Matthias Baaz
Alexander Leitsch

Trends in Logic 34

Methods of Cut-Elimination

Springer

Methods of Cut-Elimination

TRENDS IN LOGIC
Studia Logica Library

VOLUME 34

SCOPE OF THE SERIES

Trends in Logic is a bookseries covering essentially the same area as the journal *Studia Logica* – that is, contemporary formal logic and its applications and relations to other disciplines. These include artificial intelligence, informatics, cognitive science, philosophy of science, and the philosophy of language. However, this list is not exhaustive, moreover, the range of applications, comparisons and sources of inspiration is open and evolves over time.

Volume Editor
Daniele Mundici

For further volumes:
http://www.springer.com/series/6645

Matthias Baaz · Alexander Leitsch

Methods of Cut-Elimination

Springer

Matthias Baaz
Vienna University of Technology
Wiedner Hauptstraße 8-10
1040 Vienna
Austria
baaz@logic.at

Alexander Leitsch
Vienna University of Technology
Favoritenstraße 9
1040 Vienna
Austria
leitsch@logic.at

ISBN 978-94-007-0319-3 e-ISBN 978-94-007-0320-9
DOI 10.1007/978-94-007-0320-9
Springer Dordrecht Heidelberg London New York

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Contents

Chapter 1

Preface

1.1 The History of This Book

This book comprises 10 years of research by Matthias Baaz and Alexander Leitsch on the topic of cut-elimination. The aim of this research was to consider computational aspects of cut-elimination, the most important method for analyzing formal first-order proofs. During this period a new method of cut-elimination, cut-elimination by resolution (CERES), has been developed which is based on the refutations of formulas characterizing the cut-structure of the proofs. This new method connects automated theorem proving with classical proof theory, allowing the development of new methods and more efficient implementations; moreover, CERES opens a new view on cut-elimination in general. This field of research is evolving quite fast and we expect further results in the near future (in particular concerning cut-elimination in higher-order logic and in nonclassical logics).

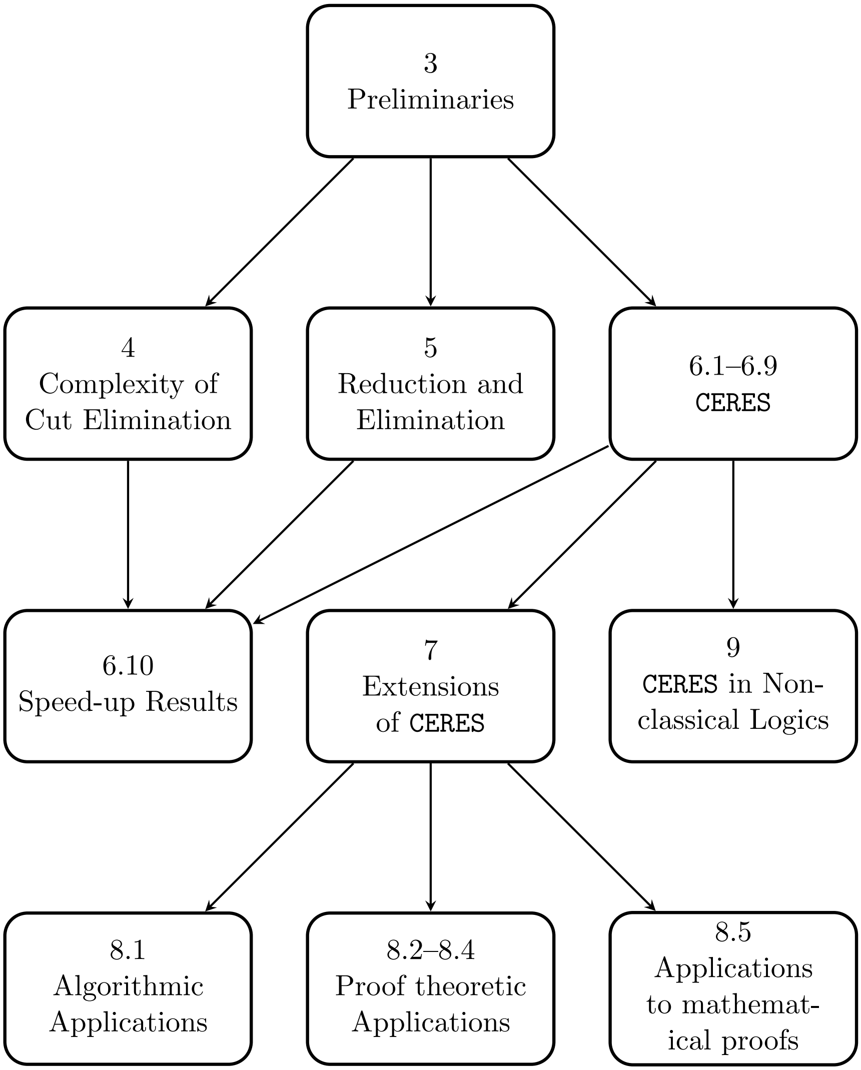
1.2 Potential Readers of This Book

This book is directed to graduate students and researchers in the field of automated deduction and proof theory. The uniform approach, developed by Alexander Leitsch, serves the purpose of importing mathematical techniques from automated deduction to proof theory, to facilitate the implementation and derivation of complexity bounds for basically indeterministic methods. Matthias Baaz has been responsible for proof theoretic considerations and for the extension of CERES to nonclassical logics.

M. Baaz, A. Leitsch, *Methods of Cut-Elimination*, Trends in Logic 34,
DOI 10.1007/978-94-007-0320-9_1, © Springer Science+Business Media B.V. 2011

1.3 How to Read This Book

The book can be read from a computer-science or from a proof-theoretic perspective as the diagram below indicates.


3
Preliminaries
4
Complexity of Cut Elimination
5
Reduction and Elimination
6.1–6.9
CERES
6.10
Speed-up Results
7
Extensions of CERES
9
CERES in Non-classical Logics
8.1
Algorithmic Applications
8.2–8.4
Proof theoretic Applications
8.5
Applications to mathematical proofs

Acknowledgments We thank Daniele Mundici for his encouragement to write a book on this topic and for his steady interest in our research during the last 15 years. We also are grateful to the Austrian Science Fund for supporting the research on cut-elimination by funding the projects P16264, P17995, and P19875.

