FAKULTÄT FÜR !NFORMATIK

Faculty of Informatics

Cut-Elimination in Functional Higher-Order Logic

MASTERARBEIT

zur Erlangung des akademischen Grades

Master of Science

im Rahmen des Studiums

Computational Logic

eingereicht von

Nika Pona
Matrikelnummer 1527977

an der Fakultät für Informatik

der Technischen Universität Wien

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

Wien, 21. September 2016

Nika Pona　　　Alexander Leitsch

Cut-Elimination in Functional Higher-Order Logic

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Computational Logic

by

Nika Pona
Registration Number 1527977

to the Faculty of Informatics

at the TU Wien

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

Vienna, 21st September, 2016

Nika Pona | Alexander Leitsch

Erklärung zur Verfassung der Arbeit

Nika Pona
Sankt-Johann-Gasse 1-5/4/9
1050 Wien

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

Wien, 21. September 2016

Nika Pona

Kurzfassung

Es ist bekannt, dass kein elementarer Beweis der Schnittelimination für den Sequenzenkalkül der Logiken höherer Stufe möglich ist. Das beruht darauf, dass ein solcher Beweis einen elementaren Beweis der Widerspruchsfreiheit der Peano Arithmetik liefern würde. Deshalb sind alle Beweise der Schnittelimination für die Logiken höherer Stufe semantisch (durch die Henkin Semantik) oder teilweise semantisch (durch die Resolution, CERES). In dieser Diplomarbeit wird eine Teil-Logik der Logik höherer Stufe mit der Quantifizierung nur über Objekte funktionalen Typs untersucht. Der funktionale Typ ist jeder Typ der einfachen Typentheorie, der ohne den Typ Boolean definiert ist. Diese Einschränkung definiert eine Logik mit einer leicht handhabbaren Beweistheorie, die dennoch einige interessante Eigenschaften der vollen Logik höherer Stufe hat. Zuerst geben wir Definitionen der Syntax und der Semantik von zwei Versionen der Logik dieser Art und beweisen die entsprechenden Sätze über Schnittelimination. Einfache Modifizierungen der Beweise der Schnittelimination für den Sequenzenkalkül der Logik erster Stufe von Gentzen sind für dieses Ziel hinreichend. Man muss nur die $\beta\eta$-Gleichheit in den Kalkül integrieren und dann das Argument für die Logik erster Stufe mit Gleichheit adaptieren. Beweistheoretisch betrachtet ist diese Logik ähnlicher der Logik erster Stufe als der Logik höherer Stufe. Das wird klar durch die Übersetzung von Beweisen der funktionalen Logik höherer Stufe in Beweise der Logik erster Stufe, die in dieser Diplomarbeit definiert ist. Solche Übersetzungen ermöglichen eine neue Sichtweise von Skolemizierung und Unifizierung, die beide in der Logik höherer Stufe problematisch sind. Weiters untersuchen wir die Sematik dieser Logik: wir beweisen ihre Unvollständigkeit bezüglich der Standardsemantik und geben einen Beweis der Vollständigkeit bezüglich der verallgemeinerten Semantik, welcher der Methode von Schütte folgt. Dieser Beweis ist einfacher und eleganter als die Spezifizierung des Beweises des Vollständigkeitssatzes für die Typentheorie von Henkin; überdies illustriert der Beweis die enge Beziehung der funktionalen Logik höherer Stufe zur Logik erster Stufe.

Abstract

It is known that there can be no elementary (reductive) proof of cut-elimination for a sequent calculus of higher-order logic, since this would provide an elementary proof of consistency of PA. Thus the cut-elimination proofs for higher-order calculi are either semantic (through Henkin semantics) or semi-semantic (through resolution, CERES). In this thesis I study a sub-logic of higher-order logic in which quantification is restricted to objects of functional type only. By functional type I mean all types of simple type theory without the occurrence of the Boolean type. Such restriction gives rise to a logic that has a manageable proof-theory and at the same time shares some interesting properties with full higher-order logic. First, I define syntax and semantics of two variants of such a logic and give the corresponding cut-elimination proofs. These are easy adaptations of the Gentzen's proof for $\mathcal{LK}$: one just has to incorporate $\beta\eta$-equality in the calculus and then repeat the argument for $\mathcal{LK}$ with equality. Thus, logic defined this way is proof-theoretically closer to $\mathcal{LK}$ than to $\mathcal{LK}_\omega$. This becomes obvious by looking at a proof-preserving translation from functional higher-order proofs to first-order proofs that is defined in this thesis. In addition, such translation provides a new perspective on Skolemization and unification, which are both problematic in higher-order logic. Second, I study semantics of this logic: I show incompleteness wrt standard semantics and give a completeness proof wrt general semantics based on Schütte's reduction tree method. This is a simpler and more elegant way of proving this result than a more obvious specialization of a Henkin-style completeness proof for full higher-order logic.

Contents

Introduction

The subject of this thesis lies within proof theory, a subfield of mathematical logic where mathematical proof is itself studied and analyzed as a mathematical object. In order to carry out such a study, a precise formulation of the notion of proof is required. Gerhard Gentzen provided such a formulation in his seminal papers *Untersuchung über das logische Schließen I+II* [Gentzen, 1935]: sequent calculus for classical and intuitionistic first-order logic (**LK** and **LJ** respectively). Sequent calculus consists of rules that allow to introduce logical connectives, as well as rules to manipulate the structure of sequents and a proof. One of the rules introduces by Gentzen is particularly important, given its intuitive meaning and further use for establishing meta-theoretic results: *the cut rule.* This rule allows one to combine proofs and intuitively it corresponds to using lemmas or intermediate statements in the proof. Formally, it is the following rule:

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

In the most simple form it corresponds to *Modus Ponens*:

$$\frac{\vdash A \qquad A \vdash B}{\vdash B}\ cut(A)$$

The main result of Gentzen's paper, now known as *Hauptsatz* or *the cut-elimination theorem*, states that any theorem of **LK** can be proved without the cut rule. In other words, every first-order theorem has an *analytic proof*, a proof in which only all the needed information is already contained in the theorem. This is due to an immediate consequence of the cut-elimination theorem: all formulae that appear in a cut-free proof of a theorem are subformulae of the theorem. This is particularly important for automatizing proof search, since one only has to only look at subformulae of the conclusion when constructing a proof. Historically, this result was important, since it was used to establish some important consistency results. Since the proof of the cut-elimination theorem is constructive (in a sense that it consists of a procedure by which one can transform any proof into a cut-free proof) it becomes a tool for proof transformation and analysis, which is valuable in its own right. Among other consequences of the cut-elimination theorem that provide methods of extracting additional mathematical information from proofs [Leitsch, 2015] are *the mid-sequent theorem* and *the interpolation lemma.*

Now, if our aim is to analyze real mathematical proofs, Gentzen's result has limited applications due to the restricted expressive power of first-order logic. In particular, typically one would want to establish results not only about the individual objects of the domain under consideration, but also about collections of such objects and their properties. In order to achieve this, one would need to go beyond the expressive power of first-order logic.

From the proof-theoretic point of view, it is straightforward to extend the sequent calculus $\mathcal{LK}$ to more powerful logical calculi, where quantification is allowed on types different from the individual type. For instance, in second order logic one adds quantification over sets or the objects of the type $i \to o$ (functions from domain to Booleans), and in simple type theory, also known as higher-order logic, one permits quantification over all finite types. Such powerful systems allow for shorter and more natural proofs in most areas of mathematics, which makes them of great interest in the field of formalizing mathematics and proof-mining. Incidentally, most proof assistants are based on versions of higher-order logic [Benzmüller and Miller, 2014].

One can prove the cut-elimination theorem and related results for the corresponding second- and higher-order sequent calculi. However, due to the high expressive power of the corresponding logics one cannot use reductive method of Gentzen and has to resort to the semantic or semi-semantic arguments [Tait, 1966], [Prawitz, 1968], [Girard et al., 1989], [Danos et al., 1997], [Hetzl et al., 2011]. Moreover, some important consequences of cut-elimination lose their meaning: subformula property and Herbrand disjunction are not providing any information when higher-order terms (which can contain lambda abstractions of formulae (e.g., $\lambda y.\forall x P(x) \wedge Q(x)$) can be used as substitution terms. This can be partially avoided if we restrict the terms available for substitution.

Thus it seems interesting to look into some fragments of higher-order logics that have high expressive power, but nicer proof theoretic properties. Some subsystems of higher-order logics are well-studied, for instance, monadic second-order logic or predicative higher order logic (cf. [Leivant, 1994]), but some didn't get much attention. In this thesis we are interested in subsystems of higher-order logic with quantification restricted to objects of functional type, that is functionals over some domain. We will define functional higher-order logics $\mathcal{L}^f_\omega$ and $\mathcal{L}^f_\omega(\mathbf{C})$ with quantification restricted to the functional types and the corresponding calculi $\mathcal{LK}^f_\omega$ and $\mathcal{LK}^f_\omega(\mathbf{C})$.

Functional higher-order logic $\mathcal{L}^f_\omega$ has increased expressive power in comparison to first-order logic. For instance, one can characterize finite structures in $\mathcal{L}^f_\omega$ (see chapter 3). However, $\mathcal{LK}^f_\omega$ is proof-theoretically less complex than the full second-order sequent calculus. Since one doesn't quantify over predicates, the cut-elimination theorem is readily available for $\mathcal{LK}^f_\omega$ via Gentzen's reductive argument similar to the one for $\mathcal{LK}$.

However, there is still more proof-theoretic complexity than in first-order logic: in particular, the usual Skolemization procedure is not validity-preserving in the presence of function quantification and thus has to be restricted. Thus this logic provides a good case study for understanding the sources of proof-theoretic and semantic complexity of higher-order logics.

Structure of the thesis

In Chapter 1 we provide some basic definitions and notational conventions needed for the rest of the thesis. We define first- and higher-order logic, as well as two sub-logics we are considering in this thesis.

In Chapter 2 we present Gentzen's cut-elimination proof for $\mathcal{LK}$ and $\mathcal{LK}_=$ and extend it to $\mathcal{LK}^f_\omega$ and $\mathcal{LK}^f_\omega(\mathbf{C})$. We also sketch the cut-elimination proof of $\mathcal{LK}_\omega$ and the proof that it implies consistency of second-order Peano Arithmetic.

In Chapter 3 we define standard and general semantics for $\mathcal{L}_\omega$ and $\mathcal{L}^f_\omega$ and give a proof of completeness of $\mathcal{L}^f_\omega$ wrt general semantics based on Schütte's reduction trees method. This is a modification of the first-order completeness proof, whereas for completeness of $\mathcal{LK}(\mathbf{C})$ one needs a method similar to the one for higher-order logic.

In Chapter 4 we define a translation from $\mathcal{LK}^f_\omega$ proofs to $\mathcal{LK}$ proofs, which makes the relation between $\mathcal{LK}^f_\omega$ and $\mathcal{LK}$ precise. Moreover, it allows to explain the restrictions needed for Skolemization in higher-order setting.

To conclude, we summarize the results of this thesis: we syntactically and semantically characterize functional higher-order logics $\mathcal{L}^f_\omega$ and $\mathcal{L}^f_\omega(\mathbf{C})$ as extensions of first-order logic.

CHAPTER 1

Preliminaries

Here we provide the basic definitions and fix the notation needed in the next chapters. In particular, we define syntax of first-order logic $\mathcal{L}_1$, higher-order logic $\mathcal{L}_\omega$ and two functional higher-order logics $\mathcal{L}^f_\omega$, $\mathcal{L}^f_\omega(\mathbf{C})$ and their corresponding sequent calculi $\mathcal{LK}$, $\mathcal{LK}_\omega$ and $\mathcal{LK}^f_\omega$, $\mathcal{LK}^f_\omega(\mathbf{C})$. Here we omit some known definitions and results (e.g., concerning substitutions); for more details consult [Takeuti, 1987], chapters 1–3. For an overview of meta-theoretic results related to higher-order logic see [Leivant, 1994], [Benzmüller and Miller, 2014] and [Van Benthem and Doets, 2001]. The semantics of corresponding logics will be defined and discussed in Chapter 3.

1.1 First-order logic $\mathcal{L}_1$ and sequent calculus $\mathcal{LK}$

Here we present Gentzen's formulation of the sequent calculus $\mathcal{LK}$ as given by [Takeuti, 1987].

1.1.1 First-order logic $\mathcal{L}_1$

Definition 1 (First-order language L_1)**.** *Let $\mathcal{FV}$ be a countably infinite set of free variables, usually denoted by $\alpha, \beta, \gamma, \ldots$ and $\mathcal{BV}$ a countably infinite set of bound variables, usually denoted by $x, y, z, \ldots$. Then a first-order language $L_1 = (\mathcal{C}, \mathcal{F}, \mathcal{P})$ is defined by:*

- *Set of constant symbols $\mathcal{C}$; usually denoted by $a, b, c, \ldots$*
- *Set of function symbols $\mathcal{F}$; usually denoted by $f, g, h, \ldots$*
- *Set of predicate symbols $\mathcal{P}$; usually denoted by $P, Q, R, \ldots$*

Each function and predicate symbols comes with a natural number n, its arity. Then we say that it is an n-ary or n-place function or predicate.

Then given a first-order language $L_1 = (\mathcal{C}, \mathcal{F}, \mathcal{P})$ we define the set of first-order logic terms and the set of first-order logic formulae $\mathcal{L}_1$ relative to L_1. Usually we assume that the language is fixed.

Remark 1. *The distinction between free and bound variables is vital to proof transformations like cut-elimination, where whole proofs have to be instantiated.*

Terms are defined as usual with the restriction that they may not contain bound variables.

Definition 2 (Semi-term, Term)**.** *We define the set of semi-terms inductively:*

- *bound and free variables are semi-terms,*
- *constants in $\mathcal{C}$ are semi-terms,*
- *if $t_1, \dots, t_n$ are semi-terms and $f \in \mathcal{F}$ is an n-place function symbol then $f(t_1, \dots, t_n)$ is a semi-term.*

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

Example 1. *$f(\alpha, \beta)$ is a term. $f(x, \beta)$ is a semi-term.*

Definition 3 (Semi-formula, Formula)**.** *We define the set of formulae $\mathcal{L}_1$ inductively:*

- *$\top$ and $\bot$ are formulas.*
- *If $t_1, \dots, t_n$ are terms and $P \in \mathcal{P}$ is an n-place predicate symbol then $P(t_1, \dots, t_n)$ is a an (atomic) formula.*
- *If A is a formula then $\neg A$ is a formula.*
- *If A, B are formulas then $(A \to B)$, $(A \wedge B)$ and $(A \vee B)$ are formulas.*
- *If $A(\alpha)$ is a formula containing a free variable α then $\forall x.A(x), \exists x.A(x)$ are formulas if $A(x)$ is the result of substitution of all occurrences of α in $A(\alpha)$ by x.*

Semi-formulas differ from formulas in containing non-bound variables from $\mathcal{BV}$.

