \Pi \vdash \Delta, \Lambda \quad c_l^*, c_r^* \\ (\mathcal{S}_{\varphi_l}\oplus\mathcal{S}_{\varphi_r})\oplus\mathcal{S}_{\varphi'_r}\end{array}$$

- $\rhd_c$ with a contraction in the right branch: As shown below, $\mathcal{S}_\varphi = \mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}$ and $\mathcal{S}_{\varphi'} = \mathcal{S}_{\varphi'_l} \oplus (\mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r})$. Since φ'_l is just a copy of φ_l with the eigenvariables renamed for the sake of proof regularity, $\mathcal{S}_{\varphi'_l} =_s \mathcal{S}_{\varphi_l}$. Therefore, $\mathcal{S}_\varphi =_s \mathcal{S}_{\varphi'}$.

$$\dfrac{\begin{matrix}{\color{red}\mathcal{S}_{\varphi_l}}\\ \varphi_l\\ \Gamma\vdash\Delta,A\end{matrix} \qquad \dfrac{\begin{matrix}{\color{red}\mathcal{S}_{\varphi_r}}\\ \varphi_r\\ A,A,\Pi\vdash\Lambda\end{matrix}}{A,\Pi\vdash\Lambda}\,c_l}{\begin{matrix}\Gamma,\Pi\vdash\Delta,\Lambda\\ {\color{red}\mathcal{S}_{\varphi_l}\oplus\mathcal{S}_{\varphi_r}}\end{matrix}}\,cut$$

$$\Downarrow$$

$$\dfrac{\dfrac{\begin{matrix}{\color{red}\mathcal{S}_{\varphi'_l}}\\ \varphi'_l\\ \Gamma\vdash\Delta,A\end{matrix} \qquad \dfrac{\begin{matrix}{\color{red}\mathcal{S}_{\varphi_l}}\\ \varphi_l\\ \Gamma\vdash\Delta,A\end{matrix} \qquad \begin{matrix}{\color{red}\mathcal{S}_{\varphi_r}}\\ \varphi_r\\ A,A,\Pi\vdash\Lambda\end{matrix}}{A,\Gamma,\Pi\vdash\Delta,\Lambda}\,cut}{\Gamma,\Gamma,\Pi\vdash\Delta,\Delta,\Lambda}\,cut}{\begin{matrix}\Gamma,\Pi\vdash\Delta,\Lambda\\ {\color{red}\mathcal{S}_{\varphi'_l}\oplus(\mathcal{S}_{\varphi_l}\oplus\mathcal{S}_{\varphi_r})}\end{matrix}}\,c_l^*,c_r^*$$

$\square$

Lemma 5.8 (Normalized Invariance under $\rhd_r$).

$$\psi[\varphi] \rhd_r \psi[\varphi'] \qquad \text{implies} \qquad \mathcal{S}_{\psi[\varphi']} = S_\psi[S']$$

such that $\mathcal{S}_{\psi[\varphi]} = S_\psi[S] \rightsquigarrow_{\oplus\otimes W} S_\psi[S']$

Proof. The case analysis below shows that no change occurs in the $\rightsquigarrow_{\oplus\otimes W}$-normalized cut-pertinent structs, when the subproof φ of ψ is rewritten according to the cut rank reduction rules $\rhd_r$.

- $\rhd_r$ with a unary rule: As shown below, $\mathcal{S}_\varphi = \mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}$ and $\mathcal{S}_{\varphi'} = \mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}$. Hence, $\mathcal{S}_\varphi = \mathcal{S}_{\varphi'}$.

$$\dfrac{\begin{matrix}{\color{red}\mathcal{S}_{\varphi_l}}\\ \varphi_l\\ \Gamma\vdash\Delta,A\end{matrix} \qquad \dfrac{\begin{matrix}{\color{red}\mathcal{S}_{\varphi_r}}\\ \varphi_r\\ A,\Pi'\vdash\Lambda'\end{matrix}}{A,\Pi\vdash\Lambda}\,\rho}{\begin{matrix}\Gamma,\Pi\vdash\Delta,\Lambda\\ {\color{red}\mathcal{S}_{\varphi_l}\oplus\mathcal{S}_{\varphi_r}}\end{matrix}}\,cut$$

$$\Downarrow$$

$$\begin{array}{c}\dfrac{\dfrac{\begin{array}{c}\mathcal{S}_{\varphi_l}\\ \varphi_l\\ \Gamma \vdash \Delta, A\end{array} \qquad \begin{array}{c}\mathcal{S}_{\varphi_r}\\ \varphi_r\\ A, \Pi' \vdash \Lambda'\end{array}}{\Gamma, \Pi' \vdash \Delta, \Lambda'}\,cut}{\Gamma, \Pi \vdash \Delta, \Lambda}\,\rho \\ \mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}\end{array}$$

$$\begin{array}{c}\dfrac{\dfrac{\begin{array}{c}\mathcal{S}_{\varphi_l}\\ \varphi_l\\ \Gamma' \vdash \Delta', A\end{array}}{\Gamma \vdash \Delta, A}\,\rho \qquad \begin{array}{c}\mathcal{S}_{\varphi_r}\\ \varphi_r\\ A, \Pi \vdash \Lambda\end{array}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut \\ \mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}\end{array}$$

$$\Downarrow$$

$$\begin{array}{c}\dfrac{\dfrac{\begin{array}{c}\mathcal{S}_{\varphi_l}\\ \varphi_l\\ \Gamma' \vdash \Delta', A\end{array} \qquad \begin{array}{c}\mathcal{S}_{\varphi_r}\\ \varphi_r\\ A, \Pi \vdash \Lambda\end{array}}{\Gamma', \Pi \vdash \Delta', \Lambda}\,cut}{\Gamma, \Pi \vdash \Delta, \Lambda}\,\rho \\ \mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_r}\end{array}$$

- $\rhd_r$ with a cut-pertinent binary rule: In all cases below, $\mathcal{S}_{\varphi} = \mathcal{S}_{\varphi'}$ by the associativity and commutativity of $\oplus$.

$$\begin{array}{c}\dfrac{\begin{array}{c}\mathcal{S}_{\varphi_l}\\ \varphi_l\\ \Pi \vdash \Lambda, A\end{array} \qquad \dfrac{\begin{array}{c}\mathcal{S}_{\varphi_1}\\ \varphi_1\\ A, \Gamma_1 \vdash \Delta_1\end{array} \qquad \begin{array}{c}\mathcal{S}_{\varphi_2}\\ \varphi_2\\ \Gamma_2 \vdash \Delta_2\end{array}}{A, \Gamma \vdash \Delta}\,\rho}{\Pi, \Gamma \vdash \Lambda, \Delta}\,cut \\ \mathcal{S}_{\varphi_l} \oplus (\mathcal{S}_{\varphi_1} \oplus \mathcal{S}_{\varphi_2})\end{array}$$

$$\Downarrow$$

$$\begin{array}{c}\dfrac{\dfrac{\begin{array}{c}\mathcal{S}_{\varphi_l}\\ \varphi_l\\ \Pi \vdash \Lambda, A\end{array} \qquad \begin{array}{c}\mathcal{S}_{\varphi_1}\\ \varphi_1\\ A, \Gamma_1 \vdash \Delta_1\end{array}}{\Pi, \Gamma_1 \vdash \Lambda, \Delta_1}\,cut \qquad \begin{array}{c}\mathcal{S}_{\varphi_2}\\ \varphi_2\\ \Gamma_2 \vdash \Delta_2\end{array}}{\Pi, \Gamma \vdash \Lambda, \Delta}\,\rho \\ (\mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_1}) \oplus \mathcal{S}_{\varphi_2}\end{array}$$

$$
\dfrac{\begin{array}{c}\mathcal{S}_{\varphi_l}\\ \varphi_l\\ \Pi\vdash\Lambda,A\end{array}\qquad \dfrac{\begin{array}{c}\mathcal{S}_{\varphi_1}\\ \varphi_1\\ \Gamma_1\vdash\Delta_1\end{array}\qquad\begin{array}{c}\mathcal{S}_{\varphi_2}\\ \varphi_2\\ A,\Gamma_2\vdash\Delta_2\end{array}}{A,\Gamma\vdash\Delta}\,\rho}{\begin{array}{c}\Pi,\Gamma\vdash\Lambda,\Delta\\ \mathcal{S}_{\varphi_l}\oplus(\mathcal{S}_{\varphi_1}\oplus\mathcal{S}_{\varphi_2})\end{array}}\,cut
$$

$$\Downarrow$$

$$
\dfrac{\begin{array}{c}\mathcal{S}_{\varphi_1}\\ \varphi_1\\ \Gamma_1\vdash\Delta_1\end{array}\qquad \dfrac{\begin{array}{c}\mathcal{S}_{\varphi_l}\\ \varphi_l\\ \Pi\vdash\Lambda,A\end{array}\qquad\begin{array}{c}\mathcal{S}_{\varphi_2}\\ \varphi_2\\ A,\Gamma_2\vdash\Delta_2\end{array}}{\Pi,\Gamma_2\vdash\Lambda,\Delta_2}\,cut}{\begin{array}{c}\Pi,\Gamma\vdash\Lambda,\Delta\\ \mathcal{S}_{\varphi_1}\oplus(\mathcal{S}_{\varphi_l}\oplus\mathcal{S}_{\varphi_2})\end{array}}\,\rho
$$

$$
\dfrac{\dfrac{\begin{array}{c}\mathcal{S}_{\varphi_1}\\ \varphi_1\\ \Gamma_1\vdash\Delta_1,A\end{array}\qquad\begin{array}{c}\mathcal{S}_{\varphi_2}\\ \varphi_2\\ \Gamma_2\vdash\Delta_2\end{array}}{\Gamma\vdash\Delta,A}\,\rho\qquad \begin{array}{c}\mathcal{S}_{\varphi_r}\\ \varphi_r\\ A,\Pi\vdash\Lambda\end{array}}{\begin{array}{c}\Gamma,\Pi\vdash\Delta,\Lambda\\ (\mathcal{S}_{\varphi_1}\oplus\mathcal{S}_{\varphi_2})\oplus\mathcal{S}_{\varphi_r}\end{array}}\,cut
$$

$$\Downarrow$$

$$
\dfrac{\dfrac{\begin{array}{c}\mathcal{S}_{\varphi_1}\\ \varphi_1\\ \Gamma_1\vdash\Delta_1,A\end{array}\qquad\begin{array}{c}\mathcal{S}_{\varphi_r}\\ \varphi_r\\ A,\Pi\vdash\Lambda\end{array}}{\Gamma_1,\Pi\vdash\Delta_1,\Lambda}\,cut\qquad \begin{array}{c}\mathcal{S}_{\varphi_2}\\ \varphi_2\\ \Gamma_2\vdash\Delta_2\end{array}}{\begin{array}{c}\Gamma,\Pi\vdash\Delta,\Lambda\\ (\mathcal{S}_{\varphi_1}\oplus\mathcal{S}_{\varphi_r})\oplus\mathcal{S}_{\varphi_2}\end{array}}\,\rho
$$

$$
\dfrac{\dfrac{\begin{array}{c}\mathcal{S}_{\varphi_1}\\ \varphi_1\\ \Gamma_1\vdash\Delta_1\end{array}\qquad\begin{array}{c}\mathcal{S}_{\varphi_2}\\ \varphi_2\\ \Gamma_2\vdash\Delta_2,A\end{array}}{\Gamma\vdash\Delta,A}\,\rho\qquad \begin{array}{c}\mathcal{S}_{\varphi_r}\\ \varphi_r\\ A,\Pi\vdash\Lambda\end{array}}{\begin{array}{c}\Gamma,\Pi\vdash\Delta,\Lambda\\ (\mathcal{S}_{\varphi_1}\oplus\mathcal{S}_{\varphi_2})\oplus\mathcal{S}_{\varphi_r}\end{array}}\,cut
$$

$$\Downarrow$$

$$\dfrac{\dfrac{\mathcal{S}_{\varphi_1}}{\dfrac{\varphi_1}{\Gamma_1 \vdash \Delta_1}} \quad \dfrac{\dfrac{\dfrac{\mathcal{S}_{\varphi_2}}{\varphi_2}}{\Gamma_2 \vdash \Delta_2, A} \quad \dfrac{\dfrac{\mathcal{S}_{\varphi_r}}{\varphi_r}}{A, \Pi \vdash \Lambda}}{\Gamma_2, \Pi \vdash \Delta_2, \Lambda}\ cut}{\Gamma, \Pi \vdash \Delta, \Lambda}\ \rho$$
$$\mathcal{S}_{\varphi_1} \oplus (\mathcal{S}_{\varphi_2} \oplus \mathcal{S}_{\varphi_r})$$

- $\rhd_r$ with a cut-impertinent binary rule:

  In the rank reduction immediately below, $\mathcal{S}_{\varphi'} = (\mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_1}) \otimes \mathcal{S}_{\varphi_2}$, and thus $\mathcal{S}_{\varphi'} \neq \mathcal{S}_{\varphi'}$. However, it is easy to see that $\mathcal{S}_{\varphi} \rightsquigarrow_{\oplus\otimes W} \mathcal{S}_{\varphi'}$, due to the fact that $\rightsquigarrow_{\oplus\otimes W}$ performs only a partial distribution[e] of $\otimes$ over $\oplus$.

$$\dfrac{\dfrac{\dfrac{\mathcal{S}_{\varphi_l}}{\varphi_l}}{\Pi \vdash \Lambda, A} \quad \dfrac{\dfrac{\dfrac{\mathcal{S}_{\varphi_1}}{\varphi_1}}{A, \Gamma_1 \vdash \Delta_1} \quad \dfrac{\dfrac{\mathcal{S}_{\varphi_2}}{\varphi_2}}{\Gamma_2 \vdash \Delta_2}}{A, \Gamma \vdash \Delta}\ \rho}{\Pi, \Gamma \vdash \Lambda, \Delta}\ cut$$
$$\mathcal{S}_{\varphi_l} \oplus (\mathcal{S}_{\varphi_1} \otimes_\rho \mathcal{S}_{\varphi_2})$$

$$\Downarrow$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{S}_{\varphi_l}}{\varphi_l}}{\Pi \vdash \Lambda, A} \quad \dfrac{\dfrac{\mathcal{S}_{\varphi_1}}{\varphi_1}}{A, \Gamma_1 \vdash \Delta_1}}{\Pi, \Gamma_1 \vdash \Lambda, \Delta_1}\ cut \quad \dfrac{\dfrac{\mathcal{S}_{\varphi_2}}{\varphi_2}}{\Gamma_2 \vdash \Delta_2}}{\Pi, \Gamma \vdash \Lambda, \Delta}\ \rho$$
$$(\mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_1}) \otimes_\rho \mathcal{S}_{\varphi_2} \rightsquigarrow_{\oplus\otimes W} \mathcal{S}_{\varphi_l} \oplus (\mathcal{S}_{\varphi_1} \otimes_\rho \mathcal{S}_{\varphi_2})$$

$$\dfrac{\dfrac{\dfrac{\mathcal{S}_{\varphi_l}}{\varphi_l}}{\Pi \vdash \Lambda, A} \quad \dfrac{\dfrac{\dfrac{\mathcal{S}_{\varphi_1}}{\varphi_1}}{\Gamma_1 \vdash \Delta_1} \quad \dfrac{\dfrac{\mathcal{S}_{\varphi_2}}{\varphi_2}}{A, \Gamma_2 \vdash \Delta_2}}{A, \Gamma \vdash \Delta}\ \rho}{\Pi, \Gamma \vdash \Lambda, \Delta}\ cut$$
$$\mathcal{S}_{\varphi_l} \oplus (\mathcal{S}_{\varphi_1} \otimes_\rho \mathcal{S}_{\varphi_2})$$

$$\Downarrow$$

[e] Not only $\rightsquigarrow_{\oplus\otimes W}$-normalized structs (and, correspondingly, swapped clause sets) but also $\rightsquigarrow_{\oplus\otimes P}$-normalized structs (and, correspondingly, profile clause sets) are invariant under $\rhd_r$. The proof of this fact can be found in [66, 67]. However, $\rightsquigarrow_{\oplus\otimes}$-normalized structs (and, correspondingly, standard clause sets) are not invariant under $\rhd_r$. This negative fact can be seen in [12].

$$\dfrac{\begin{matrix}\mathcal{S}_{\varphi_1}\\ \varphi_1\\ \Gamma_1 \vdash \Delta_1\end{matrix} \qquad \dfrac{\begin{matrix}\mathcal{S}_{\varphi_l}\\ \varphi_l\\ \Pi \vdash \Lambda, A\end{matrix} \qquad \begin{matrix}\mathcal{S}_{\varphi_2}\\ \varphi_2\\ A, \Gamma_2 \vdash \Delta_2\end{matrix}}{\Pi, \Gamma_2 \vdash \Lambda, \Delta_2}\,cut}{\Pi, \Gamma \vdash \Lambda, \Delta}\,\rho$$

$$\mathcal{S}_{\varphi_1} \otimes_\rho (\mathcal{S}_{\varphi_l} \oplus \mathcal{S}_{\varphi_2}) \rightsquigarrow_{\oplus\otimes W} \mathcal{S}_{\varphi_l} \oplus (\mathcal{S}_{\varphi_1} \otimes_\rho \mathcal{S}_{\varphi_2})$$

$$\dfrac{\dfrac{\begin{matrix}\mathcal{S}_{\varphi_1}\\ \varphi_1\\ \Gamma_1 \vdash \Delta_1, A\end{matrix} \qquad \begin{matrix}\mathcal{S}_{\varphi_2}\\ \varphi_2\\ \Gamma_2 \vdash \Delta_2\end{matrix}}{\Gamma \vdash \Delta, A}\,\rho \qquad \begin{matrix}\mathcal{S}_{\varphi_r}\\ \varphi_r\\ A, \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut$$

$$(\mathcal{S}_{\varphi_1} \otimes_\rho \mathcal{S}_{\varphi_2}) \oplus \mathcal{S}_{\varphi_r}$$

$$\Downarrow$$

$$\dfrac{\dfrac{\begin{matrix}\mathcal{S}_{\varphi_1}\\ \varphi_1\\ \Gamma_1 \vdash \Delta_1, A\end{matrix} \qquad \begin{matrix}\mathcal{S}_{\varphi_r}\\ \varphi_r\\ A, \Pi \vdash \Lambda\end{matrix}}{\Gamma_1, \Pi \vdash \Delta_1, \Lambda}\,cut \qquad \begin{matrix}\mathcal{S}_{\varphi_2}\\ \varphi_2\\ \Gamma_2 \vdash \Delta_2\end{matrix}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,\rho$$

$$(\mathcal{S}_{\varphi_1} \oplus \mathcal{S}_{\varphi_r}) \otimes_\rho \mathcal{S}_{\varphi_2} \rightsquigarrow_{\oplus\otimes W} (\mathcal{S}_{\varphi_1} \otimes_\rho \mathcal{S}_{\varphi_2}) \oplus \mathcal{S}_{\varphi_r}$$

$$\dfrac{\dfrac{\begin{matrix}\mathcal{S}_{\varphi_1}\\ \varphi_1\\ \Gamma_1 \vdash \Delta_1\end{matrix} \qquad \begin{matrix}\mathcal{S}_{\varphi_2}\\ \varphi_2\\ \Gamma_2 \vdash \Delta_2, A\end{matrix}}{\Gamma \vdash \Delta, A}\,\rho \qquad \begin{matrix}\mathcal{S}_{\varphi_r}\\ \varphi_r\\ A, \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut$$

$$(\mathcal{S}_{\varphi_1} \otimes_\rho \mathcal{S}_{\varphi_2}) \oplus \mathcal{S}_{\varphi_r}$$

$$\Downarrow$$

$$\dfrac{\begin{matrix}\mathcal{S}_{\varphi_1}\\ \varphi_1\\ \Gamma_1 \vdash \Delta_1\end{matrix} \qquad \dfrac{\begin{matrix}\mathcal{S}_{\varphi_2}\\ \varphi_2\\ \Gamma_2 \vdash \Delta_2, A\end{matrix} \qquad \begin{matrix}\mathcal{S}_{\varphi_r}\\ \varphi_r\\ A, \Pi \vdash \Lambda\end{matrix}}{\Gamma_2, \Pi \vdash \Delta_2, \Lambda}\,cut}{\Gamma, \Pi \vdash \Delta, \Lambda}\,\rho$$

$$\mathcal{S}_{\varphi_1} \otimes_\rho (\mathcal{S}_{\varphi_2} \oplus \mathcal{S}_{\varphi_r}) \rightsquigarrow_{\oplus\otimes W} (\mathcal{S}_{\varphi_1} \otimes_\rho \mathcal{S}_{\varphi_2}) \oplus \mathcal{S}_{\varphi_r}$$

□

Definition 5.2.5 (Struct Precedence).

$$\preccurlyeq \doteq \leq_s \cup =_s \cup =$$

Lemma 5.9 (Precedence under $\triangleright$). If $\psi[\varphi] \triangleright \psi[\varphi']$ then for any $\rightsquigarrow_{\oplus\otimes_W}$-normal-form S' of $\mathcal{S}_{\psi[\varphi']}$ there exists a $\rightsquigarrow_{\oplus\otimes_W}$-normal-form S of $\mathcal{S}_{\psi[\varphi]}$ such that $S \preccurlyeq S'$.

Proof. This lemma is a direct consequence of Lemmas 5.1, 5.2, 5.3, 5.7 and 5.8. □

Lemma 5.10 (Refutations of Preceding Clause Sets). Let S_1 and S_2 be structs in $\rightsquigarrow_{\oplus\otimes_W}$-normal-form such that $S_1 \preccurlyeq^* S_2$. Then, any resolution refutation δ of $\mathrm{cl}(S_2)$ is also a refutation of $\mathrm{cl}(S_1)$.

Proof. By the definition of $\preccurlyeq$, the clauses in $\mathrm{cl}(S_2)$ must be instances of $\mathrm{cl}(S_1)$. Since the leaf clauses of any refutation δ of $\mathrm{cl}(S_2)$ are instances of clauses in $\mathrm{cl}(S_2)$, they are also instances of clauses of $\mathrm{cl}(S_1)$. Therefore, δ is also a refutation of $\mathrm{cl}(S_1)$. □

Theorem 5.4. CERes_W^O CR-simulates $\triangleright_{\bar{a}}^{\downarrow}$.

Proof. Let φ be a proof with cuts and φ' such that $\varphi \triangleright_{\bar{a}}^{\downarrow} \varphi'$, and let S' be $\rightsquigarrow_{\oplus\otimes_W}$-normal-form of $\mathcal{S}_{\varphi'}$.

By Lemma 4.9, there exists a proof φ'' such that $\varphi' \gg^* \varphi''$ and $\mathcal{S}_{\varphi''} = S'$. Moreover, φ'' is in $\gg_{\oplus\otimes}$-normal-form. Hence $CR(\varphi'')$ exists and is a refutation of S'.

By an iterated use of Lemma 5.9, there exists a $\rightsquigarrow_{\oplus\otimes_W}$-normal-form S of $\mathcal{S}_\varphi$ such that $S \preccurlyeq S'$; and, by Lemma 5.10, $CR(\varphi'')$ is also a refutation of S.

Let $\varphi^* \doteq \mathrm{CERes}_W^O(\varphi, CR(\varphi''))$. Clearly, $CR(\varphi^*) = CR(\varphi'')$. Therefore, CERes_W^O CR-simulates $\triangleright_{\bar{a}}^{\downarrow}$. □

Remark 5.2.1. The proof of Theorem 5.4, together with the soundness of the Resolution calculus, can be seen as an alternative constructive proof of Theorem 4.5. A refutation of a swapped clause set is constructed by applying $\triangleright_{\bar{a}}^{\downarrow}$ to a proof and extracting its canonic refutation. Therefore, since a refutation (i.e. the extracted canonic refutation) of the swapped clause set exists and the resolution calculus is sound, the swapped clause set must be unsatisfiable.

Theorem 5.5. CERes_W^O CR-simulates $\triangleright_{\tilde{a}}^{\downarrow}$.

Proof. By Theorem 5.4, CERes_W^O CR-simulates $\triangleright_{\bar{a}}^{\downarrow}$. By Theorem 5.3, $\triangleright_{\bar{a}}^{\downarrow}$ CR-simulates $\triangleright_{\tilde{a}}^{\downarrow}$. Therefore, CERes_W^O CR-simulates $\triangleright_{\tilde{a}}^{\downarrow}$. □

Theorem 5.6. CERes_W^O cannot be CR-simulated by any cut-elimination method based on $\rhd$.

Proof. Many examples of proofs for which CR-simulation is impossible are shown in Chapter 6. $\square$

Chapter 6

Resolution Refinements for Cut-Elimination

Theorems 5.4, 5.6 in Chapter 5 have shown that CERes is better than reductive cut-elimination methods in the sense that CERes can always produce (modulo CR-equivalence) the normal forms produced by reductive methods but the converse does not hold. However, CERes pays a high price for its flexibility and power. As shown in Example 6.1, refuting a clause set of a proof can be as hard as proving its end-sequent by resolution (i.e. refuting the negation of its end-sequent) from scratch. Moreover, Theorem 6.1 shows that Example 6.1 is not an isolated case; there is actually a large class of proofs, namely proofs that do not contain certain kinds of redundancy (e.g. weakening inferences), for which refuting the clause set is as hard as proving the end-sequent from scratch.

That the difficulty of refuting clause sets is a serious issue, which jeopardizes the practical application of CERes, is empirically supported by experiments made with Fuerstenberg's proof of the existence of infinitely many primes [6]. Fuerstenberg's proof, which uses lemmas (cuts) from topology, has been formalized as a sequence of proofs where the k-th proof shows that there are more than k primes. Current theorem provers, such as Otter [83] and Prover9 [84], were unable to refute the clause sets for $k \geq 2$.

One possible approach to solve the issue of refutation search is to enrich the cut-pertinent struct with additional information (defined in Subsection 6.2.1) about the cuts in the proof. Subsequently, resolution refinements can be defined, which use the extra information to constrain the inferences of the resolution calculus and thus reduce the space of allowed resolution deductions. Subsections 6.2.4, 6.2.5, 6.2.6 and 6.2.7 show a few possible refinements, and Subsection 6.2.8 is devoted to proving that clause sets are still refutable when the refined resolution calculi are used.

6.1 Refuting Clause Sets versus Searching for Resolution Proofs of End-Sequents

Example 6.1 shows a case in which refuting the clause set of a proof is as hard as refuting the negation of the formula corresponding to the end-sequent of the proof.

Indeed, in general, a proof with only atomic cuts can always be obtained by simply reproving the end-sequent by resolution (i.e. refuting its negation) and then converting this refutation to a sequent calculus proof [39]. However, this process is normally more costly than to use CERes or reductive methods, simply because a large amount of information contained in the proof with cuts is simply disregarded and only the end-sequent is considered and reproved. But for cases such as that of Example 6.1, it is actually simpler to reprove the end-sequent by resolution; and this is certainly an undesirable result for CERes (and perhaps even for cut-elimination in general), because a cut-elimination method ought to be able to exploit the information contained in the input proof with cuts in such a way that a proof with only atomic cuts could be constructed more easily than if it were constructed simply by reproving the end-sequent without no additional information. In summary, a cut-elimination method ought to be simpler and easier than simply searching again for a cut-free proof (or a proof with only atomic cuts)[a].

A semantic proof of cut-elimination together with a complete search procedure for cut-free sequent calculus proofs could hardly be considered a cut-elimination method. Analogously, if using CERes were always as hard as reproving the end-sequent by resolution, it could be argued that CERes is not really a cut-elimination method, but just resolution proof search disguised as a cut-elimination method. Fortunately, there are many proofs for which using CERes is computationally simpler than reproving the end-sequent by resolution. Nevertheless, Theorem 6.1 characterizes a

[a] In fact, this kind of requirement could be generalized to proof transformations in general. A method that transforms a proof φ into a proof φ' satisfying certain properties ought to be computationally simpler than searching for another proof ψ having the same end-sequent as φ and satisfying those properties.

disturbingly large class of proofs on which CERes behaves as in Example 6.1.

Theorem 6.1 indirectly suggests that CERes behaves well (in comparison with pure resolution proof search) when the proofs with cuts contain weakening inferences or axiom sequents having either no cut-pertinent formula occurrences or only cut-pertinent formula occurrences. In such cases the clause sets are smaller and simpler to refute than the clause forms of the negated end-sequents. Theorem 6.2 shows that CERes also exploits the instantiations of cut-impertinent quantifiers, which are incorporated on the clause sets, making them easier to refute than the clause forms of the negated end-sequents. These theorems informally indicate that CERes is intermediary between a pure proof search method and a pure proof transformation method.

Example 6.1 (The Difficulty of Refuting Clause Sets)**.** *Let φ be the proof below:*

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{\vdash \neg A, A}\neg_r \quad B \vdash B}{A \to B \vdash \neg A, B}\to_l}{A \to B \vdash \neg A \vee B}\vee_r \quad \dfrac{\dfrac{\dfrac{A \vdash A}{\neg A \vdash \neg A}\neg^* \quad \dfrac{B \vdash B}{\neg B, B \vdash}\neg_l}{\neg B, \neg A \vee B \vdash \neg A}\vee_l}{\neg A \vee B \vdash \neg B \to \neg A}\to_r}{A \to B \vdash \neg B \to \neg A}cut$$

Then, its cut-pertinent struct is:

$$\mathcal{S}_\varphi \equiv (\neg A \otimes B) \oplus (A \oplus \neg B)$$

And its clause set is:

$$C^W_\varphi \equiv \{\ A \vdash B\ ;\ \vdash A\ ;\ B \vdash\ \}$$

The formula corresponding to end-sequent of φ is:

$$F_\varphi \doteq \mathcal{F}(A \to B \vdash \neg B \to \neg A) \equiv ((A \to B) \to (\neg B \to \neg A))$$

The clause form of $\neg F_\varphi$ is:

$$C \doteq \{\ A \vdash B\ ;\ \vdash A\ ;\ B \vdash\ \}$$

Since $C^W_\varphi = C$, refuting C^W_φ is as hard as proving the end-sequent of φ by resolution without using any information from φ.