The research on this topic began with the authors and Alessandra Carbone during the time of her Lise Meitner fellowship. In the course of the following FWF research projects P16264, P17995, and P19875, the Ph.D. students Stefan Hetzl, Clemens Richter, Hendrik Spohr, Daniel Weller, and Bruno Woltzenlogel-Paleo contributed substantially to the theoretic and, especially, to the practical development of the `CERES` method. The extension of the method to Gödel logic has been carried out together with Agata Ciabattoni and Chris Fermüller.

Our special thanks go to Tomer Libal, Daniel Weller and Bruno Woltzenlogel for their careful and critical reading of the text. Their comments and suggestions have been integrated in the text and have resulted in a substantial improvement of the book.

We are very grateful to the reviewer for his numerous critical comments and suggestions for improvements which had a substantial impact on the final version of text.

Chapter 2

Introduction

Gottfried Wilhelm Leibniz called a proof analytic iff[1] the proof is based on concepts contained in the proven statement (praedicatum inest subjecto [59]). His own example [60] shows that this notion is significant, as it is connected to the distinction between inessential derivation steps (mostly formulated as definitions) and derivation steps which may or may not be based on concepts contained in the result:

(a) $4 = 2 + 2$ (the result)

(b) $3 + 1 = 2 + 2$ (by the definition $4 = 3 + 1$)

(c) $(2 + 1) + 1 = 2 + 2$ (by the definition $3 = 2 + 1$)

(d) $2 + (1 + 1) = 2 + 2$ (by associativity)

(e) $2 + 2 = 2 + 2$ (by the definition $2 = 1 + 1$)

The interest in the notion of analytic proof and analytic provability is twofold:

- First, the reduction of the concepts constituting a proof to the concepts contained in the (desired) result is essential to construct a proof by an analysis of the result (This was the main aim of Leibniz). Therefore analytic proofs in a suitable definition are the core of any approach to automated theorem proving.
- Second, analytic proofs allow control not only of the result but also the means of the proof and admit the derivation of additional information

[1] If and only if.

M. Baaz, A. Leitsch, *Methods of Cut-Elimination*, Trends in Logic 34,
DOI 10.1007/978-94-007-0320-9_2, © Springer Science+Business Media B.V. 2011

> related to the result from the proof. In other words: a theorem with an analytic proof can be strengthened by looking at the proof.

In mathematics, the obvious counterpart to the notion of analytic proof is the notion of elementary proof. What elementary means, however, changes in time (from avoiding arguments on complex numbers in the Prime Number Theorem [37] to omitting arguments on p-adic numbers and more recently ergodic theory). In a more modern expression analyticity relates to the distiction between soft and hard analysis by Terence Tao [76].

David Hilbert introduced the concept of purity of methods (Reinheit der Methode) as an emphasis on analytic provability (and not so much on analytic proofs). He discussed for the first time whether, for a given mathematical theory, all provable statements are in fact analytically provable (this line of thought is already present in *Grundlagen der Geometrie* [50]).

The social value of mathematics (and of science in general) is connected to the establishment of verified statement i.e. theorems which can be applied without (using the) knowledge of its proofs. It is not necessary to understand the proof of the central limit theorem for working with normal distributions.

This principle also applies within mathematics w.r.t. the intermediary statements i.e. lemmata. In terms of propositional reasoning this is expressed by the rule of modus ponens

$$\frac{A \quad A \to B}{B}$$

which is the historically primal example of a cut rule. The presence of such rules in a proof, however, might hide valuable information such as an implicit constructive content.

The introduction of cut-free derivations in the sequent calculus **LK** (**LJ**) in Gerhard Gentzen's seminal papers *Über das logische Schliessen I+II* [38] provided a stable notion of analytic proof for classical (intuitionistic) first-order logic based on the subformula property. The structural rules represent the obvious derivation steps not necessarily related to the result. Gentzen was the first to actually prove, that everything derivable can be derived analytically (the Hauptsatz).

In this book we focus on cut-elimination for classical logic from a procedural point of view. In the tradition of proof theory, the emphasis is on cut-free provability with restricted means, not on the actual elimination of cuts from proofs. We develop a more radical form of cut-elimination using the fact, that the cuts after cancellation of other parts of the proof can be considered as contradictions. The method (called CERES[2] – cut-elimination by resolution) works as follows:

- extract from the parts of the axioms, leading to cuts, a set of clauses (in the sense of the resolution calculus) which is refutable. The set of clauses can be represented by clause terms, which are algebraic objects.

- For every clause, there exists a cut-free part of the original proof (the projection), which derives the original end sequent extended by the clause.

- Refute the set of clauses using resolution, construct a ground resolution proof and augment the clauses with the associated (substitution instances) of projections.

By the method CERES an essentially cut-free proof is obtained. The remaining atomic cuts are easily removable in the presence of logical axioms. This is even not necessary as they do not interfere with the extraction of desired information implicitly contained in proofs as Herbrand disjunctions, interpolants etc. To apply CERES, it is necessary to reduce compound logical axioms to atomic ones and to replace strong quantifiers in the end-sequent by adequate Skolem functions without increasing the complexity of the proof. The elimination of the Skolem functions from a cut-free proof is of at most exponential expense.

CERES simulates the usual cut elimination methods of Gentzen and Schütte/-Tait, here formulated nondeterministically. On the other hand there are sequences of proofs, whose cut-free normal forms according to Gentzen and Schütte/Tait grow nonelementarily w.r.t.[3] the cut-free normal forms according to CERES. The reason is, that usual cut-elimination methods are local in the sense that only a small part of the proof is analyzed, namely the derivation corresponding to the introduction of the uppermost logical connective. As a consequence many types of redundancies in proofs are left undetected leading to a bad computational behaviour.

The strong regularity properties of cut-free normal forms obtained by CERES (the proofs are composed from the projections) together with the simulation results (reductive methods can be simulated by CERES) allow the formulation of negative results also for the traditional methods. For example no cut-free proof, whose Herbrand disjunction is not composed from substitution

[2] http://www.logic.at/ceres

[3] With respect to.

instances of the Herbrand disjunctions of the projections can be obtained by Gentzen or Schütte/Tait cut-elimination.