Example 2. *$P(f(\alpha, \beta))$ is a formula, and so is $(\forall x)P(f(x, \beta))$. $P(f(x, \beta))$ is a semi-formula.*

We write $A(t_1, \dots, t_n)$ to indicate some occurrences of $t_1, \dots, t_n$ in A and $A\{x_1 := t_1, \dots, x_n := t_n\}$ to express that $A(t_1, \dots, t_n)$ is obtained from a semi-formula $A(x_1, \dots, x_n)$ by replacing the corresponding variables $x_1, \dots, x_n$ with terms $t_1, \dots, t_n$.

1.1.2 Sequent Calculus $\mathcal{LK}$

Definition 4 (Sequent). *Let Γ and Δ be finite (possibly empty) sequences of formulas. Then the expression $S := \Gamma \vdash \Delta$ is called a* sequent. *Γ is called the* antecedent *of S and Δ the* consequent *of S.*

Definition 5 (Semantics of sequents). *Semantically a sequent*

$$S := A_1, \ldots, A_n \vdash B_1, \ldots, B_m$$

stands for

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

In particular we define $\mathfrak{M}$ to be an interpretation of S if $\mathfrak{M}$ 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 $\mathfrak{M}$ if $F(S)$ is true in $\mathfrak{M}$. S is called valid *if $F(S)$ is valid.*

Now we want to define the notion of proof in terms of sequents. In particular, we define a proof in sequent calculus as a sequence of inferences according to logical and structural rules.

Definition 6 (Inference). *Let S_1, S_2, S_3 be sequents. Then the structures below are called inferences:*

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

S_2 is said to be the lower sequent of the inference with the rule ξ, S_1 is the upper sequents of the inference.

$$\frac{S_1 \quad S_2}{S_3}\ \xi$$

Here S_1 and S_2 are the upper sequents of the inference, and S_3 the lower sequent.

Definition 7 ($\mathcal{LK}$). *In sequent calculus 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 ξL, right versions by ξR. 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., exchange) 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.*

Axioms:

$$A \vdash A$$

for any atomic formula $A \in \mathcal{L}_1$

The logical rules:

- $\wedge$-*introduction:*

$$\frac{A, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta} \wedge L1 \quad \frac{B, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta} \wedge L2 \quad \frac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B} \wedge R$$

- $\vee$-*introduction:*

$$\frac{A, \Gamma \vdash \Delta \quad B, \Gamma \vdash \Delta}{A \vee B, \Gamma \vdash \Delta} \vee L \quad \frac{\Gamma \vdash \Delta, A}{\Gamma \vdash \Delta, A \vee B} \vee R1 \quad \frac{\Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \vee B} \vee R2$$

- $\rightarrow$-*introduction:*

$$\frac{\Gamma \vdash \Delta, A \quad B, \Pi \vdash \Lambda}{A \rightarrow B, \Gamma, \Pi \vdash \Delta, \Lambda} \rightarrow L \quad \frac{A, \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \rightarrow B} \rightarrow R$$

- $\neg$-*introduction:*

$$\frac{\Gamma \vdash \Delta, A}{\neg A, \Gamma \vdash \Delta} \neg L \quad \frac{A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg A} \neg R$$

- $\forall$-*introduction:*

$$\frac{A\{x := t\}, \Gamma \vdash \Delta}{\forall x.A, \Gamma \vdash \Delta} \forall L$$

  where t is an arbitrary term *replaced with x in the lower sequent.*

$$\frac{\Gamma \vdash \Delta, A\{x := \alpha\}}{\Gamma \vdash \Delta, \forall x.A} \forall R$$

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

- $\exists$-*introduction*

$$\frac{A\{x := \alpha\}, \Gamma \vdash \Delta}{\exists x.A, \Gamma \vdash \Delta} \exists L \quad \frac{\Gamma \vdash \Delta, A\{x := t\}}{\Gamma \vdash \Delta, \exists x.A} \exists R$$

  with the same conditions for t and α as for $\forall$ rules.

The structural rules:

- Exchange

$$\frac{\Gamma \vdash \Delta_1, B, A, \Delta_2}{\Gamma \vdash \Delta_1, A, B, \Delta_2}\, ExR \qquad \frac{\Gamma_1, B, A, \Gamma_2 \vdash \Delta}{\Gamma_1, A, B, \Gamma_2 \vdash \Delta}\, ExL$$

- Weakening*:*

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, A}\, WeakR \qquad \frac{\Gamma \vdash \Delta}{A, \Gamma \vdash \Delta}\, WeakL$$

- Contraction:

$$\frac{A, A, \Gamma \vdash \Delta}{A, \Gamma \vdash \Delta}\, ContrL \qquad \frac{\Gamma \vdash \Delta, A, A}{\Gamma \vdash \Delta, A}\, ContrR$$

- The cut rule:

$$\frac{\Gamma \vdash \Delta, A \quad A, \Pi \vdash \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda}\, cut(A)$$

  The formula A is the auxiliary formula of $cut(A)$ and there is no principal formula. A is called the cut-formula.

Definition 8 ($\mathcal{LK}_e$)**.** *To $\mathcal{LK}$ add the following rules in order to obtain the calculus for first-order logic with equality:*

Equality rules:

$$\frac{\Gamma \ \vdash \ \Delta, s = t \qquad \Gamma \ \vdash \ \Delta, A(s)}{\Gamma \ \vdash \ \Delta, A(t)} = R1$$

$$\frac{\Gamma \ \vdash \ \Delta, t = s \qquad \Gamma \ \vdash \ \Delta, A(s)}{\Gamma \ \vdash \ \Delta, A(t)} = R2$$

$$\frac{\Gamma \ \vdash \ \Delta, s = t \qquad \Gamma, A(s) \ \vdash \ \Delta}{\Gamma, A(t) \ \vdash \ \Delta} = L1$$

$$\frac{\Gamma \ \vdash \ \Delta, t = s \qquad \Gamma, A(s) \ \vdash \ \Delta}{\Gamma, A(t) \ \vdash \ \Delta} = L2$$

Alternatively, one may define $\mathcal{LK}_e$ by adding to $\mathcal{LK}$ the following axioms:

Axioms of $\mathcal{LK}_e$

$$\vdash t = t$$

$$s_1 = t_1, \dots, s_n = t_n \vdash f(s_1, \dots, s_n) = f(t_1, \dots, t_n)$$

$$s_1 = t_1, \dots, s_n = t_n, P(s_1, \dots, s_n) \vdash P(t_1, \dots, t_n)$$

for all terms t, $s_1, \dots, s_n, t_1, \dots, t_n$, function symbols $f \in \mathcal{F}$ and predicate symbols $P \in \mathcal{P}$ of $\mathcal{L}_1$.

Proposition 1. *Let $A(s_1, \dots, s_n)$ be an arbitrary formula. Then:*

$$s_1 = t_1, \dots, s_n = t_n, A(s_1, \dots, s_n) \vdash A(t_1, \dots, t_n)$$

for all terms $s_1, \dots, s_n, t_1, \dots, t_n$.

Proof. By induction on the complexity of the formula $A(s_1, \dots, s_n)$.

Base: $A(s_1, \dots, s_n)$ atomic. Then we have:

$$s_1 = t_1, \dots, s_n = t_n, P(s_1, \dots, s_n) \vdash P(t_1, \dots, t_n)$$

which is an axiom.

Induction Step: Consider cases $\wedge$ and $\forall$, other are analogous.

Case $\forall$

Assume $A(s_1, \dots, s_n) = \forall x.B(s_1, \dots, s_n)$. Since $s_1, \dots, s_n$ are terms, x doesn't occur in any of s_i. Below I use $s_i = t_i$ as a shortcut for $s_1 = t_1, \dots, s_n = t_n$.

By induction hypothesis we have a proof fo $s_1 = t_1, \dots, s_n = t_n, B(\alpha, s_1, \dots, s_n) \vdash B(\alpha, t_1, \dots, t_n)$, with α not occurring in $s_1, \dots, s_n, t_1, \dots, t_n$. We get the proof of $s_i = t_i, \forall x.B(s_1, \dots, s_n) \vdash \forall x.B(t_1, \dots, t_n)$ by $\forall$L and $\forall$R application (see next page).

Case $\wedge$

By induction hypothesis we have $s_1 = t_1, \dots, s_n = t_n, B(s_1, \dots, s_n) \vdash B(t_1, \dots, t_n)$ and $s_1 = t_1, \dots, s_n = t_n, C(s_1, \dots, s_n) \vdash C(t_1, \dots, t_n)$, from which by weakenings and $\wedge$R and $\wedge$L we get the desired result.

Case $\forall$

$$\dfrac{\dfrac{\begin{array}{c}\overset{I.H.}{\vdots}\\ s_i = t_i, B(s_1, \ldots, s_n)\{x := \alpha\} \ \vdash \ B(t_1, \ldots, t_n)\{x := \alpha\}\end{array}}{s_i = t_i, \forall x.B(s_1, \ldots, s_n) \ \vdash \ B(t_1, \ldots, t_n)\{x := \alpha\}}\ \forall L}{s_i = t_i, \forall x.B(s_1, \ldots, s_n) \ \vdash \ \forall x.B(t_1, \ldots, t_n)}\ \forall R$$

Case $\wedge$

$$\dfrac{\dfrac{\dfrac{\begin{array}{c}\overset{I.H.}{\vdots}\\ s_i = t_i, B(s_1, \ldots, s_n) \ \vdash \ B(t_1, \ldots, t_n)\end{array}}{s_i = t_i, B(s_1, \ldots, s_n), C(s_1, \ldots, s_n) \ \vdash \ B(t_1, \ldots, t_n)}\ WeakL \qquad \dfrac{\begin{array}{c}\overset{I.H.}{\vdots}\\ s_i = t_i, C(s_1, \ldots, s_n) \ \vdash \ C(t_1, \ldots, t_n)\end{array}}{s_i = t_i, B(s_1, \ldots, s_n), C(s_1, \ldots, s_n) \ \vdash \ C(t_1, \ldots, t_n)}\ WeakL}{s_i = t_i, B(s_1, \ldots, s_n), C(s_1, \ldots, s_n) \ \vdash \ (B \wedge C)(t_1, \ldots, t_n)}\ \wedge R}{s_i = t_i, (B \wedge C)(s_1, \ldots, s_n) \ \vdash \ (B \wedge C)(t_1, \ldots, t_n)}\ \wedge L$$

□

Proposition 2. *Symmetry and transitivity of equality are provable in $\mathcal{LK}_e$:*

- $s = t \vdash t = s$ *(Symmetry)*
- $s_1 = s_2, s_2 = s_3 \vdash s_1 = s_3$ *(Transitivity)*

Proof. Consider the following derivations:

[Symmetry]

$$\dfrac{\vdash\ s = s \qquad \dfrac{\text{Axiom instance: } s = t, s = s\ \vdash\ t = s}{s = s, s = t\ \vdash\ t = s}\,ExL}{s = t\ \vdash\ t = s}\,cut(s = s)$$

[Transitivity]

$$\dfrac{\begin{array}{c}\text{Symmetry:}\\ s_1 = s_2\ \vdash\ s_2 = s_1\end{array} \qquad \text{Axiom instance: } s_2 = s_1, s_2 = s_3\ \vdash\ s_1 = s_3}{s_1 = s_2, s_2 = s_3\ \vdash\ s_1 = s_3}\,cut(s_2 = s_1)$$

Note that we need atomic cuts in both proofs.

□

In this thesis we are interested in more expressible logics where one is allowed to quantify not only over individuals. Although such a logic can be viewed as a restriction of higher-order logic such that quantification is allowed only over the objects of functional type, it is more convenient to define it as an extension of first-order logic, thus we first provide definitions of basic functional higher-order logic $\mathcal{L}^f_\omega$ and Church functional higher-order logic $\mathcal{L}^f_\omega(\mathbf{C})$ and then full higher-order logic.

1.2 Functional higher-order logic $\mathcal{L}^f_\omega$ and sequent calculus $\mathcal{LK}^f_\omega$

Here we define the functional higher-order logic $\mathcal{L}^f_\omega$ and the corresponding sequent calculus $\mathcal{LK}^f_\omega$. Intuitively, $\mathcal{L}^f_\omega$ is a logic where one can quantify over functions and functionals (functions of higher order). First, we need a way to distinguish the objects of different type or order. Define functional types:

Definition 9 (Functional types $\mathbb{F}$). *The set of higher-order functional types $\mathbb{F}$ is the smallest set such that:*

- $i \in \mathbb{F}$ *(individual type);*

- $\tau \to \sigma \in \mathbb{F}$ *if* $\tau, \sigma \in \mathbb{F}$.

Definition 10 (Functional higher-order language L^f_ω). *Let $\mathcal{FV}^\omega$ be a countably infinite set of typed free variables, usually denoted by $\alpha^\tau, \beta^\tau, \gamma^\tau, \dots$ and $\mathcal{BV}$ a countably infinite set of typed free variables, usually denoted by $x^\tau, y^\tau, z^\tau, \dots$, for each functional type $\tau \in \mathbb{F}$. Then a functional higher-order language $L^f_\omega = (\mathcal{C}^\omega, \mathcal{P})$ is defined by:*

- *Set of typed constant symbols $\mathcal{C}^\omega$; usually denoted by $a^{\tau_1}, b^{\tau_2}, c^{\tau_3}, \dots$ for $\tau_1, \tau_2, \tau_3, \dots \in \mathbb{F}$*
- *Set of* untyped *predicate symbols $\mathcal{P}$; usually denoted by $P, Q, R, \dots$*

Then given a first-order language $L^f_\omega = (\mathcal{C}^\omega, \mathcal{P})$ we define the set of functional higher-order logic terms and the set of functional higher-order logic formulae $\mathcal{L}^f_\omega$ relative to L^f_ω. As in the case of first-order logic, we assume that the language is fixed.