Theorem 6.1 (Difficulty of Refuting Clause Sets of Some Proofs)**.** Let φ be a proof with a quantifier-free end-sequent, with neither contraction nor weakening inferences and such that every axiom sequent contains exactly one cut-pertinent occurrence. Let C be the clause form of the negation of the formula corresponding to the end-sequent of φ. Then $C^P_\varphi = C$.

Proof. Let $\mathcal{S}_\varphi \backslash \rho$ denote the substruct of $\mathcal{S}_\varphi$ at the inference ρ. Let $S\backslash\rho$ denote the $\rightsquigarrow_{\oplus\otimes_P}$-normal-form of $\mathcal{S}_\varphi\backslash\rho$. Let $C^P_\varphi\backslash\rho \doteq \mathrm{cl}(S\backslash\rho)$.

Let s_ρ denote the cut-impertinent subsequent of the conclusion sequent of the inference ρ. Let $\mathrm{cf}(F)$ denote the standard clause form of a formula F. Let C_ρ denote $\mathrm{cf}(\neg\mathcal{F}(s_\rho))$.

Below it is shown by induction that $C^P_\varphi\backslash\rho = C_\rho$ for all inferences ρ in φ. The theorem then follows from the facts that $C^P_\varphi = C^P_\varphi\backslash\rho^*$ and $C_{\rho^*} = C$, for ρ^* the bottommost inference of φ.

In all proofs displayed below, cut-pertinent formula occurrences are highlighted in red.

- Base case, ρ is an axiom inference: then the subproof at ρ has one of the following two forms:
  - If the cut-pertinent formula occurrence is in the consequent of the conclusion sequent of ρ:

    $$\dfrac{}{A \vdash A}\ axiom$$

    In this case, $C^P_\varphi\backslash\rho = \{\vdash A\}$ and $C_\rho = \{\vdash A\}$. Therefore $C^P_\varphi\backslash\rho = C_\rho$.
  - If the cut-pertinent formula occurrence is in the antecedent of the conclusion sequent of ρ:

    $$\dfrac{}{A \vdash A}\ axiom$$

    In this case, $C^P_\varphi\backslash\rho = \{A \vdash\}$ and $C_\rho = \{A \vdash\}$. Therefore $C^P_\varphi\backslash\rho = C_\rho$.

- ρ is a cut-pertinent unary inference: then the subproof at ρ has the following form:

  $$\dfrac{\dfrac{\vdots}{\Gamma, \Lambda \vdash \Delta, \Pi}\ \rho'}{\Gamma', \Lambda \vdash \Delta', \Pi}\ \rho$$

  By induction hypothesis, $C^P_\varphi\backslash\rho' = C_{\rho'}$. Since $s_\rho = s_{\rho'}$, $C_\rho = C_{\rho'}$. And $C^P_\varphi\backslash\rho = C^P_\varphi\backslash\rho'$, because $\mathcal{S}_\varphi\backslash\rho = \mathcal{S}_\varphi\backslash\rho'$. Therefore, $C^P_\varphi\backslash\rho = C_\rho$.

- ρ is a cut-pertinent binary inference: then the subproof at ρ has the following form:

$$\dfrac{\dfrac{\vdots}{\Gamma_1, \Lambda_1 \vdash \Delta_1, \Pi_1}\rho_1' \quad \dfrac{\vdots}{\Gamma_2, \Lambda_2 \vdash \Delta_2, \Pi_2}\rho_2'}{\Gamma', \Lambda_1, \Lambda_2 \vdash \Delta', \Pi_1, \Pi_2}\rho$$

By induction hypothesis, $C^P_\varphi \backslash \rho_1' = C_{\rho_1'}$ and $C^P_\varphi \backslash \rho_2' = C_{\rho_2'}$. $\mathcal{S}_\varphi \backslash \rho = \mathcal{S}_\varphi \backslash \rho_1' \oplus \mathcal{S}_\varphi \backslash \rho_2'$, and hence $C^P_\varphi \backslash \rho = C^P_\varphi \backslash \rho_1' \cup C^P_\varphi \backslash \rho_2'$. Moreover:

$$\begin{aligned}
C_\rho &= \mathrm{cf}(\neg\mathcal{F}(\Lambda_1, \Lambda_2 \vdash \Pi_1, \Pi_2)) \\
&= \mathrm{cf}(\neg(\bigwedge \Lambda_1 \wedge \bigwedge \Lambda_2 \rightarrow \bigvee \Pi_1 \vee \bigvee \Pi_2)) \\
&= \mathrm{cf}(\bigwedge \Lambda_1 \wedge \bigwedge \Lambda_2 \wedge \neg \bigvee \Pi_1 \wedge \neg \bigvee \Pi_2) \\
&= \mathrm{cf}(\bigwedge \Lambda_1) \cup \mathrm{cf}(\bigwedge \Lambda_2) \cup \mathrm{cf}(\neg \bigvee \Pi_1) \cup \mathrm{cf}(\neg \bigvee \Pi_2) \\
&= \mathrm{cf}(\bigwedge \Lambda_1) \cup \mathrm{cf}(\neg \bigvee \Pi_1) \cup \mathrm{cf}(\bigwedge \Lambda_2) \cup \mathrm{cf}(\neg \bigvee \Pi_2) \\
&= \mathrm{cf}(\bigwedge \Lambda_1 \wedge \neg \bigvee \Pi_1) \cup \mathrm{cf}(\bigwedge \Lambda_2 \wedge \neg \bigvee \Pi_2) \\
&= \mathrm{cf}(\neg\mathcal{F}(\Lambda_1 \vdash \Pi_1)) \cup \mathrm{cf}(\neg\mathcal{F}(\Lambda_2 \vdash \Pi_2)) \\
&= C_{\rho_1'} \cup C_{\rho_2'}
\end{aligned}$$

Therefore, $C^P_\varphi \backslash \rho = C_\rho$.

- ρ is a cut-impertinent unary inference: then ρ is a unary inference introducing a propositional connective, since by assumption contraction, weakening and quantifier inferences are not present in φ. The following cases can be distinguished:

  - ρ is a $\neg_r$ inference: Then the subproof at ρ has the following form:

$$\dfrac{\dfrac{\vdots}{\Gamma, \Lambda, F \vdash \Delta, \Pi}\rho'}{\Gamma, \Lambda \vdash \Delta, \Pi, \neg F}\neg_r(\rho)$$

    Clearly, $C^P_\varphi \backslash \rho = C^P_\varphi \backslash \rho'$, because $\mathcal{S}_\varphi \backslash \rho = \mathcal{S}_\varphi \backslash \rho'$. Moreover:

$$\begin{aligned}
C_\rho &= \mathrm{cf}(\neg\mathcal{F}(\Lambda \vdash \Pi, \neg F)) \\
&= \mathrm{cf}(\neg(\bigwedge \Lambda \rightarrow \bigvee \Pi \vee \neg F)) \\
&= \mathrm{cf}(\bigwedge \Lambda \wedge \neg \bigvee \Pi \wedge F) \\
&= \mathrm{cf}(\bigwedge \Lambda \wedge F \wedge \neg \bigvee \Pi) \\
&= \mathrm{cf}(\neg(\bigwedge \Lambda \wedge F \rightarrow \bigvee \Pi)) \\
&= \mathrm{cf}(\neg(\mathcal{F}(\Lambda, F \vdash \Pi))) \\
&= C_{\rho'}
\end{aligned}$$

    By induction hypothesis, $C^P_\varphi \backslash \rho' = C_{\rho'}$. Therefore, $C^P_\varphi \backslash \rho = C_\rho$.

  - ρ is a $\neg_l$ inference: Analogous to the previous case.

– ρ is a $\wedge_l$ inference: Then the subproof at ρ has the following form:

$$\dfrac{\dfrac{\vdots}{\Gamma, \Lambda, F_1, F_2 \vdash \Delta, \Pi}\ \rho'}{\Gamma, \Lambda, F_1 \wedge F_2 \vdash \Delta, \Pi}\ \wedge_l(\rho)$$

Clearly, $C^P_\varphi \backslash \rho = C^P_\varphi \backslash \rho'$, because $\mathcal{S}_\varphi \backslash \rho = \mathcal{S}_\varphi \backslash \rho'$. Moreover:

$$\begin{aligned} C_\rho &= \mathrm{cf}(\neg \mathcal{F}(\Lambda, F_1 \wedge F_2 \vdash \Pi)) \\ &= \mathrm{cf}(\neg(\bigwedge \Lambda \wedge F_1 \wedge F_2 \rightarrow \bigvee \Pi)) \\ &= \mathrm{cf}(\bigwedge \Lambda \wedge F_1 \wedge F_2 \wedge \neg \bigvee \Pi) \\ &= \mathrm{cf}(\neg \mathcal{F}(\Lambda, F_1, F_2 \vdash \Pi)) \\ &= C_{\rho'} \end{aligned}$$

By induction hypothesis, $C^P_\varphi \backslash \rho' = C_{\rho'}$. Therefore, $C^P_\varphi \backslash \rho = C_\rho$.

– ρ is a $\vee_r$ inference: analogous to the previous case.

– ρ is a $\rightarrow_r$ inference: analogous to the previous cases.

- ρ is a cut-impertinent binary inference: then ρ is a binary inference introducing a propositional connective. The following cases can therefore be distinguished:

– ρ is a $\wedge_r$ inference: then the subproof at ρ has the following form:

$$\dfrac{\dfrac{\vdots}{\Gamma_1, \Lambda_1 \vdash \Delta_1, \Pi_1, F_1}\ \rho'_1 \qquad \dfrac{\vdots}{\Gamma_2, \Lambda_2 \vdash \Delta_2, \Pi_2, F_2}\ \rho'_2}{\Gamma_1, \Gamma_2, \Lambda_1, \Lambda_2 \vdash \Delta_1, \Delta_2, \Pi_1, \Pi_2, F_1 \wedge F_2}\ \wedge_r(\rho)$$

Then, $C_{\rho'_1} = \mathrm{cf}(\bigwedge \Lambda_1) \cup \mathrm{cf}(\neg \bigvee \Pi_1) \cup \mathrm{cf}(\neg F_1)$, $C_{\rho'_1} = \mathrm{cf}(\bigwedge \Lambda_1) \cup \mathrm{cf}(\neg \bigvee \Pi_1) \cup \mathrm{cf}(\neg F_1)$ and:

$$\begin{aligned} C_\rho &= \mathrm{cf}(\neg \mathcal{F}(\Lambda_1, \Lambda_2 \vdash \Pi_1, \Pi_2, F_1 \wedge F_2)) \\ &= \mathrm{cf}(\neg(\bigwedge \Lambda_1 \wedge \bigwedge \Lambda_2 \rightarrow \bigvee \Pi_1 \vee \bigvee \Pi_2 \vee (F_1 \wedge F_2))) \\ &= \mathrm{cf}(\bigwedge \Lambda_1 \wedge \bigwedge \Lambda_2 \wedge \neg \bigvee \Pi_1 \wedge \neg \bigvee \Pi_2 \wedge \neg(F_1 \wedge F_2)) \\ &= \mathrm{cf}(\bigwedge \Lambda_1) \cup \mathrm{cf}(\bigwedge \Lambda_2) \cup \mathrm{cf}(\neg \bigvee \Pi_1) \cup \mathrm{cf}(\neg \bigvee \Pi_2) \cup \mathrm{cf}(\neg(F_1 \wedge F_2)) \\ &= \mathrm{cf}(\bigwedge \Lambda_1) \cup \mathrm{cf}(\bigwedge \Lambda_2) \cup \mathrm{cf}(\neg \bigvee \Pi_1) \cup \mathrm{cf}(\neg \bigvee \Pi_2) \cup \mathrm{cf}(\neg F_1 \vee \neg F_2)) \\ &= \mathrm{cf}(\bigwedge \Lambda_1) \cup \mathrm{cf}(\bigwedge \Lambda_2) \cup \mathrm{cf}(\neg \bigvee \Pi_1) \cup \mathrm{cf}(\neg \bigvee \Pi_2) \cup (\mathrm{cf}(\neg F_1) \odot \mathrm{cf}(\neg F_2)) \end{aligned}$$

where $(\mathrm{cf}(\neg F_1) \odot \mathrm{cf}(\neg F_2))$ denotes the set

$$\{c_1 \circ c_2 \mid c_1 \in \mathrm{cf}(\neg F_1), c_2 \in \mathrm{cf}(\neg F_2)\}$$

.

On the other hand, $\mathcal{S}_\varphi \backslash \rho = \mathcal{S}_\varphi \backslash \rho'_1 \otimes_\rho \mathcal{S}_\varphi \backslash \rho'_2$. By induction hypothesis, $C^P_\varphi \backslash \rho'_1 = C_{\rho'_1} = \mathrm{cf}(\bigwedge \Lambda_1) \cup \mathrm{cf}(\neg \bigvee \Pi_1) \cup \mathrm{cf}(\neg F_1)$ and

$C^P_\varphi \backslash \rho'_2 = C_{\rho'_2} = \mathrm{cf}(\bigwedge \Lambda_1) \cup \mathrm{cf}(\neg \bigvee \Pi_1) \cup \mathrm{cf}(\neg F_2)$. Then, by definition of $\rightsquigarrow_{\oplus\otimes_P}$-normalization:

$$C^P_\varphi \backslash \rho = \mathrm{cf}(\bigwedge \Lambda_1) \cup \mathrm{cf}(\bigwedge \Lambda_2) \cup \mathrm{cf}(\neg \bigvee \Pi_1) \cup \mathrm{cf}(\neg \bigvee \Pi_2) \cup (\mathrm{cf}(\neg F_1) \odot \mathrm{cf}(\neg F_2))$$

because only the substructs corresponding to $\mathrm{cf}(\neg F_1)$ and $\mathrm{cf}(\neg F_2)$ are $\Omega_\rho(\varphi)$-pertinent. Therefore, $C^P_\varphi \backslash \rho = C_\rho$.

- ρ is a $\vee_l$ inference: analogous to the previous case.
- ρ is a $\rightarrow_l$ inference: analogous to the previous cases.

$\square$

Remark 6.1.1. Since Theorem 6.1 requires the proof φ to be free of weakening inferences, $C^W_\varphi = C^P_\varphi$. Therefore, the theorem also holds for swapped clause sets. However, the theorem does not hold for standard clause sets, since the simple $\rightsquigarrow_{\oplus\otimes}$-normalization employed in the construction of standard clause sets creates redundant clauses and literals and hence the case of cut-impertinent binary inferences would fail.

Remark 6.1.2. A proof φ in the class of proofs that satisfy the conditions of Theorem 6.1 is such that it essentially as hard to compute $\mathrm{CERes}^O_P(\varphi, \delta)$ or $\mathrm{CERes}^O_W(\varphi, \delta)$ as to simply reprove the end-sequent by resolution. It is interesting to notice, moreover, that for this class of proofs, due to the redundancies introduced in the construction of standard clause sets, it is actually easier to simply reprove the end-sequent by resolution than to compute $\mathrm{CERes}^O_S(\varphi, \delta)$. Moreover, the proofs produced by reproving the end-sequent would be in general shorter and smaller than $\mathrm{CERes}^O_S(\varphi, \delta)$. This is yet another way of recognizing the superiority of profile and swapped clause sets in comparison to standard clause sets.

Theorem 6.2 (Difficulty of Refuting Clause Sets of Some Proofs)**.** Let φ be a skolemized proof with neither contraction nor weakening inferences and such that every axiom sequent contains exactly one cut-pertinent formula occurrence. Let C be the clause form of the negation of the formula corresponding to the end-sequent of φ. Then $C \leq_s C^W_\varphi$.

Proof. The proof is analogous to the proof of Theorem , with the following two additional cases when ρ is a weak quantifier inference:

- ρ is a $\forall_l$ inference: Then the subproof at ρ has the following form:

$$\dfrac{\dfrac{\vdots}{\Gamma, \Lambda, F[t] \vdash \Delta, \Pi}\,\rho'}{\Gamma, \Lambda, (\forall x)F(x) \vdash \Delta, \Pi}\,\forall_l(\rho)$$

By induction hypothesis, $C_{\rho'} \leq_s C^P_\varphi \backslash \rho'$. Clearly, $C^P_\varphi \backslash \rho = C^P_\varphi \backslash \rho'$, because $\mathcal{S}_\varphi \backslash \rho = \mathcal{S}_\varphi \backslash \rho'$; and $C_\rho <_s C_{\rho'}$. Therefore, $C_\rho \leq_s C^P_\varphi \backslash \rho$.

- ρ is a $\exists_r$ inference: analogous to the previous case.

□

Remark 6.1.3. Theorem 6.2 shows that, in the presence of weak quantifier inferences, while retaining all the other conditions of Theorem 6.1, it is easier to refute the cut-pertinent clause set than to reprove the end-sequent by resolution from scratch. This is so because the cut-pertinent clause set contains information about the instantiations that were used in the weak quantifier inferences, and hence it is more specific than the clause form of the negation of the formula corresponding to the end-sequent.

Remark 6.1.4. It might be fruitful to investigate further generalizations of Theorems 6.1 and 6.2 which would, for example, not require absence of contraction and weakening inferences. It seems that the presence of weakening inferences is always beneficial to CERes. However, Example 7.2 suggests that the presence of contractions can be bad for the CERes method. In that example, the swapped clause set has twice more clauses than the clause form of the negation of the formula corresponding to the end-sequent of the proof. Half of the clauses of the swapped clause set are mere duplicates caused by contractions. Therefore, it seems that a more detailed analysis of the effect of contractions on CERes could lead to improvements of the method.

6.2 Refinements based on Reductive Methods

In Section 6.1 it has been argued that the amount of search performed by CERes is in some cases unacceptably high; as high as to cast doubts on CERes status as a genuine cut-elimination method. The amount of search is high because the cut-pertinent structs do not contain enough information about the proofs with cuts. Therefore, to constrain the search, the natural solution is to enrich the cut-pertinent struct with additional information and define refined resolution calculi that make use of the extra information to constrain the inference rules and thus reduce the search space.

The following subsections define increasingly constrained refinements using increasingly more information from the proof with cuts. This allows a trade-off between how constrained the refutation search is and the variety of normal forms that can be produced. The more constrained the refinement and the more information is used from the proof with cuts, the farther the refined CERes is from being a pure resolution proof search method and closer it is to being a genuine proof-transformation method.

The refinements are inspired by reductive methods of cut-elimination. Basically, the aim is to define refined CERes methods that still CR-simulate reductive methods, but do not CR-Simulate the unrefined CERes. In this sense, the refined methods are intermediary between CERes and reductive methods. To achieve this aim, a simple procedure is used: firstly, example proofs with cuts are found for which reductive methods are not capable of producing normal forms that are CR-equivalent to some normal forms produced by CERes; then a refinement is defined in such a way that the refined CERes does not produce these normal forms that are not producible (modulo CR-equivalence) by reductive methods.

This procedure for designing refinements is also interesting from the point of view of clarifying the relation between CERes and reductive methods, two fundamentally different kinds of cut-elimination methods. In particular, it shows very clearly the reason why reductive methods do not CR-simulate CERes. Reductive methods are bound to preserve cut-linkage and cut-side annotations, as defined in Subsection 6.2.1 and proved in Subsections 6.2.2 and 6.2.3, while the unrefined CERes methods are not.

6.2.1 Cut-Linkage and Cut-Side

The following auxiliary definitions allow the annotation of cut-pertinent formula occurrences with information about how they relate to the cuts of a proof. These annotations persist when the cut-pertinent struct is extracted, therefore the cut-pertinent struct is automatically enriched with such annotations. Resolution refinements that exploit these annotations are defined in the following subsections.

Definition 6.2.1 (Cut-Linkage)**.** Two cut-pertinent atomic (sub)formula occurrences ν_1 and ν_2 in a proof φ are *cut-linked*, denoted $\nu_1 \smile \nu_2$, if and only if there is a cut ρ such that ν_1 is an ancestor of ν_i and ν_2 is an ancestor of ν_j where ν_i and ν_j are auxiliary occurrences of ρ.

Remark 6.2.1. ν_i and ν_j can be the same auxiliary occurrence. They do not need to be in different premises of the cut.

Example 6.2 (Cut-Linkage)**.** *In the proof φ below, occurrences highlighted with the same color are cut-linked to each other.*

$$
\begin{prooftree}
\AxiomC{$P(\alpha) \vdash P(\alpha)$}
\RightLabel{w_r}
\UnaryInfC{$P(\alpha), \neg P(\alpha) \vdash P(\alpha)$}
\RightLabel{$\rightarrow_r$}
\UnaryInfC{$P(\alpha), \vdash \neg P(\alpha) \rightarrow P(\alpha)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$(\forall x)P(x), \vdash \neg P(\alpha) \rightarrow P(\alpha)$}
\RightLabel{$\forall_r$}
\UnaryInfC{$(\forall x)P(x), \vdash (\forall x)(\neg P(x) \rightarrow P(x))$}
\AxiomC{$P(s) \vdash P(s)$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash \neg P(s), P(s)$}
\AxiomC{$P(s) \vdash P(s)$}
\RightLabel{$\rightarrow_l$}
\BinaryInfC{$\neg P(s) \rightarrow P(s) \vdash P(s), P(s)$}
\RightLabel{c_r}
\UnaryInfC{$\neg P(s) \rightarrow P(s) \vdash P(s)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$(\forall x)(\neg P(x) \rightarrow P(x)) \vdash P(s)$}
\AxiomC{$P(s) \vdash P(s)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$P(s) \vdash (\exists y)P(y)$}
\RightLabel{cut}
\BinaryInfC{$(\forall x)(\neg P(x) \rightarrow P(x)) \vdash (\exists y)P(y)$}
\RightLabel{cut}
\BinaryInfC{$(\forall x)P(x) \vdash (\exists y)P(y)$}
\end{prooftree}
$$

Definition 6.2.2 (Cut-Side)**.** Let ν be an ancestor of an auxiliary formula occurrence ν' of a cut ρ. ν has left (right) *cut-side* if and only if ν' occurs in the left (right) premise of ρ.

Definition 6.2.3 (Atomic Cut-Linkage)**.** Two cut-pertinent atomic formula occurrences ν_1 and ν_2 in a proof φ are *atomically cut-linked*, denoted $\nu_1 \smile \nu_2$, if and only if there is a cut-inference ρ such that $\nu_1 \searrow \lfloor \nu_i \rfloor_\pi$ and $\nu_2 \searrow \lfloor \nu_j \rfloor_\pi$ where π is the position of an atomic sub-formula and ν_i and ν_j are auxiliary occurrences of ρ.

Remark 6.2.2. ν_i and ν_j can be the same auxiliary occurrence. They do not need to be in different premises of the cut-inference.

Example 6.3 (Atomic Cut-Linkage)**.** *In the proof φ below, occurrences highlighted with the same color are atomically cut-linked to each other.*

$$
\begin{prooftree}
\AxiomC{$P(\alpha) \vdash P(\alpha)$}
\RightLabel{w_r}
\UnaryInfC{$P(\alpha), \neg P(\alpha) \vdash P(\alpha)$}
\RightLabel{$\rightarrow_r$}
\UnaryInfC{$P(\alpha), \vdash \neg P(\alpha) \rightarrow P(\alpha)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$(\forall x)P(x), \vdash \neg P(\alpha) \rightarrow P(\alpha)$}
\RightLabel{$\forall_r$}
\UnaryInfC{$(\forall x)P(x), \vdash (\forall x)\neg P(x) \rightarrow P(x)$}
\AxiomC{$P(s) \vdash P(s)$}
\RightLabel{$\neg_r$}
\UnaryInfC{$\vdash \neg P(s), P(s)$}
\AxiomC{$P(s) \vdash P(s)$}
\RightLabel{$\rightarrow_l$}
\BinaryInfC{$\neg P(s) \rightarrow P(s) \vdash P(s), P(s)$}
\RightLabel{c_r}
\UnaryInfC{$\neg P(s) \rightarrow P(s) \vdash P(s)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$(\forall x)\neg P(x) \rightarrow P(x) \vdash P(s)$}
\AxiomC{$P(s) \vdash P(s)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$P(s) \vdash (\exists y)P(y)$}
\RightLabel{cut}
\BinaryInfC{$(\forall x)\neg P(x) \rightarrow P(x) \vdash (\exists y)P(y)$}
\RightLabel{cut}
\BinaryInfC{$(\forall x)P(x) \vdash (\exists y)P(y)$}
\end{prooftree}
$$

Definition 6.2.4 (Proofs Annotated with Cut-Links and Cut-Sides)**.** A proof φ is said to be *annotated with cut-links and cut-sides* if and only if every atomic cut-pertinent (sub)formula occurrence is annotated with a set of labels that indicate its cut-side in φ and the set of (atomically) cut-linked occurrences of φ to which it belongs.

If ν is a cut-pertinent atomic (sub)formula occurrence in φ, then:

- *cutlink*(ν) denotes the label that indicates to which set of cut-linked occurrences ν belongs;
- *cutlink*$_a(\nu)$ denotes the label that indicates to which set of atomically cut-linked occurrences ν belongs;

- and *cutside*(v) denotes the label that indicates the cut-side of v.

Remark 6.2.3. The annotations of a formula occurrence are meant to be *persistent* in the sense that they are carried along with the occurrence when the occurrence is used for the construction of cut-pertinent structs and clause sets. In other words, when a cut-pertinent struct S is extracted from an annotated proof φ, the atomic formula occurrences in S have the same annotations as the corresponding occurrences in φ.

Moreover, if φ is transformed to a proof φ' by a proof transformation method, the annotations of a formula occurrence v in φ persist as annotations of any formula occurrence v' corresponding to v in φ'. Clearly, what it means for an occurrence in φ' to "correspond" to an occurrence of φ might be not always clear and ultimately depends on the particular proof transformations under consideration. Nevertheless, for the proof transformations considered here (i.e. reductive cut-elimination methods), the intuitive notion of correspondence of occurrences seems to be clear enough not to justify a tedious fully formal definition of correspondence. A problematic case might be reductions over contractions, in which occurrences are duplicated. In this case, both copies in φ' are considered to correspond to the occurrence in φ that was duplicated. Hence the annotations are also duplicated and carried along to both copies. Another problematic case are transformations that merge equal subproofs by adding contractions (e.g. the proof rewriting rewriting relations shown in Definition 4.3.4). In this case, the merged occurrences inherit the union of the annotations of the corresponding occurrences that have been merged.

To avoid confusion and clarify the fact that the labels have been inherited from φ, the following notation can be used:

- $\mathit{cutlink}(v', \varphi) \doteq \{\mathit{cutlink}(v) | v' \text{ corresponds to } v\}$;

- $\mathit{cutlink}_a(v', \varphi) \doteq \{\mathit{cutlink}_a(v) | v' \text{ corresponds to } v\}$;

- $\mathit{cutside}(v', \varphi) \doteq \{\mathit{cutside}(v) | v' \text{ corresponds to } v\}$;

Remark 6.2.4. To facilitate visualization, the annotations indicating cut-linkage are displayed as colors in the examples in this chapter. The annotations indicating the cut-side of an occurrence are displayed as superscripts on the occurrence (l for left cut-side and r for right cut-side).

6.2.2 Cut-Linkage Preservation under Reductive Cut-Elimination

The importance of the concepts of cut-linkage and atomic cut-linkage developed in the previous section lies in the fact that, under reductive cut-elimination, they are preserved (in a sense made precise in Theorems 6.3 and 6.4). Preservation of cut-linkage annotations is therefore an essential property of reductive cut-elimination methods, and it informally entails that canonic refutations from normal forms produced by reductive cut-elimination methods are always such that resolved literals have the same (atomic) cut-linkage annotations with respect to the input proof with cuts. This fact can be used to design resolution refinements for CERes as shown in Subsections 6.2.4, 6.2.5, 6.2.6 and 6.2.7.

Definition 6.2.5 (Positions and Subformulas)**.** $\ulcorner F \urcorner^{\pi}$ denotes the subformula of F at position π.

Remark 6.2.5. Positions can be encoded as binary strings. In this case, for example: $\ulcorner A \wedge (B \vee (\forall x)C(x))\urcorner^{0} = A$; $\ulcorner A \wedge (B \vee (\forall x)C(x))\urcorner^{1} = B \vee (\forall x)C(x)$; $\ulcorner A \wedge (B \vee (\forall x)C(x))\urcorner^{10} = B$; $\ulcorner A \wedge (B \vee (\forall x)C(x))\urcorner^{110} = C(x)$.

However, for the remainder of this chapter, it is irrelevant how positions are actually encoded.

Definition 6.2.6 (Context Formulas)**.** $F[F']_{\pi}$ denotes a formula F with the subformula F' in position π of F.

Theorem 6.3 (Atomic Cut-Linkage Preservation under Reductive Cut-Elimination)**.** Let φ be a proof with cuts and φ^* a proof such that $\varphi \rhd^* \varphi^*$. Let ρ be a cut in φ^* with auxiliary occurrences ν_l and ν_r. Then, for any position π, $cutlink_a(\ulcorner \nu_l \urcorner^{\pi}, \varphi) = cutlink_a(\ulcorner \nu_r \urcorner^{\pi}, \varphi)$.

Proof. The theorem can be proved by induction on the length n of the sequence rewriting $\varphi \rhd \varphi^1 \rhd \ldots \rhd \varphi^n = \varphi^*$. The base case, when $n = 0$ is trivial. For the inductive case, it is assumed as induction hypothesis that the theorem holds for $n = k$ and it is shown that it also holds for $n = k + 1$. To this aim, all possible cases of rewriting from φ^k to φ^{k+1} are analyzed below.