As intended, CERES is used to extract structural information implicit in proofs with cuts such as interpolants etc. It serves as a tool for the generalization of proofs (justifying the Babylonian reasoning by examples). Furthermore we demonstrate how to apply CERES to the analysis of mathematical proofs using two straightforward examples. CERES relates these proofs with cuts to the spectrum of all cut-free proofs obtainable in a reasonable way. By analyzing CERES itself, we establish easy-to-describe classes of proofs, which admit fast (i.e. elementary) cut elimination. Possibilities and limits of the extension of CERES-like methods to the realm of nonclassical, especially intermediate logics are discussed using the example of first-order Gödel-Dummett logic (i.e. the logic of linearly ordered Kripke structures with constant domains).
We finally stress that the proximity of CERES to the resolution calculus facilitates its implementation (and thereby the implementation of the traditional cut-elimination methods) using state-of-the-art automated theorem proving frameworks. Furthermore, resolution strategies might be employed to express knowledge about cut formulas obvious to mathematicians but usually algorithmically difficult to represent. This includes the difference between the proved lemma (positive occurrence of the cut formula) and its application (negative occurrence of the cut-formula).

Chapter 3

Preliminaries

3.1 Formulas and Sequents

In this chapter we present some basic concepts which will be needed throughout the whole book. We assume that the the reader is familiar with the most basic notions of predicate logic, like terms, formulas, substitutions and interpretations.

We denote predicate symbols by P, Q, R, function symbols by f, g, h, constant symbols by a, b, c. We distinguish a set of free variables V_f and a set of bound variables V_b (both sets are assumed to be countably infinite).

Remark: The distinction between free and bound variables is vital to proof transformations like cut-elimination, where whole proofs have to be instantiated. ◇

We use α, β for free variables and x, y, z for bound ones. Terms are defined as usual with the restriction that they may not contain bound variables.

Definition 3.1.1 (semi-term, term) We define the set of semi-terms inductively:

- bound and free variables are semi-terms,
- constants are semi-terms,
- if $t_1, \ldots, t_n$ are semi-terms and f is an n-place function symbol then $f(t_1, \ldots, t_n)$ is a semi-term.

◇

Semi-terms which do not contain bound variables are called terms.

M. Baaz, A. Leitsch, *Methods of Cut-Elimination*, Trends in Logic 34,
DOI 10.1007/978-94-007-0320-9_3, © Springer Science+Business Media B.V. 2011

Example 3.1.1 $f(\alpha,\beta)$ is a term. $f(x,\beta)$ is a semi-term. $P(f(\alpha,\beta))$ is a formula. $\diamond$

Replacement on positions play a central role in proof transformations. We first introduce the concept of position for terms.

Definition 3.1.2 (position) We define the positions within semi-terms inductively:

- If t is a variable or a constant symbol then ϵ is a position in t and $t.\epsilon = t$
- Let $t = f(t_1,\ldots,t_n)$ then ϵ is a position in t and $t.\epsilon = t$. Let μ be a position in a t_j (for $1 \leq j \leq n$), $\mu = (k_1,\ldots,k_l)$ and $t_j.\mu = s$; then ν, for $\nu = (j,k_1,\ldots,k_l)$, is a position in t and $t.\nu = s$.

$\diamond$

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

Example 3.1.2 Let $t = f(f(\alpha,\beta),a)$ be a term. Then

$$\begin{array}{rcl}
t.\epsilon & = & t, \\
t.(1) & = & f(\alpha,\beta), \\
t.(2) & = & a, \\
t.(1,1) & = & \alpha, \\
t.(1,2) & = & \beta, \\
t[g(a)].(1,1) & = & f(g(a),\beta).
\end{array}$$

$\diamond$

Positions in formulas can be defined in the same way (the simplest way is to consider all formulas as terms).

Definition 3.1.3 (substitution) A *substitution* is a mapping from $V_f \cup V_b$ to the set of *semi-terms* s.t. $\sigma(v) \neq v$ for only finitely many $v \in V_f \cup V_b$.

If σ is a substitution with $\sigma(x_i) = t_i$ for $x_i \neq t_i$ $(i = 1, \ldots, n)$ and $\sigma(v) = v$ for $v \notin \{x_1, \ldots, x_n\}$ then we denote σ by $\{x_1 \leftarrow t_1, \ldots, x_n \leftarrow t_n\}$. We call the set $\{x_1 \leftarrow t_1, \ldots, x_n \leftarrow t_n\}$ the *domain* of σ and denote it by $dom(\sigma)$. Substitutions are written in postfix, i.e. we write $F\sigma$ instead of $\sigma(F)$. $\diamond$

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

Definition 3.1.4 A substitution σ is called *more general* than a substitution ϑ $(\sigma \leq_s \vartheta)$ if there exists a substitution μ s.t. $\vartheta = \sigma\mu$. $\diamond$

Example 3.1.3 Let $\vartheta = \{x \leftarrow a, y \leftarrow a\}$ and $\sigma = \{x \leftarrow y\}$. Then $\sigma\mu = \vartheta$ for $\mu = \{y \leftarrow a\}$ and thus $\sigma \leq_s \vartheta$. Note that for the identical substitution we get $\emptyset \leq_s \lambda$ for all substitutions λ. $\diamond$

Definition 3.1.5 (semi-formula, formula) $\top$ and $\bot$ are formulas.
If $t_1, \ldots, t_n$ are terms and P is an n-place predicate symbol then $P(t_1, \ldots, t_n)$ is an (atomic) formula.

- If A is a formula then $\neg A$ is a formula.
- If A, B are formulas then $(A \rightarrow B)$, $(A \wedge B)$ and $(A \vee B)$ are formulas.
- If $A\{x \leftarrow \alpha\}$ is a formula then $(\forall x)A, (\exists x)A$ are formulas.

Semi-formulas differ from formulas in containing free variables in V_b. $\diamond$

Example 3.1.4 $P(f(\alpha, \beta))$ is a formula, and so is $(\forall x)P(f(x, \beta))$. $P(f(x, \beta))$ is a semi-formula. $\diamond$

Definition 3.1.6 (logical complexity of formulas) If F is a formula in PL then the complexity $comp(F)$ is the number of logical symbols occurring in F. Formally we define

$comp(F) = 0$ if F is an atomic formula,

$comp(F) = 1 + comp(A) + comp(B)$ if $F \equiv A \circ B$ for $\circ \in \{\wedge, \vee, \rightarrow\}$,

$comp(F) = 1 + comp(A)$ if $F \equiv \neg A$ or $F \equiv (Qx)A$ for $Q \in \{\forall, \exists\}$ and $x \in V_b$.