Definition 11 (Pre-term). *Pre-terms $\mathcal{T}$ of $\mathcal{L}^f_\omega$ are defined inductively:*

- *α^τ is a pre-term if $\alpha^\tau \in \mathcal{FV}_\omega$*
- *c^τ is a pre-term if $c^\tau \in \mathcal{C}_\omega$*
- *$t^{\tau_1 \to \dots \to \tau_n}(t_1^{\tau_1}, \dots, t_{n-1}^{\tau_{n-1}})$ is a pre-term of type τ_n if $t^{\tau_1 \to \dots \to \tau_n}, t_1^{\tau_1}, \dots, t_{n-1}^{\tau_{n-1}}$ are pre-terms of corresponding types*

Definition 12 (Term). *A pre-term t^τ is called a term if $\tau = i$.*

Definition 13 (Semi-term). *A semi-pre-term is defined as pre-terms, with addition of the clause:*

$$x^\tau \text{ is a pre-semi-term if } x^\tau \in \mathcal{BV}_\omega$$

Then a semi-term *is any semi-pre-term of type i.*

Definition 14 (Semi-formulae, formulae). *Formulae are defined in the usual way using the notion of the term defined above:*

- *$P(t_1, \dots, t_n)$ is a formula, if $t_1, \dots, t_n$ are terms*
- *For $\to, \neg, \wedge, \vee$ as usual*
- *If $A(\alpha^\tau)$ is a formula containing a free variable α^τ then $\forall x^\tau.A(x^\tau), \exists x^\tau.A(x^\tau)$ are formulas if $A(x^\tau)$ is the result of substitution of all occurrences of α^τ in $A(\alpha^\tau)$ by $x^\tau \in \mathcal{BV}^\omega$.*

Then semi-formulas $\mathcal{L}^f_\omega$ are defined as formulae, but they can contain bound variables x that do not occur in any $\forall x$ or $\exists x$.

Definition 15 (Order of a type). *The order of a type τ, $\mathcal{O}(\tau)$:*

- $\mathcal{O}(i) = 0$
- $\mathcal{O}(\tau \to \sigma) = max[\mathcal{O}(\tau) + 1, \mathcal{O}(\sigma)]$

Definition 16 (Order of a term). *The order of a term t^τ:*

$$\mathcal{O}(t^\tau) = max\{\mathcal{O}(\sigma) \; ; \; s^\sigma \textit{ is a subterm of } t\}$$

Example 3. *$F^{(i\to i)\to(i\to i)}$ is a second-order term of a second-order type, $Fg^{i\to i}$ is a second-order term of first-order type $i \to i$; $(Fg)x^i$ is a second-order term of the zero-order (individual) type i; $g(x)$ is a first-order term of individual type.*

Thus, terms to which a predicate is applied in a formula can be of any order but only of the individual type. Below I'll omit the type superscripts if they are redundant.

Example 4. *Given $F^{(i\to i)\to(i\to i)}$, $g^{i\to i} \in \mathcal{F}^\omega$, $P((Fg)x^i)$ is a semi-formula and $\forall x^i.P((Fg)x)$ is a formula.*

Definition 17 (Sequent Calculus $\mathcal{LK}^f_\omega$). *To obtain $\mathcal{LK}^f_\omega$ in the sequent calculus $\mathcal{LK}$ replace the quantifier rules with the following rules:*

$$\frac{\Gamma, A\{x^\tau := \boldsymbol{T}\} \vdash \Delta}{\Gamma, \forall x^\tau.A \vdash \Delta}\,\forall L$$

$$\frac{\Gamma \vdash \Delta, A\{x^\tau := \boldsymbol{T}\}}{\Gamma \vdash \Delta, \exists x^\tau.A}\,\exists R$$

where $\boldsymbol{T}$ is any pre-term of type τ replaced with x^τ in the lower sequent. Sometimes we will add subscript to the rule to emphasize that the functional quantification rule was used.

$$\frac{\Gamma \vdash \Delta, A\{x := \alpha^\tau\}}{\Gamma \vdash \Delta, \forall x^\tau.A}\,\forall_f R$$

$$\frac{\Gamma, A\{x := \alpha^\tau\} \vdash \Delta}{\Gamma, \exists x^\tau.A \vdash \Delta}\,\exists_f L$$

with $\alpha^\tau \in \mathcal{FV}^\omega$ not occurring in $\Gamma, \Delta, A(x^\tau)$. It is called an eigenvariable.

Remember that the idea was to extend $\mathcal{L}_1$ in a way that one could quantify over functions and functionals and define a corresponding calculus. With this calculus $\mathcal{LK}^f_\omega$ we added a possibility to derive formulae that express theorems involving quantification over functionals. However, the question is whether the resulting logic has the desired expressive power. In particular, we are interested in the following:

Definition 18 (Functional Comprehension)**.** *For t any pre-term of $\mathcal{L}^f_\omega$:*

$$\vdash \exists f \forall x_1 \ldots \forall x_n [f(x_1, \ldots, x_n) = t(x_1, \ldots, x_n)]$$

is a functional comprehension formula.

This formula is derivable in $\mathcal{LK}^f_\omega$, which means that we are at least quantifying over the functions defined by the terms of $\mathcal{L}^f_\omega$. These, however, are quite limited: in terms of lambda-calculus (see below), we just have applications of constants. Consider, for instance, $A = \exists f.f(x) = x$. There is no proof in $\mathcal{LK}^f_\omega$ of A. However, semantically it should be true, since the identity function should be available as one of the functions we quantify over. Thus we want to make the term language more expressible, so that we quantify over all meaningful functionals, here, all functional terms of simple type theory. All this will be made precise in the following chapters where we define semantics for these logics.

1.3 Functional higher-order $\mathcal{L}^f_\omega(\mathbf{C})$ and sequent calculus $\mathcal{LK}^f_\omega(\mathbf{C})$

Now we extend the term language to the functional fragment of simple type theory by allowing to form terms by abstraction and allowing beta-reduction in the derivation.

Definition 19 (Lambda terms)**.** *The set of simply-typed lambda terms, denoted Λ, is built up from a countably infinite set of simply typed constants $\mathcal{C}^\omega$ and variables $\mathcal{V}^\omega$ by the following rules:*

1. *If $x^\tau \in \mathcal{C}^\omega$ or $x^\tau \in \mathcal{V}^\omega$, then $x^\tau \in \Lambda$.*
2. *If $M^{\sigma\to\tau}, N^\sigma \in \Lambda$, then $(MN)^\tau \in \Lambda$.*
3. *If $x^\sigma \in \mathcal{V}^\omega$ and $M^\tau \in \Lambda$, then $(\lambda x^\sigma M)^{\sigma\to\tau} \in \Lambda$.*

The notions of free and bound variables are defined as usual (with λ being the only binder).

Definition 20 (β-reduction)**.** *The rule of β-reduction is $(\lambda x.M)N \to_\beta M[x := N]$ and $\lambda x.M$ is called a β-redex of $(\lambda x.M)N$. A term which does not contain a β-redex is said to be in β-normal form.*

To define the extended language $L^f_\omega(\mathbf{C})$ we take all simply typed lambda terms over L^f_ω; that is, all simply typed terms with types restricted to $\mathbb{F}$.

Definition 21 (Pre-term)**.** *Let $\mathcal{T}$ be pre-terms of $\mathcal{L}^f_\omega$. Then Church pre-terms $\mathcal{T}(\boldsymbol{C})$ of a language L^f_ω are define in the following way:*

- *if $t \in \mathcal{T}$ then $t \in \mathcal{T}(\boldsymbol{C})$*
- *if $x^\tau \in \mathcal{BV}_\omega$ and $t^\sigma \in \mathcal{T}(\boldsymbol{C})$, then $(\lambda x.t)^{\tau\to\sigma} \in \mathcal{T}(\boldsymbol{C})$*

Terms and formulae defined in the same way as for $\mathcal{L}^f_\omega$. Note that now terms can contain bound variables, however, only λ-bound.

To make use of the λ-terms we need the corresponding rules. Consider the functional comprehension again:

Definition 22 (Functional Comprehension)**.** *For t any pre-term of $\mathcal{L}^f_\omega(\boldsymbol{C})$:*

$$\vdash \exists f \forall x_1 \ldots \forall x_n [f(x_1, \ldots, x_n) = t(x_1, \ldots, x_n)]$$

is a functional comprehension formula.

In this form it will not help us deriving $\vdash \exists f.f(x) = x$, since for no term $t(x) = x$. If we incorporate β-equality, then it is a direct consequence of the functional comprehension:

$$\frac{\vdash \vdash \exists f \forall x[f(x) = (\lambda z.z)(x))]}{\vdash\ \exists f \forall x.f(x) = x}\ \exists_f\text{R, } \beta\text{-reduction}$$

Definition 23 ($\mathcal{LK}^f_\omega(\mathbf{C})$ via β-reduction)**.** *To obtain sequent calculus $\mathcal{LK}^f_\omega(\boldsymbol{C})$ to the rules of $\mathcal{LK}^f_\omega$ one could add the following rules:*

$$\frac{\Gamma\ \vdash\ \Delta, A(\boldsymbol{T})}{\Gamma\ \vdash\ \Delta, A(\boldsymbol{T'})}\ \beta R$$

$$\frac{\Gamma, A(\boldsymbol{T})\ \vdash\ \Delta}{\Gamma, A(\boldsymbol{T'})\ \vdash\ \Delta}\ \beta L$$

if $\boldsymbol{T} \rightsquigarrow^\beta \boldsymbol{T'}$ (if $\boldsymbol{T}$ in some steps β-reduces to $\boldsymbol{T'}$).

Example 5.

$$\dfrac{\dfrac{\vdash\ (\lambda z.z)x = (\lambda z.z)x}{\vdash\ (\lambda z.z)x = x}\ \beta R}{\vdash\ \exists f.f(x) = x}\ \exists_f R$$

Alternatively one could incorporate β-reduction via addition of equality axioms: whenever $\mathbf{T} \rightsquigarrow_\beta \mathbf{T}'$, add $\mathbf{T} = \mathbf{T}'$ as an axiom to $\mathcal{LK}_e$. Note that we have to treat the equality predicate differently now: it can apply to pre-terms as well as terms. Formally we define this by adding to the language a symbol $=_\tau$ for each $\tau \in \mathbb{F}$. Then we redefine the notion of formula:

Definition 24 (Formulae of $\mathcal{L}^f_\omega(\mathbf{C})$)**.** *Intuitively, we allow $s =_\tau t$ as atoms:*

- *for every predicate symbol P other than $=_\tau$ for each $\tau \in \mathbb{F}$, $P(t_1, \ldots, t_n)$ is a formula if $t_1, \ldots, t_n$ are terms (that is, pre-terms of type i)*
- *For $=_\tau$ for each $\tau \in \mathbb{F}$, $s =_\tau t$ is a formula if s and t are pre-terms of type τ.*
- *Then the definition is as before.*

Then we add the equality axioms to $\mathcal{LK}^f_\omega$ for each $=_\tau$ and get $\mathcal{LK}^f_\omega(\mathbf{C})$. We will use the following calculus in this thesis.

Definition 25 (Sequent Calculus $\mathcal{LK}^f_\omega(\mathbf{C})$)**.** *Take the rules of $\mathcal{LK}^f_\omega$ and:*

- *$\vdash t =_\tau t$, for all pre-terms t of $\mathcal{L}^f_\omega$ of type τ;*
- *$\vdash s =_\tau t$, for all pre-terms s, t of $\mathcal{L}^f_\omega$ of type τ such that $s \rightsquigarrow_\beta t$;*
- *$s_1 =_{\tau_1} t_1, \ldots, s_n =_{\tau_n} t_n \vdash f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n)$, for all pre-terms $s_1, \ldots, s_n, t_1, \ldots, t_n$ and functional symbols $f \in \mathcal{C}^\omega$ of corresponding types;*
- *$s_1 =_{\tau_1} t_1, \ldots, s_n =_{\tau_n} t_n, P(s_1, \ldots, s_n) \vdash P(t_1, \ldots, t_n)$, for all pre-terms $s_1, \ldots, s_n, t_1, \ldots, t_n$ and predicate symbols $P \in \mathcal{P}$ of corresponding types.*

Proposition 3. *Let $A(s_1, \ldots, s_n)$ be an arbitrary formula. Then:*

$$s_1 =_{\tau_1} t_1, \ldots, s_n =_{\tau_n} t_n, A(s_1, \ldots, s_n) \vdash A(t_1, \ldots, t_n)$$

for all pre-terms $s_1, \ldots, s_n, t_1, \ldots, t_n$ of corresponding types.

Proof. As for simple equality $=$. □

Proposition 4. *Symmetry and transitivity of equality are provable from the above axioms:*

- $s =_\tau t \vdash t =_\tau s$
- $s_1 =_\tau s_2, s_2 =_\tau s_3 \vdash s_1 =_\tau s_3$

Proof. As for simple equality $=$. □

Example 6.

$$\frac{\vdash\ (\lambda z.z)x =_i x\ (\beta\text{-axiom})}{\vdash\ \exists f.f(x) =_i x}\ \exists_f R$$

From this point of view, extending the term language to other type systems (system T, for instance) means adding additional equation axioms.

1.4 Higher-order logic $\mathcal{L}_\omega$ and sequent calculus $\mathcal{LK}_\omega$

Here we define the set of higher-order logic formulae $\mathcal{L}_\omega$.

Definition 26 (Higher-order types $\mathbb{T}$)**.** *Define the set of higher-order types $\mathbb{T}$:*

1. $i, o \in \mathbb{T}$, *(individual and boolean types)*
2. $\tau \to \sigma \in \mathbb{T}$, *if* $\tau, \sigma \in \mathbb{T}$.

The difference with functional type $\mathbb{F}$ is that we have two base types: individual and boolean. This gives us a possibility to quantify over objects other than functionals. For instance, sets (objects of type $i \to o$) or formulae (objects of type o).

Then the terms of $\mathcal{L}_\omega$ are all simply typed lambda-terms. The formulae of $\mathcal{L}_\omega$ are all simply typed lambda-terms of type o given some simply-typed language $L_\omega = (\mathcal{C}^\omega, \to^{o\to o\to o}, \forall_\tau^{(\tau\to o)\to o})$, where $\mathcal{C}^\omega$ are any typed constants and $\to$ and $\forall$ are distinguished logical constants. If in the above one replaces $\forall_\tau$ with $\forall_{i^n}$ we get the second-order logic $\mathcal{L}_2$. We also distinguish between free variables $\alpha^\tau, \beta^\sigma, \ldots$ and bound variables $x^\tau, y^\sigma, \ldots$. Then $\forall x^\tau.A$ is a shortcut for $\forall_\tau \lambda x^\tau.A$.

As a sub-system of higher-order logic $\mathcal{L}_\omega$, we could have defined $\mathcal{L}_\omega^f$ as all simply typed terms of type o, with all sub-pre-terms other than logical connectives restricted to types in $\mathbb{F} \cup \{i \to o\}$.