In all cases, it suffices to analyze what happens to a cut ρ of φ^{k+1} if it was modified (shifted, created, reduced, ...) by the rewriting of φ^k to φ^{k+1} (for if ρ is not modified, then $cutlink_a(\ulcorner \nu_l \urcorner^{\pi}, \varphi) = cutlink_a(\ulcorner \nu_r \urcorner^{\pi}, \varphi)$ holds trivially).

- $\varphi^k \rhd_p \varphi^{k+1}$: This case can be subdivided in four cases, depending on which connective is eliminated:

– $\varphi^k \rhd_{p_\wedge} \varphi^{k+1}$: Notice that in the lowermost of the two new cuts of φ^{k+1}, for any position π_2, the occurrences of the atomic formula C occupying that position in both auxiliary occurrences of the cut have the same atomic cut-linkage annotations (the same color blue). The same analogously holds for the uppermost of the two new cuts. Therefore, for any cut ρ of φ^{k+1} with auxiliary occurrences ν_l and ν_r and for any position π, $cutlink_a(\ulcorner \nu_l \urcorner^\pi, \varphi) = cutlink_a(\ulcorner \nu_r \urcorner^\pi, \varphi)$.

$$\dfrac{\dfrac{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1, F_1[B]_{\pi_1}} \qquad \overset{\varphi_2}{\Gamma_2 \vdash \Delta_2, F_2[C]_{\pi_2}}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2, F_1[B]_{\pi_1} \wedge F_2[C]_{\pi_2}} \wedge_r \qquad \dfrac{\overset{\varphi_r}{F_1[B]_{\pi_1}, F_2[C]_{\pi_2}, \Pi \vdash \Lambda}}{F_1[B]_{\pi_1} \wedge F_2[C]_{\pi_2}, \Pi \vdash \Lambda} \wedge_l}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda} cut$$

$$\Downarrow$$

$$\dfrac{\overset{\varphi_2}{\Gamma_2 \vdash \Delta_2, F_2[C]_{\pi_2}} \qquad \dfrac{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1, F_1[B]_{\pi_1}} \qquad \overset{\varphi_r}{F_1[B]_{\pi_1}, F_2[C]_{\pi_2}, \Pi \vdash \Lambda}}{F_2[C]_{\pi_2}, \Gamma_1, \Pi \vdash \Delta_1, \Lambda} cut}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda} cut$$

– $\varphi^k \rhd_{p_\vee} \varphi^{k+1}$: Analogous to the case for $\rhd_{p_\wedge}$.

$$\dfrac{\dfrac{\overset{\varphi_l}{\Pi \vdash \Lambda, F_1[B]_{\pi_1}, F_2[C]_{\pi_2}}}{\Pi \vdash \Lambda, F_1[B]_{\pi_1} \vee F_2[C]_{\pi_2}} \vee_r \qquad \dfrac{\overset{\varphi_1}{F_1[B]_{\pi_1}, \Gamma_1 \vdash \Delta_1} \qquad \overset{\varphi_2}{F_2[C]_{\pi_2}, \Gamma_2 \vdash \Delta_2}}{F_1[B]_{\pi_1} \vee F_2[C]_{\pi_2}, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2} \vee_l}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda} cut$$

$$\Downarrow$$

$$\dfrac{\dfrac{\overset{\varphi_l}{\Pi \vdash \Lambda, F_1[B]_{\pi_1}, F_2[C]_{\pi_2}} \qquad \overset{\varphi_2}{F_2[C]_{\pi_2}, \Gamma_2 \vdash \Delta_2}}{\Pi, \Gamma_2 \vdash \Delta_2, \Lambda, F_1[B]_{\pi_1}} cut \qquad \overset{\varphi_1}{F_1[B]_{\pi_1}, \Gamma_1 \vdash \Delta_1}}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda} cut$$

– $\varphi^k \rhd_{p_\rightarrow} \varphi^{k+1}$: Analogous to the case for $\rhd_{p_\wedge}$.

$$\dfrac{\dfrac{\overset{\varphi_l}{F_1[B]_{\pi_1}, \Pi \vdash \Lambda, F_2[C]_{\pi_2}}}{\Pi \vdash \Lambda, F_1[B]_{\pi_1} \rightarrow F_2[C]_{\pi_2}} \rightarrow_r \qquad \dfrac{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1, F_1[B]_{\pi_1}} \qquad \overset{\varphi_2}{F_2[C]_{\pi_2}, \Gamma_2 \vdash \Delta_2}}{F_1[B]_{\pi_1} \rightarrow F_2[C]_{\pi_2}, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2} \rightarrow_l}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda} cut$$

$$\Downarrow$$

$$\dfrac{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1, F_1[B]_{\pi_1}} \quad \dfrac{\overset{\varphi_l}{F_1[B]_{\pi_1}, \Pi \vdash \Lambda, F_2[C]_{\pi_2}} \quad \overset{\varphi_2}{F_2[C]_{\pi_2}, \Gamma_2 \vdash \Delta_2}}{F_1[B]_{\pi_1}, \Pi, \Gamma_2 \vdash \Delta_2, \Lambda}\, cut}{\Gamma_1, \Gamma_2, \Pi \vdash \Delta_1, \Delta_2, \Lambda}\, cut$$

– $\varphi^k \rhd_{p_\neg} \varphi^{k+1}$: Analogous to the case for $\rhd_{p_\wedge}$.

$$\dfrac{\dfrac{\overset{\varphi_l}{F[A]_\pi, \Gamma \vdash \Delta}}{\Gamma \vdash \Delta, \neg F[A]_\pi}\, \neg_r \quad \dfrac{\overset{\varphi_r}{\Pi \vdash \Lambda, F[A]_\pi}}{\neg F[A]_\pi, \Pi \vdash \Lambda}\, \neg_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut$$

$$\Downarrow$$

$$\dfrac{\overset{\varphi_r}{\Pi \vdash \Lambda, F[A]_\pi} \quad \overset{\varphi_l}{F[A]_\pi, \Gamma \vdash \Delta}}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut$$

- $\varphi^k \rhd_q \varphi^{k+1}$: Similar to the case for $\rhd_p$. It can also be subdivided in two cases, according to which quantifier is eliminated:

  – $\varphi^k \rhd_{q_\forall} \varphi^{k+1}$: Notice that in the cut of φ^{k+1} that has been changed by the rewriting, for any position π, the occurrences of the atomic formula A occupying that position in both auxiliary occurrences of the cut have the same atomic cut-linkage annotations (the same color red). Therefore, for any cut ρ of φ^{k+1} with auxiliary occurrences ν_l and ν_r and for any position π, $cutlink_a(\ulcorner \nu_l \urcorner^\pi, \varphi) = cutlink_a(\ulcorner \nu_r \urcorner^\pi, \varphi)$.

$$\dfrac{\dfrac{\overset{\varphi_l}{\Gamma \vdash \Delta, F[A]_\pi\{x \leftarrow \alpha\}}}{\Gamma \vdash \Delta, (\forall x)F[A]_\pi}\, \forall_r \quad \dfrac{\overset{\varphi_r}{F[A]_\pi\{x \leftarrow t\}, \Pi \vdash \Lambda}}{(\forall x)F[A]_\pi, \Pi \vdash \Lambda}\, \forall_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut$$

$$\Downarrow$$

$$\dfrac{\overset{\varphi_l\{\alpha \leftarrow t\}}{\Gamma \vdash \Delta, F[A]_\pi\{x \leftarrow t\}} \quad \overset{\varphi_r}{F[A]_\pi\{x \leftarrow t\}, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut$$

  – $\varphi^k \rhd_{q_\exists} \varphi^{k+1}$: Analogous to the case for $\rhd_{q_\forall}$.

$$\dfrac{\dfrac{\overset{\varphi_l}{\Gamma \vdash \Delta, F[A]_\pi\{x \leftarrow t\}}}{\Gamma \vdash \Delta, \exists x F[A]_\pi}\, \exists_r \quad \dfrac{\overset{\varphi_r}{F[A]_\pi\{x \leftarrow \alpha\}, \Pi \vdash \Lambda}}{\exists x F[A]_\pi, \Pi \vdash \Lambda}\, \exists_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut$$

$$\Downarrow$$

$$\dfrac{\begin{array}{c}\varphi_l\\ \Gamma \vdash \Delta, F[A]_\pi\{x \leftarrow t\}\end{array} \qquad \begin{array}{c}\varphi_r\{\alpha \leftarrow t\}\\ F[A]_\pi\{x \leftarrow t\}, \Pi \vdash \Lambda\end{array}}{\Gamma, \Pi \vdash \Delta, \Lambda}\ cut$$

- $\varphi^k \rhd_r \varphi^{k+1}$: This is the simplest case. Notice that in the cut of φ^{k+1} that has been shifted upward by the rewriting, for any position π, the occurrences of the atomic formula A occupying that position in both auxiliary occurrences of the cut have the same atomic cut-linkage annotations (the same color red). Therefore, for any cut ρ of φ^{k+1} with auxiliary occurrences ν_l and ν_r and for any position π, $cutlink_a(\ulcorner \nu_l \urcorner^\pi, \varphi) = cutlink_a(\ulcorner \nu_r \urcorner^\pi, \varphi)$.

$$\dfrac{\begin{array}{c}\varphi_l\\ \Gamma \vdash \Delta, F[A]_\pi\end{array} \qquad \dfrac{\begin{array}{c}\varphi_r\\ F[A]_\pi, \Pi' \vdash \Lambda'\end{array}}{F[A]_\pi, \Pi \vdash \Lambda}\ \rho}{\Gamma, \Pi \vdash \Delta, \Lambda}\ cut$$

$$\Downarrow$$

$$\dfrac{\dfrac{\begin{array}{c}\varphi_l\\ \Gamma \vdash \Delta, F[A]_\pi\end{array} \qquad \begin{array}{c}\varphi_r\\ F[A]_\pi, \Pi' \vdash \Lambda'\end{array}}{\Gamma, \Pi' \vdash \Delta, \Lambda'}\ cut}{\Gamma, \Pi \vdash \Delta, \Lambda}\ \rho$$

$$\dfrac{\dfrac{\begin{array}{c}\varphi_l\\ \Gamma' \vdash \Delta', F[A]_\pi\end{array}}{\Gamma \vdash \Delta, F[A]_\pi}\ \rho \qquad \begin{array}{c}\varphi_r\\ F[A]_\pi, \Pi \vdash \Lambda\end{array}}{\Gamma, \Pi \vdash \Delta, \Lambda}\ cut$$

$$\Downarrow$$

$$\dfrac{\dfrac{\begin{array}{c}\varphi_l\\ \Gamma' \vdash \Delta', F[A]_\pi\end{array} \qquad \begin{array}{c}\varphi_r\\ F[A]_\pi, \Pi \vdash \Lambda\end{array}}{\Gamma', \Pi \vdash \Delta', \Lambda}\ cut}{\Gamma, \Pi \vdash \Delta, \Lambda}\ \rho$$

$$\dfrac{\overset{\varphi_l}{\Pi \vdash \Lambda, F[A]_\pi} \qquad \dfrac{\overset{\varphi_1}{F[A]_\pi, \Gamma_1 \vdash \Delta_1} \qquad \overset{\varphi_2}{\Gamma_2 \vdash \Delta_2}}{F[A]_\pi, \Gamma \vdash \Delta}\,\rho}{\Pi, \Gamma \vdash \Lambda, \Delta}\,cut$$

$$\Downarrow$$

$$\dfrac{\dfrac{\overset{\varphi_l}{\Pi \vdash \Lambda, F[A]_\pi} \qquad \overset{\varphi_1}{F[A]_\pi, \Gamma_1 \vdash \Delta_1}}{\Pi, \Gamma_1 \vdash \Lambda, \Delta_1}\,cut \qquad \overset{\varphi_2}{\Gamma_2 \vdash \Delta_2}}{\Pi, \Gamma \vdash \Lambda, \Delta}\,\rho$$

$$\dfrac{\overset{\varphi_l}{\Pi \vdash \Lambda, F[A]_\pi} \qquad \dfrac{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1} \qquad \overset{\varphi_2}{F[A]_\pi, \Gamma_2 \vdash \Delta_2}}{F[A]_\pi, \Gamma \vdash \Delta}\,\rho}{\Pi, \Gamma \vdash \Lambda, \Delta}\,cut$$

$$\Downarrow$$

$$\dfrac{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1} \qquad \dfrac{\overset{\varphi_l}{\Pi \vdash \Lambda, F[A]_\pi} \qquad \overset{\varphi_2}{F[A]_\pi, \Gamma_2 \vdash \Delta_2}}{\Pi, \Gamma_2 \vdash \Lambda, \Delta_2}\,cut}{\Pi, \Gamma \vdash \Lambda, \Delta}\,\rho$$

$$\dfrac{\dfrac{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1, F[A]_\pi} \qquad \overset{\varphi_2}{\Gamma_2 \vdash \Delta_2}}{\Gamma \vdash \Delta, F[A]_\pi}\,\rho \qquad \overset{\varphi_r}{F[A]_\pi, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut$$

$$\Downarrow$$

$$\dfrac{\dfrac{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1, F[A]_\pi} \qquad \overset{\varphi_r}{F[A]_\pi, \Pi \vdash \Lambda}}{\Gamma_1, \Pi \vdash \Delta_1, \Lambda}\,cut \qquad \overset{\varphi_2}{\Gamma_2 \vdash \Delta_2}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,\rho$$

$$\dfrac{\dfrac{\overset{\varphi_1}{\Gamma_1 \vdash \Delta_1} \qquad \overset{\varphi_2}{\Gamma_2 \vdash \Delta_2, F[A]_\pi}}{\Gamma \vdash \Delta, F[A]_\pi}\,\rho \qquad \overset{\varphi_r}{F[A]_\pi, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\,cut$$

$$\Downarrow$$

$$\dfrac{\begin{array}{c}\varphi_1\\ \Gamma_1 \vdash \Delta_1\end{array} \qquad \dfrac{\begin{array}{c}\varphi_2\\ \Gamma_2 \vdash \Delta_2, F[A]_\pi\end{array} \qquad \begin{array}{c}\varphi_r\\ F[A]_\pi, \Pi \vdash \Lambda\end{array}}{\Gamma_2, \Pi \vdash \Delta_2, \Lambda}\, cut}{\Gamma, \Pi \vdash \Delta, \Lambda}\, \rho$$

- $\varphi^k \rhd_c \varphi^{k+1}$: Notice that in the two new cuts of φ^{k+1} that have been created by the rewriting of φ^k to φ^{k+1}, for any position π, the occurrences of the atomic formula A occupying that position in all auxiliary occurrences of the cuts have the same atomic cut-linkage annotations (the same color red). This is so because atomic cut-linkage annotations persist, being distributed to all copies of an occurrence when it is duplicated (i.e. when $\rhd_c$ duplicates some occurrences, all copies of an occurrence are assumed to retain the same cut-linkage annotations as the original occurrence). Therefore, for any cut ρ of φ^{k+1} with auxiliary occurrences ν_l and ν_r and for any position π, $cutlink_a(\ulcorner \nu_l \urcorner^\pi, \varphi) = cutlink_a(\ulcorner \nu_r \urcorner^\pi, \varphi)$.

$$\dfrac{\dfrac{\begin{array}{c}\varphi_l\\ \Gamma \vdash \Delta, F[A]_\pi, F[A]_\pi\end{array}}{\Gamma \vdash \Delta, F[A]_\pi}\, c_r \qquad \begin{array}{c}\varphi_r\\ F[A]_\pi, \Pi \vdash \Lambda\end{array}}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut$$

$$\Downarrow$$

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\varphi_l\\ \Gamma \vdash \Delta, F[A]_\pi, F[A]_\pi\end{array} \qquad \begin{array}{c}\varphi_r\\ F[A]_\pi, \Pi \vdash \Lambda\end{array}}{\Gamma, \Pi \vdash \Delta, \Lambda, F[A]_\pi}\, cut \qquad \begin{array}{c}\varphi_r'\\ F[A]_\pi, \Pi \vdash \Lambda\end{array}}{\Gamma, \Pi, \Pi \vdash \Delta, \Lambda, \Lambda,}\, cut}{\Gamma, \Pi \vdash \Delta, \Lambda}\, c_l^*, c_r^*$$

$$\dfrac{\begin{array}{c}\varphi_l\\ \Gamma \vdash \Delta, F[A]_\pi\end{array} \qquad \dfrac{\begin{array}{c}\varphi_r\\ F[A]_\pi, F[A]_\pi, \Pi \vdash \Lambda\end{array}}{F[A]_\pi, \Pi \vdash \Lambda}\, c_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut$$

$$\Downarrow$$

$$\dfrac{\dfrac{\varphi_l'}{\Gamma \vdash \Delta, F[A]_\pi} \quad \dfrac{\dfrac{\varphi_l}{\Gamma \vdash \Delta, F[A]_\pi} \quad \dfrac{\varphi_r}{F[A]_\pi, F[A]_\pi, \Pi \vdash \Lambda}}{F[A]_\pi, \Gamma, \Pi \vdash \Delta, \Lambda}\, cut}{\dfrac{\Gamma, \Gamma, \Pi \vdash \Delta, \Delta, \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda}\, c_l^*, c_r^*}\, cut$$

- $\varphi^k \rhd_w \varphi^{k+1}$: When φ^k is rewritten to φ^{k+1} via $\rhd_w$, the only thing that happens is the elimination of one cut. All cuts the remain in φ^{k+1} are left unchanged. Therefore, for any cut ρ of φ^{k+1} with auxiliary occurrences ν_l and ν_r and for any position π, $cutlink_a(\ulcorner \nu_l \urcorner^\pi, \varphi) = cutlink_a(\ulcorner \nu_r \urcorner^\pi, \varphi)$.

$$\dfrac{\dfrac{\dfrac{\varphi_l}{\Gamma \vdash \Delta}}{\Gamma \vdash \Delta, F[A]_\pi}\, w_r \quad \dfrac{\varphi_r}{F[A]_\pi, \Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut$$

$$\Downarrow$$

$$\dfrac{\dfrac{\varphi_l}{\Gamma \vdash \Delta}}{\Gamma, \Pi \vdash \Delta, \Lambda}\, w_r^*, w_l^*$$

$$\dfrac{\dfrac{\varphi_l}{\Gamma \vdash \Delta, F[A]_\pi} \quad \dfrac{\dfrac{\varphi_r}{\Pi \vdash \Lambda}}{F[A]_\pi, \Pi \vdash \Lambda}\, w_l}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut$$

$$\Downarrow$$

$$\dfrac{\dfrac{\varphi_r}{\Pi \vdash \Lambda}}{\Gamma, \Pi \vdash \Delta, \Lambda}\, w_r^*, w_l^*$$

- $\varphi^k \rhd_a \varphi^{k+1}$: Analogous to the case for $\rhd_w$.

$$\dfrac{A \vdash A \quad \dfrac{\varphi_r}{A, \Pi \vdash \Lambda}}{A, \Pi \vdash \Lambda}\, cut$$

$$\Downarrow$$

$$\begin{array}{c} \varphi_r \\ A, \Pi \vdash \Lambda \end{array}$$

$$\dfrac{\begin{array}{c}\varphi_l \\ \Gamma \vdash \Delta, {\color{red}A}\end{array} \qquad {\color{red}A} \vdash A}{\Gamma, \vdash \Delta, A}\ cut$$

$$\Downarrow$$

$$\begin{array}{c} \varphi_l \\ \Gamma \vdash \Delta, A \end{array}$$

$\square$

Theorem 6.4 (Cut-Linkage Preservation under Reductive Cut-Elimination)**.** Let φ be a proof with cuts and φ^* a proof such that $\varphi \rhd^* \varphi^*$. Let ρ be a cut in φ^* with auxiliary occurrences ν_l and ν_r. Then, for any position π, $cutlink(\ulcorner \nu_l \urcorner^\pi, \varphi) = cutlink(\ulcorner \nu_r \urcorner^\pi, \varphi)$.

Proof. This theorem follows from Theorem 6.3 and from noting that atomic cut-linkage is a sub-relation of cut-linkage. $\square$

6.2.3 Cut-Side Preservation under Reductive Cut-Elimination

A similar preservation result can be proved for cut-side annotations, as shown in Theorem 6.5.

Theorem 6.5 (Cut-Side Preservation under Reductive Cut-Elimination)**.** Let φ be a proof with cuts and φ^* a proof such that $\varphi \rhd^* \varphi^*$. Let ρ be a cut in φ^* with auxiliary occurrences ν_l and ν_r. Then, for any position π, $cutside(\ulcorner \nu_l \urcorner^\pi, \varphi) \neq cutside(\ulcorner \nu_r \urcorner^\pi, \varphi)$.

Proof. Analogous to the proof of Theorem 6.3. $\square$

6.2.4 Using Cut-Linkage

The refinement described in this subsection (Definition 6.2.7) uses only cut-linkage annotations and yet in a very loose way. Nevertheless, it is already strict enough to prevent certain refutations, as shown in Examples 6.4 and 6.5.

Example 6.4. *Let φ be the proof of Example 6.2. Then the clause set of φ is:*

$$C_\varphi \equiv \{\underbrace{\vdash P(\alpha)}_{c_1}; \underbrace{P(s) \vdash P(s)}_{c_2}; \underbrace{P(s) \vdash P(s)}_{c_3}; \underbrace{P(s) \vdash}_{c_4}\}$$

Let δ be the following refutation of C_φ:

$$\frac{c_1 \qquad c_4}{\vdash}\ r$$

There is no $\triangleright_{\overline{a}}$-normal-form φ' of φ such that $CR(\varphi') = \delta$ (this fact can be easily verified by performing reductive cut-elimination on φ or by noting that otherwise this would lead to a contradiction of Theorem 6.4).

Definition 6.2.7 (Refined Resolution using Cut-Linkage)**.** The *inference rules* of the resolution calculus refined by using cut-linkage $\mathbf{R_{cl}}$ are the same as the rules of the unrestricted resolution calculus, except for an additional restriction on the resolution rule:

- **Resolution Rule** (restricted by using cut-linkage):

$$\frac{\Gamma_1 \vdash \Delta_1, A_1 \qquad A_2, \Gamma_2 \vdash \Delta_2}{\Gamma_1\sigma\eta, \Gamma_2\sigma\eta \vdash \Delta_1\sigma\eta, \Delta_2\sigma\eta}\ r(\sigma)$$

  only if there exist formula occurrences $F_1 \in \Gamma_1 \cup \Delta_1 \cup \{A_1\}$ and $F_2 \in \Gamma_2 \cup \Delta_2 \cup \{A_2\}$ such that $cutlink(F_1) \cap cutlink(F_2) \neq \emptyset$.

  where σ is a most general unifier of A_1 and A_2 and η is a substitution that renames all variables to globally new fresh ones.

Example 6.5 (Refined Resolution using Cut-Linkage)**.** *The* **R***-refutation δ shown in Example 6.4 is not a* $\mathbf{R_{cl}}$*-refutation, because c_1 and c_4 do not have any literals that have the same cut-linkage annotations (i.e. the same colors). There are four minimal* $\mathbf{R_{cl}}$*-refutations of C_φ, which are shown below:*

$$\frac{\dfrac{c_1 \qquad c_2}{\vdash P(s)}\ r \qquad c_4}{\vdash}\ r$$

$$\frac{\dfrac{c_1 \qquad c_3}{\vdash P(s)}\ r \qquad c_4}{\vdash}\ r$$

$$\frac{c_1 \qquad \dfrac{c_2 \qquad c_4}{P(s) \vdash}\ r}{\vdash}\ r$$

$$\cfrac{c_1 \qquad \cfrac{c_3 \qquad c_4}{P(s) \vdash}\, r}{\vdash}\, r$$

Infinitely many non-minimal $\mathbf{R}_{\mathbf{cl}}$*-refutations of* C_φ *can be obtained by resolving* c_2 *with* c_3 *many times. An example of such a non-minimal refutation is shown in Example 6.6.*

6.2.5 Using Cut-Linkage (More Strictly)

The refinement described in this subsection also uses only cut-linkage annotations, but it does so in a more strict way than the refinement described in the previous subsection. Here it does not suffice for two clauses to contain literals having equal cut-linkage annotations, but the resolved literals themselves must have equal cut-linkage annotations.

Example 6.6. *Consider the proof* φ *of Example 6.4 and let* δ^* *be the following* $\mathbf{R}_{\mathbf{cl}}$*-refutation of* C_φ*:*

$$\cfrac{\cfrac{\vdash P(\alpha) \qquad \cfrac{P(s) \vdash P(s) \qquad P(s) \vdash P(s)}{P(s) \vdash P(s)}\, r}{\vdash P(s)}\, r \qquad P(s) \vdash}{\vdash}\, r$$

There is no $\rhd_{\overline{a}}$*-normal-form* φ' *of* φ *such that* $CR(\varphi') = \delta^*$*. Otherwise, Theorem 6.4 would be contradicted. In any case, this fact can also be seen by analyzing the cut reduction at the moment when the lowermost cut has been shifted up to the* $\rightarrow_l$ *and* $\rightarrow_r$ *inferences of* φ*. At this moment, the proof has the following form* φ_1*, where* ψ *is a partial proof:*

$$\cfrac{\cfrac{\cfrac{P(s) \vdash P(s)}{P(s), \neg P(s) \vdash P(\alpha)}\, w_r}{P(s), \vdash \neg P(s) \rightarrow P(s)}\, \rightarrow_r \qquad \cfrac{\cfrac{P(s) \vdash P(s)}{\vdash \neg P(s), P(s)}\, \neg_r \qquad P(s) \vdash P(s)}{\neg P(s) \rightarrow P(s) \vdash P(s), P(s)}\, \rightarrow_l}{\psi}\, cut$$

Applying $\rhd_{p_\rightarrow}$ *yields the following proof* φ_2*:*

$$\cfrac{\cfrac{\cfrac{P(s) \vdash P(s)}{\vdash \neg P(s), P(s)}\, \neg_r \qquad \cfrac{P(s) \vdash P(s)}{P(s), \neg P(s) \vdash P(s)}\, w_r}{P(s), \vdash P(s), P(s)}\, cut \qquad P(s) \vdash P(s)}{\psi}\, cut$$

The reduction displayed above shows that the elimination of the cut with cut-formula $\neg P(s) \rightarrow P(s)$ creates two new cuts such that their cut-formula occurrences have all equal cut-linkage annotations (colors). Therefore, in $CR(\varphi')$, resolved literals necessarily have the same cut-linkage annotation. This is not the case with δ^*, because the first resolution inference resolves two literals with different cut-linkage annotations.

Definition 6.2.8 (Refined Resolution using Cut-Linkage (More Strictly))**.** The *inference rules* of the resolution calculus refined by using cut-linkage more strictly, $\mathbf{R_{cls}}$, are the rules of the unrestricted resolution calculus with the following additional restrictions on the resolution and factoring rules:

- **Resolution Rule** (using cut-linkage more strictly):

$$\frac{\Gamma_1 \vdash \Delta_1, A_1 \qquad A_2, \Gamma_2 \vdash \Delta_2}{\Gamma_1\sigma\eta, \Gamma_2\sigma\eta \vdash \Delta_1\sigma\eta, \Delta_2\sigma\eta}\ r(\sigma)$$

  only if $cutlink(A_1) \cap cutlink(A_2) \neq \emptyset$.

  where σ is a most general unifier of A_1 and A_2 and η is a substitution that renames all variables to globally new fresh ones.

- **Factoring Rules**:

$$\frac{A, A', \Gamma \vdash \Delta}{A\sigma\eta, \Gamma\sigma\eta \vdash \Delta\sigma\eta}\ f_l(\sigma) \qquad\qquad \frac{\Gamma \vdash \Delta, A, A'}{\Gamma\sigma\eta \vdash \Delta\sigma\eta, A\sigma\eta}\ f_r(\sigma)$$

  and then $cutlink(A\sigma\eta) \doteq cutlink(A) \cup cutlink(A')$.

  where σ is a most general unifier of A and A' and η is a substitution that renames all variables to globally new fresh ones.

6.2.6 Using Cut-Side and Cut-Linkage

Example 6.7 shows that, even when $\mathbf{R_{cls}}$ is used, CERes could still produce normal forms not obtainable by reductive methods modulo CR-equivalence, because $\mathbf{R_{cls}}$ still allows the resolution of literals having equal cut-side annotations. Such cases can be avoided by refining the resolution calculus in a way that uses cut-side annotations as well, as described in Definition 6.2.9

Example 6.7. *Let φ be the following annotated proof:*

$$
\dfrac{
\dfrac{\dfrac{\dfrac{\dfrac{P(\alpha) \vdash P(\alpha)}{P(\alpha) \vdash \neg P(\alpha), P(\alpha)}\, w_r}{P(\alpha) \vdash \neg P(\alpha) \lor P(\alpha)}\, \lor_r}{(\forall x)P(x) \vdash \neg P(\alpha) \lor P(\alpha)}\, \forall_l}{(\forall x)P(x) \vdash (\forall x)(\neg P(x) \lor P(x))}\, \forall_r
\qquad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{P(t) \vdash P(t)}{\neg P(t), P(t) \vdash}\, \neg_l}{\neg P(t) \vdash \neg P(t)}\, \neg_r \quad P(t) \vdash P(t)}{\neg P(t) \lor P(t) \vdash P(t), \neg P(t)}\, \lor_l}{\neg P(t) \lor P(t) \vdash P(t) \lor \neg P(t)}\, \lor_r}{(\forall x)(\neg P(x) \lor P(x)) \vdash P(t) \lor \neg P(t)}\, \forall_l
}{(\forall x)P(x) \vdash P(t) \lor \neg P(t)}\, cut
$$

Then the clause set of φ is:

$$C_\varphi \equiv \{\underbrace{\vdash P(\alpha)}_{c_1}; \underbrace{\vdash P(t)}_{c_2}; \underbrace{P(t) \vdash}_{c_3}\}$$

A possible $\mathbf{R_{cls}}$-refutation of C_φ is shown below:

$$\dfrac{c_2 \qquad c_3}{\vdash}\, r$$

On the other hand, any $\rhd_{\overline{a}}$-normal form ψ such that $\varphi \rhd^{\downarrow}_{\overline{a}} \psi$ is such that $CR(\psi)$ is the following:

$$\dfrac{c_1\{\alpha \mapsto t\} \qquad c_3}{\vdash}\, r$$

Definition 6.2.9 (Refined Resolution using Cut-Linkage (Strictly) and Cut-Sides)**.** The *inference rules* of the resolution calculus refined by using cut-linkage and cut-sides, $\mathbf{R_{scl}}$, are the rules of the unrestricted resolution calculus with the following additional restrictions on the resolution and factoring rules:

- **Resolution Rule** (using cut-side and cut-linkage):

  $$\dfrac{\Gamma_1 \vdash \Delta_1, A_1 \qquad A_2, \Gamma_2 \vdash \Delta_2}{\Gamma_1\sigma\eta, \Gamma_2\sigma\eta \vdash \Delta_1\sigma\eta, \Delta_2\sigma\eta}\, r(\sigma)$$

  only if $cutlink(A_1) \cap cutlink(A_2) \neq \emptyset$ and $cutside(A_1) \cap cutside(A_2) = \emptyset$.

  where σ is a most general unifier of A_1 and A_2 and η is a substitution that renames all variables to globally new fresh ones.