$\diamond$

Gentzen's famous calculus **LK** is based on so called sequents; sequents are structures with sequences of formulas on the left and on the right hand side of a symbol which does not belong to the syntax of formulas. We call this symbol *the sequent sign* and denote it by $\vdash$.

Definition 3.1.7 (sequent) Let Γ and Δ be finite (possibly empty) sequences of formulas. Then the expression $S\colon \Gamma \vdash \Delta$ is called a *sequent.* Γ is called the *antecedent* of S and Δ the *consequent* of S. $\diamond$

Let

$$\bigwedge_{i=1}^{1} A_i = A_1,\ \bigwedge_{i=1}^{n+1} A_i = A_{n+1} \wedge \bigwedge_{i=1}^{n} A_i \text{ for } n \geq 1,$$

and analogous for $\bigvee$.

Definition 3.1.8 (semantics of sequents) Semantically a sequent

$$S\colon A_1, \ldots, A_n \vdash B_1, \ldots, B_m$$

stands for

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

In particular we define $\mathcal{M}$ to be an interpretation of S if $\mathcal{M}$ is an interpretation of $F(S)$. If $n = 0$ (i.e. there are no formulas in the antecedent of S) we assign $\top$ to $\bigwedge_{i=1}^{n} A_i$, if $m = 0$ we assign $\bot$ to $\bigvee_{j=1}^{m} B_j$. Note that the empty sequent is represented by $\top \to \bot$ which is equivalent to $\bot$ and represents falsum. We say that S is true in $\mathcal{M}$ if $F(S)$ is true in $\mathcal{M}$. S is called *valid* if $F(S)$ is valid. $\diamond$

Example 3.1.5

$$S\colon\ P(a), (\forall x)(P(x) \to P(f(x))) \vdash P(f(a))$$

is a sequent. The corresponding formula

$$F(S)\colon\ (P(a) \wedge (\forall x)(P(x) \to P(f(x)))) \to P(f(a))$$

is valid; so S is a valid sequent. $\diamond$

Definition 3.1.9 A sequent $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$ is called *atomic* if the A_i, B_j are atomic formulas. $\diamond$

Definition 3.1.10 (composition of sequents) If $S = \Gamma \vdash \Delta$ and $S' = \Pi \vdash \Lambda$ we define the composition of S and S' by $S \circ S'$, where $S \circ S' = \Gamma, \Pi \vdash \Delta, \Lambda$. ◇

Definition 3.1.11 Let Γ be a sequence of formulas. Then we write $\Gamma - A$ for Γ after deletion of all occurrences of A. Formally we define

$$\begin{aligned}(A_1, \ldots A_n) - A &= (A_2, \ldots A_n) - A \text{ for } A = A_1,\\ &= A_1, ((A_2, \ldots A_n) - A) \text{ for } A \neq A_1,\\ \epsilon - A &= \epsilon.\end{aligned}$$

◇

Definition 3.1.12 (permutation of sequents) Let S be the sequent $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$, π be a permutation of $\{1, \ldots, n\}$, and π' be a permutation of $\{1, \ldots, m\}$. Then the sequent

$$S' \colon A_{\pi(1)}, \ldots, A_{\pi(n)} \vdash B_1, \ldots, B_m$$

is called a *left permutation* of S (based on π), and

$$S'' \colon A_1, \ldots, A_n \vdash B_{\pi'(1)}, \ldots, B_{\pi'(m)}$$

is called a *right permutation* of S (based on π'). A *permutation* of S is a left permutation of a right permutation of S. ◇

Definition 3.1.13 (subsequent) Let S, S' be sequents. We define $S' \sqsubseteq S$ if there exists a sequent S'' s.t. $S' \circ S''$ is a permutation of S and call S' a *subsequent* of S. ◇

Example 3.1.6 S': $P(b) \vdash Q(a)$ is a subsequent of

$$S\colon\ P(a), P(b), P(c) \vdash Q(a), Q(b).$$

S'' has to be defined as $P(a), P(c) \vdash Q(b)$. Then clearly

$$S' \circ S'' = P(b), P(a), P(c) \vdash Q(a), Q(b).$$

The left permutation (12) then gives S. ◇

By definition of the semantics of sequents, every sequent is implied by all of its subsequents. The empty sequent (which stands for falsum) implies every sequent.

Definition 3.1.14 Substitutions can be extended to sequents in an obvious way. If $S = A_1, \dots, A_n \vdash B_1, \dots, B_m$ and σ is a substitution then

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

◇

Definition 3.1.15 (polarity) Let λ be an occurrence of a formula A in B. If $A \equiv B$ then λ is a positive occurrence in B. If $B \equiv (C \wedge D), B \equiv (C \vee D), B \equiv (\forall x)C$ or $B \equiv (\exists x)C$ and λ is a positive (negative) occurrence of A in C (or in D respectively) then the corresponding occurrence λ' of A in B is positive (negative). If $B \equiv (C \rightarrow D)$ and λ is a positive (negative) occurrence of A in D then the corresponding occurrence λ' in B is positive (negative); if, on the other hand, λ is a positive (negative) occurrence of A in C then the corresponding occurrence λ' of A in B is negative (positive). If $B \equiv \neg C$ and λ is a positive (negative) occurrence of A in C then the corresponding occurrence λ' of A in B is negative (positive). If there exists a positive (negative) occurrence of a formula A in B we say that A is of positive (negative) polarity in B. ◇

Definition 3.1.16 (strong and weak quantifiers)
If $(\forall x)$ occurs positively (negatively) in B then $(\forall x)$ is called a strong (weak) quantifier. If $(\exists x)$ occurs positively (negatively) in B then $(\exists x)$ is called a weak (strong) quantifier. ◇

Note that (Qx) may occur several times in a formula B; thus it may be strong and weak at the same time. If confusion might arise we refer to the specific position of (Qx) in B. In particular we may replace every formula A by a logically equivalent "variant" A' s.t. every (Qx) (for $Q \in \{\forall, \exists\}$ and $x \in V$) occurs at most once in A'. In this case the term "(Qx) is a strong (weak) quantifier" has a unique meaning.