Definition 27 ($\mathcal{LK}_\omega$)**.** *$\mathcal{LK}_\omega$ is $\mathcal{LK}$ with the following quantifier rules:*

- $\forall$*-introduction:*

$$\frac{A(\boldsymbol{T}), \Gamma \vdash \Delta}{\forall x^\tau.A(x^\tau), \Gamma \vdash \Delta}\ \forall L$$

where $\boldsymbol{T}$ *is an arbitrary* simply-typed term *of type* τ*.*

$$\frac{\Gamma \vdash \Delta, A(\alpha^\tau)}{\Gamma \vdash \Delta, \forall x^\tau.A(x^\tau)}\ \forall R$$

where α^τ *is a free variable which may not occur in* $\Gamma, \Delta, A(x^\tau)$*.* α^τ *is called the* eigenvariable *of this inference.*

- $\exists$-*introduction*

$$\frac{A(\alpha^\tau), \Gamma \vdash \Delta}{\exists x^\tau.A(x^\tau), \Gamma \vdash \Delta}\ \exists L \qquad \frac{\Gamma \vdash \Delta, A(\boldsymbol{T})}{\Gamma \vdash \Delta, \exists x^\tau.A(x^\tau)}\ \exists R$$

Example 7.

$$\frac{\Gamma_2, P(\beta) \wedge Q(\beta)\ \vdash\ \Delta_2}{\Gamma_2, \forall x^{i \to o}.x(\beta)\ \vdash\ \Delta_2}\ \forall\text{-}L$$

It can be rewritten in the following way:

$$\frac{\Gamma_2, x\{x^{i \to o} := \lambda y^i.P(y) \wedge Q(y)\}(\beta)\ \vdash\ \Delta_2}{\Gamma_2, \forall x^{i \to o}.x(\beta)\ \vdash\ \Delta_2}\ \forall\text{-}L$$

Thus we identify $\beta\eta$-equal terms; sometimes we will explicitly mention this fact. We will perform the β-abstraction and expansion implicitly, for simplicity. That is, we identify $P(\beta) \wedge Q(\beta)$ and $(\lambda y.P(y) \wedge Q(y))(\beta)$, or derive the latter from the former by "β-rule".

Alternatively, one could obtain the sequent calculus $\mathcal{LK}_\omega$ by adding to $\mathcal{LK}$ the comprehension axiom schema:

Definition 28 (Comprehension Schema)**.**

$$\vdash \forall \overrightarrow{z} \exists X^\tau \forall \overrightarrow{y}[X^\tau(\overrightarrow{y}) \leftrightarrow A(\overrightarrow{y}, \overrightarrow{z})]$$

for A *any formula in* $\mathcal{L}_\omega$ *with all free variables replaced by* $\overrightarrow{y}$, $\overrightarrow{z}$*, and* τ *corresponding to the term* $[\lambda \overrightarrow{u}.A(\overrightarrow{u}, \overrightarrow{z})]^\tau(\overrightarrow{y}) =_\beta A(\overrightarrow{y}, \overrightarrow{z})$

Note that the schema is derivable in $\mathcal{LK}_\omega$ in the following way:

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\vdash \forall \overrightarrow{y}[A(\overrightarrow{y}, \overrightarrow{\alpha}) \leftrightarrow A(\overrightarrow{y}, \overrightarrow{\alpha})]}}{\vdash \forall \overrightarrow{y}[(\lambda \overrightarrow{z}.A(\overrightarrow{z}, \overrightarrow{\alpha}))(\overrightarrow{y}) \leftrightarrow A(\overrightarrow{y}, \overrightarrow{\alpha})]}\ \beta\eta\text{-expansion}}{\vdash \exists X^\tau \forall \overrightarrow{y}[X^\tau(\overrightarrow{y}) \leftrightarrow A(\overrightarrow{y}, \overrightarrow{\alpha})]}\ \exists\text{R}}{\dfrac{\vdots}{\vdash \forall \overrightarrow{z} \exists X^\tau \forall \overrightarrow{y}[X^\tau(\overrightarrow{y}) \leftrightarrow A(\overrightarrow{y}, \overrightarrow{z})]}\ \forall\text{R}}\ \forall\text{R}$$

CHAPTER 2

Cut-Elimination in First-Order and Higher-Order Logics

2.1 The cut-elimination theorem for $\mathcal{LK}$ and $\mathcal{LK}_e$

Here I describe the reductive proof of cut-elimination for $\mathcal{LK}$ and $\mathcal{LK}_e$. This is a form of the original Gentzen's proof. Later we will see that for $\mathcal{LK}_\omega^f$ we can use a very similar argument, although in general it is not possible to give a reductive cut-elimination proof for higher-order sequent calculus. For the detailed description of the proof see [Takeuti, 1987], Chapter 1.

Theorem 1. *Every $\mathcal{LK}$-proof of S can be transformed into a cut-free proof of S.*

First of all, introduce a rule equivalent to cut, called mix :

Definition 29 (Mix rule)**.** *Assume that Δ_2, Λ_1 contain at least one occurrence of A:*

$$\frac{\Delta_1 \vdash \Lambda_1 \qquad \Delta_2 \vdash \Lambda_2}{\Delta_1, \Delta_2^* \vdash \Lambda_1^*, \Lambda_2}\, mix(A)$$

where Δ_2^*, Λ_1^* are obtained from Δ_2, Λ_1 by removing all occurrences of A. A is called a mix formula.

It is easy to see that in $\mathcal{LK}$ this rule is equivalent to cut. From now on by $\mathcal{LK}$ we mean $\mathcal{LK}$ with the mix rule instead of the cut rule.

Proof of Theorem 2. Below we show that if a proof of S contains just one mix as the last inference, then this mix can be eliminated. Then, by induction, we can remove all of mixes in any proof and therefore obtain a cut-free proof of S.

□

Theorem 2 (Cut-elimination for $\mathcal{LK}$)**.** *If an $\mathcal{LK}$-proof of S contains one mix as the last inference, then there is an $\mathcal{LK}$-proof of S without mix.*

The proof is done by double induction on grade and rank of the proof determined by the unique mix we are trying to remove. Let P be a proof of S that contains only one mix as the last inference. Define:

Definition 30 (Grade)**.** *Grade of the proof:*

- *Grade of the formula A is the number of occurrences of logical connectives in A.*
- *Grade of a mix is the grade of the corresponding mix formula.*
- *Grade of the proof $Grade(P)$ is the grade of the mix.*

By assumption the last inference of P has the following form:

$$\dfrac{\overbrace{\Delta_1 \ \vdash \ \Lambda_1}^{S_1} \qquad \overbrace{\Delta_2 \ \vdash \ \Lambda_2}^{S_2}}{\underbrace{\Delta_1, \Delta_2^* \ \vdash \ \Lambda_1^*, \Lambda_2}_{S}}\ mix(A)$$

A branch of P containing S_1 as end-sequent is called a left branch of P and a branch of P containing S_2 as end-sequent is called a right branch of P. Then define rank of P.

Definition 31 (Rank)**.** *The rank of the proof is defined as follows:*

- *The rank of a left (right) branch is the number of consecutive occurrences of the mix formula in the branch, starting from S_1 (S_2);*
- *$Rank_l(P)$ is the maximal rank among left branches;*
- *$Rank_r(P)$ is the maximal rank among right branches;*

$$Rank(P) = Rank_l(P) + Rank_r(P)$$

Note that $Rank(P)$ is always ≥ 2.

Proof. By induction on grade and rank of the proof P, examining the possible structure of S_1, S_2 and S. Apply the induction hypothesis to P' when $Grade(P') < Grade(P)$ or $Grade(P') = Grade(P)$ and $Rank(P') < Rank(P)$.

Case 1: $Rank(P) = 2$.

Consider all cases corresponding to the forms of proofs of S_1 and S_2.

S_1 or S_2 initial sequent: obtain S by some exchanges and contraction from S_2 or S_1.

S_1 or S_2 structural inference: obtain S by weakenings and exchanges from S_1's predecessor.

Below I will only show one propositional case in detail, other cases are analogous.

Case $\to$

$$\dfrac{\to\text{-R}\,\dfrac{\begin{matrix}Q\\ \vdots\\ \Gamma, A \vdash \Delta, B\end{matrix}}{\Gamma \vdash \Delta, A \to B} \qquad \dfrac{\begin{matrix}P_1\\ \vdots\\ \Gamma_1 \vdash \Delta_1, A\end{matrix} \qquad \begin{matrix}P_2\\ \vdots\\ \Gamma_2, B \vdash \Delta_2\end{matrix}}{A \to B, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\,\to\text{-L}}{\Gamma, \Gamma_1^*, \Gamma_2^* \vdash \Delta^*, \Delta_1, \Delta_2}\,\text{mix } (A \to B)$$

Then we can "move the cut upwards" and by applying the induction hypothesis, first to the proof of the mix (B) lower sequent, then to the mix (A) lower sequent we obtain a mix-free proof of S:

$$\dfrac{\dfrac{\dfrac{\begin{matrix}P_1\\ \vdots\\ \Gamma_1 \vdash \Delta_1, A\end{matrix} \qquad \dfrac{\begin{matrix}Q\\ \vdots\\ \Gamma, A \vdash \Delta, B\end{matrix} \qquad \begin{matrix}P_2\\ \vdots\\ \Gamma_2, B \vdash \Delta_2\end{matrix}}{A, \Gamma, \Gamma_2^{\#} \vdash \Delta^{\#}, \Delta_2}\,\text{mix } (B)}{\Gamma^\circ, \Gamma_1, (\Gamma_2^{\#})^\circ \vdash \Delta^{\#}, \Delta_1^\circ, \Delta_2}\,\text{mix } (A)}{\text{Some weakenings}}}{\Gamma, \Gamma_1^*, \Gamma_2^* \vdash \Delta^*, \Delta_1, \Delta_2}$$

Case $\forall$

For treating the quantifier cases, we will need the following lemma:

Lemma 1. *If $S(\alpha)$ is $\mathcal{LK}$-provable by the proof $P(\alpha)$, then $S(t)$ is $\mathcal{LK}$-provable by $P(t)$, where α is a free variable different from all eigenvariables of the proof and doesn't occur in t, and t is an arbitrary term.*

Proof. By induction on number of inferences. See [Takeuti, 1987], pp. 16–17. □

$$\dfrac{\forall\text{-R}\,\dfrac{\begin{matrix}P'(\alpha)\\ \vdots\\ \Gamma_1 \vdash \Delta_1, A(\alpha)\end{matrix}}{\Gamma_1 \vdash \Delta_1, \forall x A(x)} \qquad \dfrac{\begin{matrix}Q\\ \vdots\\ \Gamma_2, A(\mathbf{T}) \vdash \Delta_2\end{matrix}}{\Gamma_2, \forall x A(x) \vdash \Delta_2}\,\forall\text{-L}}{\Gamma_1, \Gamma_2^* \vdash \Delta_1^*, \Delta_2}\,\text{mix } (\forall x A(x))$$

By lemma above, whenever we have a proof of $\Gamma_1 \vdash \Delta_1, A(\alpha)$, we also have a proof $\Gamma_1 \vdash \Delta_1, A(\mathbf{T})$. Then we can "move the cut upwards" in the following way:

$$\dfrac{\dfrac{\dfrac{\overset{P'(\mathbf{T})}{\vdots}}{\Gamma_1 \vdash \Delta_1, A(\mathbf{T})} \quad \dfrac{\overset{Q}{\vdots}}{\Gamma_2, A(\mathbf{T}) \vdash \Delta_2}}{\Gamma_1, \Gamma_2^{\circ} \vdash \Delta_1^{\circ}, \Delta_2}\ \text{mix } (A(\mathbf{T}))}{\dfrac{\text{Some weakenings}}{\Gamma_1, \Gamma_2^{*} \vdash \Delta_1^{*}, \Delta_2}}$$

Then, by induction hypothesis, since the grade of the mix formula decreased, we get mix-free proofs of $\Gamma_1 \vdash \Delta_1, A(\mathbf{T})$ and $\Gamma_2, A(\mathbf{T}) \vdash \Delta_2$.

Case 2: Rank$(P) > 2$.

Here proceed by induction on *Rank*(P). *Rank*$(P) > 2$ means that either left or right rank is > 1. Assume that *Rank*$_r(P) > 1$. The other case is treated analogously. Remember that we have the inference of this form as the last inference of the proof P, which contains exactly one mix:

$$\dfrac{\overbrace{\Delta_1 \vdash \Lambda_1}^{S_1} \qquad \overbrace{\Delta_2 \vdash \Lambda_2}^{S_2}}{\underbrace{\Delta_1, \Delta_2^{*} \vdash \Lambda_1^{*}, \Lambda_2}_{S}}\ mix(A)$$

Now, as previously, proceed by analyzing the structure of S_1 and S_2:

Δ_1 in S_1 contains A: Assume Δ_1 in S_1 contains A. Construct the proof as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\Delta_2 \vdash \Lambda_2}}{\text{Exchanges and contractions}}}{\dfrac{A, \Delta_2^{*} \vdash \Lambda_2}{\text{Weakenings and exchanges}}}}{\Delta_1, \Delta_2^{*} \vdash \Lambda_1^{*}, \Lambda_2}$$

***A* not the principal formula of S_2:** Now assume that S_2 is the lower sequent of some unary inference ξ, however, A is not a principal formula of ξ. That is, the inference looks like this:

$$\dfrac{\begin{array}{c}P_1\\ \vdots\\ \Delta_1 \vdash \Lambda_1\end{array} \qquad \dfrac{\begin{array}{c}P_2\\ \vdots\\ \Delta \vdash \Lambda\end{array}}{\Delta_2 \vdash \Lambda_2}\,\xi}{\Delta_1, \Delta_2^* \vdash \Lambda_1^*, \Lambda_2}\,mix(A)$$

P_1 and P_2 do not contain mixes by assumption, and Δ contains at least one occurrence of A, since the $Rank_r(P) > 1$. Consider the following proof:

$$\dfrac{\begin{array}{c}P_1\\ \vdots\\ \Delta_1 \vdash \Lambda_1\end{array} \qquad \begin{array}{c}P_2\\ \vdots\\ \Delta \vdash \Lambda\end{array}}{\Delta_1, \Delta^* \vdash \Lambda_1^*, \Lambda}\,mix(A)$$