- **Factoring Rules** (using cut-side and cut-linkage):

  $$\dfrac{A, A', \Gamma \vdash \Delta}{A\sigma\eta, \Gamma\sigma\eta \vdash \Delta\sigma\eta}\, f_l(\sigma) \qquad\qquad \dfrac{\Gamma \vdash \Delta, A, A'}{\Gamma\sigma\eta \vdash \Delta\sigma\eta, A\sigma\eta}\, f_r(\sigma)$$

only if $cutside(A_1) \cap cutside(A_2) = \emptyset$. And then $cutlink(A\sigma\eta) \doteq cutlink(A') \cup cutlink(A')$.

where σ is a most general unifier of A and A' and η is a substitution that renames all variables to globally new fresh ones.

6.2.7 Using Atomic Cut-Linkage

The most restrictive refinement studied in this chapter is obtained when atomic cut-linkage is used. More precisely, resolved literals are required to have equal atomic cut-linkage annotations, as described in Definition 6.2.10. It is capable of preventing refutations such as the one shown in Example 6.8.

Example 6.8. *Let φ be the proof below:*

$$
\begin{prooftree}
\AxiomC{$P(\alpha) \vdash P(\alpha)^l$}
\RightLabel{$\forall_l$}
\UnaryInfC{$(\forall z)P(z) \vdash P(\alpha)^l$}
\RightLabel{$\forall_r$}
\UnaryInfC{$(\forall z)P(z) \vdash (\forall x)P(x)^l$}
\RightLabel{w_r}
\UnaryInfC{$(\forall z)P(z) \vdash (\forall x)P(x)^l, (\forall y)P(y)^l$}
\RightLabel{$\vee_r$}
\UnaryInfC{$(\forall z)P(z) \vdash (\forall x)P(x)^l \vee (\forall y)P(y)^l$}
\AxiomC{$P(t)^r \vdash P(t)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$P(t)^r \vdash (\exists w)P(w)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$(\forall x)P(x)^r \vdash (\exists w)P(w)$}
\AxiomC{$P(s)^r \vdash P(s)$}
\RightLabel{$\exists_r$}
\UnaryInfC{$P(s)^r \vdash (\exists w)P(w)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$(\forall y)P(y)^r \vdash (\exists w)P(w)$}
\RightLabel{$\vee_l$}
\BinaryInfC{$(\forall x)P(x)^r \vee (\forall y)P(y)^r \vdash (\exists w)P(w), (\exists w)P(w)$}
\RightLabel{c_r}
\UnaryInfC{$(\forall x)P(x)^r \vee (\forall y)P(y)^r \vdash (\exists w)P(w)$}
\RightLabel{cut}
\BinaryInfC{$(\forall z)P(z) \vdash (\exists w)P(w)$}
\end{prooftree}
$$

Its clause set is:

$$C_\varphi \equiv \{\underbrace{\vdash P(\alpha)^l}_{c_1}; \underbrace{P(t)^r \vdash}_{c_2}; \underbrace{P(s)^r \vdash}_{c_3}\}$$

Let δ be the following $\mathbf{R_{scl}}$-refutation of C_φ:

$$\frac{c_1 \qquad c_3}{\vdash}\, r$$

The refutation δ cannot be the canonic refutation of any normal form φ^* obtained by reductive cut-elimination methods. This can be seen by an analysis of the first cut-reduction, on $\vee$, which matches each subformula of the left cut-occurrence with its corresponding subformula in the right cut-occurrence. To obtain δ as a canonic refutation, on the other hand, it would be necessary to match the left subformula of the left cut-occurrence with the right subformula of the right cut-occurrence when doing the cut-reduction on $\vee$. Indeed, the only canonic refutation for a reductive normal form φ^* is the following δ^*:

$$\frac{c_1\{\alpha \leftarrow t\} \qquad c_2}{\vdash}\, r$$

Definition 6.2.10 (Refined Resolution using Atomic Cut-Linkage)**.** The *inference rules* of the resolution calculus refined by using atomic cut-linkage, $\mathbf{R_{acl}}$, are the rules of the unrestricted resolution calculus with the following additional restrictions on the resolution and factoring rules:

- **Resolution Rule** (using cut-linkage more strictly):

$$\frac{\Gamma_1 \vdash \Delta_1, A_1 \qquad A_2, \Gamma_2 \vdash \Delta_2}{\Gamma_1\sigma\eta, \Gamma_2\sigma\eta \vdash \Delta_1\sigma\eta, \Delta_2\sigma\eta}\, r(\sigma)$$

  only if $\mathit{cutlink}_a(A_1) \cap \mathit{cutlink}_a(A_2) \neq \emptyset$.

  where σ is a most general unifier of A_1 and A_2 and η is a substitution that renames all variables to globally new fresh ones.

- **Factoring Rules**:

$$\frac{A, A', \Gamma \vdash \Delta}{A\sigma\eta, \Gamma\sigma\eta \vdash \Delta\sigma\eta}\, f_l(\sigma) \qquad\qquad \frac{\Gamma \vdash \Delta, A, A'}{\Gamma\sigma\eta \vdash \Delta\sigma\eta, A\sigma\eta}\, f_r(\sigma)$$

  and then $\mathit{cutlink}_a(A\sigma\eta) \doteq \mathit{cutlink}_a(A') \cup \mathit{cutlink}_a(A)$.

  where σ is a most general unifier of A and A' and η is a substitution that renames all variables to globally new fresh ones.

6.2.8 Refined Refutability

Sections 6.2.4, 6.2.5, 6.2.6, 6.2.7 have shown how cut-linkage and cut-side annotations and refined resolution calculi that use these annotations can be used to prevent certain kinds of refutations. A question that naturally arises from this approach is whether clause sets remain refutable if these refined resolution calculi are used.

For unrestricted resolution, the refutability is a consequence of the unsatisfiability of clause sets (Theorems 4.1 and 4.5) and the refutational completeness of unrestricted resolution. However, refined resolution calculi are generally not refutationally complete, and hence the refutability of clause sets must be proved by other means. In particular, the proofs shown here are constructive; they actually show how to construct a resolution proof for a clause set.

Theorem 6.6 shows that at least one of the swapped clause sets of a proof is indeed refutable by the most refined resolution calculus $\mathbf{R_{acl}}$. Theorem 6.7 shows that, in fact, any swapped clause set is $\mathbf{R_{acl}}$-refutable. This result can be transfered to less refined calculi by using Lemma 6.1.

Although these theorems hold for swapped clause sets, it is conjectured the similar results could also be proved for standard clause sets and profile clause sets. The proofs, however, would be more complicated because standard clause sets and profile clause sets are not as invariant as swapped clause sets.

Theorem 6.6 ($\mathbf{R_{acl}}$-Refutability of a Swapped Clause Set)**.** For any proof φ, there exists a $\mathbf{R_{acl}}$-refutation of a swapped clause set $C^W_{\varphi|S^*}$ of φ with respect to a $\rightsquigarrow_{\oplus\otimes_W}$-normal-form S^* of $\mathcal{S}_\varphi$.

Proof. Let φ' be such that $\varphi \rhd^{\downarrow}_{\bar{a}} \varphi'$. By Definitions 5.1.4 and 5.1.5 and Theorem 5.1, there exists $\delta \in S_{CR}(\varphi')$ such that δ is a $\mathbf{R}$-refutation of $C^W_{\varphi'|S'}$ where S' is a $\rightsquigarrow_{\oplus\otimes_W}$-normal-form of $\mathcal{S}_{\varphi'}$. By Lemma 5.9, there exists a $\rightsquigarrow_{\oplus\otimes_W}$-normal-form S^* of $\mathcal{S}_\varphi$ such that $S^* \preccurlyeq S'$. By Lemma 5.10, δ is also a refutation of $C^W_{\varphi|S^*}$. By Theorem 6.3, the resolved literals of any resolution inference ρ of δ must have a non-empty intersection of atomic cut-linkage annotations. Therefore, δ is a $\mathbf{R_{acl}}$-refutation of $C^W_{\varphi|S^*}$. □

Theorem 6.7 ($\mathbf{R_{acl}}$-Refutability of any Swapped Clause Set)**.** For any proof φ, there exists a $\mathbf{R_{acl}}$-refutation of the swapped clause set $C^W_{\varphi|S^*}$ of φ with respect to any $\rightsquigarrow_{\oplus\otimes_W}$-normal-form S^* of $\mathcal{S}_\varphi$.

Proof. By Lemma 4.9 there exists a proof φ^* such that $\mathcal{S}_{\varphi^*} = S^*$. Let φ' be such that $\varphi^* \rhd^{\downarrow}_{\bar{a}} \varphi'$. Since φ^* is in $\gg_{\oplus\otimes}$-normal-form and $\rhd_{\bar{a}}$ does not introduce any degenerate inference, φ' has only one $\gg_{\oplus\otimes}$-normal-form. Let φ'' be the $\gg_{\oplus\otimes}$-normal-form of φ'. Let δ be the canonic refutation from φ''. By Definition 5.1.5 and Theorem 5.1 δ is a $\mathbf{R}$-refutation of $C^W_{\varphi'|S'}$ where S' is the $\rightsquigarrow_{\oplus\otimes_W}$-normal-form of $\mathcal{S}_{\varphi'}$. By Lemma 5.9, there exists a $\rightsquigarrow_{\oplus\otimes_W}$-normal-form S of $\mathcal{S}_{\varphi^*}$ such that $S \preccurlyeq S'$. Moreover, since φ^* is in $\gg_{\oplus\otimes}$-normal-form, there is only one $\rightsquigarrow_{\oplus\otimes_W}$-normal-form of $\mathcal{S}_{\varphi^*}$. Hence, $S = S^*$ and $S^* \preccurlyeq S'$. Therefore, by Lemma 5.10, δ is also a refutation of $\mathrm{cl}(S^*) \equiv C^W_{\varphi|S^*}$. By Theorem 6.3, the resolved literals of any resolution inference ρ of δ must have a non-empty intersection of atomic cut-linkage annotations. Therefore, δ is a $\mathbf{R_{acl}}$-refutation of $C^W_{\varphi|S^*}$. □

Lemma 6.1 (Restrictiveness of Refinements)**.** The following hold with respect to a refutation δ of a clause set C_φ of a proof φ enriched with atomic cut-linkage, cut-linkage and cut-side annotations:

- If δ is a $\mathbf{R_{acl}}$-refutation, then δ is a $\mathbf{R_{scl}}$-refutation.
- If δ is a $\mathbf{R_{scl}}$-refutation, then δ is a $\mathbf{R_{cls}}$-refutation.

- If δ is a $\mathbf{R_{cls}}$-refutation, then δ is a $\mathbf{R_{cl}}$-refutation.
- If δ is a $\mathbf{R_{cl}}$-refutation, then δ is a $\mathbf{R}$-refutation.

Proof. The last three items of the lemma are evident; they follow directly from the definitions of each refined resolution calculus. The first item, though, is slightly less evident and hence is shown below:

For simplicity, assume that δ contains only resolution inferences (the analysis for the general case containing factoring inferences is analogous). Assume, for the sake of contradiction, that δ is a $\mathbf{R_{acl}}$-refutation but not a $\mathbf{R_{scl}}$-refutation. Then, by the definition of $\mathbf{R_{acl}}$, for any resolution inference ρ with resolved literals A_1 and A_2, $cutlink_a(A_1) = cutlink_a(A_2)$. Consequently, $cutlink(A_1) = cutlink(A_2)$. Moreover, since δ is not a $\mathbf{R_{scl}}$-refutation, by the definition of $\mathbf{R_{scl}}$, there must be a resolution inference ρ' with resolved literals A'_1 and A'_2 such that $cutside(A'_1) = cutside(A'_2)$. And this, together with the fact that $cutlink_a(A'_1) = cutlink_a(A'_2)$, implies that A'_1 and A'_2 must have the same polarity. But this is a contradiction, because only literal of opposite polarity can be resolved. □

Theorem 6.8 ($\mathbf{R_{scl}}$-Refutability of any Swapped Clause Set)**.** For any proof φ, there exists a $\mathbf{R_{scl}}$-refutation of the swapped clause set $C^W_{\varphi|S^*}$ of φ with respect to any $\rightsquigarrow_{\oplus\otimes_W}$-normal-form S^* of $\mathcal{S}_\varphi$.

Proof. This theorem follows from Theorem 6.7 and Lemma 6.1 □

Theorem 6.9 ($\mathbf{R_{cls}}$-Refutability of any Swapped Clause Set)**.** For any proof φ, there exists a $\mathbf{R_{cls}}$-refutation of the swapped clause set $C^W_{\varphi|S^*}$ of φ with respect to any $\rightsquigarrow_{\oplus\otimes_W}$-normal-form S^* of $\mathcal{S}_\varphi$.

Proof. This theorem follows from Theorem 6.8 and Lemma 6.1 □

Theorem 6.10 ($\mathbf{R_{cl}}$-Refutability of any Swapped Clause Set)**.** For any proof φ, there exists a $\mathbf{R_{cl}}$-refutation of the swapped clause set $C^W_{\varphi|S^*}$ of φ with respect to any $\rightsquigarrow_{\oplus\otimes_W}$-normal-form S^* of $\mathcal{S}_\varphi$.

Proof. This theorem follows from Theorem 6.9 and Lemma 6.1 □

Theorem 6.11 ($\mathbf{R}$-Refutability of any Swapped Clause Set)**.** For any proof φ, there exists a $\mathbf{R}$-refutation of the swapped clause set $C^W_{\varphi|S^*}$ of φ with respect to any $\rightsquigarrow_{\oplus\otimes_W}$-normal-form S^* of $\mathcal{S}_\varphi$.

Proof. This theorem follows from Theorem 4.5 and the refutational completeness of $\mathbf{R}$. However, notice that such a proof does not construct a

R-refutation of the swapped clause set. It just proves that it exists. A proof according to the proof of Theorem 6.10 and using Lemma 6.1, on the other hand, does construct a **R**-refutation. □

6.2.9 CR-Simulation among Refinements

The refutability results established in Section 6.2.8 can be reformulated as CR-simulation results.

Theorem 6.12 (CR-Simulation among Refinements)**.** The following CR-simulation results hold:

- CERes with $\mathbf{R_{scl}}$ CR-simulates CERes with $\mathbf{R_{acl}}$.
- CERes with $\mathbf{R_{cls}}$ CR-simulates CERes with $\mathbf{R_{scl}}$.
- CERes with $\mathbf{R_{cl}}$ CR-simulates CERes with $\mathbf{R_{cls}}$.
- CERes CR-simulates CERes with $\mathbf{R_{cl}}$.

Proof. These results follow immediately from Lemma 6.1. □

Theorem 6.13 (CR-Simulation of Reductive Methods by Refined CERes)**.** CERes_W^O with $R \in \{\mathbf{R_{scl}}, \mathbf{R_{acl}}, \mathbf{R_{cls}}, \mathbf{R_{cl}}\}$ CR-simulates $\rhd^{\downarrow}_{\tilde{a}}$.

Proof. This is clear from an analysis of the proof of Theorem 6.7. □

Conjecture 6.1 (CR-Simulation of CERes with $\mathbf{R_{acl}}$ by Reductive Methods)**.** $\rhd^{\downarrow}_{\tilde{a}}$ CR-simulates CERes_W^O with $\mathbf{R_{acl}}$.

Proof. Proving or disproving this conjecture remains for future work. In case it is disproved, it would be interesting to define a more restricted refinement for which this conjecture could be proved. □

Chapter 7

Cut-Introduction

It is well-known that eliminating cuts frequently increases the size and length of proofs. In the worst case, cut-elimination can produce non-elementarily larger and longer proofs [109, 93]. This fact naturally leads to a desire to devise methods that could introduce cuts and compress proofs. However, this has been a notoriously difficult task. Indeed, the problem of answering, given a proof φ and a number l such that $l \leq length(\varphi)$, whether there is a proof ψ such that $length(\psi) < l$ is known to be undecidable [14]. Nevertheless, a lower bound for compressibility based on specific cut-introduction methods that are inverse of reductive cut-elimination methods is known [65][a], and some ad-hoc methods to introduce cuts of restricted forms have been proposed. They are based on techniques from automated theorem proving, such as conflict-driven formula learning [42], and from logic programming, such as tabling [92, 86].

Besides the concern of compression, cut-introduction is also interesting as a way of structuring and extracting interesting concepts from proofs. In [40], for example, it is shown that many translation techniques of automated deduction can be seen as introduced cuts. Furthermore, in mathematical proofs, automatic introduction of cuts would correspond to the automatic discovery of potentially useful lemmas. And Chapter 9 shows how cuts and the introduction of cuts corresponds to various procedures in other areas of science.

This Chapter shows that, with very simple modifications, the CERes method can be converted into a method for the introduction of atomic

[a] A cut-introduction method g is inverse of a reductive cut-elimination method if and only if, for any cut-free proof φ, the proof with cuts $g(\varphi)$ rewrites to φ (i.e. $g(\varphi) \rhd \varphi$).

cuts that can compress proofs.

7.1 Cut-Introduction by Resolution

The CIRes method of cut-introduction by resolution is based on two simple observations:

- In a naive attempt to introduce cuts by applying the proof rewriting rules of reductive cut-elimination methods in an inverse direction, the first step, which is the introduction of atomic cuts in the top of cut-free proofs, is trivial. However, pushing the cuts down (by applying inverse rank reduction rules), combining the cuts to make more complex cuts, and exploiting redundancies in the form of contractions is non-trivial.

- If applied to a proof containing atomic cuts in the top, CERes outputs a proof containing atomic cuts in the bottom, because a CERes-normal-form is constructed by plugging all the projections on the top of the refutation, and all atomic cuts are converted resolution inferences from the refutation.

The CIRes method then simply consists of adding atomic cuts to every leaf of the cut-free proof and applying CERes to push these cuts down. Compression can be achieved thanks mainly to three ways by which CERes is able to exploit or avoid redundancies:

- It is possible that the refutation uses only some clauses of the clause set. The effect is that large parts of the proof (i.e. the projections with respect to the unused clauses) can be deleted, replaced by weakening. Clearly, this kind of deletion can lead to compression.

- If a refutation contains factoring inferences, the corresponding proof with cuts has cuts whose cut-formula occurrences are being contracted. Since the presence of cut-pertinent contractions is a major reason for the increase of size and length during cut-elimination, adding cut-pertinent contractions (via factoring) can lead to compression.

- The improved normalizations used in the construction of profile or swapped clause sets and the careful construction of O-projections minimize redundancy. (It is interesting to note, on the other hand, that in the case of standard clause sets and S-projections, parts of the proof frequently occur repeated in several projections, jeopardizing the potential for compression. The evolution from standard clause sets to profile or swapped clause sets and from S-projections to O-projections has been therefore a critical step to enable the development of CIRes from CERes.)

Definition 7.1.1 (Naive Introduction of Atomic Cuts)**.** Let φ be a cut-free proof. Then φ^a denotes the proof obtained from φ by replacing every axiom inference with conclusion sequent of the form $A \vdash A$ by a subproof of the form:

$$\dfrac{A \vdash A \qquad A \vdash A}{A \vdash A}\, cut$$

Definition 7.1.2 (CIRes-Normal-Form)**.** The CIRes-*normal-form* of a cut-free proof φ with respect to a refutation δ of a clause set C_{φ^a} is:

$$\mathrm{CIRes}(\varphi, \delta) \doteq \mathrm{CERes}(\varphi^a, \delta)$$

Remark 7.1.1. Analogously to CERes, subscripts and superscripts are also used to explicitly state which variants of clause sets and projections are used. So, $\mathrm{CIRes}^O_W(\varphi, \delta)$ denotes a CIRes-normal-form in which a swapped clause set and O-projections are used, for example.

Example 7.1 (CIRes Applied to a Very Simple Compressible Proof)**.** *Let φ be the proof below:*

$$\dfrac{\dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A, A \vdash A \wedge A}\,\wedge_r}{A \vdash A \wedge A}\,c_l \qquad \dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A, A \vdash A \wedge A}\,\wedge_r}{A \vdash A \wedge A}\,c_l}{\dfrac{A \vee A \vdash A \wedge A, A \wedge A}{A \vee A \vdash A \wedge A}\,c_r}\,\vee_l$$

Then φ^a is:

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A \vdash A}\,cut \qquad \dfrac{A \vdash A \qquad A \vdash A}{A \vdash A}\,cut}{A, A \vdash A \wedge A}\,\wedge_r}{A \vdash A \wedge A}\,c_l \qquad \dfrac{\dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A \vdash A}\,cut \qquad \dfrac{A \vdash A \qquad A \vdash A}{A \vdash A}\,cut}{A, A \vdash A \wedge A}\,\wedge_r}{A \vdash A \wedge A}\,c_l}{\dfrac{A \vee A \vdash A \wedge A, A \wedge A}{A \vee A \vdash A \wedge A}\,c_r}\,\vee_l$$

Its cut-pertinent struct is:

$$\mathcal{S}_{\varphi^a} \equiv ((A \oplus \neg A) \otimes^{**} (A \oplus \neg A)) \otimes^{****} ((A \oplus \neg A) \otimes^{**} (A \oplus \neg A))$$

And it can be $\rightsquigarrow_{\oplus\otimes_W}$-normalized as follows:

$$\begin{array}{rl}
\mathcal{S}_{\varphi^a} & \equiv ((A \oplus \neg A) \otimes^{**} (A \oplus \neg A)) \otimes^{****} ((A \oplus \neg A) \otimes^{**} (A \oplus \neg A)) \\
\rightsquigarrow_{\oplus\otimes_W} & (A \oplus A \oplus (\neg A \otimes^{**} \neg A)) \otimes^{****} ((A \oplus \neg A) \otimes^{**} (A \oplus \neg A)) \\
\rightsquigarrow_{\oplus\otimes_W} & (A \oplus A \oplus (\neg A \otimes^{**} \neg A)) \otimes^{****} (A \oplus A \oplus (\neg A \otimes^{**} \neg A)) \\
\rightsquigarrow_{\oplus\otimes_W} & (A \otimes^{****} A) \oplus (A \otimes^{****} A) \oplus (A \otimes^{****} A) \oplus (A \otimes^{****} A) \oplus (\neg A \otimes^{**} \neg A) \oplus (\neg A \otimes^{**} \neg A)
\end{array}$$

The swapped clause set is therefore:

$$C^W_{\varphi^a} \equiv \{\ \vdash A, A\ ;\ \vdash A, A\ ;\ \vdash A, A\ ;\ \vdash A, A\ ;\ A, A \vdash\ ;\ A, A \vdash\ \}$$

And δ shown below is one of the possible refutations of $C^W_{\varphi^a}$:

$$\dfrac{\dfrac{\vdash A, A}{\vdash A}\,f_r \qquad \dfrac{A, A \vdash}{A \vdash}\,f_l}{\vdash}\,r$$

The O-projections are:

$\lfloor \varphi^a \rfloor^O_{\vdash A, A}$:

$$\dfrac{A \vdash A \qquad A \vdash A}{A \vee A \vdash A, A}\,\vee_l$$

$\lfloor \varphi^a \rfloor^O_{A, A \vdash}$:

$$\dfrac{A \vdash A \qquad A \vdash A}{A, A \vdash A \wedge A}\,\wedge_r$$

Finally, the O-projections and the refutation can be combined in order to produce $\mathrm{CERes}^O_W(\varphi^a, \delta)$*, which is, by definition,* $\mathrm{CIRes}^O_W(\varphi, \delta)$*:*

$$\dfrac{\dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A \vee A \vdash A, A}\,\vee_l}{A \vee A \vdash A}\,c_r \qquad \dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A, A \vdash A \wedge A}\,\wedge_r}{A \vdash A \wedge A}\,c_l}{A \vee A \vdash A \wedge A}\,cut$$

The following hold regarding sizes and length of φ:

- $\mathit{length}(\varphi) = 6$
- $|\varphi| = 41$
- $|\varphi|_a = 32$

Regarding sizes and length of $\mathrm{CIRes}^O_W(\varphi, \delta)$*, on the other hand:*

- $\mathit{length}(\mathrm{CIRes}^O_W(\varphi, \delta)) = 5$
- $|\mathrm{CIRes}^O_W(\varphi, \delta)| = 32$

- $|\mathrm{CIRes}^{O}_{W}(\varphi, \delta)|_a = 26$

This shows that CIRes *is indeed able to compress proofs. It is interesting to note that* $\mathrm{CIRes}^{O}_{W}(\varphi, \delta)$ *is the shortest proof of* $A \lor A \vdash A \land A$ *in the sequent calculus used. The proof* φ*, on the other hand is one of the two shortest cut-free proofs of* $A \lor A \vdash A \land A$*. The other shortest proof is the proof* φ' *shown below:*

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A \lor A \vdash A, A}\ \lor_l}{A \lor A \vdash A}\ c_r \qquad \dfrac{\dfrac{A \vdash A \qquad A \vdash A}{A \lor A \vdash A, A}\ \lor_l}{A \lor A \vdash A}\ c_r}{A \lor A, A \lor A \vdash A \land A}\ \land_r}{A \lor A \vdash A \land A}\ c_l$$

Note that $\mathrm{CIRes}^{O}_{W}(\varphi, \delta) \rhd^{*} \varphi$ *and* $\mathrm{CIRes}^{O}_{W}(\varphi, \delta) \rhd^{*} \varphi'$*, and it is also not so hard to verify that* $\mathrm{CIRes}^{O}_{W}(\varphi', \delta) = \mathrm{CIRes}^{O}_{W}(\varphi, \delta)$ *(not considering the colors/labels).*

In fact, there are infinitely many φ^{*} *such that* $\mathrm{CIRes}^{O}_{W}(\varphi, \delta) \rhd^{*} \varphi^{*}$ *[66], and for any of them* $\mathrm{CIRes}^{O}_{W}(\varphi^{*}, \delta) = \mathrm{CIRes}^{O}_{W}(\varphi, \delta)$*. Considering that the size of* φ^{*} *is arbitrary, the amount of compression achievable with* CIRes *is unbounded.*

Example 7.2 (CIRes Applied to a More Interesting Proof). *Although Example 7.1 shows that* CIRes *can compress proofs, the proof used is very simple. In this example, a more interesting proof*[b] φ *shown below is used:*

$$
\begin{prooftree}
\AxiomC{$P^1 \vdash P^1$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^1, \neg P^1 \vdash$}
\AxiomC{$P^1 \vdash P^1$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^1, \neg P^1 \vdash$}
\AxiomC{$P^2_- \vdash P^2_-$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^2_-, \neg P^2_- \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^2_-, P^1, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^1, P^1, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{c_l}
\UnaryInfC{$P^1, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\AxiomC{$P^1 \vdash P^1$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^1, \neg P^1 \vdash$}
\AxiomC{$P^1 \vdash P^1$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^1, \neg P^1 \vdash$}
\AxiomC{$P^2_- \vdash P^2_-$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^2_-, \neg P^2_- \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^2_-, P^1, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^1, P^1, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{c_l}
\UnaryInfC{$P^1, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\AxiomC{$P^2_+ \vdash P^2_+$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^2_+, \neg P^2_+ \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^2_+, P^1 \vee \neg P^2_+, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^1 \vee P^2_+, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_-, P^1 \vee \neg P^2_+, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{c_l}
\UnaryInfC{$P^1 \vee P^2_+, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_-, P^1 \vee \neg P^2_+, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{c_l}
\UnaryInfC{$P^1 \vee P^2_+, \neg P^1 \vee P^2_-, P^1 \vee \neg P^2_+, \neg P^1 \vee \neg P^2_- \vdash$}
\end{prooftree}
$$

The subproofs highlighted below are equal. This redundant repetition suggests that cut-introduction might be able to compress this proof.