Definition 3.1.17 A sequent S is called *weakly quantified* if all quantifier occurrences in S are weak. ◇

3.2 The Calculus LK

Like most other calculi Gentzen's **LK** is based on axioms and rules.

Definition 3.2.1 (axiom set) A (possibly infinite) set $\mathcal{A}$ of sequents is called an *axiom set* if it is closed under substitution, i.e., for all $S \in \mathcal{A}$ and for all substitutions θ we have $S\theta \in \mathcal{A}$. If $\mathcal{A}$ consists only of atomic sequents we speak about an *atomic axiom set.* ◇

Remark: The closure under substitution is required for proof transformations, in particular for cut-elimination. ◇

Definition 3.2.2 (standard axiom set) Let $\mathcal{A}_T$ be the smallest axiom set containing all sequents of the form $A \vdash A$ for arbitrary atomic formulas A. A_T is called the *standard axiom set.* ◇

Definition 3.2.3 (LK) Basically we use Gentzen's version of **LK** (see [38]) with the exception of the permutation rule. There are two groups of rules, the logical and the structural ones. All rules with the exception of cut have left and right versions; left versions are denoted by $\xi\colon l$, right versions by $\xi\colon r$. Every logical rule introduces a logical operator on the left or on the right side of a sequent. Structural rules serve the purpose of making logical inferences possible (e.g. permutation) or to put proofs together (cut). A and B denote formulas, $\Gamma, \Delta, \Pi, \Lambda$ sequences of formulas. In the rules there are introducing or *auxiliary formulas* (in the premises) and introduced or *principal formulas* in the conclusion. We indicate these formulas for all rules. In particular we mark the auxiliary formula occurrences by $+$ and the principal ones by $\star$. We frequently say auxiliary (main) *formula* instead of auxiliary (main) formula occurrence.

The logical rules:

- $\wedge$-introduction:

$$\frac{A^+, \Gamma \vdash \Delta}{(A \wedge B)^\star, \Gamma \vdash \Delta}\ \wedge\colon l_1 \quad \frac{B^+, \Gamma \vdash \Delta}{(A \wedge B)^\star, \Gamma \vdash \Delta}\ \wedge\colon l_2 \quad \frac{\Gamma \vdash \Delta, A^+ \quad \Gamma \vdash \Delta, B^+}{\Gamma \vdash \Delta, (A \wedge B)^\star}\ \wedge\colon r$$

- $\vee$-introduction:

$$\frac{A^+, \Gamma \vdash \Delta \quad B^+, \Gamma \vdash \Delta}{(A \vee B)^\star, \Gamma \vdash \Delta}\ \vee\colon l \quad \frac{\Gamma \vdash \Delta, A^+}{\Gamma \vdash \Delta, (A \vee B)^\star}\ \vee\colon r_1 \quad \frac{\Gamma \vdash \Delta, B^+}{\Gamma \vdash \Delta, (A \vee B)^\star}\ \vee\colon r_2$$

- $\rightarrow$-introduction:

$$\frac{\Gamma \vdash \Delta, A^+ \quad B^+, \Pi \vdash \Lambda}{(A \rightarrow B)^\star, \Gamma, \Pi \vdash \Delta, \Lambda}\ \rightarrow\colon l \quad \frac{A^+, \Gamma \vdash \Delta, B^+}{\Gamma \vdash \Delta, (A \rightarrow B)^\star}\ \rightarrow\colon r$$

- $\neg$-introduction:

$$\frac{\Gamma \vdash \Delta, A^+}{\neg A^\star, \Gamma \vdash \Delta}\ \neg\colon l \quad \frac{A^+, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg A^\star}\ \neg\colon r$$

- $\forall$-introduction:

$$\frac{A\{x \leftarrow t\}^{+}, \Gamma \vdash \Delta}{(\forall x)A^{\star}, \Gamma \vdash \Delta}\ \forall\colon l$$

  where t is an arbitrary *term.*

$$\frac{\Gamma \vdash \Delta, A\{x \leftarrow \alpha\}^{+}}{\Gamma \vdash \Delta, (\forall x)A^{\star}}\ \forall\colon r$$

  where α is a free variable which may not occur in Γ, Δ, A. α is called an *eigenvariable.*

- The logical rules for $\exists$-introduction (the variable conditions for $\exists : l$ are the same as those for $\forall\colon r$, and similarly for $\exists : r$ and $\forall\colon l$):

$$\frac{A\{x \leftarrow \alpha\}^{+}, \Gamma \vdash \Delta}{(\exists x)A^{\star}, \Gamma \vdash \Delta}\ \exists\colon l \qquad \frac{\Gamma \vdash \Delta, A\{x \leftarrow t\}^{+}}{\Gamma \vdash \Delta, (\exists x)A^{\star}}\ \exists\colon r$$

The structural rules:

- *permutation*

$$\frac{S}{S'}\ \pi\colon l \qquad \frac{S}{S''}\ \pi'\colon r$$

  where S' is a left permutation of S based on π, and S'' is a right permutation of S based on π' . In $(\colon l\pi) : l$ all formulas on the left side of S' are principal formulas and all formulas on the left side of S are auxiliary formulas; similarly for $p(\pi) : r$. Mostly we write the rules in the form

$$\frac{S}{S'}\ p\colon l \qquad \frac{S}{S''}\ p\colon r$$

  when we not interested in specifying the particular permutation.

- *weakening*:

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, A^{\star}}\ w\colon r \qquad \frac{\Gamma \vdash \Delta}{A^{\star}, \Gamma \vdash \Delta}\ w\colon l$$

- *contraction:*

$$\frac{A^{+}, A^{+}, \Gamma \vdash \Delta}{A^{\star}, \Gamma \vdash \Delta}\ c\colon l \qquad \frac{\Gamma \vdash \Delta, A^{+}, A^{+}}{\Gamma \vdash \Delta, A^{\star}}\ c\colon r$$

- The *cut rule:* Let us assume that A occurs in Δ and in Π. Then we define
$$\frac{\Gamma \vdash \Delta \quad \Pi \vdash \Lambda}{\Gamma, \Pi^* \vdash \Delta^*, \Lambda}\ cut(A)$$
where Π^* is Π after deletion of at least one occurrence of A, and Δ^* is Δ after deletion of at least one occurrence of A. The formula A is the auxiliary formula of $cut(A)$ and there is no principal one. If $\Pi^* = \Pi - A$ and $\Delta^* = \Delta - A$, i.e. we delete all occurrences of A in Π and Δ we speak about a *mix.* If A is not an atomic formula we call the cut *essential*, and *inessential* if A is an atom.