Rank of this proof is $Rank(P) - 1$, thus we can apply the induction hypothesis and eliminate the mix. Then we can construct the following mix-free proof of S:

$$\begin{array}{c}\vdots\\ \dfrac{\dfrac{\dfrac{\Delta_1, \Delta^* \vdash \Lambda_1^*, \Lambda}{\text{Exchanges}}}{\Delta_2^*, \Delta_1 \vdash \Lambda_1^*, \Lambda_2}\,\xi}{\dfrac{\text{Exchanges}}{\Delta_1, \Delta_2^* \vdash \Lambda_1^*, \Lambda_2}}\end{array}$$

In case that S_2 is the lower sequent of some binary inference ξ and A is not a principal formula of ξ, we make a similar argument:

$$\dfrac{\begin{array}{c}P_1\\ \vdots\\ \Delta_1 \vdash \Lambda_1\end{array} \qquad \dfrac{\begin{array}{c}Q_1\\ \vdots\\ \Psi_1 \vdash \Pi_1\end{array} \qquad \begin{array}{c}Q_2\\ \vdots\\ \Psi_2 \vdash \Pi_2\end{array}}{\Delta_2 \vdash \Lambda_2}\,\xi}{\Delta_1, \Delta_2^* \vdash \Lambda_1^*, \Lambda_2}\,mix(A)$$

P_1, Q_1 and Q_2 do not contain mixes by assumption, and Ψ_1 or Ψ_2 contains at least one occurrence of A, since the $Rank_r(P) > 1$. Consider the following proofs:

$$[P_l : \text{if } \Psi_1 \text{ contains } A] \quad \dfrac{\overset{P_1}{\vdots} \atop \Delta_1 \vdash \Lambda_1 \qquad \overset{Q_1}{\vdots} \atop \Psi_1 \vdash \Pi_1}{\Delta_1, \Psi_1^* \vdash \Lambda_1^*, \Pi_1}\, mix(A)$$

$$[P_r : \text{if } \Psi_2 \text{ contains } A] \quad \dfrac{\overset{P_1}{\vdots} \atop \Delta_1 \vdash \Lambda_1 \qquad \overset{Q_2}{\vdots} \atop \Psi_2 \vdash \Pi_2}{\Delta_1, \Psi_2^* \vdash \Lambda_1^*, \Pi_2}\, mix(A)$$

$$[P_i' : \text{if } \Psi_i \text{ doesn't contain } A]$$

$$\dfrac{\dfrac{\overset{Q_i}{\vdots} \atop \Psi_i \vdash \Pi_i}{\text{Exchanges and weakenings}}}{\Delta_1, \Psi_i^* \vdash \Lambda_1^*, \Pi_i}$$

In the last case $\Psi_i^* = \Psi_i$. Rank of all above proofs is $Rank(P) - 1$, thus we can apply the induction hypothesis and eliminate the mix. Then we can construct the following mix-free proof of S:

$$\dfrac{\dfrac{\dfrac{\overset{P_l \text{ or } P_1'}{\vdots} \atop \Delta_1, \Psi_1^* \vdash \Lambda_1^*, \Pi_1}{\text{Some exchanges}} \qquad \dfrac{\overset{P_r \text{ or } P_2'}{\vdots} \atop \Delta_1, \Psi_2^* \vdash \Lambda_1^*, \Pi_2}{\text{Some exchanges}}}{\Delta_1, \Delta_2^* \vdash \Lambda_2, \Lambda_1^*}\,\xi}{\dfrac{\text{Some exchanges}}{\Delta_1, \Delta_2^* \vdash \Lambda_1^*, \Lambda_2}}$$

***A* is the principal formula of S_2:** Now consider the cases where the mix formula is the principal formula of the last inference in S_2.

Case $\rightarrow$

$$\dfrac{\overset{Q}{\vdots} \atop \Gamma \vdash \Delta \qquad \dfrac{\overset{P_1}{\vdots} \atop \Gamma_1 \vdash \Delta_1, A \qquad \overset{P_2}{\vdots} \atop \Gamma_2, B \vdash \Delta_2}{A \rightarrow B, \Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\,\rightarrow\text{-L}}{\Gamma, \Gamma_1^{\#}, \Gamma_2^{\#} \vdash \Delta^{\#}, \Delta_1, \Delta_2}\,\text{mix } (A \rightarrow B)$$

Again, we want to move the cuts upwards in order to decrease the rank measure and then apply the induction hypothesis:

$$[Q_1]$$

$$\dfrac{\begin{array}{c}Q\\ \vdots\\ \Gamma \vdash \Delta\end{array} \qquad \begin{array}{c}P_1\\ \vdots\\ \Gamma_1 \vdash \Delta_1, A\end{array}}{\Gamma, \Gamma_1^{\#} \vdash \Delta^{\#}, \Delta_1, A}\ \text{mix } (A \to B)$$

$$[Q_2]$$

$$\dfrac{\begin{array}{c}Q\\ \vdots\\ \Gamma \vdash \Delta\end{array} \qquad \begin{array}{c}P_2\\ \vdots\\ \Gamma_2, B \vdash \Delta_2\end{array}}{\Gamma, B, \Gamma_2^{\#} \vdash \Delta^{\#}, \Delta_2}\ \text{mix } (A \to B)$$

In both Q_1 and Q_2, $Rank_r(P_i)$ decreased, thus by induction hypothesis, we get a mix free proofs Q_1' and Q_2' of their end-sequents. From Q_1 and Q_2 we construct a proof P' in the following way:

$$[P']$$

$$\text{mix } (A \to B)\ \dfrac{\dfrac{\begin{array}{c}Q\\ \vdots\\ \Gamma \vdash \Delta\end{array} \qquad \dfrac{\begin{array}{c}Q_1'\\ \vdots\\ \Gamma, \Gamma_1^{\#} \vdash \Delta^{\#}, \Delta_1, A\end{array} \qquad \begin{array}{c}Q_2'\\ \vdots\\ \Gamma, B, \Gamma_2^{\#} \vdash \Delta^{\#}, \Delta_2\end{array}}{A \to B, \Gamma, \Gamma_1^{\#}, \Gamma_2^{\#} \vdash \Delta^{\#}, \Delta_1, \Delta_2}\ \to\text{L}}{\text{Some weakenings}}}{\Gamma, \Gamma_1^{\#}, \Gamma_2^{\#} \vdash \Delta^{\#}, \Delta_1, \Delta_2}$$

Observe that $Rank_r(P') = 1$, thus we can apply the induction hypothesis, since by assumption, the right rank of the original proof was more than 1.

Case $\forall$ A is the principal formula in S_2 introduced by $\forall$L. The last inference of the proof P looks like this:

$$\dfrac{\begin{array}{c}Q\\ \vdots\\ \Gamma \vdash \Delta\end{array} \qquad \dfrac{\begin{array}{c}Q_1\\ \vdots\\ \Gamma_2, A(\mathbf{T}) \vdash \Delta_2\end{array}}{\Gamma_2, \forall x A(x) \vdash \Delta_2}\ \forall\text{-L}}{\Gamma_1, \Gamma_2^{*} \vdash \Delta_1^{*}, \Delta_2}\ \text{mix } (\forall x A(x))$$

Move the cut upwards in the following way:

$$\begin{array}{c}[Q']\\ \dfrac{\begin{array}{c}Q\\ \vdots\\ \Gamma \vdash \Delta\end{array} \qquad \begin{array}{c}Q_1\\ \vdots\\ \Gamma_2, A(\mathbf{T}) \vdash \Delta_2\end{array}}{\Gamma, \Gamma_2^*, A(\mathbf{T}) \vdash \Delta^*, \Delta_2}\ \text{mix } (\forall x A(x))\end{array}$$

Then, by induction hypothesis, there is a mix-free proof of $\Gamma, \Gamma_2^*, A(\mathbf{T})\Delta^*, \Delta_2$. Now construct a proof P' of S with only one mix as the last inference:

$$\dfrac{\begin{array}{c}Q\\ \vdots\\ \Gamma \vdash \Delta\end{array} \qquad \dfrac{\begin{array}{c}Q'\\ \vdots\\ \Gamma, \Gamma_2^*, A(\mathbf{T}) \vdash \Delta^*, \Delta_2\end{array}}{\Gamma, \Gamma_2^*, \forall x A(x) \vdash \Delta^*, \Delta_2}\ \forall\text{L}}{\Gamma, \Gamma_2^* \vdash \Delta^*, \Delta_2}\ \text{mix } (\forall x A(x))$$

Since now $\mathit{Rank}_r(P')$ is smaller than the original rank, we can apply the induction hypothesis and get a mix-free proof of S.

Case $\exists$ Same as the $\forall$ case. The last inference of the proof P looks like this:

$$\dfrac{\begin{array}{c}Q\\ \vdots\\ \Gamma \vdash \Delta\end{array} \qquad \dfrac{\begin{array}{c}Q_1(\alpha)\\ \vdots\\ \Gamma_2, A(\alpha) \vdash \Delta_2\end{array}}{\Gamma_2, \exists x A(x) \vdash \Delta_2}\ \exists\text{-L}}{\Gamma_1, \Gamma_2^* \vdash \Delta_1^*, \Delta_2}\ \text{mix } (\exists x A(x))$$

By above lemma we have a proof of $\Gamma_2, A(\beta) \vdash \Delta_2$, since by eigenvariable condition α is not in Γ_2 nor Δ_2. Then move the mix upwards:

$$\dfrac{\begin{array}{c}Q\\ \vdots\\ \Gamma \vdash \Delta\end{array} \qquad \begin{array}{c}Q_1(\beta)\\ \vdots\\ \Gamma_2, A(\beta) \vdash \Delta_2\end{array}}{\Gamma, \Gamma_2^*, A(\beta) \vdash \Delta^*, \Delta_2}\ \text{mix } (\exists x A(x))$$

Then, by induction hypothesis, there is a mix-free proof Q' of $\Gamma, \Gamma_2^*, A(\beta) \vdash \Delta^*, \Delta_2$. Now construct P':

$$\dfrac{\begin{array}{c} Q \\ \vdots \\ \Gamma \vdash \Delta \end{array} \qquad \dfrac{\begin{array}{c} [P'] \\ Q' \\ \vdots \\ \Gamma, \Gamma_2^*, A(\beta) \vdash \Delta^*, \Delta_2 \end{array}}{\Gamma, \Gamma_2^*, \exists x A(x) \vdash \Delta^*, \Delta_2}\,\exists\text{-L}}{\Gamma, \Gamma_2^* \vdash \Delta^*, \Delta_2}\,\text{mix } (\exists x A(x))$$

Since now $Rank_r(P')$ is smaller than the original right rank, we can apply the induction hypothesis and get a mix-free proof of S. □

Corollary 1 (Subformula property)**.** *All formulae occurring in a cut-free proof of S are subformulae of S.*

Proof. By induction on the number of inferences of the cut-free proof. □

2.1.1 Cut-elimination for $\mathcal{LK}_e$

In general if one adds axioms to $\mathcal{LK}$, the cut-elimination theorem doesn't hold, as a simple example below demonstrates. Assume that we have $\mathcal{LK}$ with two axioms $\vdash A$ and $\vdash A \to B$, with A and B distinct atomic formulae. Consider the following proof:

Example 8.

$$\dfrac{\vdash A \qquad \dfrac{\vdash A \to B \qquad \dfrac{B \vdash B \qquad A \vdash A}{A, A \to B \vdash B}\,{\to}L}{A \vdash B}\,\textit{cut } (A \to B)}{\vdash B}\,\textit{cut } (A)$$

If there were a cut-free proof of $\vdash B$, then by the subformula property it would have to consist only of formulae B and the proper axioms could not be used, thus it should have been a pure $\mathcal{LK}$ proof. Clearly, $\vdash B$ is not provable for an atomic formula.

Theorem 3. *Let $\boldsymbol{S}$ be a set of sequents closed under substitution. Add $\boldsymbol{S}$ to $\mathcal{LK}$. Then if S is provable in this extended calculus, then there is a proof where all cuts operate on sequents in $\boldsymbol{S}$.*

Proof. See [Girard, 1991], p.123: repeat the above cut-elimination proof, just "forget the cuts" that operate on sequents in **S** when defining the grade and rank. Below we provide the proof for **S** the equality axioms.

□

$\mathcal{LK}_e$ admits restricted cut-elimination in the sense described above: if S is $\mathcal{LK}_e$ provable, we can always find a proof of S where all cuts operate only on equational atoms.

Definition 32. *If the cut formula in an $\mathcal{LK}_e$ proof is of the form $s = t$ call it inessential.*

Theorem 4. *If a sequent S is $\mathcal{LK}_e$-provable, then there is an $\mathcal{LK}_e$ proof of S where all cuts are inessential.*

Proof. Apply the $\mathcal{LK}$ reduction to remove all the essential mixes. We only consider the new cases here:

Case $Rank(P) = 2$

Assume S_2 is an equality axiom. Then, since we are only removing essential cuts, the cut formula must be of the form $P(t_1, \dots, t_n)$, for P predicate symbol other than $=$.

If S_1 is also an equality axiom, then we have:

$$\dfrac{s_i = t_i, P(s_1, \dots, s_n) \vdash P(t_1, \dots, t_n) \qquad t_i = r_i, P(t_1, \dots, t_n) \vdash P(r_1, \dots, r_n)}{s_1 = t_1, \dots, s_n = t_n, t_1 = r_1, \dots, r_n = t_n, P(s_1, \dots, s_n) \vdash P(r_1, \dots, r_n)}\,\text{cut}$$

where $s_i = t_i$ is a short-cut for $s_1 = t_1, \dots, s_n = t_n$. We can obtain the same end-sequent using only inessential cuts with the cut-formulae $s_1 = r_1, \dots, s_n = r_n$ in the following way:

$$\begin{array}{c}
\dfrac{s_1 = t_1, t_1 = r_1 \vdash s_1 = r_1 \qquad s_1 = r_1, \dots, s_n = r_n, P(s_1, \dots, s_n) \vdash P(r_1, \dots, r_n)}{s_1 = t_1, t_1 = r_1, s_2 = r_2 \dots, s_n = r_n, P(s_1, \dots, s_n) \vdash P(r_1, \dots, r_n)}\,\text{cut} \\
\hline
\vdots \\
\text{cuts } (s_i = r_i) \\
\hline
s_1 = t_1, \dots, s_n = t_n, t_1 = r_1, \dots, r_n = t_n, P(s_1, \dots, s_n) \vdash P(r_1, \dots, r_n)
\end{array}$$

The rest of the proof is the same as for $\mathcal{LK}$.