[b] The end-sequent of this proof was adapted from an instance of a sequence of clause sets used in [26] to show that the resolution calculus can produce significantly shorter proofs than the analytic tableaux calculus. .

$$
\begin{prooftree}
\AxiomC{$P^1 \vdash P^1$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^1, \neg P^1 \vdash$}
\AxiomC{$P^1 \vdash P^1$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^1, \neg P^1 \vdash$}
\AxiomC{$P^2_- \vdash P^2_-$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^2_-, \neg P^2_- \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^2_-, P^1, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^1, P^1, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{c_l}
\UnaryInfC{$P^1, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\AxiomC{$P^1 \vdash P^1$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^1, \neg P^1 \vdash$}
\AxiomC{$P^1 \vdash P^1$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^1, \neg P^1 \vdash$}
\AxiomC{$P^2_- \vdash P^2_-$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^2_-, \neg P^2_- \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^2_-, P^1, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^1, P^1, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{c_l}
\UnaryInfC{$P^1, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\AxiomC{$P^2_+ \vdash P^2_+$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^2_+, \neg P^2_+ \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^2_+, P^1 \vee \neg P^2_+, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^1 \vee P^2_+, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_-, P^1 \vee \neg P^2_+, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{c_l}
\UnaryInfC{$P^1 \vee P^2_+, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_-, P^1 \vee \neg P^2_+, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{c_l}
\UnaryInfC{$P^1 \vee P^2_+, \neg P^1 \vee P^2_-, P^1 \vee \neg P^2_+, \neg P^1 \vee \neg P^2_- \vdash$}
\end{prooftree}
$$

Following the steps of the CIRes *method, φ^a is obtained by adding atomic cuts to the leaves of φ:*

$$
\begin{prooftree}
\AxiomC{φ^a_l}
\AxiomC{φ^a_r}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^1 \vee P^2_+, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_-, P^1 \vee \neg P^2_+, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{c_l}
\UnaryInfC{$P^1 \vee P^2_+, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_-, P^1 \vee \neg P^2_+, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{c_l}
\UnaryInfC{$P^1 \vee P^2_+, \neg P^1 \vee P^2_-, P^1 \vee \neg P^2_+, \neg P^1 \vee \neg P^2_- \vdash$}
\end{prooftree}
$$

where φ^a_l is:

$$
\begin{prooftree}
\AxiomC{$P^1 \vdash P^1$}
\AxiomC{$P^1 \vdash P^1$}
\RightLabel{cut}
\dashedLine
\BinaryInfC{$P^1 \vdash P^1$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^1, \neg P^1 \vdash$}
\AxiomC{$P^1 \vdash P^1$}
\AxiomC{$P^1 \vdash P^1$}
\RightLabel{cut}
\dashedLine
\BinaryInfC{$P^1 \vdash P^1$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^1, \neg P^1 \vdash$}
\AxiomC{$P^2_- \vdash P^2_-$}
\AxiomC{$P^2_- \vdash P^2_-$}
\RightLabel{cut}
\dashedLine
\BinaryInfC{$P^2_- \vdash P^2_-$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^2_-, \neg P^2_- \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^2_-, P^1, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^1, P^1, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{c_l}
\UnaryInfC{$P^1, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\end{prooftree}
$$

and φ^a_r is:

$$
\begin{prooftree}
\AxiomC{$P^1 \vdash P^1$}
\AxiomC{$P^1 \vdash P^1$}
\dashedLine
\RightLabel{cut}
\BinaryInfC{$P^1 \vdash P^1$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^1, \neg P^1 \vdash$}
\AxiomC{$P^1 \vdash P^1$}
\AxiomC{$P^1 \vdash P^1$}
\dashedLine
\RightLabel{cut}
\BinaryInfC{$P^1 \vdash P^1$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^1, \neg P^1 \vdash$}
\AxiomC{$P^2_- \vdash P^2_-$}
\AxiomC{$P^2_- \vdash P^2_-$}
\dashedLine
\RightLabel{cut}
\BinaryInfC{$P^2_- \vdash P^2_-$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^2_-, \neg P^2_- \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^2_-, P^1, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^1, P^1, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{c_l}
\UnaryInfC{$P^1, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\AxiomC{$P^2_+ \vdash P^2_+$}
\AxiomC{$P^2_+ \vdash P^2_+$}
\dashedLine
\RightLabel{cut}
\BinaryInfC{$P^2_+ \vdash P^2_+$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^2_+, \neg P^2_+ \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^2_+, P^1 \vee \neg P^2_+, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\end{prooftree}
$$

Its cut-pertinent struct is:

$$\mathcal{S}_{\varphi^a} \equiv ((P^1 \oplus \neg P^1) \otimes^{**} ((P^1 \oplus \neg P^1) \otimes^{**} (P^2_- \oplus \neg P^2_-))) \otimes^{***} (((P^1 \oplus \neg P^1) \otimes^{**} ((P^1 \oplus \neg P^1) \otimes^{**} (P^2_- \oplus \neg P^2_-))) \otimes^{***} (P^2_+ \oplus \neg P^2_+))$$

And it can be normalized as follows:

$$
\begin{array}{lcl}
\mathcal{S}_{\varphi^a} & \equiv & ((P^1 \oplus \neg P^1) \otimes^{**} ((P^1 \oplus \neg P^1) \otimes^{**} (P^2_- \oplus \neg P^2_-))) \otimes^{***} (((P^1 \oplus \neg P^1) \otimes^{**} ((P^1 \oplus \neg P^1) \otimes^{**} (P^2_- \oplus \neg P^2_-))) \otimes^{***} (P^2_+ \oplus \neg P^2_+)) \\
 & \rightsquigarrow_{\oplus\otimes W} & ((P^1 \oplus \neg P^1) \otimes^{**} (P^1 \oplus P^2_- \oplus (\neg P^1 \otimes^{**} \neg P^2_-))) \otimes^{***} (((P^1 \oplus \neg P^1) \otimes^{**} ((P^1 \oplus \neg P^1) \otimes^{**} (P^2_- \oplus \neg P^2_-))) \otimes^{***} (P^2_+ \oplus \neg P^2_+)) \\
 & \rightsquigarrow_{\oplus\otimes W} & (P^1 \oplus P^1 \oplus (\neg P^1 \otimes^{**} P^2_-) \oplus (\neg P^1 \otimes^{**} \neg P^2_-)) \otimes^{***} (((P^1 \oplus \neg P^1) \otimes^{**} ((P^1 \oplus \neg P^1) \otimes^{**} (P^2_- \oplus \neg P^2_-))) \otimes^{***} (P^2_+ \oplus \neg P^2_+)) \\
 & \rightsquigarrow_{\oplus\otimes W} & (P^1 \oplus P^1 \oplus (\neg P^1 \otimes^{**} P^2_-) \oplus (\neg P^1 \otimes^{**} \neg P^2_-)) \otimes^{***} (((P^1 \oplus \neg P^1) \otimes^{**} (P^1 \oplus P^2_- \oplus (\neg P^1 \otimes^{**} \neg P^2_-))) \otimes^{***} (P^2_+ \oplus \neg P^2_+)) \\
 & \rightsquigarrow_{\oplus\otimes W} & (P^1 \oplus P^1 \oplus (\neg P^1 \otimes^{**} P^2_-) \oplus (\neg P^1 \otimes^{**} \neg P^2_-)) \otimes^{***} ((P^1 \oplus P^1 \oplus (\neg P^1 \otimes^{**} P^2_-) \oplus (\neg P^1 \otimes^{**} \neg P^2_-)) \otimes^{***} (P^2_+ \oplus \neg P^2_+)) \\
 & \rightsquigarrow_{\oplus\otimes W} & (P^1 \oplus P^1 \oplus (\neg P^1 \otimes^{**} P^2_-) \oplus (\neg P^1 \otimes^{**} \neg P^2_-)) \otimes^{***} ((P^1 \otimes^{***} \neg P^2_+) \oplus (P^1 \otimes^{***} \neg P^2_+) \oplus (\neg P^1 \otimes^{**} P^2_-) \oplus (\neg P^1 \otimes^{**} \neg P^2_-) \oplus P^2_+) \\
 & \rightsquigarrow_{\oplus\otimes W} & ((P^1 \otimes^{***} P^2_+) \oplus (P^1 \otimes^{***} P^2_+) \oplus (\neg P^1 \otimes^{**} P^2_-) \oplus (\neg P^1 \otimes^{**} \neg P^2_-)) \oplus ((P^1 \otimes^{***} \neg P^2_+) \oplus (P^1 \otimes^{***} \neg P^2_+) \oplus (\neg P^1 \otimes^{**} P^2_-) \oplus (\neg P^1 \otimes^{**} \neg P^2_-)) \\
 & \equiv & (P^1 \otimes^{***} P^2_+) \oplus (P^1 \otimes^{***} P^2_+) \oplus (\neg P^1 \otimes^{**} P^2_-) \oplus (\neg P^1 \otimes^{**} \neg P^2_-) \oplus (P^1 \otimes^{***} \neg P^2_+) \oplus (P^1 \otimes^{***} \neg P^2_+) \oplus (\neg P^1 \otimes^{**} P^2_-) \oplus (\neg P^1 \otimes^{**} \neg P^2_-)
\end{array}
$$

The swapped clause set is therefore:

$$C^W_{\varphi^a} \equiv \{\ \vdash P^1, P^2_+ \ ;\ \vdash P^1, P^2_+ \ ;\ P^1 \vdash P^2_- \ ;\ P^1, P^2_- \vdash \ ;\ P^2_+ \vdash P^1 \ ;\ P^2_+ \vdash P^1 \ ;\ P^1 \vdash P^2_- \ ;\ P^1, P^2_- \vdash\ \}$$

$C^W_{\varphi^a}$ can be refuted by the refutation δ below:

$$\dfrac{\dfrac{\dfrac{\vdash P^1, P^2_+ \qquad P^2_+ \vdash P^1}{\vdash P^1, P^1}\,r}{\vdash P^1}\,f_r \qquad \dfrac{\dfrac{P^1 \vdash P^2_- \qquad P^1, P^2_- \vdash}{P^1, P^1 \vdash}\,r}{P^1 \vdash}\,f_r}{\vdash}\,r$$

The O-projections are:

$\lfloor \varphi^a \rfloor^O_{\vdash P^1, P^2_+}$:

$$\dfrac{P^1 \vdash P^1 \qquad P^2_+ \vdash P^2_+}{P^1 \vee P^2_+ \vdash P^1, P^2_+}\,\vee_l$$

$\lfloor \varphi^a \rfloor^O_{P^2_+ \vdash P^1}$:

$$\dfrac{P^1 \vdash P^1 \qquad \dfrac{P^2_+ \vdash P^2_+}{P^2_+, \neg P^2_+ \vdash}\,\neg_l}{P^2_+, P^1 \vee \neg P^2_+ \vdash P^1}\,\vee_l$$

$\lfloor \varphi^a \rfloor^O_{P^1 \vdash P^2_-}$:

$$\dfrac{\dfrac{P^1 \vdash P^1}{P^1, \neg P^1 \vdash}\,\neg_l \qquad P^2_- \vdash P^2_-}{P^1, \neg P^1 \vee P^2_- \vdash P^2_-}\,\vee_l$$

$\lfloor \varphi^a \rfloor^O_{P^1, P^2_- \vdash}$:

$$\dfrac{\dfrac{P^1 \vdash P^1}{P^1, \neg P^1 \vdash}\,\neg_l \qquad \dfrac{P^2_- \vdash P^2_-}{P^2_-, \neg P^2_- \vdash}\,\neg_l}{P^1, P^2_-, \neg P^1 \vee \neg P^2_- \vdash}\,\vee_l$$

$\mathrm{CIRes}^{O}_{W}(\varphi,\delta)$, *which is by definition equal to* $\mathrm{CERes}^{O}_{S}(\varphi^{a},\delta)$, *is:*

$$
\begin{prooftree}
\AxiomC{$P^1 \vdash P^1$}
\AxiomC{$P^2_+ \vdash P^2_+$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^1 \vee P^2_+ \vdash P^1, P^2_+$}
\AxiomC{$P^1 \vdash P^1$}
\AxiomC{$P^2_+ \vdash P^2_+$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^2_+, \neg P^2_+ \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^2_+, P^1 \vee \neg P^2_+ \vdash P^1$}
\RightLabel{cut}
\BinaryInfC{$P^1 \vee P^2_+, P^1 \vee \neg P^2_+ \vdash P^1, P^1$}
\RightLabel{c_r}
\UnaryInfC{$P^1 \vee P^2_+, P^1 \vee \neg P^2_+ \vdash P^1$}
\AxiomC{$P^1 \vdash P^1$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^1, \neg P^1 \vdash$}
\AxiomC{$P^2_- \vdash P^2_-$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^1, \neg P^1 \vee P^2_- \vdash P^2_-$}
\AxiomC{$P^1 \vdash P^1$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^1, \neg P^1 \vdash$}
\AxiomC{$P^2_- \vdash P^2_-$}
\RightLabel{$\neg_l$}
\UnaryInfC{$P^2_-, \neg P^2_- \vdash$}
\RightLabel{$\vee_l$}
\BinaryInfC{$P^1, P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{cut}
\BinaryInfC{$P^1, P^1, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{c_r}
\UnaryInfC{$P^1, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\RightLabel{cut}
\BinaryInfC{$P^1 \vee P^2_+, P^1 \vee \neg P^2_+, \neg P^1 \vee P^2_-, \neg P^1 \vee \neg P^2_- \vdash$}
\end{prooftree}
$$

Comparing the lengths and sizes, the compression achieved is now much more impressive than in Example 7.1:

- $\mathit{length}(\varphi) = 17$
- $|\varphi| = 169$
- $|\varphi|_a = 97$
- $\mathit{length}(\mathrm{CIRes}^{O}_{W}(\varphi,\delta)) = 13$
- $|\mathrm{CIRes}^{O}_{W}(\varphi,\delta)| = 105$
- $|\mathrm{CIRes}^{O}_{W}(\varphi,\delta)|_a = 70$

7.2 Remarks

Even though CIRes can successfully introduce atomic cuts and compress proofs, the method still has a major drawback. Refuting the swapped clause sets is, at least in the examples considered in the previous section, as hard as refuting the clause form of the negation of the end-sequent. This means that introducing cuts via CIRes is approximately as computationally expensive as simply reproving the theorem by resolution and then converting the resolution proof to a sequent calculus. Nevertheless, it might be possible to design resolution refinements for CIRes that could use more information contained in the cut-free proof in ways that would facilitate the introduction of cuts.

Another drawback is the fact that CIRes produces normal forms in which all atomic cuts are in the bottom. In the examples considered in this chapter, this was not a problem, because the shortest proofs with cuts happened to be proofs in which the atomic cuts occur in the bottom. However, this is not generally the case, and then CIRes will sometimes produce sub-optimally compressed normal forms, because the projections will contain redundancies in order to allow the atomic cuts to be in the bottom. This drawback could only be solved by using radically different notions of projections, which would allow them to be combined in more flexible ways with the refutation of the clause set, so that the atomic cuts do not necessarily appear in the bottom of the normal forms. However, such an improvement of projections and of the CERes method itself is far from trivial and has yet to be developed.

Finally, it must be remarked that much more significant compression (e.g. non-elementary compression) could in principle be obtained via introduction of quantified cuts. The CIRes method described in this chapter introduces only atomic cuts and is therefore just a small first step toward the harder task of introducing general complex quantified cuts. An intermediary step could be the introduction of propositional cuts, which could be perhaps possible by using definitional and swapped definitional clause sets. But even then, there would still be a long way to achieve compressive quantified-cut-introduction. And in any case, an algorithm that would generally guarantee optimal compression cannot exist, due to the undecidability results in [14].

Chapter 8

Complexity

In previous chapters, several variants of the CERes method were developed, but comparisons were purely qualitative. The purpose of this chapter is to complement this qualitative analysis with a few quantitative results and conjectures regarding the asymptotic sizes of proofs, clause sets and refutation search spaces when various variants are used.

8.1 Number of Clauses in Clause Sets

In this section, the asymptotic number of clauses in different variants of clause sets is compared.

Theorem 8.1 (Number of Clauses in Profile and Swapped Clause Sets)**.** There exists a sequence of proofs $\varphi_1, \varphi_2, \ldots, \varphi_n, \ldots$ such that:

- $|\varphi_n| = kn$, for some constant k.
- $|C^P_{\varphi_n}| = 3n$.
- $|C^W_{\varphi_n|S}| = 3$, for any $\rightsquigarrow_{\oplus\otimes_W}$-normal-form S of $\mathcal{S}_\varphi$.

Proof. Let φ_1 be the proof below:

$$\dfrac{\dfrac{A_1 \vdash A_1 \qquad B_1 \vdash B_1}{A_1, B_1 \vdash A_1 \wedge B_1}\wedge_r \qquad \dfrac{\dfrac{A_1 \vdash A_1 \qquad B_1 \vdash B_1}{A_1, B_1 \vdash A_1 \wedge B_1}\wedge_r}{A_1 \wedge B_1 \vdash A_1 \wedge B_1}\wedge_l}{A_1, B_1 \vdash A_1 \wedge B_1}\mathit{cut}$$

And let φ_n, for $n > 1$, be:

$$\dfrac{\dfrac{\begin{array}{c}\varphi_{n-1}\\ A_1, \ldots, A_{n-1}, B_1, \ldots, B_{n-1} \vdash A_1 \wedge B_1, \ldots, A_{n-1} \wedge B_{n-1}\end{array}}{A_1, \ldots, A_{n-1}, B_1, \ldots, B_{n-1} \vdash A_1 \wedge B_1, \ldots, A_{n-1} \wedge B_{n-1}, C}w_r \qquad \dfrac{\dfrac{\dfrac{A_n \vdash A_n \qquad B_n \vdash B_n}{A_n, B_n \vdash A_n \wedge B_n}\wedge_r \qquad \dfrac{\dfrac{A_n \vdash A_n \qquad B_n \vdash B_n}{A_n, B_n \vdash A_n \wedge B_n}\wedge_r}{A_n \wedge B_n \vdash A_n \wedge B_n}\wedge_l}{A_n, B_n \vdash A_n \wedge B_n}\mathit{cut}}{C, A_n, B_n \vdash A_n \wedge B_n}w_l}{A_1, \ldots, A_n, B_1, \ldots, B_n \vdash A_1 \wedge B_1, \ldots, A_n \wedge B_n}\mathit{cut}_{n-1}$$

Then:

$$\mathcal{S}_{\varphi_n} = (\ldots((A_1 \oplus B_1) \oplus (\neg A_1 \otimes \neg B_1)) \oplus \ldots) \oplus ((A_n \oplus B_n) \oplus (\neg A_n \otimes \neg B_n))$$

$\mathcal{S}_{\varphi_n}$ is already in $\rightsquigarrow_{\oplus\otimes_P}$-normal-form. Therefore:

$$C^P_{\varphi_n} = \{ \vdash A_1 \,;\, \vdash B_1 \,;\, A_1, B_1 \vdash \,;\, \vdash A_2 \,;\, \vdash B_2 \,;\, A_2, B_2 \vdash \,;\, \ldots \,;\, \vdash A_n \,;\, \vdash B_n \,;\, A_n, B_n \vdash \,;\}$$

And hence, $|C^P_{\varphi_n}| = 3n$.

However, $\mathcal{S}_{\varphi_n}$ is not in $\rightsquigarrow_{\oplus\otimes_W}$-normal-form, because the inferences cut_i ($1 \leq i < n$) are degenerate. Therefore, there are n $\rightsquigarrow_{\oplus\otimes_W}$-normal-forms S_j (($1 \leq j \leq n$)) of $\mathcal{S}_{\varphi_n}$:

$$S_j = (A_j \oplus B_j) \oplus (\neg A_j \otimes \neg B_j)$$

Consequently:

$$\mathcal{C}^{W}_{\varphi_n|S_j} = \{\vdash A_j \,;\, \vdash B_j \,;\, A_j, B_j \vdash\}$$

And hence, $|\mathcal{C}^{W}_{\varphi_n|S_j}| = 3$, for any $\rightsquigarrow_{\oplus\otimes_W}$-normal-form S_j of $\mathcal{S}_{\varphi_n}$. $\square$

Theorem 8.2 (Number of Clauses in Definitional and non-Definitional Clause Sets)**.** There exists a sequence of proofs $\varphi_1, \varphi_2, \ldots, \varphi_n, \ldots$ such that:

- $|\varphi_n| \leq k4^n$, for some rational constant k.
- $|C^S_{\varphi_n}| \geq k'2^{2^n}$, for some rational constant k'.
- $|C^D_{\varphi_n}| \leq k''2^{(k'''n)}$, for some rational constants k'' and k'''.

Proof. Let $\psi_1(s)$ be the following proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{A_1(s) \vdash A_1(s) \qquad B_1(s) \vdash B_1(s)}{A_1(s), B_1(s) \vdash A_1(s) \wedge B_1(s)}\wedge_r}{A_1(s) \wedge B_1(s) \vdash A_1(s) \wedge B_1(s)}\wedge_l \qquad \dfrac{\dfrac{C_1(s) \vdash C_1(s) \qquad D_1(s) \vdash D_1(s)}{C_1(s), D_1(s) \vdash C_1(s) \wedge D_1(s)}\wedge_r}{C_1(s) \wedge D_1(s) \vdash C_1(s) \wedge D_1(s)}\wedge_l}{(A_1(s) \wedge B_1(s)) \vee (C_1(s) \wedge D_1(s)) \vdash A_1(s) \wedge B_1(s), C_1(s) \wedge D_1(s)}\vee_l}{(A_1(s) \wedge B_1(s)) \vee (C_1(s) \wedge D_1(s)) \vdash (A_1(s) \wedge B_1(s)) \vee (C_1(s) \wedge D_1(s))}\vee_r$$

And let $\psi_n(s)$ $(n > 1)$ be:

$$\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\psi_{n-1}(a.s)\\ A_n(s) \vdash A_n(s)\end{array} \qquad \begin{array}{c}\psi_{n-1}(b.s)\\ B_n(s) \vdash B_n(s)\end{array}}{A_n(s), B_n(s) \vdash A_n(s) \wedge B_n(s)}\wedge_r}{A_n(s) \wedge B_n(s) \vdash A_n(s) \wedge B_n(s)}\wedge_l \qquad \dfrac{\dfrac{\begin{array}{c}\psi_{n-1}(c.s)\\ C_n(s) \vdash C_n(s)\end{array} \qquad \begin{array}{c}\psi_{n-1}(d.s)\\ D_n(s) \vdash D_n(s)\end{array}}{C_n(s), D_n(s) \vdash C_n(s) \wedge D_n(s)}\wedge_r}{C_n(s) \wedge D_n(s) \vdash C_n(s) \wedge D_n(s)}\wedge_l}{(A_n(s) \wedge B_n(s)) \vee (C_n(s) \wedge D_n(s)) \vdash A_n(s) \wedge B_n(s), C_n(s) \wedge D_n(s)}\vee_l}{(A_n(s) \wedge B_n(s)) \vee (C_n(s) \wedge D_n(s)) \vdash (A_n(s) \wedge B_n(s)) \vee (C_n(s) \wedge D_n(s))}\vee_r$$

where (for $n > 1$):

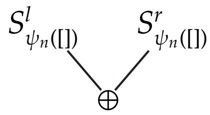
$$\begin{array}{rcl}
A_n(s) & \doteq & (A_{n-1}(a.s) \wedge B_{n-1}(a.s)) \vee (C_{n-1}(a.s) \wedge D_{n-1}(a.s)) \\
B_n(s) & \doteq & (A_{n-1}(b.s) \wedge B_{n-1}(b.s)) \vee (C_{n-1}(b.s) \wedge D_{n-1}(b.s)) \\
C_n(s) & \doteq & (A_{n-1}(c.s) \wedge B_{n-1}(c.s)) \vee (C_{n-1}(c.s) \wedge D_{n-1}(c.s)) \\
D_n(s) & \doteq & (A_{n-1}(d.s) \wedge B_{n-1}(d.s)) \vee (C_{n-1}(d.s) \wedge D_{n-1}(d.s))
\end{array}$$

and $x.l$ denotes the result of appending the character x in the beginning of the list of characters l.

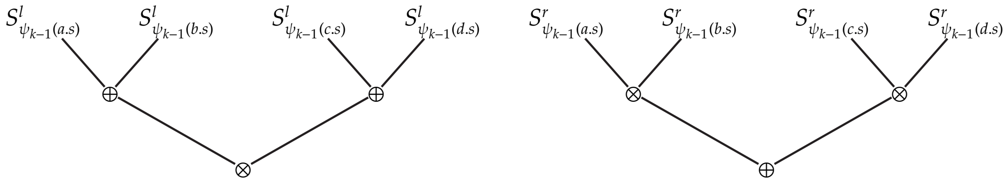
Moreover, let φ_n be the proof below:

$$\dfrac{\psi_n([]) \qquad \psi_n([])}{(A_n([]) \wedge B_n([])) \vee (C_n([]) \wedge D_n([])) \vdash (A_n([]) \wedge B_n([])) \vee (C_n([]) \wedge D_n([]))}\,cut$$

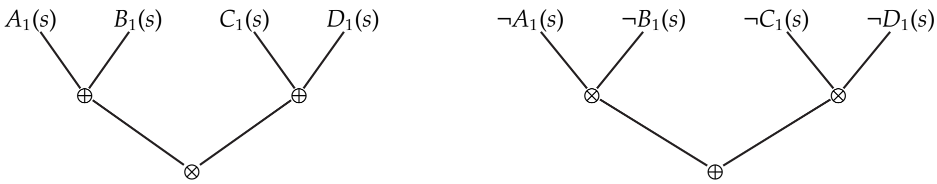
Let $S^l_{\psi_k(s)}$ be the substruct of $\mathcal{S}_{\varphi_n}$ at the root inference of the subproof $\psi_k(s)$ in left side of φ_n. Analogously, let $S^r_{\psi_k(s)}$ be the substruct of $\mathcal{S}_{\varphi_n}$ at the root inference of the subproof $\psi_k(s)$ in the right side of φ_n. Then $\mathcal{S}_{\varphi_n}$ is (displayed as a tree):



where $S^l_{\psi_k(s)}$ and $S^r_{\psi_k(s)}$ ($k > 1$) are:



and $S^l_{\psi_1(s)}$ and $S^r_{\psi_1(s)}$ are:



Let $f_l(n)$ be the number of clauses in $C^S_{\varphi_n}$ whose formula occurrences occur in the left branch of the cut. Analogously, let $f_r(n)$ be the number of clauses stemming from the right branch of the cut. Clearly, $|C^S_{\varphi_n}| = f_l(n) + f_r(n)$. A careful analysis of the structs displayed above shows that:

$$f_l(n) = \begin{cases} 4 & \text{, if } n = 1 \\ 4(f_l(n-1))^2 & \text{, otherwise} \end{cases}$$

$$f_r(n) = \begin{cases} 2 & \text{, if } n = 1 \\ 2(f_r(n-1))^2 & \text{, otherwise} \end{cases}$$

It can be easily proved by induction that $f_l(n) = 4^{(2^n-1)}$ and $f_r(n) = 2^{(2^n-1)}$. Therefore:

$$|C^S_{\varphi_n}| = 4^{(2^n-1)} + 2^{(2^n-1)} \geq 2^{(2^n)}$$

An inspection of the structs displayed above also shows that the number of non-literal substructs in $\mathcal{S}_{\varphi_n}$ is $2(4^n) - 1$ ($4^n - 1$ substructs in each of $S^l_{\psi_n([])}$ and $S^r_{\psi_n([])}$ plus the whole unproper substruct $\mathcal{S}_{\varphi_n}$). Each substruct generates a new defined predicate, and for each defined predicate, three clauses are formed. An additional clause corresponds to the whole struct. Therefore $|C^D_{\varphi_n}| = 3(2(4^n) - 1) + 1 = 6(4^n) + 2 \leq 7(2^{2n})$.

□

Remark 8.1.1. Theorem 8.2 could be modified so that $C^P_{\varphi_n}$ or $C^W_{\varphi_n}$ were used instead of $C^S_{\varphi_n}$. For φ_n shown in the proof of the theorem, $C^P_{\varphi_n} = C^W_{\varphi_n} = C^S_{\varphi_n}$. Therefore, exactly the same bounds on the sizes would apply.

Remark 8.1.2. Theorem 8.2 could be modified so that $C^D_{\varphi_n}$ were used instead of $C^D_{\varphi_n}$. Clearly, the upper bound for $|C^D_{\varphi_n}|$ would still be exponential, but the constants k'' and k''' could be smaller.

8.2 Search Space

In this section, the search spaces when using different variants of the CERes method are compared.

Definition 8.2.1 (Search Space[a])**.** The *search space* $s_R(C)$ used for refuting a set of clauses C in a refined resolution calculus R is the minimum number of resolvents generated by a breadth-first search strategy until the empty clause is obtained.

Remark 8.2.1. For Theorems 8.3 and 8.4, it is assumed that no kind of subsumption algorithm is used together with the resolution search procedure.

Theorem 8.3 (Search Space Improvement by Using Swapped Clause Sets)**.** There exists a sequence of proofs $\varphi_1, \varphi_2, \ldots, \varphi_n, \ldots$ such that:

- $|\varphi_n| = kn$, for some constant k.
- $s_{\mathbf{R}}(C^P_{\varphi_n}) = 2n^2$.
- $s_{\mathbf{R}}(C^W_{\varphi_n|S}) = 2$, for any $\rightsquigarrow_{\oplus\otimes_W}$-normal-form S of $\mathcal{S}_\varphi$.

[a] A much more general and deeper investigation of the complexity of resolution refutation search can be found in [80].

Proof. In the proofs below, subscript indexes are used to distinguish occurrences of the same predicate symbol that occur in different subproofs. Thus, A_1 and A_2 are considered to be the same predicate symbol A, but A_1 occurs in the subproofs φ_i, for $i \geq 1$, and A_2 occurs in the subproofs φ_i, for $i \geq 2$ only.