  The cut rule can be simulated by mix and other structural rules. Indeed let ψ be the proof
$$\frac{\begin{matrix}(\psi_1) & (\psi_2)\\ \Gamma \vdash \Delta & \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi^* \vdash \Delta^*, \Lambda}\ cut(A)$$
  Then the proof ψ':
$$\frac{\dfrac{\begin{matrix}(\psi_1) & (\psi_2)\\ \Gamma \vdash \Delta & \Pi \vdash \Lambda\end{matrix}}{\Gamma, \Pi - A \vdash \Delta - A, \Lambda}\ mix(A)}{\Gamma, \Pi^* \vdash \Delta^*, \Lambda}\ w^* + p^*$$
  is a derivation of the same end sequent. The number of additional weakenings is bounded by the number of occurrences of A in Π and Δ. At most two permutations are necessary to obtain the desired end sequent.

  Note that the version of cut we are defining here is more general than the cut and mix rules in Gentzen's original paper. If we delete only one occurrence of A in Π and Δ we obtain the cut rule (according to Gentzen's terminology); if we delete all occurrences in Π and Δ we get a mix (which corresponds to Gentzen's terminology). As we are dealing with classical logic only this version of cut does not lead to problems and makes the analysis of cut-elimination more comfortable.

$\diamond$

Definition 3.2.4 Let
$$\frac{S_1 \quad S_2}{S}\ \xi$$

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

Example 3.2.1 Consider the rule

$$\frac{\Gamma \vdash \Delta, A^+ \quad \Gamma \vdash \Delta, B^+}{\Gamma \vdash \Delta, (A \wedge B)^\star} \wedge\colon r$$

Then

$$\frac{(\forall x)P(x), (\forall x)Q(x) \vdash P(a)^+ \quad (\forall x)P(x), (\forall x)Q(x) \vdash Q(b)^+}{(\forall x)P(x), (\forall x)Q(x) \vdash (P(a) \wedge Q(b))^\star} \wedge\colon r$$

is an instance of $\wedge\colon r$. ◇

Definition 3.2.5 (LK-derivation) An **LK**-*derivation* is defined as a finite directed labeled tree where the nodes are labelled by sequents (via the function *Seq*) and the edges by the corresponding rule applications. The label of the root is called the *end-sequent*. Sequents occurring at the leaves are called *initial sequents* or *axioms*. We give a formal definition:

- Let ν be a node and $Seq(\nu) = S$ for an arbitrary sequent S. Then ν is an **LK**-derivation and ν is the root node (and also a leaf).

- Let φ be a derivation tree and ν be a leaf in φ. Let (S_1, S_2, S) be an instance of the binary **LK**-rule ξ. We extend φ to φ' by appending the edges $e_1\colon(\nu, \mu_1)$, $e_2\colon(\nu, \mu_2)$ to ν s.t. $Seq(\mu_1) = S_1$, $Seq(\mu_2) = S_2$, and the label of e_1, e_2 is ξ. Then φ' is an **LK**-derivation with the same root as φ. μ_1, μ_2 are leaves in φ', but ν is not. ν is called a ξ-node in φ'.

- Let φ be a derivation tree and ν be a leaf in φ. Let (S', S) be an instance of a unary **LK**-rule ξ. We extend φ to φ' by appending the edge $e\colon(\nu, \mu)$ to ν s.t. $Seq(\mu) = S'$, and the label of e is ξ. Then φ' is an **LK**-derivation with the same root as φ. μ is a leaf in φ', but ν is not. Again ν is called a ξ-node in φ'.

We write

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

to express that ψ is an **LK**- derivation with end sequent S. ◇

Definition 3.2.6 Let φ be an **LK**-derivation with initial sequent S and end sequent S' s.t. all edges are labelled by unary structural rules (these are all structural rules with the exception of cut). Then we may represent φ by

$$\frac{S}{S'}\ s^*$$

Moreover, if the structural rules are only weakenings we may write w^* instead of s^*, for weakenings and permutations $(w+p)^*$, for arbitrary weakenings and one permutation w^*+p. This notation applies to any combination of unary structural rules, where w stands for weakening, p for permutation and c for contraction. ◇

Example 3.2.2 Let φ be the **LK**-derivation

$$\cfrac{\cfrac{\nu_1\colon P(a) \vdash P(a)}{\nu_2\colon (\forall x)P(x) \vdash P(a)}\ \forall\colon l \qquad \cfrac{\nu_3\colon P(a) \vdash Q(a)}{\nu_4\colon P(a) \vdash (\exists x)Q(x)}\ \exists\colon r}{\cfrac{\nu_5\colon (\forall x)P(x) \vdash (\exists x)Q(x)}{\nu_6\colon \vdash (\forall x)P(x) \to (\exists x)Q(x)}\ \to\colon r}\ \mathit{cut}$$

The ν_i denote the nodes in φ. The leaf nodes are ν_1 and ν_3, the end node is ν_6. $\mathit{Seq}(\nu_2) = (\forall x)P(x) \vdash P(a)$. In practice the representation of nodes is omitted in writing down **LK**-proofs. ◇

Definition 3.2.7 (cut-complexity) Let φ be an **LK**-derivation with cuts and $\mathcal{C}$ be the set of all cut-formulas occurring in φ . Then $\max\{\mathit{comp}(A) \mid A \in \mathcal{C}\}$ is called the *cut-complexity* of φ and is denoted by $\mathit{cutcomp}(\varphi)$. If φ is cut-free (i.e. $\mathcal{C} = \emptyset$) we define $\mathit{cutcomp}(\varphi) = -1$ ◇