□

2.1.2 Midsequent theorem for $\mathcal{LK}$ and $\mathcal{LK}_e$

Theorem 5 (Gentzen's midsequent theorem for $\mathcal{LK}$)**.** *Let S be a sequent which consists of prenex formulas only and is provable in $\mathcal{LK}$. Then there is a cut-free proof of S which contains a sequent (called a mid-sequent) S', which satisfies the following:*

1. *S' is quantifier-free.*
2. *Every inference above S' is either structural or propositional.*
3. *Every inference below S' is either structural or a quantifier inference.*

Thus a midsequent splits the proof into an upper part, which contains the propositional inferences, and a lower part, which contains the quantifier inferences.

Proof. By induction on the number of propositional inferences occurring below a quantifier inference. The strategy is to move upwards the propositional inferences. Consider ∀R. Let I be the first propositional inference occurring after this ∀R application. Assume that I is the first logical inference below ∀R. We have a proof of this form:

$$\dfrac{\dfrac{\Gamma \vdash \Delta, A(\alpha)}{\Gamma \vdash \Delta, \forall x A(x)}\ \forall\mathrm{R}}{\dfrac{\vdots}{\Delta \vdash \Lambda}\ \mathrm{I}}$$

Transform it in the following way:

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash \Delta, A(\alpha)}{\text{Structural Inferences}}}{\Gamma \vdash \Delta, A(\alpha), \forall x A(x)}}{\dfrac{\dfrac{\dfrac{\vdots}{\Delta \vdash A(\alpha), \Lambda}\ \mathrm{I}}{\Delta \vdash \forall x A(x), \Lambda}\ \forall\text{-R}}{\Delta \vdash \Lambda}\ \text{Contraction}}$$

□

The mid-sequent is also called a Herbrand sequent. It provides an example of how cut-elimination can be used as a proof analysis tool: Herbrand sequents provide information about the substitution terms for quantifiers used in the end-sequent.

2.2 Cut-elimination theorem for $\mathcal{LK}_\omega$

One could try repeating the same argument for proving the cut-elimination theorem for $\mathcal{LK}_\omega$. Most of the argument will go through. However, consider one of the quantifier cases in the case where $Rank(P) = 2$:

Case $\forall$

$$\dfrac{\forall\text{R}\,\dfrac{\Gamma_1 \vdash \Delta_1, A(\alpha)}{\Gamma_1 \vdash \Delta_1, \forall x A(x)} \qquad \dfrac{\Gamma_2, A(\mathbf{T}) \vdash \Delta_2}{\Gamma_2, \forall x A(x) \vdash \Delta_2}\,\forall\text{L}}{\Gamma_1, \Gamma_2^* \vdash \Delta_1^*, \Delta_2}\,\text{mix } \forall x A(x)$$

Here we cannot apply the induction hypothesis on grade of the formula to $A(\mathbf{T})$, since $\mathbf{T}$ can be an abstraction of a formula more complex than $\forall x A(x)$. For instance:

Example 9.

$$\dfrac{\forall\text{-}R\,\dfrac{\Gamma_1 \vdash \Delta_1, \alpha(\beta)}{\Gamma_1 \vdash \Delta_1, \forall X X(\beta)} \qquad \dfrac{\Gamma_2, P(\beta) \wedge Q(\beta) \vdash \Delta_2}{\Gamma_2, \forall X X(\beta) \vdash \Delta_2}\,\forall L}{\Gamma_1, \Gamma_2^* \vdash \Delta_1^*, \Delta_2}\,\textit{mix } \forall X X(\beta)$$

One could try to modify the proof by adding another induction measure. However, there is no chance that this will work and that in the end we obtain a reductive or elementary proof (that is, a proof formalizable in Peano Arithmetic). This follows from Gödel's second incompleteness theorem. Below we provide the sketch of the proof of this fact from [Girard, 1991].

2.2.1 Cut-elimination for $\mathcal{LK}_\omega$ implies consistency of second-order arithmetic

The proofs of cut-elimination for $\mathcal{LK}_\omega$ are either semantic or semi-semantic, due to the fact that cut-elimination for $\mathcal{LK}_\omega$ implies consistency of second-order arithmetic: thus no elementary proof can exist, by Second Gödel's Incompleteness theorem. *A forteriori*, such a proof cannot be formalized neither in $\mathcal{LK}_\omega$ nor in second-order arithmetic.

Definition 33 (PRA). *Primitive recursive arithmetic* **PRA** *consists of the following axioms, added to $\mathcal{LK}$:*

1. *Equality axioms;*
2. *Axioms of elementary arithmetic;*
3. *Defining equations for all primitive recursive functions and predicates.*

   *The above axioms define elementary arithmetic **EA**.*
4. *Quantifier-free induction.*

All of the above axioms can be expressed by atomic axioms, and then cut-elimination theorem is proved in the same way as for $\mathcal{LK}_e$: in such systems all but inessential cuts can be removed in the same way as in $\mathcal{LK}$.

Proposition 5. ***PRA*** $\vdash$ *$Con(\mathcal{LK}^+)$, where $\mathcal{LK}^+$ is $\mathcal{LK}$ with equality and elementary arithmetic axioms.*

Proof. See [Takeuti, 1987].

□

Definition 34 ($\mathbf{PA}_2$)**.** *Second-order arithmetic $\boldsymbol{PA}_2$ consists of the following axioms, added to $\mathcal{LK}$:*

1. *Axioms of **PRA**;*
2. *Equality axioms for second-order variables;*
3. *Induction axioms for any formula in $\mathcal{L}_2$;*
4. *Comprehension axioms for all $A \in \mathcal{L}_2$:*

   $$\exists X \forall x (A(x) \leftrightarrow X(x))$$

   with X not free in A.

Theorem 6 (Girard, 1987)**.** *Cut-elimination in $\mathcal{LK}_2$ implies consistency of $\boldsymbol{PA}_2$. Hence, there is no elementary proof of cut-elimination for $\mathcal{LK}_2$ and $\mathcal{LK}_\omega$.*

In other words, the reductive cut-elimination proof cannot be adapted to $\mathcal{LK}_\omega$.

Proof sketch. The idea is to show, by an argument formalizable in **PA**, that cut-elimination in $\mathcal{LK}_2$ implies consistency of $\mathbf{PA}_2$. Since $\mathbf{PA}_2 \nvdash Cons(\mathbf{PA}_2)$ and $\mathbf{PA} \nvdash Cons(\mathbf{PA}_2)$, by Gödel's second incompleteness theorem, we have that cut-elimination for $\mathcal{LK}_2$ cannot be proved neither in **PA** nor $\mathbf{PA}_2$. Hence, more complex methods are needed for the proof (for instance, reducibility candidates method of [Girard et al., 1989], chapter 4).

One proceeds by showing that $\mathbf{PA} \vdash$ Cut-elimination in $\mathcal{LK}_2 \wedge Cons(\mathbf{EA}) \rightarrow Cons(\mathbf{PA}_2)$. The result follows from the fact that $\mathbf{PA} \vdash Cons(\mathbf{EA})$.

To start, show that $\mathbf{PA}_2$ can be interpreted in $\mathcal{LK}_2$ with second-order equality and elementary **EA** axioms. That is, that induction and comprehension axioms can be interpreted in $\mathcal{LK}_2$. It is easy to see that $\mathcal{LK}_2$ proves comprehension axioms, thus they are redundant:

$$\dfrac{\dfrac{\begin{matrix}\vdots\\ \vdash\ A(\alpha) \leftrightarrow A(\alpha)\end{matrix}}{\vdash\ \forall x[A(x) \leftrightarrow A(x)]}\,\forall\text{R}}{\vdash\ \exists X \forall x[A(x) \leftrightarrow X(x)]}\,\exists_2\text{R},\ \mathbf{T}\text{=}\ \lambda x A(x)$$

Then, we can define the predicate for natural numbers in $\mathcal{LK}_2$. Let

$$N(x) = \forall X(X(0) \wedge \forall z(X(z) \to X(s(z))) \to X(x))$$

and

$$\mathbb{N} = \lambda x N(x).$$

Then relativize all quantifiers to $\mathbb{N}$. $\mathbb{N}$-translation of induction and comprehension is then provable in $\mathcal{LK}_2$ + **EA** + second-order equality. Then similarly, second-order equality can be eliminated by E-translation: relativize the quantifiers to predicates closed under second-order equality.

Thus we have that, whenever $\Delta \vdash \Lambda$ is provable in $\mathbf{PA}_2$, $\Delta^{\mathbb{N},E} \vdash \Lambda^{\mathbb{N},E}$ is provable in $\mathcal{LK}_2$+**EA**, where $\mathbb{N}, E$ are corresponding relativizations of quantifiers. Via this translation we reduce cut-elimination in $\mathcal{LK}_2$ to consistency in $\mathbf{PA}_2$:

Assume $\mathbf{PA}_2$ is inconsistent, i.e., the empty sequent is derivable in $\mathbf{PA}_2$. Then it's translation $(\vdash)^{\mathbb{N},E}$ should be derivable in $\mathcal{LK}_2$ + **EA**. If $\mathcal{LK}_2$ admits cut-elimination, then we can obtain a cut-free derivation of the empty sequent $\vdash$ in $\mathcal{LK}_2$ from axioms **EA**. Since the latter axioms are in the first-order language, by the subformula property, which is the consequence of the cut-elimination theorem, we have a derivation of an empty sequent in $\mathcal{LK}$ from **EA**. We know that $\mathbf{PA} \vdash \mathrm{Cons}(\mathbf{EA})$. Therefore, if the proof of cut-elimination is elementary, then also $\mathbf{PA} \vdash Cons(\mathbf{PA}_2)$, which is impossible by Gödel's second incompleteness theorem.

□

2.3 Cut-elimination theorem for $\mathcal{LK}^f_\omega$ and $\mathcal{LK}^f_\omega(\mathbf{C})$

Theorem 7. *$\mathcal{LK}^f_\omega$ admits cut-elimination.*

Proof. Use the same proof as the one above for $\mathcal{LK}$. We should only consider the cases for the new quantifier rules. When $Rank(P) = 2$ we have the following situation:

$$\dfrac{\forall_f\mathrm{R}\,\dfrac{\Gamma_1 \vdash \Delta_1, A(\alpha)}{\Gamma_1 \vdash \Delta_1, \forall f A(f)} \qquad \dfrac{\Gamma_2, A(\mathbf{T}) \vdash \Delta_2}{\Gamma_2, \forall f A(f) \vdash \Delta_2}\,\forall_f\mathrm{L}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}\,\mathrm{mix}\ (\forall f A(f))$$

We can move the mix upwards and then use induction on the grade, as in the first order case. Note that we don't have the same difficulty as in $\mathcal{LK}_\omega$ due to the fact that the term $\mathbf{T}$ cannot increase grade complexity of the proof. Using Lemma 2 below, we can construct the following proof:

$$\dfrac{\dfrac{\Gamma_1 \vdash \Delta_1, A(\mathbf{T}) \qquad \Gamma_2, A(\mathbf{T}) \vdash \Delta_2}{\Gamma_1, \Gamma_2^{\#} \vdash \Delta_1^{\#}, \Delta_2}\ \text{mix}\ (A(\mathbf{T}))}{\dfrac{\text{Structural inferences}}{\Gamma_1, \Gamma_2 \vdash \Delta_1, \Delta_2}}$$

When $Rank(P) > 2$, the proof remains the same as in the first-order case.

□

Lemma 2. *If $\Gamma(\alpha^\tau) \vdash_{\mathcal{LK}_\omega} \Delta(\alpha^\tau)$, then $\Gamma(\mathbf{T}) \vdash_{\mathcal{LK}_\omega} \Delta(\mathbf{T})$ (replace all α's by $\mathbf{T}$ in the sequent), where α^τ is a free variable different from all eigenvariables of the proof and doesn't occur in $\boldsymbol{T}$, and $\boldsymbol{T}$ an arbitrary term of type τ of $\mathcal{L}^f_\omega$.*

Proof. By induction on the number of inferences in the proof. □

2.3.1 Cut-elimination in $\mathcal{LK}^f_\omega(\mathbf{C})$

In the case of adding β-rules to obtain $\mathcal{LK}^f_\omega(\mathbf{C})$ from $\mathcal{LK}^f_\omega$ one would need to take care of the β-rules during cut elimination. When $Rank(P) = 2$, when moving the cut upwards, the grade of the cut-formula stays the same and thus the induction doesn't go through and the same proof as previously cannot be used. In this case it is easier to adapt the reductive proof of first-order logic with equality to $\mathcal{LK}^f_\omega(\mathbf{C})$ as an $\mathcal{LK}^f_\omega$ with equality axioms. Moreover, such proof would work for any type system that can be formulated as an equational theory of the sort described below.

Definition 35. *A cut is called inessential if its cut formula is of the form $\mathbf{T} =_\tau \mathbf{T}'$. Otherwise it is called essential.*

Theorem 8. *If a sequent S is $\mathcal{LK}^f_\omega(\mathbf{C})$ provable, then it is provable without essential cut.*

Proof. Same as for $\mathcal{LK}_e$, just add the cases for $=_\tau$ axioms, which are the same as for simple equality $=$.

□

CHAPTER 3

Standard and General Semantics for $\mathcal{L}_\omega$ and $\mathcal{L}_\omega^f$

3.1 Standard and general semantics for $\mathcal{L}_\omega$

Usually one considers a simplified formulation of type theory and its semantics, called relational type theory. Since one can express functions as special type of relation, this is not a real restriction.

Definition 36 (Relational types)**.** *The relational types $\mathbb{T}$ are:*

1. $i, o \in \mathbb{T}$,
2. $\tau_1 \to \ldots \to \tau_n \to o \in \mathbb{T}$, *if* $\tau_1, \ldots, \tau_n \in \mathbb{T}$.

Below we will omit mentioning o and write:

1. $i \in \mathbb{T}$,
2. $(\tau_1 \ldots \tau_n) \in \mathbb{T}$, *if* $\tau_1, \ldots, \tau_n \in \mathbb{T}$.

For instance, the second order variables have types of the form $(i, \ldots, i)$. (Alternatively $i \to \ldots \to i \to o$.)