Let φ_1 be the proof below:

$$\dfrac{\dfrac{A_1 \vdash A_1 \qquad B_1 \vdash B_1}{A_1, B_1 \vdash A_1 \wedge B_1}\wedge_r \qquad \dfrac{\dfrac{A_1 \vdash A_1 \qquad B_1 \vdash B_1}{A_1, B_1 \vdash A_1 \wedge B_1}\wedge_r}{A_1 \wedge B_1 \vdash A_1 \wedge B_1}\wedge_l}{A_1, B_1 \vdash A_1 \wedge B_1}\,cut$$

And let φ_n, for $n > 1$, be:

$$\dfrac{\dfrac{\begin{matrix}\varphi_{n-1}\\ A_1, \ldots, A_{n-1}, B_1, \ldots, B_{n-1} \vdash A_1 \wedge B_1, \ldots, A_{n-1} \wedge B_{n-1}\end{matrix}}{A_1, \ldots, A_{n-1}, B_1, \ldots, B_{n-1} \vdash A_1 \wedge B_1, \ldots, A_{n-1} \wedge B_{n-1}, C}w_r \qquad \dfrac{\dfrac{\dfrac{A_n \vdash A_n \qquad B_n \vdash B_n}{A_n, B_n \vdash A_n \wedge B_n}\wedge_r \qquad \dfrac{\dfrac{A_n \vdash A_n \qquad B_n \vdash B_n}{A_n, B_n \vdash A_n \wedge B_n}\wedge_r}{A_n \wedge B_n \vdash A_n \wedge B_n}\wedge_l}{A_n, B_n \vdash A_n \wedge B_n}\,cut}{C, A_n, B_n \vdash A_n \wedge B_n}w_l}{A_1, \ldots, A_n, B_1, \ldots, B_n \vdash A_1 \wedge B_1, \ldots, A_n \wedge B_n}\,cut_{n-1}$$

Then:

$$C^P_{\varphi_n} = \{\vdash A_1\ ;\ \vdash B_1\ ;\ A_1, B_1 \vdash\ ;\ \vdash A_2\ ;\ \vdash B_2\ ;\ A_2, B_2 \vdash\ ;\ \ldots\ ;\ \vdash A_n\ ;\ \vdash B_n\ ;\ A_n, B_n \vdash\ ;\}$$

And hence, $s_{\mathbf{R}}(C^P_{\varphi_n}) = 2n^2$ (n^2 resolvents with resolved atom A and n^2 resolvents with resolved atom B.).

On the other hand:

$$C^W_{\varphi_n|S_j} = \{\vdash A_j\ ;\ \vdash B_j\ ;\ A_j, B_j \vdash\}$$

And hence, $s_{\mathbf{R}}(C^W_{\varphi_n|S}) = 2$ (one resolvent with resolved atom A and one resolvent with resolved atom B.). □

Theorem 8.4 (Search Space Improvement by Using Refinements)**.** There exists a sequence of proofs $\varphi_1, \varphi_2, \ldots, \varphi_n, \ldots$ such that:

- $|\varphi_n| = kn$, for some constant k.
- $s_{\mathbf{R}}(C^P_{\varphi_n}) = 2n^2$.
- $s_R(C^P_{\varphi_n}) = 2n$, for $R \in \{\mathbf{R_{cl}}, \mathbf{R_{cls}}, \mathbf{R_{scl}}, \mathbf{R_{acl}}\}$.

Proof. Consider again the proofs shown in Theorem 8.3. It was already shown there that $s_{\mathbf{R}}(C^P_{\varphi_n}) = 2n^2$. It is easy to see that, when using any of the refinements, two atoms A_i and A_j (or B_i and B_j) can only be resolved if $i = j$. Therefore, there are only n resolvents with resolved atom A and only n resolvents with resolved atom B, and hence $s_R(C^P_{\varphi_n}) = 2n$, for $R \in \{\mathbf{R_{cl}}, \mathbf{R_{cls}}, \mathbf{R_{scl}}, \mathbf{R_{acl}}\}$. □

Remark 8.2.2. Theorem 8.4 holds not only for profile clause sets, but also for standard clause sets, because both variants of clause sets coincide for the sequence of proofs used in the proof of the theorem. However, the proof of theorem cannot be adapted to work for swapped clause sets, simply because they handle degenerate inferences more cleverly than standard and profile clause sets.

8.3 Proof Sizes and Lengths

In this section, the sizes and lengths of proofs using different variants of CERes are compared.

Theorem 8.5 (Sizes of CERes-Normal-Forms using Definitional Clause Sets)**.** There exists a sequence of proofs $\varphi_1, \varphi_2, \ldots, \varphi_n, \ldots$ such that:

- $|\varphi_n| \leq k_1 4^n$, for some rational constant k_1.
- $|\mathrm{CERes}^O_S(\varphi_n, \delta_S)| \geq k_2 2^{2^n}$, for some rational constant k_2, for any refutation δ_S.
- $|\mathrm{CERes}^D_D(\varphi_n, \delta_D)| \leq 2^{(k_3 n)}$, for some rational constant k_3, for some refutation δ_D.

Proof. Consider the proofs φ_n defined in the proof of the Theorem 8.2. As proved there, $|C^S_{\varphi_n}| \geq k' 2^{2^n}$, for some rational constant k'. Moreover, in any refutation δ_S of $C^S_{\varphi_n}$, every clause of $C^S_{\varphi_n}$ has to be used at least once. Therefore, $|\delta| \geq k' 2^{2^n}$. Since $|\mathrm{CERes}^O_S(\varphi_n, \delta_S)| \geq |\delta_S|$, $|\mathrm{CERes}^O_S(\varphi_n, \delta_S)| \geq k' 2^{2^n}$ too.

It was also proved in Theorem 8.2 that $|C^D_{\varphi_n}| \leq k''2^{(k'''n)}$. It is not too hard to see that a refutation δ_D of $C^D_{\varphi_n}$ could be constructed so that every clause of $C^D_{\varphi_n}$ is used at most once, with the exception of the clause containing only the defined predicate symbol corresponding to the whole struct, which must be used three times. Therefore, there is a rational constant c such that $|\delta_D| \leq 2^{(cn)}$. The sizes of all definitional D-projections is constant. The size of the proper D-projection $\lfloor \varphi_n \rfloor^{D_P}_{\vdash D_{S_{\varphi_n}}}$, which has to be used three times, is linear on the size of φ_n. Hence $|\lfloor \varphi_n \rfloor^{D_P}_{\vdash D_{S_{\varphi_n}}}| \leq c'4^n$, for some rational constant c'. Therefore $|\mathrm{CERes}^D_D(\varphi_n, \delta_D)| \leq 2^{(k_3 n)}$, for some rational constant k_3. □

Theorem 8.6 (Sizes of QCERes-Normal-Forms)**.** There exists a sequence of proofs $\zeta_1, \zeta_2, \ldots, \zeta_n, \ldots$ such that:

- $|\zeta_n| \leq k_1 4^n$, for some rational constant k_1.
- $|\mathrm{CERes}^O_S(\zeta_n, \delta_S)| \geq k_2 2^{2^n}$, for some rational constant k_2, for any refutation δ_S.
- $|\mathrm{QCERes}(\zeta_n, \delta)| \leq k_3 4^n$, for some rational constant k_3, for any refutation δ.

Proof. Let ξ be the proof below:

$$\dfrac{\dfrac{\dfrac{P(\alpha) \vdash P(\alpha)}{(\forall x)P(x) \vdash P(\alpha)}\,\forall_l}{(\forall x)P(x) \vdash (\forall x)P(x)}\,\forall_r \qquad \dfrac{\dfrac{P(c) \vdash P(c)}{(\forall x)P(x) \vdash P(c)}\,\forall_l}{(\forall x)P(x) \vdash (\exists x)P(x)}\,\exists_r}{(\forall x)P(x) \vdash (\exists x)P(x)}\,cut$$

Let ζ_n be the proof below:

$$\dfrac{\xi \qquad \varphi_n}{(\forall x)P(x) \vee ((A_n([]) \wedge B_n([])) \vee (C_n([]) \wedge D_n([]))) \vdash (A_n([]) \wedge B_n([])) \vee (C_n([]) \wedge D_n([])), (\exists x)P(x)}\,\vee_l$$

where φ_n is defined in the proof of Theorem 8.2.

$C^S_{\zeta_n}$ is similar to $C^S_{\varphi_n}$, except for the additional clause $P(c) \vdash$ and the fact that the literal $P(\alpha)$ is merged to every clause stemming from the left branch of φ_n. By essentially the same argument used in the proof of Theorem 8.5, $|\mathrm{CERes}^O_S(\zeta_n, \delta_S)| \geq k_2 2^{2^n}$, for some rational constant k_2, for any refutation δ_S.

The clause set according to QCERes, though, is simply:

$$\{ \vdash P(\alpha)\ ;\ P(c) \vdash \}$$

The only ground refutation of this clause set is:

$$\frac{\vdash P(c) \qquad P(c) \vdash}{\vdash} r$$

And the QCERes-normal-form QCERes(ζ_n, δ) is:

$$\dfrac{\dfrac{\dfrac{P(c) \vdash P(c)}{(\forall x)P(x) \vdash P(c)} \forall_l \qquad \varphi_n}{(\forall x)P(x) \vee T \vdash T, P(c)} \vee_l \qquad \dfrac{P(c) \vdash P(c)}{P(c) \vdash (\exists x)P(x)} \exists_r}{(\forall x)P(x) \vee T \vdash T, (\exists x)P(x)} cut$$

where:

$$T \doteq ((A_n([]) \wedge B_n([])) \vee (C_n([]) \wedge D_n([])))$$

Clearly, the size of QCERes(ζ_n, δ) is therefore linear on the size of $zeta_n$. Therefore, $|\text{QCERes}(\zeta_n, \delta)| \leq k_3 4^n$. □

Definition 8.3.1 (A Non-Elementary Function)**.** The function $s()$ is defined as follows:

$$s(n) \doteq \begin{cases} 1 & \text{, if } n = 0 \\ 2^{s(n-1)} & \text{, otherwise} \end{cases}$$

Conjecture 8.1 (Sizes of CERes-Normal-Forms using Profile and Swapped Clause Sets)**.** There exists a sequence of proofs $\varphi_1, \varphi_2, \ldots, \varphi_n, \ldots$ such that:

- $|\varphi_n| \leq 2^{kn}$, for some rational constant k.
- $|\text{CERes}^O_W(\varphi_n, \delta_W)| \geq k' s(n)$, for some rational constant k', for any refutation δ_W.
- $|\text{CERes}^O_P(\varphi_n, \delta_P)| = k''$, for some constant k', for some refutation δ_P.

Proof. Let φ_n be a proof of the following form:

$$\dfrac{\dfrac{\dfrac{\psi^l_n}{\Gamma^l_n \vdash \Delta^l_n}}{\Gamma^l_n \vdash \Delta^l_n, A} w_r \qquad \dfrac{\dfrac{\psi^r_n}{\Gamma^r_n \vdash \Delta^r_n}}{\Gamma^r_n \vdash \Delta^r_n, B} w_r}{\Gamma^l_n, \Gamma^r_n \vdash \Delta^l_n, \Delta^r_n, A \wedge B} \wedge_r$$

Let C_l be the subset of C^P_φ having the clauses stemming from ψ^l_n, and let C_r be the subset of C^P_φ having the clauses stemming from ψ^r_n.

Due to the degenerate inference $\wedge_r$, φ_n has two swapped clause sets: $C^W_{\varphi_n|S_1} = C_l$ and $C^W_{\varphi_n|S_2} = C_r$.

Assuming that both ψ^l_n and ψ^r_n admit only non-elementary cut-elimination, any refutation δ_W of any of the two swapped clause sets will have non-elementary size.

On the other hand, $C^P_\varphi = C_l \cup C_r$, and hence it might be possible, if ψ^l_n and ψ^r_n are in a certain sense partially dual, that there are clauses $c_l \in C_l$ and $c_r \in C_r$ such that the following refutation of C^P_φ exists:

$$\frac{c_l \qquad c_r}{\vdash}\,r$$

However, the construction of ψ_n^l and ψ_n^r has not been done yet. □

8.4 Proof Compression by Cut-Introduction

This section discusses a conjecture regarding the amount of compression that can be achieved by the CIRes method.

Conjecture 8.2 (Compression via CIRes)**.** There exists a sequence of sequents $s_1, s_2, \ldots, s_n, \ldots$ such that: such that:

- $|\varphi_n| \geq 2^{2^{(cn)}}$, for some rational constant c.
- $|\mathrm{CIRes}_W^O(\varphi_n, |) \leq 2^{(km)}$, for some rational constant k.

where $\varphi_1, \varphi_2, \ldots, \varphi_n, \ldots$ is the sequence of smallest cut-free proofs having, respectively, $s_1, s_2, \ldots, s_n, \ldots$ as their end-sequents.

Proof. The idea for this proof is to use a sequence of sets of clauses given in [26]:

$$T_n = \{\pm P^1 \vee \pm P^2_{\pm} \vee \pm P^3_{\pm\pm} \vee \ldots \vee \pm P^n_{\pm\ldots\pm}\}$$

where ± preceding a predicate symbol is either empty or ¬, and the string of ± in the subscript of a predicate symbol corresponds to the values taken by ± before the previous predicate symbols. For example:

$$T_2 = \{P^1 \vee P^2_+, \neg P^1 \vee P^2_-, P^1 \vee \neg P^2_+, \neg P^1 \vee \neg P^2_-\}$$

It is known (from [26]) that:

- There are refutations δ_n of T_n such that $|\delta_n| \leq 2^{(k_1 m)}$.
- For any analytic tableaux refutation ψ_n of T_n $|\psi_n| \geq 2^{2^{(c_1 n)}}$).

Example 7.2 shows that CIRes is able to do the desired compression for T_2. And there is no reason to believe that it would not work for T_n in general. □

Chapter 9

Applications of Cut

In the previous chapters, the cut rule and methods for its elimination and introduction were studied in a quite abstract, technical and theoretical way. Even though motivations for the study of cut were given in Section 3.1, they were still mostly theoretical and restricted to the field of proof theory. The purpose of this chapter is to illustrate that proof theory, and hence also this thesis, is actually not as isolated and closed in itself as the previous chapters might suggest. The main aim is to suggest that proof theory is highly cross-disciplinary, since it can serve as a tool for foundational investigations of a wide range of scientific disciplines.

Applications of proof theory to linguistics, for example, are well known [23, 88]. In the following sections, informal correspondences between proof theory and a few other scientific disciplines are discussed. It is shown how typical concepts, methods and tasks in these disciplines can be understood from a proof-theoretical perspective as cuts, cut-elimination, cut-introduction, proof search and extraction of Herbrand disjunctions.

9.1 Physics: Energy Conservation as Cut

To solve problems of physics, certain invariants (such as energy) are frequently used. This is so because solving problems by using a derived principle (such as the principle of energy conservation) is usually easier than solving them by using the most basic physical laws. This section intends to exemplify how problem solution can generally be seen from a proof-theoretic perspective in which the use of derived principles correspond to an implicit use of cut. The following simple problem of physics shall be considered:

> An object of mass m is dropped from height h_0 and with initial velocity equal to zero. The only force acting on the object is the force of gravity (with an intensity mg). What is the velocity of the object when its height is equal to zero?

A typical solution (Solution 1) to this problem uses the principle of energy conservation, as follows:

1. Let t_f be the time when the object reaches height zero.
2. According to the principle of energy conservation, $e(t_f) = e(0)$, i.e. the energy at t_f is equal to the initial energy.
3. Hence, by definition of gravitational potential energy in a uniform gravitational field and by definition of kinetic energy, $mgh(t_f) + m\frac{\dot{h}(t_f)^2}{2} = mgh(0) + m\frac{\dot{h}(0)^2}{2}$.
4. According to the initial conditions, $h(0) = h_0$ and $\dot{h}(0) = 0$. Moreover, by assumption, $h(t_f) = 0$. Therefore, $m\frac{\dot{h}(t_f)^2}{2} = mgh_0$.
5. Clearly, $m\frac{\dot{h}(t_f)^2}{2} = mgh_0$ holds if $\dot{h}(t_f) = \sqrt{2gh_0}$ or $\dot{h}(t_f) = -\sqrt{2gh_0}$.
6. Since it is known that the height is decreasing and velocity is the rate of change of height, the only solution is: $\dot{h}(t_f) = -\sqrt{2g_0}$.

Another solution (Solution 2) computes the velocity as a function of time by integrating the acceleration produced by the gravitational force. Then it determines the time when the object reaches height zero, and computes the velocity at that time. The details are shown below:

1. According to Newton's second law of motion, $f(t) = m\ddot{h}(t)$ at any time t. Moreover, the uniform gravitational field produces a force $f(t) = -mg$. Hence, $\ddot{h}(t) = -g$.
2. By integration, $\dot{h}(t) = -gt + \dot{h}(0)$.
3. According to the initial conditions, $\dot{h}(0) = 0$, and hence $\dot{h}(t) = -gt$.
4. By integration again, $h(t) = -g\frac{t^2}{2} + h(0)$.

5. According to the initial conditions, $h(0) = h_0$, and hence $h(t) = -g\frac{t^2}{2} + h_0$.

6. For $h(t_f) = 0$ to hold, it must be the case that $t_f = \sqrt{\frac{2h_0}{g}}$.

7. Hence $\dot{h}(t_f) = -g\sqrt{\frac{2h_0}{g}}$, which can be simplified to $\dot{h}(t_f) = -\sqrt{2gh_0}$.

Solution 2 is more basic in the sense that it uses only the basic physical laws of motion (here assumed to be Newton's laws of motion) and of uniform gravitational fields. Solution 1, on the other hand, assumes that energy is conserved, without actually proving it from Newton's basic laws.

In order to view problem solving from a proof theoretic perspective, it is necessary to formalize problem solving as theorem proving. In the example above, the problem can be stated as the following theorem to be proved:

$$(\exists t')(h(t') = 0 \wedge (\exists v)\,\dot{h}(t') = v)$$

Solving the given problem then consists of finding a proof of the theorem above such that v is instantiated by a ground term. Interestingly, formalizing the problem as a theorem to be proved enforces the explicit mention of the hidden assumption that the height eventually becomes zero; otherwise the variable t' would be free and the theorem would be open.

Solution 1 can be easily formalized as the proof φ_1 below. To keep the size of the proof small enough, the inference rule s_l was used to encode simple algebraic simplifications. Clearly, this rule is admissible in the sense that it could be replaced by several applications of more formal and more basic inference rules such as paramodulation and appropriate axioms.

$$
\begin{prooftree}
\AxiomC{$h(t_f) = 0 \vdash h(t_f) = 0$}
\AxiomC{$\dot{h}(t_f) = -\sqrt{2gh_0} \vdash \dot{h}(t_f) = -\sqrt{2gh_0}$}
\RightLabel{$\exists_r$}
\UnaryInfC{$\dot{h}(t_f) = \sqrt{2gh_0} \vdash (\exists v)\, \dot{h}(t_f) = v$}
\RightLabel{s_l}
\UnaryInfC{$mg0 + m\frac{\dot{h}(t_f)^2}{2} = mgh_0 + m\frac{0^2}{2} \vdash (\exists v)\, \dot{h}(t_f) = v$}
\doubleLine
\RightLabel{w_l}
\UnaryInfC{$h(t_f) = 0, h(0) = h_0, \dot{h}(0) = 0, mg0 + m\frac{\dot{h}(t_f)^2}{2} = mgh_0 + m\frac{0^2}{2} \vdash (\exists v)\, \dot{h}(t_f) = v$}
\doubleLine
\RightLabel{$=_l$}
\UnaryInfC{$h(t_f) = 0, h(0) = h_0, \dot{h}(0) = 0, mgh(t_f) + m\frac{\dot{h}(t_f)^2}{2} = mgh(0) + m\frac{\dot{h}(0)^2}{2} \vdash (\exists v)\, \dot{h}(t_f) = v$}
\doubleLine
\RightLabel{d_l}
\UnaryInfC{$h(t_f) = 0, h(0) = h_0, \dot{h}(0) = 0, e(t_f) = e(0) \vdash (\exists v)\, \dot{h}(t_f) = v$}
\doubleLine
\RightLabel{$\forall_l$}
\UnaryInfC{$h(t_f) = 0, h(0) = h_0, \dot{h}(0) = 0, (\forall t_i)(\forall t_j)\, e(t_i) = e(t_j) \vdash (\exists v)\, \dot{h}(t_f) = v$}
\doubleLine
\RightLabel{$\forall_l$}
\UnaryInfC{$h(t_f) = 0, h(0) = h_0, \dot{h}(0) = 0, (\forall t_i)(\forall t_j)\, e(t_i) = e(t_j) \vdash (\exists v)\, \dot{h}(t_f) = v$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$h(t_f) = 0, h(t_f) = 0, h(0) = h_0, \dot{h}(0) = 0, (\forall t_i)(\forall t_j)\, e(t_i) = e(t_j) \vdash h(t_f) = 0 \wedge (\exists v)\, \dot{h}(t_f) = v$}
\RightLabel{c_l}
\UnaryInfC{$h(t_f) = 0, h(0) = h_0, \dot{h}(0) = 0, (\forall t_i)(\forall t_j)\, e(t_i) = e(t_j) \vdash h(t_f) = 0 \wedge (\exists v)\, \dot{h}(t_f) = v$}
\RightLabel{$\exists_r$}
\UnaryInfC{$h(t_f) = 0, h(0) = h_0, \dot{h}(0) = 0, (\forall t_i)(\forall t_j)\, e(t_i) = e(t_j) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)$}
\RightLabel{$\exists_l$}
\UnaryInfC{$(\exists t)\, h(t) = 0, h(0) = h_0, \dot{h}(0) = 0, (\forall t_i)(\forall t_j)\, e(t_i) = e(t_j) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$(\exists t)\, h(t) = 0, h(0) = h_0 \wedge \dot{h}(0) = 0, (\forall t_i)(\forall t_j)\, e(t_i) = e(t_j) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)$}
\RightLabel{d_l}
\UnaryInfC{$Fall, Init, EnergyConservation \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)$}
\end{prooftree}
$$

Solution 2 can be formalized as the following proof φ_2. Again, due to size constraints, integration has been encoded only semi-formally by means of the inference rule $\int_l$.

$$
\begin{prooftree}
\AxiomC{$h\left(\sqrt{\frac{2h_0}{g}}\right) = 0 \vdash h\left(\sqrt{\frac{2h_0}{g}}\right) = 0$}
\RightLabel{w_l}
\UnaryInfC{$h(0) = h_0, h\left(\sqrt{\frac{2h_0}{g}}\right) = 0 \vdash h\left(\sqrt{\frac{2h_0}{g}}\right) = 0$}
\AxiomC{$\dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = -\sqrt{2gh_0} \vdash \dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = -\sqrt{2gh_0}$}
\RightLabel{$\exists_r$}
\UnaryInfC{$\dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = -\sqrt{2gh_0} \vdash (\exists v)\, \dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = v$}
\RightLabel{s_l}
\UnaryInfC{$\dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = -g\sqrt{\frac{2h_0}{g}} \vdash (\exists v)\, \dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = v$}
\RightLabel{$\forall_l$}
\UnaryInfC{$(\forall t)(\dot{h}(t) = -gt) \vdash (\exists v)\, \dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = v$}
\RightLabel{w_l}
\UnaryInfC{$\dot{h}(0) = 0, (\forall t)(\dot{h}(t) = -gt) \vdash (\exists v)\, \dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = v$}
\RightLabel{$\wedge_r$}
\BinaryInfC{$h(0) = h_0, \dot{h}(0) = 0, h\left(\sqrt{\frac{2h_0}{g}}\right) = 0, (\forall t)(\dot{h}(t) = -gt) \vdash h\left(\sqrt{\frac{2h_0}{g}}\right) = 0 \wedge (\exists v)\, \dot{h}\left(\sqrt{\frac{2h_0}{g}}\right) = v$}
\RightLabel{$\exists_r$}
\UnaryInfC{$h(0) = h_0, \dot{h}(0) = 0, h\left(\sqrt{\frac{2h_0}{g}}\right) = 0, (\forall t)(\dot{h}(t) = -gt) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)$}
\RightLabel{s_l}
\UnaryInfC{$h(0) = h_0, \dot{h}(0) = 0, h\left(\sqrt{\frac{2h_0}{g}}\right) = -g\frac{\left(\sqrt{\frac{2h_0}{g}}\right)^2}{2} + h_0, (\forall t)(\dot{h}(t) = -gt) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)$}
\RightLabel{$\forall_l$}
\UnaryInfC{$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(h(t) = -g\frac{t^2}{2} + h_0), (\forall t)(\dot{h}(t) = -gt) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)$}
\RightLabel{$=_l$}
\UnaryInfC{$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(h(t) = -g\frac{t^2}{2} + h(0)), (\forall t)(\dot{h}(t) = -gt) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)$}
\RightLabel{$\int_l$}
\UnaryInfC{$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(\dot{h}(t) = -gt), (\forall t)(\dot{h}(t) = -gt) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)$}
\RightLabel{c_l}
\UnaryInfC{$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(\dot{h}(t) = -gt) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)$}
\RightLabel{s_l}
\UnaryInfC{$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(\dot{h}(t) = -gt + 0) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)$}
\RightLabel{$=_l$}
\UnaryInfC{$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(\dot{h}(t) = -gt + \dot{h}(0)) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)$}
\RightLabel{$\int_l$}
\UnaryInfC{$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(\ddot{h}(t) = -g) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)$}
\RightLabel{s_l}
\UnaryInfC{$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(m\ddot{h}(t) = -mg) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)$}
\RightLabel{d_l}
\UnaryInfC{$h(0) = h_0, \dot{h}(0) = 0, (\forall t)(f(t) = -mg) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)$}
\RightLabel{$\wedge_l$}
\UnaryInfC{$h(0) = h_0 \wedge \dot{h}(0) = 0, (\forall t)(f(t) = -mg) \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)$}
\RightLabel{d_l}
\doubleLine
\UnaryInfC{$Init, Gravity \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)$}
\end{prooftree}
$$

As expected φ_1 is not only smaller than φ_2, but also simpler, since it does not need to use integration. Furthermore, while in φ_2 the time when the object hits the floor has to be computed explicitly (i.e. t' is instantiated to a ground term), in φ_1 this is not so (i.e. t' is instantiated to a variable).

The idea of formalizing a problem as a theorem and in such a way that its solution is in the instances used for the quantified variables in the proof is the fundamental principle behind the logic programming paradigm, of which Prolog [107] is the most prominent language. However, while logic programming languages usually use refined resolution calculi, which are suitable for proof search, here a high-level sequent calculus was used, since the aim was not to search for proofs automatically but to formalize existing solutions as proofs in the simplest and most natural possible way. The fact that both φ_1 and φ_2 use sophisticated simplification rules and φ_2 even uses an integration rule shows that even such a simple physics problem would be hard to solve (and even to state) in pure logic programming. It illustrates the practical necessity of deduction modulo [36] and the integration of logic programming (or automated deduction in general) with computer algebra systems [114].