Example 3.2.3 Let φ be the **LK**-derivation in Example 3.2.2. Then

$$\mathit{cutcomp}(\varphi) = 0.$$

In fact the only cut formula in φ is $P(a)$ which is atomic. ◇

Definition 3.2.8 Let $\mathcal{A}$ be an axiom set. An **LK**-*proof* φ of S from $\mathcal{A}$ is an **LK**-derivation of S with initial sequents in $\mathcal{A}$. If $\mathcal{A}$ is the standard axiom set we simply call φ a proof of S. The set of all **LK**-proofs from $\mathcal{A}$ is denoted by $\Phi^{\mathcal{A}}$. If the axiom set $\mathcal{A}$ is clear from the context we frequently write Φ. For all $i \geq 0$ we define:

$$\Phi_i^{\mathcal{A}} \quad = \quad \{\varphi \mid \varphi \in \Phi^{\mathcal{A}},\ \mathit{cutcomp}(\varphi) \leq i\}.$$

The set of cut-free proofs is denoted by $\Phi_{\emptyset}^{\mathcal{A}}$. ◇

Example 3.2.4 Let $\mathcal{A} = \{P(a) \vdash P(a),\ P(a) \vdash Q(a)\}$. Then $\mathcal{A}$ is an axiom set (indeed there are no variables in the sequents of $\mathcal{A}$). The **LK**-derivation φ, defined in Example 3.2.2, is an **LK**-proof of $\mathit{Seq}(\nu_6)$ from $\mathcal{A}$, i.e. $\varphi \in \Phi^{\mathcal{A}}$. Moreover $\varphi \in \Phi_0^{\mathcal{A}}$. Note that $\mathcal{A}$ is not a subset of the standard axiom set. ◇

Definition 3.2.9 (path) Let π: $\mu_1, \ldots, \mu_n$ be a sequence of nodes in an **LK**-derivation φ s.t. for all $i \in \{1, \ldots, n-1\}$ (μ_i, μ_{i+1}) is an edge in φ. Then π is called a *path* from μ_1 to μ_n in φ of *length* $n-1$ (denoted by $\mathit{lp}(\pi) = n-1$). If $n = 1$ and $\pi = \mu_1$ then ψ is called a trivial path. π is called a *branch* if μ_1 is the root of φ and μ_n is a leaf in φ. We use the terms *predecessor* and *successor* contrary to the direction of edges in the tree: if there exists a path from μ_1 to μ_2 then μ_2 is called a *predecessor* of μ_1. The successor relation is defined in a analogous way. E.g. every initial sequent is a predecessor of the end sequent. ◇

Example 3.2.5 Let $\varphi =$

$$\cfrac{\cfrac{\nu_1\colon P(a) \vdash P(a)}{\nu_2\colon (\forall x)P(x) \vdash P(a)}\;\forall\colon l \qquad \cfrac{\nu_3\colon P(a) \vdash Q(a)}{\nu_4\colon P(a) \vdash (\exists x)Q(x)}\;\exists\colon r}{\cfrac{\nu_5\colon (\forall x)P(x) \vdash (\exists x)Q(x)}{\nu_6\colon \vdash (\forall x)P(x) \to (\exists x)Q(x)}\;\to\colon r}\;\mathit{cut}$$

as in Example 3.2.2. $\nu_6, \nu_5, \nu_2, \nu_1$ is a path in φ which is also a branch. ν_2 is a predecessor of ν_6. ν_1 is not a predecessor of ν_4. ◇

Definition 3.2.10 (subderivation) Let φ' be the subtree of an **LK**-derivation φ with root node ν (where ν is a node in φ). Then φ' is called a *subderivation* of φ and we write $\varphi' = \varphi.\nu$.
Let ρ be an (arbitrary) **LK**-derivation of $\mathit{Seq}(\nu)$. Then we write $\varphi[\rho]_\nu$ for the deduction φ after the replacement of the subderivation $\varphi.\nu$ by ρ on the node ν in φ (under the restriction that $\varphi.\nu$ and ρ have the same end-sequent). ◇

Example 3.2.6 Let $\varphi =$

$$\cfrac{\cfrac{\nu_1\colon P(a) \vdash P(a)}{\nu_2\colon (\forall x)P(x) \vdash P(a)}\;\forall\colon l \qquad \cfrac{\nu_3\colon P(a) \vdash Q(a)}{\nu_4\colon P(a) \vdash (\exists x)Q(x)}\;\exists\colon r}{\cfrac{\nu_5\colon (\forall x)P(x) \vdash (\exists x)Q(x)}{\nu_6\colon \vdash (\forall x)P(x) \to (\exists x)Q(x)}\;\to\colon r}\;\mathit{cut}$$

$\varphi.\nu_4 =$

$$\cfrac{\nu_3\colon P(a) \vdash Q(a)}{\nu_4\colon P(a) \vdash (\exists x)Q(x)}\;\exists\colon r$$

Let $\rho =$

$$\begin{prooftree}
\AxiomC{$\nu_8\colon P(a), P(a) \vdash Q(a)$}
\RightLabel{$\exists\colon r$}
\UnaryInfC{$\nu_9\colon\ P(a), P(a) \vdash (\exists x)Q(x)$}
\RightLabel{$c\colon l$}
\UnaryInfC{$\nu_{10}\colon\ P(a) \vdash (\exists x)Q(x)$}
\end{prooftree}$$

Then $\varphi[\rho]_{\nu_4} =$

$$\begin{prooftree}
\AxiomC{$\nu_1\colon P(a) \vdash P(a)$}
\RightLabel{$\forall\colon l$}
\UnaryInfC{$\nu_2\colon (\forall x)P(x) \vdash P(a)$}
\AxiomC{$\nu_8\colon P(a), P(a) \vdash Q(a)$}
\RightLabel{$\exists\colon r$}
\UnaryInfC{$\nu_9\colon\ P(a), P(a) \vdash (\exists x)Q(x)$}
\RightLabel{$c\colon l$}
\UnaryInfC{$\nu_{10}\colon\ P(a) \vdash (\exists x)Q(x)$}
\RightLabel{cut}
\BinaryInfC{$\nu_5\colon (\forall x)P(x) \vdash (\exists x)Q(x)$}
\RightLabel{$\to\colon r$}
\UnaryInfC{$\nu_6\colon\ \vdash (\forall x)P(x) \to (\exists x)Q(x)$}
\end{prooftree}$$

Note that $\varphi[\rho]_{\nu_4}$ is an **LK**-proof from the axiom set

$$\{P(a) \vdash P(a);\ P(a), P(a) \vdash Q(a)\}.$$

$\diamond$

Definition 3.2.11 (depth) Let φ be an **LK**-derivation and ν be a node in φ. Then the *depth* of ν (denoted by $\mathrm{depth}(\nu)$) is defined by the maximal length of a path from ν to a leaf of $\varphi.\nu$. The depth of any leaf in φ is zero. $\diamond$

Definition 3.2.12 (regularity) An **LK**-derivation φ is called *regular* if

- all eigenvariables of quantifier introductions $\forall\colon r$ and $\exists\colon l$ in φ are mutually different.
- If an eigenvariable α occurs as an eigenvariable in a proof node ν then α occurs only above ν in the proof tree.