Definition 37 (Standard semantics)**.** *Let $\tau \in \mathbb{T}$ and let D be a set. Define*

1. $D_\iota = D$,
2. $D_o = \{true, false\}$,

3. $D_{(\tau_1 \dots \tau_n)} = \mathcal{P}(D_{\tau_1} \times \dots \times D_{\tau_n})$

Let σ be a variable assignment, that is, some function $\sigma : \mathcal{V}^\omega \to \bigcup_{\tau \in \mathbb{F}} D_\tau$. Then $\mathfrak{M} = (D, I)$ is a standard $\mathcal{L}^f_\omega$-model if the interpretation I maps the variables and constants to the members of the appropriate domains D_τ. Formally:

- $I(\alpha^\tau) = \sigma(\alpha^\tau)$ *for* $\alpha^\tau \in \mathcal{V}^\omega$[1]
- $I(c^\tau) \in D_\tau$ *for* $c^\tau \in \mathcal{C}^\omega$
- $I^\sigma((t^{\sigma \to \tau} s^\sigma)^\tau) = I^\sigma(t^{\sigma \to \tau}) I(s^\sigma) \in D_\tau$
- $I^\sigma(\lambda x^\sigma . t^\tau) \in D_{\sigma \to \tau}$ *such that* $I^\sigma(\lambda x^\sigma . t^\tau)(m) = I^{\sigma \cup \{x := m\}}(t^\tau)$, *where* $\sigma \cup \{x := m\}$ *means* σ *with* x *mapped to* m.

Let $\models^s_{\mathcal{L}^f_\omega}$ denote the standard semantics satisfiability relation: we say that $\mathfrak{M}, \sigma \models^s_{\mathcal{L}^f_\omega} F^o$ iff $I(F) = true$. We say that a formula $A \in \mathcal{L}^f_\omega$ is $\models^s_{\mathcal{L}^f_\omega}$-satisfiable iff there is a standard model $\mathfrak{M}$ and a variable assignment σ such that $\mathfrak{M}, \sigma \models^s_{\mathcal{L}^f_\omega} A$. The quantified formulae is interpreted in the following way:

$$\mathfrak{M} = (D, I), \sigma \models^s_{\mathcal{L}^f_\omega} \forall x^\tau F(x)$$

$$\textit{iff}$$

$$\textit{for all } d \in D_\tau : (D, I), \sigma \cup \{x := d\} \models^s_{\mathcal{L}^f_\omega} F(x)$$

Unsatisfiability and validity are defined as usual.

Theorem 9. *The set of $\mathcal{L}_\omega$-validities wrt standard semantics is not recursively enumerable.*

Proof. Use the fact that Peano Arithmetic formulated as a second-order theory is a categorical theory: every two standard models of Peano Arithmetic are isomorphic. Then the idea is that one can express validity in the standard model of arithmetic using one $\mathcal{L}_2$ sentence, and we know that the former is not recursively enumerable, by Gödel's first incompleteness theorem.

$\square$

In general semantics everything stays the same, except that D_τ is defined differently:

Definition 38 (General semantics)**.** *Let $\tau \in \mathbb{T}$ and let D be a set.*

[1]Sometimes it is convenient to write I^σ to mean an interpretation I given the variable assignment σ

1. $D_\iota = D$

2. $D_{(\tau_1 \ldots \tau_n)} \subseteq \mathcal{P}(D_{\tau_1} \times \ldots \times D_{\tau_n})$.

Then (D, I) with the interpretation as above is a general pre-structure. When we restrict the subsets to the relations definable in higher-order logic, we get the general semantics. Let $\models^g_{\mathcal{L}_\omega}$ denote the general semantics consequence. We will omit the subscript when it is clear which relation is meant.

Theorem 10. *The set of $\mathcal{L}_\omega$-validities wrt general semantics is recursively enumerable.*

Proof. See below. In the case of the general semantics we have more structures, since for each $D_\tau = D_{(\tau_1 \ldots \tau_n)} \subseteq \mathcal{P}(D_{\tau_1} \times \ldots \times D_{\tau_n})$ we have a corresponding structure. Since with general semantics there are more models available, less sentences are valid, and in fact we end up with such a "small" set that it becomes recursively enumerable. This follows from the fact that we can reduce $\models^g_{\mathcal{L}_\omega}$-validity to first-order consequence from a recursive set of axioms $\mathcal{TT}$.

□

Definition 39 ($\mathcal{TT}$)**.** *We define a first-order theory $\mathcal{TT}$ that characterizes the general models. Extend a given first-order language L_1 to L^s with predicate symbols E_τ (of different arities) and unary predicates T_τ for all $\tau \in \mathbb{T}$. Then the theory $\mathcal{TT}$ consists of the following axioms:*

1. $\exists x T_i(x)$

2. *a)* $T_i(c)$, *for constants c of L_1*

   b) $\forall x_1 \ldots \forall x_n [\bigwedge_{j=1}^n T_i(x_j) \rightarrow T_i(f(x_1, \ldots, x_n))]$, *for functions f of L_1*

   c) $\forall x_1 \ldots \forall x_n [R(x_1, \ldots, x_n) \rightarrow \bigwedge_{j=1}^n T_i(x_j)]$, *and predicates R of L_1.*

3. $\forall x (T_\tau(x) \rightarrow \neg T_\sigma(x))$, *for $\sigma \neq \tau$*

4. $\forall r \forall x_1 \ldots \forall x_n [E_\tau(r, x_1, \ldots, x_n) \rightarrow T_\tau(r) \wedge T_{\tau_1}(x_1) \wedge \ldots \wedge T_{\tau_n}(x_n)]$, *for $\tau = \tau_1 \rightarrow \ldots \rightarrow \tau_n \rightarrow \tau_{n+1}$ and the designated predicate E_τ*

5. *Extensionality axioms:*

$$\forall p \forall r [T_\tau(p) \wedge T_\tau(r) \wedge \forall x_1 \ldots \forall x_n (T_{\tau_1}(x_1) \wedge \ldots \wedge T_{\tau_n}(x_n) \rightarrow$$
$$(E_\tau(p, x_1, \ldots, x_n) \leftrightarrow E_\tau(r, x_1, \ldots, x_n)) \rightarrow p = r]$$

Moreover, one adds the comprehension axioms formulated in the language L^s.

Theorem 11. *There is a translation function $s : \mathcal{L}_\omega \to L^s$, such that for all $\phi \in \mathcal{L}_\omega$:*

$$\models^g_{\mathcal{L}_\omega} \phi \text{ iff } \mathcal{TT} \models \phi^s$$

The translation consists in relativizing the quantifiers to appropriate types. Then the predicate E_τ is intended to mean the application predicate. For more details see, for instance, [Van Benthem and Doets, 2001]. Transform a general model into a first order model in the following way:

Definition 40 (s-transformation)**.** *Let (D, I) be a general pre-structure, we obtain a first-order L^s-structure $(D, I)^s = (D^s, I^s)$ as follows:*

1. $D^s = \bigcup_{\tau\in\mathbb{T}} I(\tau)$
2. $I^s(c) = I(c)$ *for* $c \in L$.
3. $I^s(T_\tau) = I(\tau)$.
4. $I^s(E_\tau) = \in$.

Lemma 3. *For all general models (D, I): $(D, I)^s \models \mathcal{TT}$.*

Proof. One can just have to check the axioms (1)–(5) and the comprehension axioms. Consider the comprehension axioms. We want to show that the translations of comprehension axioms are true in $(D, I)^s$. $(D, I)^s \models \exists y \forall \overline{x}(E_\tau(y, \overline{x}) \leftrightarrow \psi^s))$ iff there is $d \in D = \bigcup_{\tau\in\mathbb{T}} I(\tau)$ such that $(D^s, I^s \cup \{y := d\}) \models \forall \overline{x}(E_\tau(y, \overline{x}) \leftrightarrow \psi))$. In general models we have that $d \in I(\tau)$ for $d : \tau$, and the latter exists and contains all the elements specified by ψ by higher-order comprehension being true on (D, I).

□

One can easily prove by induction that the translation is truth-preserving:

Lemma 4. *For all $\phi \in L_\omega : (D, I) \models_{L_\omega} \phi$ iff $(D, I)^s \models \phi^s$*

We can characterize $(D, I)^s$-structures using $\mathcal{TT}$ up to isomorphism.

Lemma 5. *Let S be an L^s-structure. Then $S \models \mathcal{TT}$ iff there is a general structure (D, I) such that $(D, I)^s$ is isomorphic to S.*

Proof. ($\Leftarrow$) Follows from Lemma 4.

($\Rightarrow$) For a $S = (H, J)$ such that $S \models \mathcal{TT}$ construct a general model (D, I) and an isomorphism $h : (H, J) \to (D^s, I^s)$, by induction on τ. Idea: in S we have $I(T_\tau) \subseteq D$, which we want to transform into $D_\tau \subseteq \mathcal{P}(D_{\tau_1} \times \ldots \times D_{\tau_n})$.

(*Base case*) $D_i = I(T_i)$; h_i is identity . We know D will be non-empty and that the interpretation of non-logical symbols will be defined in (D^s, I^s) in the same way as in S by axioms of $\mathcal{TT}$ concerning T_i (1)–(2).

(*Inductive Step*) Assume $D_{(\tau_1 \dots \tau_n)}$ and $h_{(\tau_1 \dots \tau_n)}$ are already defined. By (4) (which describes correct typing in E_τ predicate) we can set h_τ with $\tau = (\tau_1 \dots \tau_n)$:

$$h_\tau(b) := \{(h_{\tau_1}(a_1) \dots h_{\tau_n}(a_n)); (b, a_1, \dots, a_n) \in I(E_\tau)\}$$

which is well-defined by extensionality (5). Set $D_\tau := h_\tau[I(T_\tau)]$. We have that $D_\tau \subseteq \mathcal{P}(D_{\tau_1} \times \dots \times D_{\tau_n})$ and thus (D, I) is a general structure. Note that since the translation of the comprehension axiom is true in S, the above clause amounts to adding all definable relations. That h_τ is an isomorphism should be clear by construction, since we have that $(b, a_1, \dots, a_n) \in I(E_\tau)$ iff $(h_{\tau_1}(a_1), \dots, h_{\tau_n}(a_n)) \in h_\tau(b)$. Then set $h := \bigcup_{\tau \in \mathbb{T}} h_\tau$ and we are done. This construction is important, since it can be used for the completeness proof. $\square$

Then the proof of the Theorem 10 immediately follows from the lemmas above:

Proof. ($\Leftarrow$) Assume $\mathcal{TT} \models \phi$. It means that for all L^s structures S: $S \models \mathcal{TT} \Rightarrow S \models \phi^s$. Then it follows that for all $A^s \models \mathcal{TT} \Rightarrow A^s \models \phi^s$, since A^s are L^s structures. Since for all A^s: $A^s \models \mathcal{TT}$ (by Lemma 3), we have that for all A^s: $A^s \models \phi^s$ and therefore for all general structures $A \models \phi$ (by Lemma 4).

($\Rightarrow$) Assume $\models \phi$. It means that for all general models $A \models \phi$. Take a model $S \models \mathcal{TT}$ and prove that $S \models \phi^s$. $S \models \mathcal{TT}$ implies that $S \simeq A^s$ for some A general model (Lemma 5). And since $A \models \phi$, it means that $A^s \models \phi^s$ (by Lemma 4) and thus $S \models \phi^s$. $\square$

Theorem 12. *$\mathcal{LK}_\omega$ is sound and complete wrt general semantics.*

Proof. See the original Henkin paper, for instance – the proof can be adapted to natural deduction or sequent calculus.

$\square$

3.2 Standard and general semantics for $\mathcal{L}^f_\omega$

In this section I define standard and general semantics for the functional higher-order logic $\mathcal{L}^f_\omega$, show that $\mathcal{LK}^f_\omega$ is incomplete wrt standard semantics for $\mathcal{L}^f_\omega$, show general incompleteness of $\mathcal{L}^f_\omega$-c, a simple extension of $\mathcal{L}^f_\omega$, then prove soundness and completeness of $\mathcal{LK}^f_\omega$ wrt general semantics. To establish these results, I use @-translation defined in the previous chapter, as well adapt some techniques available in the literature. I point to the relation of $\mathcal{L}^f_\omega$ to some class of unification problems, as defined by Miller for full type theory – this will be relevant for defining appropriate Skolemization procedure for $\mathcal{L}^f_\omega$ in the last chapter. For proving incompleteness of $\mathcal{LK}^f_\omega$ wrt standard semantics, I adapt Schütte's completeness proof for $\mathcal{LK}$ to $\mathcal{LK}^f_\omega$.

3.2.1 Standard semantics for $\mathcal{L}^f_\omega$

Definition 41 (Standard semantics). *Let $\tau \in \mathbb{F}$ (functional type) and let D be some set. Define*

1. $D_\iota = D$,
2. $D_{\tau\to\sigma} = \{f \ ; \ f : D_\tau \to D_\sigma$ *is a set-theoretic function*$\}$[2]

Let σ be a variable assignment, that is, some function $\sigma : \mathcal{V}^\omega \to \bigcup_{\tau\in\mathbb{F}} D_\tau$. Then $\mathfrak{M} = (D, I)$ is a standard $\mathcal{L}^f_\omega$-model if the interpretation I maps the variables and constants to the members of the appropriate domains D_τ. Formally:

- $I(\alpha^\tau) = \sigma(\alpha^\tau)$ *for* $\alpha^\tau \in \mathcal{V}^\omega$[3]
- $I(c^\tau) \in D_\tau$ *for* $c^\tau \in \mathcal{C}^\omega$
- $I((t^{\sigma\to\tau}s^\sigma)^\tau) = I(t^{\sigma\to\tau})I(s^\sigma) \in D_\tau$
- $I(P^n) \subseteq D^n$ *for* P^n $n-$*ary predicate*

Let $\models^s_{\mathcal{L}^f_\omega}$ denote the standard semantics consequence relation. We say that a formula $A \in \mathcal{L}^f_\omega$ is $\models^s_{\mathcal{L}^f_\omega}$-satisfiable iff there is a standard model $\mathfrak{M}$ and a variable assignment σ such that $\mathfrak{M}, \sigma \models^s_{\mathcal{L}^f_\omega} A$. The satisfiability relation is defined as usual:

$\mathfrak{M} = (D, I), \sigma \models^s_{\mathcal{L}^f_\omega} P(t_1, \ldots, t_n)$ *iff* $(I^\sigma(t_1), \ldots, I^\sigma(t_n)) \in I(P)$ *for atomic formulae;*

$\mathfrak{M} = (D, I), \sigma \models^s_{\mathcal{L}^f_\omega} A \wedge B$ *iff* $\mathfrak{M} = (D, I), \sigma \models^s_{\mathcal{L}^f_\omega} A$ *and* $\mathfrak{M} = (D, I), \sigma \models^s_{\mathcal{L}^f_\omega} B$*; Similarly for other propositional connectives. For the quantified formulae it is interpreted in the following way:*

$$\mathfrak{M} = (D, I), \sigma \models^s_{\mathcal{L}^f_\omega} \forall x^\tau F(x)$$

$$\textit{iff}$$

$$\textit{for all } d \in D_\tau : (D, I), \sigma \cup \{x := d\} \models^s_{\mathcal{L}^f_\omega} F(x)$$

Unsatisfiability and validity are defined as usual.