Solution 1 implicitly uses cuts, because *EnergyConservation* and *Fall* are not considered to be basic laws of physics. In principle, φ_1 must be composed with a proof φ_E of *EnergyConservation* and a proof φ_F of *Fall*. This is done with two cuts, as shown in the proof φ below:

$$
\dfrac{\dfrac{\begin{array}{c}\varphi_F\\ \mathit{Init}, \mathit{Gravity} \vdash \mathit{Fall}\end{array} \qquad \dfrac{\begin{array}{c}\varphi_E\\ \mathit{Gravity} \vdash \mathit{EnergyConservation}\end{array} \qquad \begin{array}{c}\varphi_P\\ \mathit{Init}, \mathit{Fall}, \mathit{EnergyConservation} \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)\end{array}}{\mathit{Init}, \mathit{Gravity}, \mathit{Fall} \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)}}{\mathit{Init}, \mathit{Init}, \mathit{Gravity}, \mathit{Gravity} \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)}}{\mathit{Init}, \mathit{Gravity} \vdash (\exists t')(h(t') = 0 \wedge (\exists v)\, \dot{h}(t') = v)}\ c_l
$$

Where φ_F is the proof below, proving that the object will eventually fall to height zero under the gravitational field and the initial conditions specified in the description of the problem:

$$
\begin{array}{cl}
h(0) = h_0, h\left(\sqrt{\frac{2h_0}{g}}\right) = 0 \vdash h\left(\sqrt{\frac{2h_0}{g}}\right) = 0 & \\ \hline
h\left(\sqrt{\frac{2h_0}{g}}\right) = 0 \vdash (\exists t')\, h(t') = 0 & \exists_r \\ \hline
h\left(\sqrt{\frac{2h_0}{g}}\right) = -g\frac{\left(\sqrt{\frac{2h_0}{g}}\right)^2}{2} + h_0 \vdash (\exists t')\, h(t') = 0 & s_l \\ \hline
(\forall t)(h(t) = -g\frac{t^2}{2} + h_0) \vdash (\exists t')\, h(t') = 0 & \forall_l \\ \hline
h(0) = h_0, (\forall t)(h(t) = -g\frac{t^2}{2} + h_0) \vdash (\exists t')\, h(t') = 0 & w_l \\ \hline
h(0) = h_0, (\forall t)(h(t) = -g\frac{t^2}{2} + h(0)) \vdash (\exists t')\, h(t') = 0 & =_l \\ \hline
h(0) = h_0, (\forall t)(\dot{h}(t) = -gt) \vdash (\exists t')\, h(t') = 0 & \int_l \\ \hline
h(0) = h_0, \dot{h}(0) = 0, (\forall t)(\dot{h}(t) = -gt + 0) \vdash (\exists t')\, h(t') = 0 & w_l \\ \hline
h(0) = h_0, \dot{h}(0) = 0, (\forall t)(\dot{h}(t) = -gt + 0) \vdash (\exists t')\, h(t') = 0 & s_l \\ \hline
h(0) = h_0, \dot{h}(0) = 0, (\forall t)(\dot{h}(t) = -gt + \dot{h}(0)) \vdash (\exists t')\, h(t') = 0 & =_l \\ \hline
h(0) = h_0, \dot{h}(0) = 0, (\forall t)(\ddot{h}(t) = -g) \vdash (\exists t')\, h(t') = 0 & \int_l \\ \hline
h(0) = h_0, \dot{h}(0) = 0, (\forall t)(m\ddot{h}(t) = -mg) \vdash (\exists t')\, h(t') = 0 & s_l \\ \hline
h(0) = h_0, \dot{h}(0) = 0, (\forall t)(f(t) = -mg) \vdash (\exists t')\, h(t') = 0 & d_l \\ \hline
h(0) = h_0 \wedge \dot{h}(0) = 0, (\forall t)(f(t) = -mg) \vdash (\exists t')\, h(t') = 0 & \wedge_l \\ \hline\hline
\mathit{Init}, \mathit{Gravity} \vdash \mathit{Fall} & d
\end{array}
$$

And φ_E is the proof that energy is conserved in a uniform gravitational field:

$$\vdash gh(0) + \frac{\dot{h}(0)^2}{2} = gh(0) + \frac{\dot{h}(0)^2}{2}$$

$$\overline{\overline{\hphantom{xxxxxxxxxxxxxxxxxxxx}}}\ w_l$$

$$(\dot{h}(\alpha) = -g\alpha + \dot{h}(0)), (h(t) = -g\frac{\alpha^2}{2} + \dot{h}(0)\alpha + h(0)), (\dot{h}(\beta) = -g\beta + \dot{h}(0)), (h(\beta) = -g\frac{\beta^2}{2} + \dot{h}(0)\beta + h(0)) \vdash gh(0) + \frac{\dot{h}(0)^2}{2} = gh(0) + \frac{\dot{h}(0)^2}{2}$$

$$\overline{\overline{\hphantom{xxxxxxxxxxxxxxxxxxxx}}}\ =^*_r, s^*_r$$

$$(\dot{h}(\alpha) = -g\alpha + \dot{h}(0)), (h(t) = -g\frac{\alpha^2}{2} + \dot{h}(0)\alpha + h(0)), (\dot{h}(\beta) = -g\beta + \dot{h}(0)), (h(\beta) = -g\frac{\beta^2}{2} + \dot{h}(0)\beta + h(0)) \vdash gh(\alpha) + \frac{\dot{h}(\alpha)^2}{2} = gh(\beta) + \frac{\dot{h}(\beta)^2}{2}$$

$$\overline{\overline{\hphantom{xxxxxxxxxxxxxxxxxxxx}}}\ \forall_l$$

$$(\forall t)(\dot{h}(t) = -gt + \dot{h}(0)), (\forall t)(h(t) = -g\frac{t^2}{2} + \dot{h}(0)t + h(0)), (\forall t)(\dot{h}(t) = -gt + \dot{h}(0)), (\forall t)(h(t) = -g\frac{t^2}{2} + \dot{h}(0)t + h(0)) \vdash gh(\alpha) + \frac{\dot{h}(\alpha)^2}{2} = gh(\beta) + \frac{\dot{h}(\beta)^2}{2}$$

$$\overline{\overline{\hphantom{xxxxxxxxxxxxxxxxxxxx}}}\ c_l$$

$$(\forall t)(\dot{h}(t) = -gt + \dot{h}(0)), (\forall t)(h(t) = -g\frac{t^2}{2} + \dot{h}(0)t + h(0)) \vdash gh(\alpha) + \frac{\dot{h}(\alpha)^2}{2} = gh(\beta) + \frac{\dot{h}(\beta)^2}{2}$$

$$\overline{\hphantom{xxxxxxxxxxxxxxxxxxxx}}\ c_l$$

$$(\forall t)(\dot{h}(t) = -gt + \dot{h}(0)) \vdash gh(\alpha) + \frac{\dot{h}(\alpha)^2}{2} = gh(\beta) + \frac{\dot{h}(\beta)^2}{2}$$

$$\overline{\overline{\hphantom{xxxxxxxxxxxxxxxxxxxx}}}\ \int_l$$

$$(\forall t)(\ddot{h}(t) = -g) \vdash gh(\alpha) + \frac{\dot{h}(\alpha)^2}{2} = gh(\beta) + \frac{\dot{h}(\beta)^2}{2}$$

$$\overline{\overline{\hphantom{xxxxxxxxxxxxxxxxxxxx}}}\ s$$

$$(\forall t)(m\ddot{h}(t) = -mg) \vdash mgh(\alpha) + m\frac{\dot{h}(\alpha)^2}{2} = mgh(\beta) + m\frac{\dot{h}(\beta)^2}{2}$$

$$\overline{\overline{\hphantom{xxxxxxxxxxxxxxxxxxxx}}}\ d_r$$

$$(\forall t)(m\ddot{h}(t) = -mg) \vdash e(\alpha) = e(\beta)$$

$$\overline{\hphantom{xxxxxxxxxxxxxxxxxxxx}}\ d_l$$

$$(\forall t)(f(t) = -mg) \vdash e(\alpha) = e(\beta)$$

$$\overline{\overline{\hphantom{xxxxxxxxxxxxxxxxxxxx}}}\ \forall_r$$

$$(\forall t)(f(t) = -mg) \vdash e(\alpha) = e(\beta)$$

$$\overline{\overline{\hphantom{xxxxxxxxxxxxxxxxxxxx}}}\ \forall_r$$

$$(\forall t)(f(t) = -mg) \vdash (\forall t_i)(\forall t_j)\ e(t_i) = e(t_j)$$

$$\overline{\overline{\hphantom{xxxxxxxxxxxxxxxxxxxx}}}\ d$$

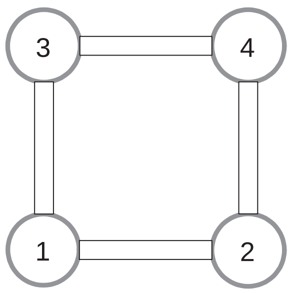
$$Gravity \vdash EnergyConservation$$

The proofs discussed above illustrate that an essential task of theoretical science is to invent and discover concepts, such as the principle of energy conservation, that turn out to be useful cuts. In other words, a significant part of the usual scientific activity can be described as cut-introduction. Different levels of Science can also be related to cuts. When reasoning about chemical reactions, the Chemistry laws used can be seen as cuts derivable from more fundamental laws of Quantum Mechanics. Major scientific revolutions, on the other hand, usually involve radical transformations of proofs. For example, while in the proofs above, which assume Newtonian Mechanics, energy is a mere derived concept, in Relativistic Mechanics energy is equivalent to mass, and hence a much more fundamental concept.

9.2 Robotics and Agent Theory: Plan Reuse as Cut

In this section, a simple example of a mobile robot in an environment with rooms and corridors is used to investigate tasks like planning from a proof-theoretical perspective.

The picture below depicts the environment where the robot is located. It consists of four rooms connected by corridors.



The robot is assumed to have acquired, by sensory experience, a set of *beliefs* regarding the existence of rooms and corridors in the environment:

$$\text{Beliefs} \doteq \left\{ \begin{array}{l} E(1), E(2), E(3), E(4), \\ C((1,e),(2,w)), C((3,e),(4,w)), \\ C((1,n),(3,s)), C((2,n),(4,s)) \end{array} \right\}$$

where, for example, $E(1)$ can be interpreted as "x is a room", and $C((1,e),(2,w))$ can be interpreted as "there is a corridor connecting the east of room 1 with the west of room 2.

Moreover, the robot assumes, in the form of rules of inference, that it can move from one room to another by moving in a certain direction, if there is a corridor connecting the rooms with an appropriate direction:

$$\frac{\vdash E(x) \wedge E(y) \wedge C((x,z),(y,\bar{z}))}{\vdash M(x,y,z)}\, M_r$$

$$\frac{\vdash E(x) \wedge E(y) \wedge C((y,\bar{z}),(x,z))}{\vdash M(x,y,z)}\, M_r$$

$$\frac{\vdash E(x) \wedge E(x)}{\vdash M(x,x,o)}\, M_r$$

$$\frac{E(x) \wedge E(y) \wedge C((x,z),(y,\bar{z})) \vdash}{M(x,y,z) \vdash}\, M_l$$

$$\frac{E(x) \wedge E(y) \wedge C((y,\bar{z}),(x,z)) \vdash}{M(x,y,z) \vdash}\, M_l$$

$$\frac{E(x) \wedge E(x) \vdash}{M(x,x,o) \vdash}\, M_l$$

The robot can also reason about composed movements:

$$\frac{\vdash (\exists z)(M(x,z,s_1) \wedge M(z,y,s_2))}{\vdash M(x,y,s_1 : s_2)}\, M'_r$$

$$\frac{(\exists z)(M(x,z,s_1) \wedge M(z,y,s_2)) \vdash}{M(x,y,s_1 : s_2) \vdash}\, M'_l$$

So, based on its beliefs and on its reasoning capabilities, the robot *knows* that it can go from room 1 to room 2, because it can construct the following *proof* (*justification*) φ_{12}:

$$\frac{\dfrac{\vdash E(1) \qquad \vdash E(2) \qquad \vdash C((1,e),(2,w))}{\vdash E(1) \wedge E(2) \wedge C((1,e),(2,w))}\, \wedge_r}{\vdash M(1,2,e)}\, M_r$$

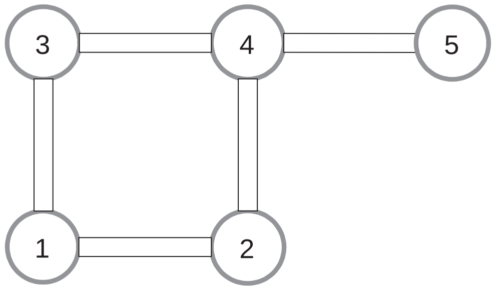
It also knows that it can move from room 1 to room 4, because of proof φ_{14} below:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdash E(1) \qquad \vdash E(2) \qquad \vdash C((1,e),(2,w))}{\vdash E(1) \wedge E(2) \wedge C((1,e),(2,w))}\,\wedge_r}{\vdash M(1,2,e)}\,M_r \qquad \dfrac{\dfrac{\vdash E(2) \qquad \vdash E(4) \qquad \vdash C((2,n),(4,s))}{\vdash E(2) \wedge E(4) \wedge C((2,n),(4,s))}\,\wedge_r}{\vdash M(2,4,n)}\,M_r}{\vdash M(1,2,e) \wedge M(2,4,n)}\,M_r}{\vdash (\exists z)(M(1,z,e) \wedge M(z,4,n))}\,\exists_r}{\vdash M(1,4,e:n)}\,M'_r
$$

In fact, the robot knows that, wherever he is, he can go wherever he wants. Essentially, the robot has the following proof φ stored in his memory:

$$
\dfrac{
\dfrac{\dfrac{\dfrac{\vdash E(1) \qquad \vdash E(2) \qquad \vdash C((1,e),(2,w))}{\vdash E(1) \wedge E(2) \wedge C((1,e),(2,w))}\,\wedge_r}{\vdash M(1,2,r)}\,M_r}{\vdash (\exists p)M(1,2,p)}\,\exists_r
\qquad \ldots \qquad
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdash E(1) \qquad \vdash E(2) \qquad \vdash C((1,e),(2,w))}{\vdash E(1) \wedge E(2) \wedge C((1,e),(2,w))}\,\wedge_r}{\vdash M(1,2,e)}\,M_r \qquad \dfrac{\dfrac{\vdash E(2) \qquad \vdash E(4) \qquad \vdash C((2,n),(4,s))}{\vdash E(2) \wedge E(4) \wedge C((2,n),(4,s))}\,\wedge_r}{\vdash M(2,4,n)}\,M_r}{\vdash M(1,2,e) \wedge M(2,4,n)}\,M_r}{\vdash (\exists z)(M(1,z,e) \wedge M(z,4,n))}\,\exists_r}{\vdash M(1,4,e:n)}\,M_r'}{\vdash (\exists p)M(1,4,p)}\,\exists_r
}{\vdash \bigwedge_{i \in \{1,2,3,4\}} \bigwedge_{j \in \{1,2,3,4\}} (\exists p)M(i,j,p)}\,\bigwedge_r
$$

Now the environment changes. A new room and a new corridor are added:



And the robot is informed of the changes and updates its beliefs accordingly:

$$\text{Beliefs} \doteq \left\{ \begin{array}{l} E(1), E(2), E(3), E(4), E(5), \\ C((1,e),(2,w)), C((3,e),(4,w)), \\ C((1,n),(3,s)), C((2,n),(4,s)), C((4,e),(5,w)) \end{array} \right\}$$

Now assume that the robot has the *desire* to go from his current room 1 to the new room 5. Naturally, if a desire ever becomes a goal, it is because the agent knows that is possible to realize the desire. So, the agent will be interested in proving the possibility of going from room 1 to room 5. In doing so, he can of course use his previous knowledge that he can move everywhere in the four previous rooms, as shown in proof φ'_{15}:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{M(1,4,\alpha) \vdash M(1,4,\alpha) \qquad \dfrac{\dfrac{\vdash E(4) \qquad \vdash E(5) \qquad \vdash C((4,e),(5,w))}{\vdash E(4) \wedge E(5) \wedge C((4,e),(5,w))}\wedge_r}{\vdash M(4,5,e)} M_r}{M(1,4,\alpha) \vdash M(1,4,\alpha) \wedge M(4,5,e)}\wedge_r}{M(1,4,\alpha) \vdash (\exists z)(M(1,z,\alpha) \wedge M(z,5,e))}\exists_r}{M(1,4,\alpha) \vdash M(1,5,\alpha : e)} M'_r}{M(1,4,\alpha) \vdash (\exists p)M(1,5,p)}\exists_r}{(\exists p)M(1,4,p) \vdash (\exists p)M(1,5,p)}\exists_l}{\bigwedge_{i\in\{1,2,3,4\}} \bigwedge_{j\in\{1,2,3,4\}} (\exists p)M(i,j,p) \vdash (\exists p)M(1,5,p)} \bigwedge_l$$

Strictly, the robot has used cut to show that its desire is realizable, as shown in proof φ_{15}:

$$\dfrac{\begin{array}{c}\varphi \\ \vdash \bigwedge_{i\in\{1,2,3,4\}} \bigwedge_{j\in\{1,2,3,4\}} (\exists p)M(i,j,p)\end{array} \qquad \begin{array}{c}\varphi'_{15} \\ \bigwedge_{i\in\{1,2,3,4\}} \bigwedge_{j\in\{1,2,3,4\}} (\exists p)M(i,j,p) \vdash (\exists p)M(1,5,p)\end{array}}{\vdash (\exists p)M(1,5,p)} cut$$

After knowing that the desire is realizable, the robot must come to a *plan* to realize it. In other words, the robot is now not only interested in showing that $(\exists p)M(1,5,p)$ holds, but also in a finding a ground term p' that instantiates the quantified p in such a way that $M(1,5,p')$ also holds. As p' is a sequence of atomic actions, it can be considered a plan. Proof-theoretically, planning correspond to extraction of the terms of Herbrand-disjunctions from proofs of the realizability of the corresponding desire.

However, proofs with quantified cut-formulas do not admit this extraction. For example, in φ_{15}, p is instantiated by $\alpha : e$, which is not a ground term. The eigenvariable α does not tell which actions should be taken. $\alpha : e$ can be understood as a *high-level underspecified plan* composed of lower level plans α to go from room 1 to a certain room x and a lowest level plan e to go east from there. α should be substituted by an actual plan. One approach would be to *replan* by reproving $(\exists p)M(1,5,p)$ without cuts from scratch, so that a ground instance for p could be extracted from the cut-free proof. A better approach, however, is simply to eliminate the cut in the proof φ_{15}. The process of cut-elimination substitutes α by a ground plan to go from room 1 to room 4, thus effectively composing a previously known plan with a new small action. Cut-elimination corresponds to plan reuse. The proof φ^*_{15} shows the result of cut-elimination.

φ^*_{15}:

$$
\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\varphi_{14}\\ \vdash M(1,4,e:n)\end{array} \qquad \dfrac{\dfrac{\vdash E(4) \quad \vdash E(5) \quad \vdash C((4,e),(5,w))}{\vdash E(4) \wedge E(5) \wedge C((4,e),(5,w))}\wedge_r}{\vdash M(4,5,e)}M_r}{\vdash M(1,4,e:n) \wedge M(4,5,e)}\wedge_r}{\vdash (\exists z)(M(1,z,e:n) \wedge M(z,5,e))}M'_r}{\vdash M(1,5,e:n:e)}M'_r}{\vdash (\exists p)M(1,5,p)}\exists_r
$$

9.3 Machine Learning: Decision Tree as Cut

Machine learning [87] is concerned with the study and development of algorithms that allow computers to *learn* from *data*. Usually, such algorithms try to construct structures (e.g. decision trees, neural networks, induced logic programs, clusters, bayesian networks, support vector machines ...) that are more compact than the data, and which can be used instead of the data to answer certain particular *queries* of interest about instances of the data (e.g. queries about the class to which a particular instance of the data belongs, in the case of *supervised learning* algorithms).

More importantly, these compact structures learned from training data set can usually be used to accurately answer queries about previously unseen instances that were not in the training data set, assuming that the training data contains a statistically significant sample of the whole data. In other words, it is possible to *reason inductively* about the whole data based on experience limited to a small training subset of the data. It is possible and easy to extrapolate the compact structure that has been learned from the training data set to the whole data set.

Clearly, a notion of *inference* is ubiquitous to machine learning, as one can argue that learned compact structures are inferred from the training data and that the classification of new data is inferred from the compact structure. However, this notion of inference seems to be broader and more informal than notions of inference used in proof theory, and the relation between them is not entirely clear. The purpose of this section is to illustrate how these notions of inference coincide and how the learned compact structures can be seen as cuts. To this aim, a simple example of decision tree is used.

The training data is shown in Table 9.1. Every instance has two attributes, which have values “low”, “medium” or “high”, and is classified either as belonging (“yes”) or not belonging (“no”) to the class C. The attributes could represent certain symptoms, and the class a disease. Or the attributes could represent performance in certain tasks in a test or evaluation, and the class whether the individual passed the test.

i	A1	A2	C
1	low	high	no
2	medium	high	yes
3	medium	medium	yes
4	high	low	yes
5	low	low	no
6	medium	low	no

Table 9.1: Training Data

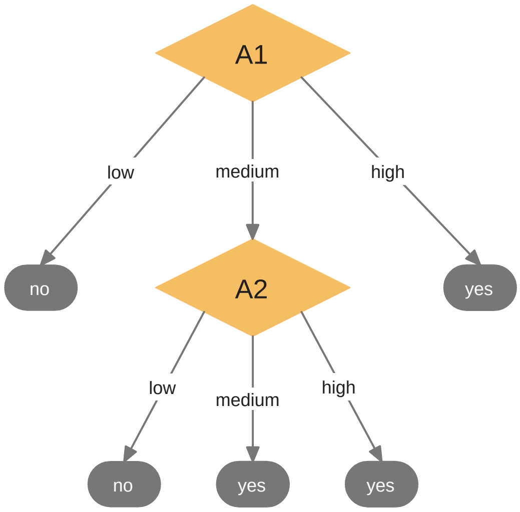
The given data can be represented as a formula:

$$\begin{aligned} Data'_T \doteq\ & (A_1(1,l) \wedge A_2(1,h) \wedge C(1,n))\wedge \\ & (A_1(2,m) \wedge A_2(2,h) \wedge C(2,y))\wedge \\ & \ldots \wedge \\ & (A_1(6,m) \wedge A_2(6,l) \wedge C(6,n)) \end{aligned}$$

By using a closed-world assumption, $Data'_T$ can be transformed to:

$Data_T \doteq Data'_T \wedge (\neg A_1(1,m) \wedge \neg A_1(1,h) \wedge \neg A_2(1,l) \wedge \neg A_2(1,m) \wedge \neg C(1,y)) \wedge \ldots$

From the training data, decision tree learning algorithms (e.g. ID3 and C4.5) can be used to generate a decision tree. In general, there might be many possible decision trees for a given training data set, and consequently there are various optimality criteria and heuristics to choose good decision trees. In this section, it is assumed that the following decision tree has been chosen:



The decision tree above can be expressed as the following formula:

$$DT \doteq \bigwedge_{i \in \{1,2,3,4,5,6\}} DT_1(i)$$

where:

$$DT_1 \doteq \bigwedge_{i \in \{1,2,3,4,5,6\}} \begin{pmatrix} (A_1(i,l) \rightarrow C(i,n)) \\ \wedge \\ (A_1(i,m) \rightarrow DT_2) \\ \wedge \\ (A_1(i,h) \rightarrow C(i,y)) \end{pmatrix}$$

and:

$$DT_2(i) \doteq \begin{pmatrix} (A_2(i,l) \rightarrow C(i,n)) \\ \wedge \\ (A_2(i,m) \rightarrow C(i,y)) \\ \wedge \\ (A_2(i,h) \rightarrow C(i,y)) \end{pmatrix}$$

It is interesting to note that the formulas above can be seen as logic programs composed of Horn clauses, and in principle, these formulas could have been obtained by inductive logic programming techniques from $Data_T$. So, decision tree learning algorithms are essentially inductive logic programming algorithms that induce logic programs of a specific form.

As desired, DT is (slightly) more compact than $Data_T$. DT does not summarize all information in $Data_T$, though, because it is not the case that $DT \to Data_T$. However, $Data_T \to DT$ holds, as sketched by the proof φ_{DT} only partially shown below:

$$\cfrac{\cfrac{\varphi^1_{DT}}{DataT \vdash (A_1(1,l) \to C(i,n)) \wedge (A_1(1,m) \to DT_2) \wedge (A_1(1,h) \to C(1,y))} \quad \ldots \quad \cfrac{\varphi^6_{DT}}{Data_T \vdash (A_1(6,l) \to C(6,n)) \wedge (A_1(6,m) \to DT_2) \wedge (A_1(6,h) \to C(6,y))}}{Data_T \vdash DT} \wedge_r, \wedge_r$$

However, DT contains enough information to classify the instances, as exemplified by the proof φ_2 below, which correctly instantiates the queried class variable x to y (class "yes"):

$$\cfrac{\cfrac{A_1(2,m) \vdash A_1(2,m) \qquad \cfrac{\cfrac{\cfrac{A_2(2,h) \vdash A_2(2,h)}{\cfrac{C(2,y) \vdash C(2,y)}{C(2,y) \vdash (\exists x)C(2,x)}\exists_r}}{A_2(2,h) \to C(2,y), A_2(2,h) \vdash (\exists x)C(2,x)}\to_l}{DT_2(2), A_2(2,h) \vdash (\exists x)C(2,x)}\wedge_l}{\cfrac{A_1(2,m) \to DT_2(2), A_1(2,m), A_2(2,h) \vdash (\exists x)C(2,x)}{(A_1(2,l) \to C(2,n)) \wedge (A_1(2,m) \to DT_2(2)) \wedge (A_1(2,h) \to C(2,y)), A_1(2,m), A_2(2,h) \vdash (\exists x)C(2,x)}\wedge_l}\to_l}{DT, A_1(2,m), A_2(2,h) \vdash (\exists x)C(2,x)}\bigwedge_l$$

The decisions taken at the decision nodes of the decision tree correspond to the pairs of $\wedge_l$ and $\to_l$ inferences in φ_1.

The use of the decision tree is an implicit use of cut, because the cut rule has to be used to actually show that the classification of instance 2 follows from the data, as shown in the proof φ_2' below:

$$\dfrac{\overset{\varphi_{DT}}{Data_T \vdash DT} \qquad \overset{\varphi_2}{DT, A_1(2,m), A_2(2,h) \vdash (\exists x)C(2,x)}}{Data_T \vdash (\exists x)C(2,x)}\ cut$$

Moreover, since the decision tree itself can be seen as a cut-formula in certain proofs , the construction of a decision tree can be seen as special kind of cut-introduction.

The generalization of the decision tree to data not belonging to the training data set corresponds simply to replacing the conjunction over all instances of the training data set by universal quantification, and results in the following formula:

$$DT^* \doteq (\forall i)DT_1(i)$$

To classify previously unseen instances, such as an instance k for which $A_1(k,h)$ and $A_2(k,m)$, the decision tree can be used as it is, resulting in a the classification "yes". By using DT^* instead of DT, the inferential use of the decision tree on a previously unseen instance to deduce its class can also be mapped to a formal proof φ_k:

$$\dfrac{\dfrac{\dfrac{A_1(k,h) \vdash A_1(k,h) \qquad \dfrac{\dfrac{C(k,y) \vdash C(k,y)}{C(k,y) \vdash (\exists x)C(k,x)}\ \exists_r}{C(k,y), A_2(k,m) \vdash (\exists x)C(k,x)}\ w_l}{A_1(k,h) \to C(k,y), A_1(k,h), A_2(k,m) \vdash (\exists x)C(k,x)}\ \to_l}{(A_1(k,l) \to C(k,n)) \land (A_1(k,m) \to DT_2(k)) \land (A_1(k,h) \to C(k,y)), A_1(k,m), A_2(k,h) \vdash (\exists x)C(k,x)}\ \land_l}{DT^*, A_1(k,h), A_2(k,m) \vdash (\exists x)C(k,x)}\ \forall_l$$

However, the classification of k only follows from the training data if a principle of induction allowing the generalization of conjunctions to universal quantification is incorporated into the calculus. This can be done by relaxing the cut rule, so that the right cut-formula is allowed to be a generalization of the left cut-formula. In this sense, the (relaxed) cut rule provides an easy way of incorporating inductive reasoning to a calculus. Evidently, this relaxed cut rule cannot be eliminated.

$$\dfrac{\overset{\varphi_{DT}}{Data_T \vdash DT} \qquad \overset{\varphi_k}{DT^*, A_1(k,h), A_2(k,m) \vdash (\exists x)C(k,x)}}{Data_T \vdash (\exists x)C(k,x)}\ cut_{relaxed}$$

Chapter 10

Conclusion

The initial goal of developing resolution refinements for cut-elimination by resolution was successfully achieved, as shown in Chapter 6. Moreover, the process of development of the refinements proved to be very fruitful, leading to many other major contributions to the understanding and improvement of the CERes method, which are summarized below:

- The CERes method is now a family of methods, all described within a unified framework of normalization of cut-pertinent structs, where different variants of the method correspond to different normalizations.

- An intuitive explanation of the notion of profile clause set was given by showing that the struct normalization that is used for the construction of the profile roughly corresponds to the swapping of inferences.

- Pushing further the correspondence with swapping of inferences led to the development of swapped clause sets, which are better than the profile clause sets in terms of size.

- The problem of the exponential size of clause sets with respect to the size of the proofs from which they are extracted was solved with the development of definitional and swapped definitional clause set, which do not have this problem because they use a normalization that is analogous to structural conjunctive normal form transformation.

- A less redundant kind of projection called O-projection was developed.
- The differences and similarities between normalized proofs produced by CERes and by reductive methods, two methods that previously seemed to be so fundamentally different and hence incomparable, are now clearer, thanks to the definition of intermediary refined methods.
- A method for introducing atomic cuts by resolution, CIRes, was developed. The method is capable of compressing proofs significantly.

Perhaps more importantly than the results that have been achieved so far, though, this thesis opens many doors for future research:

- Regarding refinements, there are at least four directions for further research:
  - The study of refinements inspired by reductive methods could be continued by trying to define refinements resembling particular strategies (e.g. Gentzen's or Tait's) for reductive cut-elimination.
  - Other kinds of refinements could be pursued. For example: the unifications allowed during resolution could be restricted according to the instantiations of quantifiers present in the input proof; semantic resolution could be considered in the case of interactive refutation search; bounds on the number of times that a clause should be used could perhaps be obtained by an analysis of the structure of the input proof . . .
  - While the refinements studied here are adequate for pure logic, mathematical applications usually use specialized inference rules like equality rules and rules incorporating deduction modulo theories. Therefore, refinements that restrict not only resolution and factoring rules but also paramodulation and demodulation rules must be developed.
  - While this thesis focused on first-order logic, CERes is already being extended to higher-order logics. Considering that the search for refutations is even harder in higher-order logics, refinements will definitely be necessary. While the refinements described here can probably serve as good starting point, it is

likely that specific refinements for the higher-order case could be investigated.

  – The implementation, experimentation and use of the refinements in the **CERes** system also remains to be done, since this thesis focused on their theoretical development and analysis only.

- The unified framework used to describe all the variants of CERes paves the way for the systematic study of other variants in the future. Definitional clause sets, for example, were still defined in a rather naive way, corresponding to the simplest structural conjunctive normal form transformations. More sophisticated definitional clause sets could be defined analogously to existing more sophisticated structural transformations.

- It is clear that, even though O-projections are much less redundant than S-projections, they still contain redundancies in certain cases. Trying to develop better projections seems to be a very challenging direction for future research. It is likely that it would require a major modification of the CERes method involving more complex ways of combining projections and refutations. A good benchmark for this kind of improvement are substructural logics without weakening and without contraction, because the still existing redundancies in current variants of CERes are due to the unnecessary use of contraction and weakening. In order to modify CERes to work for substructural logics, this source of redundancy would have to be eliminated.

- Less redundant projections would also imply a greater potential for compression of proofs via CIRes. Atomic cut-introduction by resolution is still just a first step toward the introduction of more complex cuts, and therefore there is still a lot of work to be done in this direction.