$\diamond$

There exists a straightforward transformation from **LK**-derivations into regular ones: just rename the eigenvariables in different subderivations. The necessity of renaming variables was the main motivation for changing Hilbert's linear format to the tree format of **LK**. From now on we assume, without mentioning the fact explicitly, that all **LK**-derivations we consider are regular.

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

Example 3.2.7 Instead of using special symbols for formula occurrences we mark the occurrences of a formula in different sequents by numbers. Let $\varphi =$

$$\dfrac{\dfrac{\dfrac{\nu_1\colon P(a)^4 \vdash P(a)}{\nu_2\colon (\forall x)P(x)^5 \vdash P(a)}\ \forall\colon l \qquad \dfrac{\nu_3\colon P(a) \vdash Q(a)^1}{\nu_4\colon P(a) \vdash (\exists x)Q(x)^2}\ \exists\colon r}{\nu_5\colon (\forall x)P(x)^6 \vdash (\exists x)Q(x)^3}\ \mathit{cut}}{\vdash (\forall x)P(x) \to (\exists x)Q(x)^7}\ \to\colon r$$

1 is ancestor of 2, 2 is ancestor of 3, 3 is ancestor of 7. 1 is ancestor of 3 and of 7. 4 is ancestor of 5, 5 of 6 and 6 of 7. 4 is ancestor of 7, but not of 2. ◇

Definition 3.2.13 (ancestor path) A sequence $\bar{\alpha}\colon (\alpha_1, \ldots, \alpha_n)$ for formula occurrences α_i in an **LK**-derivation φ is called an *ancestor path* in φ if for all $i \in \{1, \ldots, n-1\}$ α_i is an immediate ancestor of α_{i+1}. If $n = 1$ then α_1 is called a (trivial) ancestor path. ◇

Example 3.2.8 In Example 3.2.7 the sequence $4, 5, 6, 7$ is an ancestor path. ◇

Definition 3.2.14 Let Ω be a set of formula occurrences in an **LK**-derivation φ and ν be a node in φ. Then $S(\nu, \Omega)$ is the subsequent of $\mathit{Seq}(\nu)$ obtained by deleting all formula occurrences which are not ancestors of occurrences in Ω. ◇

Example 3.2.9 Let $\varphi =$

$$\dfrac{\dfrac{\dfrac{\nu_1\colon P(a) \vdash P(a)}{\nu_2\colon (\forall x)P(x) \vdash P(a)}\ \forall\colon l \qquad \dfrac{\nu_3\colon P(a) \vdash Q(a)}{\nu_4\colon P(a) \vdash (\exists x)Q(x)}\ \exists\colon r}{\nu_5\colon (\forall x)P(x) \vdash (\exists x)Q(x)}\ \mathit{cut}}{\nu_6\colon\ \vdash (\forall x)P(x) \to (\exists x)Q(x)}\ \to\colon r$$

and α the left occurrence of the cut formula in φ, and β the right occurrence. Let $\Omega = \{\alpha, \beta\}$. Then

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

◇

Remark: If Ω consists just of the occurrences of all cut formulas which occur "below" ν then $S(\nu, \Omega)$ is the subsequent of $Seq(\nu)$ consisting of all formulas which are ancestors of a cut. These subsequents are crucial for the definition of the characteristic set of clauses and of the method CERES in Chapter 6. ◇

Definition 3.2.15 The *length* of a proof φ is defined by the number of nodes in φ and is denoted by $l(\varphi)$. ◇

Definition 3.2.16 (cut-derivation) Let ψ be an **LK**-derivation of the form

$$\dfrac{\begin{array}{cc} (\psi_1) & (\psi_2) \\ \Gamma_1 \vdash \Delta_1 & \Gamma_2 \vdash \Delta_2 \end{array}}{\Gamma_1, \Gamma_2^* \vdash \Delta_1^*, \Delta_2}\; cut(A)$$

Then ψ is called a *cut-derivation*; note that ψ_1 and ψ_2 may contain cuts. If the cut is a mix we speak about a *mix-derivation*. ψ is called *essential* if $comp(A) > 0$ (i.e. if the cut is essential). ◇

Definition 3.2.17 (rank, grade) Let ψ be a cut-derivation of the form

$$\dfrac{\begin{array}{cc} (\psi_1) & (\psi_2) \\ \Gamma_1 \vdash \Delta_1 & \Gamma_2 \vdash \Delta_2 \end{array}}{\Gamma_1, \Gamma_2^* \vdash \Delta_1^*, \Delta_2}\; cut(A)$$

Then we define the *grade* of ψ as $comp(A)$.
Let μ be the root node of ψ_1 and ν be the root node of ψ_2. An A-right path in ψ_1 is a path in ψ_1 of the form $\mu, \mu_1, \ldots, \mu_n$ s.t. A occurs in the consequents of all $Seq(\mu_i)$ (note that A clearly occurs in Δ_1). Similarly an A-left path in ψ_2 is a path in ψ_2 of the form $\nu, \nu_1, \ldots, \nu_m$ s.t. A occurs in the antecedents of all $Seq(\nu_j)$. Let P_1 be the set of all A-right paths in ψ_1 and P_2 be the set of all A-left paths in ψ_2. Then we define the *left-rank* of ψ ($\mathrm{rank}_l(\psi)$) and the right-rank of ψ ($\mathrm{rank}_r(\psi)$) as

$$\begin{array}{rcl} \mathrm{rank}_l(\psi) & = & \max\{lp(\pi) \mid \pi \in P_1\} + 1, \\ \mathrm{rank}_r(\psi) & = & \max\{lp(\pi) \mid \pi \in P_2\} + 1. \end{array}$$

The *rank* of ψ is the sum of right-rank and left-rank, i.e. $\mathrm{rank}(\psi) = \mathrm{rank}_l(\psi) + \mathrm{rank}_r(\psi)$. ◇