Bibliography

[1] W. Ackermann. Begruendung des "tertium non datur" mittles der hilbertschen theorie der widerspruchfreiheit. *Mathematiche Annalen*, 93:1–36, 1924. [cited at p. 26]

[2] M. Baaz, U. Egly, and A. Leitsch. Normal form transformations. In A. Voronkov A. Robinson, editor, *Handbook of Automated Reasoning*, pages 275–333. Elsevier, 2001. [cited at p. 17, 48, 94]

[3] M. Baaz, C. G. Fermüller, and A. Leitsch. A non-elementary speed-up in proof length by structural clause form transforamation. In *Proc. 9th Annual IEEE Symposium on Logic in Computer Science, LICS'94*, pages 213–219. IEEE Computer Society Press, 1994. [cited at p. 20]

[4] M. Baaz, C. G. Fermüller, A. Ovrutcki, and R. Zach. Multilog: A system for axiomatizing many-valued logics. In A. Voronkov, editor, *LPAR 93; LNAI 698*, pages 345–347. Springer, 1993. [cited at p. 45]

[5] M. Baaz and A. Leitsch. Ceres in many-valued logics. In F. Baader and A. Vornokov, editors, *LPAR 2004, LNAI 3452*, pages 1–20. Springer-Verlag, 2005. [cited at p. 45, 49]

[6] Matthias Baaz, Stefan Hetzl, Alexander Leitsch, Clemens Richter, and Hendrik Spohr. Ceres: An Analysis of Fürstenberg's Proof of the Infinity of Primes. to appear in Theoretical Computer Science. [cited at p. 43, 49, 145]

[7] Matthias Baaz, Stefan Hetzl, Alexander Leitsch, Clemens Richter, and Hendrik Spohr. Proof Transformation by CERES. In Jonathan M. Borwein and William M. Farmer, editors, *Mathematical Knowledge Management (MKM) 2006*, volume 4108 of *Lecture Notes in Artificial Intelligence*, pages 82–93. Springer, 2006. [cited at p. 50]

[8] Matthias Baaz and Alexander Leitsch. On skolemization and proof complexity. *Fundamenta Informaticae*, 20(4):353–379, 1994. [cited at p. 110]

[9] Matthias Baaz and Alexander Leitsch. Methods of cut-elimination. Lecture Notes of a Course in the European Summer School of Language, Logic and Information, 1999. [cited at p. 24, 41, 43, 49, 50]

[10] Matthias Baaz and Alexander Leitsch. Cut-elimination and Redundancy-elimination by Resolution. *Journal of Symbolic Computation*, 29(2):149–176, 2000. [cited at p. 13, 46, 49, 50]

[11] Matthias Baaz and Alexander Leitsch. Comparing the complexity of cut-elimination methods. In *Proof Theory in Computer Science*, pages 49–67, 2001. [cited at p. 43]

[12] Matthias Baaz and Alexander Leitsch. Towards a clausal analysis of cut-elimination. *Journal of Symbolic Computation*, 41(3–4):381–410, 2006. [cited at p. 49, 125, 129, 140]

[13] Matthias Baaz and Alexander Leitsch. *Methods of Cut-Elimination*. To Appear, 2009. [cited at p. 41, 42, 46, 49, 50]

[14] Matthias Baaz and Richard Zach. Algorithmic structuring of cut-free proofs. In *CSL '92: Selected Papers from the Workshop on Computer Science Logic*, pages 29–42, London, UK, 1993. Springer-Verlag. [cited at p. 173, 183]

[15] U. Berger, W. Buchholz, and H. Schwichtenberg. Refined program extraction from classical proofs. *Annals of Pure and Applied Logic*, 114:3–25, 2002. [cited at p. 26]

[16] Bombieri. [cited at p. 26]

[17] Paul Brauner, Clement Houtmann, and Claude Kirchner. Principles of Superdeduction. In *Twenty-Second Annual IEEE Symposium on Logic in Computer Science (LiCS)*, 2007. [cited at p. 15, 45]

[18] Kai Bruennler and Stephane Lengrand. On two forms of bureaucracy in derivations. 2007. [cited at p. 25]

[19] Paola Bruscoli and Alessio Guglielmi. On the proof complexity of deep inference. *ACM Transactions on Computational Logic*, 10:1–34, 2009. [cited at p. 25]

[20] S. Buss. The undecidability of k-provability. *Annals of Pure and Applied Logic*, 53:72–102, 1991. [cited at p. 25]

[21] S. R. Buss. On Herbrand's theorem. *Lecture Notes in Computer Science*, 960:195, 1995. [cited at p. 24]

[22] S.R. Buss. *Handbook of Proof Theory*, chapter An Introduction to Proof Theory, pages 1–78. Elsevier (Amsterdam), 1998. [cited at p. 13, 24, 41, 42]

[23] Wojciech Buszkowski. Mathematical linguistics and proof theory. In *Handbook of Logic and Language*, pages 683–736. Elsevier, 1997. [cited at p. 197]

[24] A. Carbone. Interpolants, cut elimination and flow graphs for the propositional calculus. *Annals of Pure and Applied Logic*, 83:249–299, 1997. [cited at p. 25]

[25] K. Clark. Negation as failure. *Readings in Nonmonotonic Reasoning*, pages 311–325, 1978. [cited at p. 14]

[26] Stephen Cook and Robert Reckhow. On the lengths of proofs in the propositional calculus (preliminary version). In *STOC '74: Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 135–148, New York, NY, USA, 1974. ACM. [cited at p. 178, 195]

[27] W. Craig. Three uses of the herbrand-gentzen theorem in relating model theory and proof theory. *Journal of Symbolic Logic*, 22(3):269–285, 1957. [cited at p. 24]

[28] Haskell Curry. Functionality in combinatory logic. In *Proceedings of the National Academy of Sciences*, volume 20, pages 584–590, 1934. [cited at p. 25]

[29] Haskell Curry and Robert Feys. *Combinatory Logic*, volume 1. North-Holland, Amsterdam, 1958. [cited at p. 25]

[30] Marcello D'Agostino, Dov M. Gabbay, Reiner Hhnle, and Joachim Posegga, editors. *Handbook of Tableau Methods*. Springer, 1999. [cited at p. 20, 23]

[31] Vincent Danos, Jean-Baptiste Joinet, and Harold Schellinx. A new deconstructive logic: Linear logic. *Journal of Symbolic Logic*, 62(3):755–807, 1997. [cited at p. 40]

[32] John W. Dawson Jr. The compactness of first-order logic: from gödel to lindström. *History and Philosophy of Logic*, 14(1):15–37, 1993. [cited at p. 20]

[33] Philippe De Groote, editor. *The Curry-Howard Isomorphism*. 1995. [cited at p. 25]

[34] A. Degtyarev and A. Voronkov. *Handbook of Automated Reasoning*, volume 1, chapter Equality Reasoning in Sequent-Based Methods, pages 611–706. Elsevier, 2001. [cited at p. 13]

[35] Anatoli Degtyarev and A. Voronkov. *Handbook of Automated Reasoning*, volume 1, chapter The Inverse Method, pages 179–272. Elsevier, 2001. [cited at p. 16, 23]

[36] Gilles Dowek, Thérèse Hardin, and Claude Kirchner. Theorem proving modulo. Rapport de Recherche 3400, Institut National de Recherche en Informatique et en Automatique, April 1998. [cited at p. 13, 45, 202]

[37] Gilles Dowek, Therese Hardin, and Claude Kirchner. HOL-$\lambda\sigma$ - an intentional first-order expression of higher-order logic. *Mathematical Structures in Computer Science*, 11(1):1–25, 2001. [cited at p. 13]

[38] Tsvetan Chavdarov Dunchev. Simplification of herbrand sequents. Master's thesis, Vienna University of Technology, 2009. [cited at p. 2]

[39] E. Eder. *Relative complexities of first-order calculi*. Vieweg, 1992. [cited at p. 20, 146]

[40] Uwe Egly and Karin Genther. Structuring of computer-generated proofs by cut introduction. In *KGC '97: Proceedings of the 5th Kurt Gödel Colloquium on Computational Logic and Proof Theory*, pages 140–152, London, UK, 1997. Springer-Verlag. [cited at p. 173]

[41] T. Eiter and G. Gottlob. Propositional circumscription and extended closed world reasoning are π_2^p-complete. *Theoretical Computer Science*, 114:231–245, 1993. [cited at p. 14]

[42] Marcelo Finger and Dov M. Gabbay. Equal rights for the cut: Computable non-analytic cuts in cut-based proofs. *Logic Journal of the IGPL*, 15(5-6):553–575, 2007. [cited at p. 173]

[43] M. Fitting. *First-Order Logic and Automated Theorem Proving*. Springer-Verlag, 1990. [cited at p. 23]

[44] D.M. Gabbay and U. Reyle. Labelled resolution for classical and non-classical logics. *Studia Logica*, 59(2):179–216, September 1997. [cited at p. 11]

[45] Dov M. Gabbay. *Labelled Deductive Systems*. Oxford University Press, 1996. [cited at p. 11, 14]

[46] Jean Gallier. Constructive logics, part 1: A tutorial on proof systems and typed λ-calculi. *Theoretical Computer Science*, 110(2):249–339, 1993. [cited at p. 40]

[47] M. Gelfond. Answer sets. In *Handbook of Knowledge Representation*, pages 285–316. Elsevier, 2008. [cited at p. 14]

[48] G. Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210,405–431, 1934–1935. [cited at p. 5, 12, 16, 23, 24, 41]

[49] G. Gentzen. Untersuchungen über das logische Schließen. In M.E.Szabo, editor, *The Collected Papers of Gerhard Gentzen*, pages 68–131. North-Holland Publishing Company, Amsterdam - London, 1969. [cited at p. 5, 12, 24]

[50] Philipp Gerhardy. *Improved Complexity Analysis for Cut-Elimination and Herbrand's Theorem (second corrected version)*. PhD thesis, 2003. [cited at p. 42]

[51] Philipp Gerhardy and Ulrich Kohlenbach. Extracting herbrand disjunctions by functional interpretation. *Archive for Mathematical Logic*, 44(5):633–644, 2005. [cited at p. 26]

[52] J.-Y. Girard. Linear logic: Its syntax and semantics. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic (Proc. of the Workshop on Linear Logic, Cornell University, June 1993)*, number 222. Cambridge University Press, 1995. [cited at p. 12, 14]

[53] Jean-Yves Girard. Linear logic. *Theor. Comput. Sci.*, 50(1):1–102, 1987. [cited at p. 12, 25]

[54] J.Y. Girard. *Proof Theory and Logical Complexity*. Studies in Proof Theory. Bibliopolis, Napoli, 1987. [cited at p. 25]

[55] K. Goedel. Ueber eine bisher noch nicht benuetzte erweiterung des finiten standpunktes. *Dialectica*, 12:280–287, 1958. [cited at p. 26]

[56] Kurt Goedel. ber formal unentscheidbare stze der principia mathematica und verwandter systeme. *Monatshefte fr Mathematik und Physik*, 38(1):173–198, 1931. [cited at p. 9]

[57] Kurt Goedel and Solomon Feferman. *Collected Works*. Oxford University Press, New York, 1986. [cited at p. 9]

[58] G. Gottlob. Translating default logic into standard autoepistemic logic. *Journal of the ACM Transactions on Computational Logic*, 42:711–740, 1995. [cited at p. 14]

[59] Nicolas Guenot. Concurrency and permutability in the sequent calculus. In Michel Parigot and Lutz Strassburger, editors, *Structures and Deduction (ESSLLI Workshop)*, pages 39–52, 2009. [cited at p. 67]

[60] Alessio Guglielmi and Tom Gundersen. Normalisation control in deep inference via atomic flows. *Logical Methods in Computer Science*, 4:1–36, 2008. [cited at p. 25]

[61] Leon Henkin. Completeness in the theory of types. *The Journal of Symbolic Logic*, 15:81–91, 1950. [cited at p. 25]

[62] J. Herbrand. Recherches sur la théorie de la démonstration. *Travaux de la Societé des Sciences et des Lettres de Varsovie, Class III, Sciences Mathématiques et Physiques*, 33, 1930. [cited at p. 24]

[63] J. Herbrand. *Recherches sur la Theorie de la Demonstration*. PhD thesis, University of Paris, 1930. [cited at p. 24]

[64] J. Herbrand. *Collected Works*. Harvard University Press, 1971. [cited at p. 24]

[65] Stefan Hetzl. Proof Fragments, Cut-Elimination and Cut-Introduction. manuscript. [cited at p. 173]

[66] Stefan Hetzl. *Characteristic Clause Sets and Proof Transformations*. PhD thesis, Vienna University of Technology, 2007. [cited at p. 6, 40, 80, 89, 90, 94, 129, 134, 140, 177]

[67] Stefan Hetzl. *Proof Profiles. Characteristic Clause Sets and Proof Transformations*. VDM, 2008. [cited at p. 40, 89, 94, 134, 140]

[68] Stefan Hetzl. Describing proofs by short tautologies. *Annals of Pure and Applied Logic*, 159(1–2):129–145, 2009. [cited at p. 25, 49]

[69] Stefan Hetzl, Alexander Leitsch, Daniel Weller, and Bruno Woltzenlogel Paleo. Cut-Elimination by Resolution and Skolemization in Second-Order Logic [Extended Abstract]. In *LPAR Workshop: Analytic Proof Systems*, 2008. [cited at p. 14]

[70] Stefan Hetzl, Alexander Leitsch, Daniel Weller, and Bruno Woltzenlogel Paleo. Herbrand sequent extraction. In *Proceedings of the Conferences on Intelligent Computer Mathematics*, number 5144 in LNAI, 2008. [cited at p. 25]

[71] Stefan Hetzl, Alexander Leitsch, Daniel Weller, and Bruno Woltzenlogel Paleo. A clausal approach to proof analysis in second-order logic. In Sergei Artemov and Anil Nerode, editors, *Logical Foundations of Computer Science*, volume 5407 of *Lecture Notes in Computer Science*, pages 214–229. Springer Berlin, 2009. [cited at p. 14]

[72] D. Hilbert. Neubegruendung der mathematik: Erste mitteilung. Abhandlungen aus dem Seminar der Hamburgischen Universitt, 1922. [cited at p. 26]

[73] D. Hilbert and P. Bernays. *Grundlagen der Mathematik*, volume 2. 1939. [cited at p. 26]

[74] David Hilbert. *David Hilbert's Lectures on the Foundations of Mathematics and Physics*. Springer. [cited at p. 16, 24]

[75] S.C. Klenne. *Two Papers on the Predicate Calculus*, chapter Permutability of Inferences in Gentzen's calculi LK and LJ, pages 1–26. American Mathematical Society, 1952. [cited at p. 67]

[76] U. Kohlenbach. *Applied Proof Theory: Proof Interpretations and their Use in Mathematics*. Springer, 2008. [cited at p. 26]

[77] Dénes König. *Theorie der endlichen und unendlichen Graphen*. Leipzig, 1936. [cited at p. 20]

[78] Franois Lamarche and Lutz Strassburger. Naming proofs in classical logic. In *Proceedings of the 7th International Conference on Typed Lambda Calculi and Applications*, 2005. [cited at p. 25]

[79] G. W. Leibniz. *Philosophical Papers and Letters*. 1969. [cited at p. 25]

[80] Alexander Leitsch. *The resolution calculus*. Springer-Verlag New York, Inc., New York, NY, USA, 1997. [cited at p. 11, 17, 18, 20, 21, 190]

[81] Tomer Libal. *Automated Deduction in Second-Order Arithmetic*. PhD thesis, Technische Universitaet Wien (Vienna University of Technology), Under Preparation. [cited at p. 14]

[82] J. McCarthy. Circumscription - a form of non-monotonic reasoning. *Artificial Intelligence*, 13:27–39, 1980. [cited at p. 14]

[83] William McCune. Otter: First-order logic automated theorem prover. Website: http://www-unix.mcs.anl.gov/AR/otter/. [cited at p. 145]

[84] William McCune. Prover9: First-order logic automated theorem prover. Website: http://www.cs.unm.edu/ mccune/prover9/. [cited at p. 145]

[85] Richard McKinley. A sequent calculus demonstration of herbrand's theorem. [cited at p. 24]

[86] Dale Miller and Vivek Nigam. Incorporating tables into proofs. In J. Duparc and T.A. Henzinger, editors, *CSL 2007: Computer Science Logic*, volume 4646, pages 466–480. Springer, 2007. [cited at p. 173]

[87] Tom Mitchell. *Machine Learning*. McGraw Hill, 1997. [cited at p. 26, 211]

[88] Michael Moortgat. Categorial type logics. In *Handbook of Logic and Language*, pages 93–177. Elsevier, 1997. [cited at p. 197]

[89] M. Morris and G. H. R. Parkinson. *Leibniz Philosophical Writings*. London: Dent, 1973. [cited at p. 25]

[90] G. Moser. The epsilon substitution method. Master's thesis, University of Leeds, 2000. [cited at p. 26]

[91] G. Moser and R. Zach. The epsilon calculus and herbrand complexity. *Studia Logica*, 82:133–155, 2006. [cited at p. 26]

[92] Vivek Nigam. Using tables to construct non-redundant proofs. In *CiE 2008: Abstracts and extended abstracts of unpublished papers*, 2008. [cited at p. 173]

[93] V.P. Orevkov. Lower bounds for increasing complexity of derivations after cut-elimination (translation). *Journal Sov. Math.*, (1982):2337–2350, 1979. [cited at p. 26, 42, 173]

[94] Bruno Woltzenlogel Paleo. Herbrand sequent extraction. Master's thesis, Technische Universitaet Dresden; Technische Universitaet Wien, Dresden, Germany; Wien, Austria, 07 2007. [cited at p. 7, 24, 25, 44, 121]

[95] Bruno Woltzenlogel Paleo. *Herbrand Sequent Extraction*. VDM-Verlag, Saarbruecken, Germany, 1 2008. [cited at p. 7, 24, 25, 121]

[96] Michel Parigot. Lambda-mu-calculus: An algorithmic interpretation of classical natural deduction. In *Logic Programming and Automated Reasoning: International Conference LPAR*, number 624 in Lecture Notes in Computer Science, pages 190–201. Springer-Verlag, 1992. [cited at p. 25]

[97] Dag Prawitz. *Natural Deduction: A Proof-Theoretic Study*. Dover Publications, 1965. [cited at p. 16]

[98] Pavel Pudlak. *The Length of Proofs*, chapter The Length of Proofs, pages 548–637. Elsevier Science B.V., 1998. [cited at p. 42]

[99] R. Reiter. On closed world data bases. *Logic and Data Bases*, pages 119–140, 1978. [cited at p. 14]

[100] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980. [cited at p. 14]

[101] Greg Restall. *On Logics without Contraction*. PhD thesis, University of Queensland, 1994. [cited at p. 14]

[102] Clemens Richter. *Proof Transformations by Resolution – Computational Methods of Cut-Elimination*. PhD thesis, Vienna University of Technology, 2006. [cited at p. 12, 13, 50]

[103] Martin Riener. Provisory title: Incorporation of deduction modulo ac-theories into atp. Master's thesis, Technische Universitaet Wien (Vienna University of Technology), 2010. [cited at p. 13]

[104] J.A. Robinson. A machine oriented logic based on the resolution principle. *Journal of the ACM*, 12(1):23–41, 1965. [cited at p. 17]

[105] Mikheil Rukhaia. Ceres and fast cut-elimination. Master's thesis, Vienna University of Technology, 2009. [cited at p. 2]

[106] Helmut Schwichtenberg. *Handbook of Mathematical Logic*, chapter Proof Theory: Some Applications of Cut-Elimination, pages 868–894. North-Holland, 1977. [cited at p. 42]

[107] Ehud Shapiro and Leon Sterling. *The Art of PROLOG: Advanced Programming Techniques*. The MIT Press, April 1994. [cited at p. 201]

[108] Stephen G. Simpson. *Subsystems of Second Order Arithmetic*. Springer-Verlag, Heidelberg, Germany, 1999. [cited at p. 20]

[109] Richard Statman. Lower bounds on Herbrand's theorem. *Proceedings of the American Mathematical Society*, 75:104–107, 1979. [cited at p. 26, 42, 173]

[110] Leon Sterling and Ehud Shapiro. *The Art of Prolog*. MIT Press, 1994. [cited at p. 24]

[111] Lutz Straßburger. What is a logic, and what is a proof? In Jean-Yves Beziau, editor, *Logica Universalis*, pages 135–145. Birkhäuser, 2005. Updated version at http://www.lix.polytechnique.fr/~lutz/papers/WhatLogicProof.pdf. [cited at p. 25]

[112] G. Takeuti. *Proof Theory*. North-Holland, Amsterdam, 2 edition, 1987. [cited at p. 16, 24, 41]

[113] Trifon Trifonov and Diana Ratiu. Esxploring the computational content of the infinite pigeonhole principle. *Journal of Logic and Computation*, to appear, 2009. [cited at p. 26, 43]

[114] Michael Trott. *The Mathematica Guidebook: Programming*. Springer, October 2004. [cited at p. 202]

[115] Christian Urban. *Classical Logic and Computation*. PhD thesis, University of Cambridge, October 2000. [cited at p. 40]

[116] Daniel Weller. *Skolemization Problems and Cut-elimination (In Preparation)*. PhD thesis, Technische Universitaet Wien (Vienna University of Technology), 2009. [cited at p. 14]

[117] Ryszard Wójcicki. *Theory of Logical Calculi: Basic Theory of Consequence Operations*. Number 199 in Synthese Library. Kluwer Academic Publishers, 1988. [cited at p. 17]

[118] Richard Zach. Proof theory of finite-valued logics. Master's thesis, Technische Universitaet Wien, 1993. [cited at p. 45]

[119] Ernst Zermelo. Untersuchungen ueber die grundlagen der mengenlehre i. *Mathematische Annalen*, 65:261–281, 1908. [cited at p. 20]

List of Notations

Notations for Reductive Cut-Elimination

Notation	Description	Definition
$\rhd_{p_\wedge}$	Cut-reduction over $\wedge$	page 30
$\rhd_{p_\vee}$	Cut-reduction over $\vee$	page 31
$\rhd_{p_\rightarrow}$	Cut-reduction over $\rightarrow$	page 31
$\rhd_{p_\neg}$	Cut-reduction over $\neg$	page 31
$\rhd_{p}$	Cut-reduction over any propositional connective	page 32
$\rhd_{q_\forall}$	Cut-reduction over universal quantifier	page 32
$\rhd_{q_\exists}$	Cut-reduction over existential quantifier	page 32
$\rhd_{q}$	Cut-reduction over any quantifier	page 32
$\rhd_{c}$	Cut-reduction over contraction	page 33
$\rhd_{c_{non-atomic}}$	Non-atomic-cut-reduction over contraction	page 33
$\rhd_{w}$	Cut-reduction over weakening	page 33
$\rhd_{w_{non-atomic}}$	Non-atomic-cut-reduction over weakening	page 33
$\rhd_{r_1}$	Cut-reduction: Rank-reduction over unary inference	page 29
$\rhd_{r_2}$	Cut-reduction: Rank-reduction over binary inference	page 29
$\rhd_{r}$	Cut-reduction: Rank-reduction	page 30
$\rhd_{r_{non-atomic}}$	Non-atomic-cut-reduction: Rank-reduction	page 30
$\rhd_{a}$	Cut-reduction over axiom inference	page 34
$\rhd$	Cut-reduction	page 34
$\rhd_{\overline{a}}$	Cut-reduction except over axiom inference	page 34
$\rhd_{\tilde{a}}$	Non-atomic-cut-reduction	page 34

Notations for Cut-Elimination by Resolution

Notation	Description	Definition
CERes	The method of cut-elimination by resolution	page 109
$\mathcal{S}_\varphi$	Cut-pertinent struct of the proof φ	page 47
$\mathcal{C}_\varphi$	Cut-pertinent clause set of the proof φ	page 49
$\mathcal{C}^S_\varphi$	Clause set using the standard struct normalization	page 49
$\mathcal{C}^P_\varphi$	Clause set using the profile struct normalization	page 92
$\mathcal{C}^W_{\varphi\|S}$	Clause set using swapped struct normalization, with respect to the struct normal form S	page 84
$\mathcal{C}^W_\varphi$	Clause set using swapped struct normalization, in case it is unique	page 84
$\mathcal{C}^D_\varphi$	Clause set using definitional struct normalization	page 97
$\mathcal{C}^D_{\varphi\|S}$	Clause set using swapped definitional struct normalization, with respect to the struct normal form S	page 106
$\lfloor\varphi\rfloor^S_c$	S-projection of the proof φ with respect to the clause c	page 51
$\lfloor\varphi\rfloor^O_c$	O-projection of the proof φ with respect to the clause c	page 64
$\lfloor\varphi\rfloor^{D_D}_c$	Definitional D-projection of the proof with respect to the clause c	page 97
$\lfloor\varphi\rfloor^{D_P}_c$	Proper D-projection of the proof φ with respect to the clause c	page 100
$\lfloor\varphi\rfloor^{DW_D}_c$	Definitional DW-projection of the proof φ with respect to the clause c	page 108
$\lfloor\varphi\rfloor^{DW_P}_c$	Pure DW-projection of the proof φ with respect to the clause c	page 108
$\lfloor\varphi\rfloor^{DW_M}_c$	Mixed DW-projection of the proof φ with respect to the clause c	page 50
$\mathrm{CERes}(\varphi,\delta)$	CERes-normal-form of the proof φ with respect to the refutation δ.	page 109
$\mathrm{CERes}^S_S(\varphi,\delta)$	CERes-normal-form using the standard clause set and S-Projections.	page 109
$\mathrm{CERes}^O_S(\varphi,\delta)$	CERes-normal-form using the standard clause set and O-Projections.	page 109
$\mathrm{CERes}^O_P(\varphi,\delta)$	CERes-normal-form using the profile clause set and O-Projections.	page 109
$\mathrm{CERes}^O_W(\varphi,\delta)$	CERes-normal-form using swapped clause sets and O-Projections.	page 109
$\mathrm{CERes}^D_D(\varphi,\delta)$	CERes-normal-form using the definitional clause set and D-Projections.	page 109
$\mathrm{CERes}^{DW}_{DW}(\varphi,\delta)$	CERes-normal-form using swapped definitional clause sets and DW-Projections.	page 109
$\mathrm{CCERes}(\varphi,\delta)$	CERes-normal-form ignoring atomic cuts.	page 118

Notation	Description	Definition
QCERes(φ, δ)	CERes-normal-form ignoring quantifier-free cuts.	page 121

Notations for Cut-Introduction by Resolution

Notation	Description	Definition
CIRes	The method of cut-introduction by resolution	page 109
CIRes(φ, δ)	CIRes-normal-form of the proof φ with respect to the refutation δ	page 109
$\mathrm{CIRes}_W^O(\varphi, \delta)$	CIRes-normal-form using swapped clause sets and O-projections.	page 109

Notations for Auxiliary Proof Transformations

Notation	Description	Definition
$Y_{\oplus}^{\Omega}(\varphi)$	Ω-pertinent Y-replacement applied to the proof φ	page 56
$Y_{\otimes}^{\Omega}(\varphi)$	Ω-impertinent Y-replacement applied to the proof φ	page 57
$(\!\|\varphi\|\!)_{\Omega}$	The proofoid of φ with respect to Ω	page 59
$\wr\varphi\int_{\{\Omega\}}$	The proof-slice of φ with respect to Ω	page 59
$W_{\mathrm{fix}}(\varphi)$	The result of fixing broken inferences of φ by adding weakening	page 60
$[\![\varphi]\!]$	The result of eliminating Y-inferences from φ	page 61
$[\![\varphi]\!]_M$	The result of eliminating Y-inferences from φ by merging branches	page 124
$r_{cut}^{r}(\varphi)$	The result of replacing cuts by resolution inferences in φ	page 125
$CR(\varphi)$	The canonic refutation extracted from φ	page 125

Index

Y-Elimination, 61
    Merging, 124
CERes-Normal-Form, 109
    Ignoring Atomic Cuts, 118
    Ignoring Quantifier-Free Cuts, 121
CIRes, 175
Ω-Pertinence, 46

Ancestor Relation, 15
Argmin, 61
Axiom Linkage, 58

Canonic Refutation, 125
Clause, 18
Clause Set
    Definitional, 97
    Profile, 92
    Standard, 49
    Swapped, 84
    Swapped Definitional, 106
Clausification, 48
Confluence, 28
CR-Simulation, 128
Cut-Elimination by Resolution, 43
Cut-Introduction, 173
Cut-Linkage, 153
    Atomic Cut-Linkage, 154
Cut-Pertinence, 45, 46
Cut-Side, 154

Factoring Rules, 19
Formula, 10
    Annotated Formula, 11
    Formula Corresponding to a Sequent, 12

Inference, 15
    Broken Inference, 60
Inference Dependence, 68
Inference Replacement, 56
    Eigen-Y-Replacement, 57
    Impertinent Y-Replacement, 57
    Pertinent Y-Replacement, 56
Inference Rule, 13
    Y-Rule, 55
Inference Swapping, 67
    Cut Conversion, 93
    Swapping of Independent Inferences, 69
    Swapping of Indirectly Dependent Contractions, 75
    Swapping of Indirectly Dependent Inferences, 74
    Swapping of Weakening, 79
    Swapping over Contractions, 76

Normal Form, 28
Normalization, 28

Polarity of Formula Occurrences, 11
Projection, 50
    Definitional D-Projection, 97
    Definitional DW-Projection, 108
    Mixed DW-Projection, 108
    O-Projection, 64
    Proper D-Projection, 100
    Pure DW-Projection, 108
    S-Projection, 51
Proof, 15
    Skolemized Proof, 53
Proof Size, 17
Proof-Slice, 59
Proofoid, 59

Quantifier, 12

Reductive Cut-Elimination, 27
Refinement for CERes

Using Atomic Cut-Linkage, 168
Using Cut-Linkage, 163
Using Cut-Linkage More Strictly, 165
Using Cut-Side and Cut-Linkage, 166
Refutation, 19
Resolution Calculus, 17
Resolution Refinement, 21
Resolution Rule, 18

Sentence, 11
Sequent, 12
Sequent Calculus, 12
LJ, 12
LK, 12
LKD, 15
Signature, 10
Struct, 46
Cut-Pertinent Struct, 47
Struct Normalization
Definitional, 95
Profile, 91
Standard, 48
Swapped, 82
Swapped Definitional, 104
Substitution, 13

Term, 10
Bound Variable, 11
Free Variable, 11

Unifier, 18

W-Fixing, 60