[2]That is, the available functions such as choice function will depend on the underlying set-theoretic assumptions.

[3]Sometimes it is convenient to write I^σ to mean an interpretation I given the variable assignment σ

Example 10.

$$(D,I),\sigma \models^s_{\mathcal{L}^f_\omega} \exists f^{i\to i}\forall x^i.\ x = f(x)$$

$$\textit{iff}$$

$$(D,I),\sigma \cup \{f := m\} \models^s_{\mathcal{L}^f_\omega} \forall x^i.\ x = m(x)$$

$$\textit{for some } m \in D_{i\to i} \textit{ (that is, for some function on } D_i\textit{)}$$

$$\textit{iff}$$

$$\textit{for all } d \in D_i : (D,I),\sigma \cup \{f := m, x := d\} \models^s_{\mathcal{L}^f_\omega} x = f(x).$$

For any D_i and σ, one can take $f \mapsto m$ with $m \in D_{i\to i}$ the identity function on D_i to satisfy this formula, thus it $\exists f^{i\to i}\forall x^i.\ x = f(x)$ valid wrt standard $\mathcal{L}^f_\omega$ semantics.

Standard semantics is very strong, even for this restricted language. In particular, the Compactness theorem fails for the standard semantics consequence relation $\models^s_{\mathcal{L}^f_\omega}$.

Proposition 6. *The Compactness theorem fails for $\models^s_{\mathcal{L}^f_\omega}$: there is an infinite set of $\mathcal{L}^f_\omega$-sentences Γ that is $\models^s_{\mathcal{L}^f_\omega}$-unsatisfiable, however, all finite subsets of Γ are $\models^s_{\mathcal{L}^f_\omega}$-satisfiable.*

Definition 42 (ψ_n)**.** $\psi_n = \exists x^i_1 \ldots \exists x^i_n.\ \bigwedge_{i\neq j\in[n]} x_i \neq x_j$ *(Intuitively, the sentence says: "the are at least n objects in the domain D_i")*

Observation 1. *Let $Inj(f) = \forall x^i \forall y^i (f(x) = f(y) \to x = y)$. Then for any standard model and variable assignment, with $F \in D_{i\to i} : (D,I),\sigma \cup \{f := F\} \models Inj(f)$ iff F is an injective function on D_i.*

[*The fact is trivial, however, it can serve as another example of evaluation wrt standard semantics, thus I provide the details:*

$$(D,I),\sigma \cup \{f := F\} \models Inj(f)$$

$$\textit{iff}$$

for all $d, d' \in D_i$: $(D,I),\sigma' \models f(x) = f(y)$ then $(D,I),\sigma' \models x = y$, with

$$\sigma' = \sigma \cup \{f := F, x := d, y := d'\}$$

Now take any $a \neq a' \in D_i$, then $(D,I),\sigma\cup\{f := F, x := a, y := a'\} \not\models x = y$ $F(a) \neq F(a')$ and thus $(D,I),\sigma \cup \{f := F, x := a, y := a'\} \not\models f(x) = f(y)$, that is $F(a) \neq F(a')$. Thus F is injective.]

Observation 2. *Let $Surj(f) = \forall z^i \exists v^i.\ f(v) = z$. Then for any standard model and variable assignment, with $F \in D_{i\to i} : (D,I),\sigma \cup \{f := F\} \models Surj(f)$ iff F is a surjective function on D_i.*

Definition 43 (ψ_{fin})**.** $\psi^{fin} = \forall f^{i\to i}.\, Inj(f) \to Surj(f)$

Fact 1. *All injective functions on D are surjective iff D is finite.*

Thus using ψ^{fin} one can characterize finite structures in $\mathcal{L}^f_\omega$, from which the failure of Compactness follows.

Lemma 6. *For any σ: $\mathfrak{M}, \sigma \models_{\mathcal{L}^f_\omega} \psi^{fin}$ if and only if the domain of $\mathfrak{M}$ is finite.*

Proof. ($\Rightarrow$)

$$\mathfrak{M} \models_{\mathcal{L}^f_\omega} \psi^{fin}$$

iff

for all $F \in D_{i\to i}$: $(D, I), \sigma \cup \{f := F\} \models Inj(f)$ then $(D, I), \sigma \cup \{f := F\} \models Surj(f)$

iff

all injective functions on D_i are surjective, i.e., D_i is finite

($\Leftarrow$) Assume that $\mathfrak{M}, \sigma \not\models \psi^{fin}$. Show that the domain of $\mathfrak{M}$ has to be infinite.

$$\mathfrak{M}, \sigma \not\models \psi^{fin}$$

iff

$\mathfrak{M}, \sigma \cup \{f := F\} \models Inj(f)$ and $\mathfrak{M}, \sigma \cup \{f := F\} \not\models Surj(f)$

for some $F \in D_{i\to i}$

By observation above, it means that some function F on D_i is injective, but not surjective. This can only happen if D_i is infinite.

□

Proof of non-compactness. Take the sentence ψ^{fin} and sentences ψ^n for each $n \in \mathbb{N}$:

$$\Gamma = \{\psi^{fin}\} \cup \{\psi^n; n \in \mathbb{N}\}$$

Any finite set $\Gamma' \subset \Gamma$ is clearly satisfiable, however, Γ is not satisfiable, since $\{\psi^n; n \in \mathbb{N}\}$ requires the domain to be at least as big as any n and ψ^{fin} the domain to be finite.

□

3.2.2 Soundness

To prove soundness of $\mathcal{LK}^f_\omega$ wrt standard semantics, first extend the notion of $\models^{st}_{\mathcal{L}^f_\omega}$-satisfiability to sequent. Let S be an $\mathcal{LK}^f_\omega$ sequent $\Delta \vdash \Lambda$ We say that $\mathfrak{M}, \sigma \models^s_{\mathcal{L}^f_\omega} S$ iff for some $A \in \Delta$, $\mathfrak{M}, \sigma \not\models^s_{\mathcal{L}^f_\omega} A$ or for some $A \in \Lambda$, $\mathfrak{M}, \sigma \models^s_{\mathcal{L}^f_\omega} A$.

Theorem 13 (Soundness of $\mathcal{LK}^f_\omega$)**.** *$\mathcal{LK}^f_\omega$ is sound wrt standard semantics: that is, if a sequent S is $\mathcal{LK}^f_\omega$-provable then for all standard models $\mathfrak{M}$ with variable assignments σ: $\mathfrak{M}, \sigma \models^s_{\mathcal{L}^f_\omega} S$.*

Proof. By induction on the number of inferences in the $\mathcal{LK}^f_\omega$-proof of S.

Base: S is an axiom, clearly holds in all standard models.

Induction Step: Induction step is easy. Consider the case $\exists$-R:

Case $\Delta \vdash \Lambda, \exists x^\tau A$:

$$\cfrac{\cfrac{\vdots}{\Delta \ \vdash\ \Lambda, A(\mathbf{T}^\tau)}}{\Delta \ \vdash\ \Lambda, \exists x^\tau A(x)}\ \exists\text{-R}$$

By induction hypothesis we know that if $\mathfrak{M}, \sigma \not\models \Lambda$, then $\mathfrak{M}, \sigma \models A(\mathbf{T}^\tau)$ for any σ and $\mathfrak{M} = (D, I)$. We know that then $I(\mathbf{T}^\tau) = F \in D_\tau$ for some $F \in D_\tau$. Thus $\mathfrak{M}, \sigma \cup \{x := F\} \models A(x)$ for any $\mathfrak{M}$ and thus $\mathfrak{M}, \sigma \models \exists x^\tau A(x)$.

Case $\Delta \vdash \Lambda, \forall x^\tau A$:

$$\cfrac{\cfrac{\vdots}{\Delta \ \vdash\ \Lambda, A(\alpha^\tau)}}{\Delta \ \vdash\ \Lambda, \forall x^\tau A(x)}\ \forall\text{-R}$$

Take any $\mathfrak{M}, \sigma' \not\models \Lambda$, show that $\mathfrak{M}, \sigma' \models \forall x^\tau A(x)$:

$$\mathfrak{M}, \sigma' \models \forall x^\tau A(x)$$

$$\text{iff}$$

$$\text{for all } F \in D_\tau : \mathfrak{M}, \sigma' \cup \{x := F\} \models A(x)$$

By induction hypothesis we know that $\mathfrak{M}, \sigma \models A(\alpha^\tau)$ for any σ, thus for any extension of σ' to α, that is for all $F \in D_\tau$: $A(F)$ holds in $\mathfrak{M}$. □

Now extend the standard semantics to treat terms of $\mathcal{L}_\omega^f(\mathbf{C})$:

$$I^\sigma(\lambda x^\tau.t)F = I^{\sigma'}(t), \text{ for } F \in D_\tau \text{ and } \sigma' = \sigma \cup \{x := F\}.$$

Then similarly one can prove soundness of $\mathcal{LK}_\omega^f(\mathbf{C})$, since clearly the standard model will contain all functionals of simple type theory.

Theorem 14 (Soundness of $\mathcal{LK}_\omega^f(\mathbf{C})$)**.** *$\mathcal{LK}_\omega^f(\boldsymbol{C})$ is sound wrt standard semantics: that is, if a sequent S is $\mathcal{LK}_\omega^f(\boldsymbol{C})$-provable then for all standard models $\mathfrak{M}$ with variable assignments σ: $\mathfrak{M}, \sigma \models^s_{\mathcal{L}_\omega^f} S$.*

3.3 Higher-order unification and incompleteness of $\mathcal{L}_\omega^f$

To show the limits of $\mathcal{LK}_\omega^f$ with respect to the standard semantics and its closer resemblance to first-order logic rather than higher-order logic, it is instructive to look at some subset of $\mathcal{L}_\omega^f$ formulae, known as unification problems. The intuition is that some formula cannot be proven in $\mathcal{LK}_\omega^f$ because there is no way of unifying terms involved, but we have to start from the axioms of the form $A(t) \vdash A(t)$ or $\vdash t = t$. We follow [Miller, 1992] who defines and characterizes $\mathcal{L}_\omega^f(\mathbf{C})$ unification problems.

Definition 44 (Higher-order unification problem)**.** *Let $Q_i \in \{\forall x_i^{\tau_i}, \exists x_i^{\tau_i}\}$. Then $Q_1 \ldots Q_n A$ is a $\mathcal{L}_\omega^f$-unification problem if $A \in \mathcal{L}_\omega^f$ and is of the form $\bigwedge_{i=0}^n s_i = t_i$ for some $n \in \mathbb{N}$.*

Intuitively, the universally quantified variables occurring in a given unification problem correspond to the functional and individual constants, whereas the existentially quantified variables are free variables of the unification problem. Moreover, the quantifier prefix also specifies the domains of admissible substitutions for the variables: they can only use the symbols from the quantifiers in whose scope they appear. The usual first-order unification is then a special case where the prefix is of the $\forall\exists$ form.

Definition 45 (Solution to a $\mathcal{L}_\omega^f$-unification problem)**.** *Given a $\mathcal{L}_\omega^f$-unification problem $Q_1 \ldots Q_n C$, let E_x be the set of existentially bound variables in $Q_1 \ldots Q_n C$ and A_x the set of universally bound variables in B. Then a solution to B is a substitution $\theta : E_x \to A_x$ such that $\theta s_i = \theta t_i$ and whenever $x \mapsto_\theta r$ with x bound by Q_i, then r contains only variables bound by Q_j, such that $j < i$.*

Example 11. *$\forall c^{i \to i} \exists x^i \forall d^i \exists y^{i \to i}.\ c(x) = y(d)$ is a $\mathcal{L}_\omega^f$-unification problem with constants c, d and variables x, y. y can be unified with c. However, since x^i is not ins scope of $\forall d^i$ it cannot take value d^i and thus the terms cannot be unified under this prefix. However the first order-unification $\forall c^{i \to i} \forall d^i \exists x^i \exists y^{i \to i}.\ c(x) = y(d)$ has a solution. Note that it is also $\mathcal{LK}_\omega^f$-provable.*

Lemma 7 (Miller 1992)**.** *Let A be an $\mathcal{LK}_\omega^f(\boldsymbol{C})$-unification problem. Then A is $\mathcal{LK}_\omega^f(\boldsymbol{C})$-provable iff A has a solution, provided all types are assumed to be non-empty.*

Proof. See Theorem 2.10 and Theorem 3.8 in [Miller, 1992].

□

Example 12. *Consider a formula $\forall_{i\to i} f \exists_i x \forall_i w.\ fw = x$. There is no solution to the corresponding unification problem, since there is no substitution for x, which would not use w and solve the equation. To see how this is related to provability in $\mathcal{LK}_\omega^f$, consider a possible cut-free $\mathcal{LK}_\omega^f$-proof of this statement. The proof would have to contain quantifier introduction sequence like this:*

$$\dfrac{\dfrac{\dfrac{\vdash\ \alpha_{i\to i}\beta_i = \boldsymbol{T}}{\vdash\ \forall_i w.\ \alpha_{i\to i} w = \boldsymbol{T}}\ \forall\text{-}r,\ \beta_i\ \textit{eigen: } \boldsymbol{T} \textit{ cannot contain } \beta_i}{\vdash\ \exists_i x \forall_i w.\ \alpha_{i\to i} w = x}\ \exists\text{-}r}{\vdash\ \forall_{i\to i} f \exists_i x \forall_i w.\ fw = x}\ \forall\text{-}r,\ \alpha_{i\to i}\ \textit{eigen}$$

Thus $\boldsymbol{T}$ has to be of the form $\alpha\beta'$ for $\beta' \neq \beta$. Then, however, one cannot prove a statement $\alpha_{i\to i}\beta_i = \alpha_{i\to i}\beta'_i$ in $\mathcal{LK}_\omega^f$.

3.3.1 Incompleteness

It is known that:

Theorem 15. *The set of $\mathcal{L}_\omega$-validities (full higher-order validities) is not recursively enumerable.*

Proof. See, for instance [Leivant, 1994]. The idea is that one can express truth in the standard model of arithmetic with a second-order sentence, since the induction axioms can be expressed with one second-order sentence. Thus we get the conclusion by Gödel's incompleteness theorem.

□

Knowing the general incompleteness result for second-order and higher-order logics and having seen the @-embedding $\mathcal{LK}_\omega^f$ into $\mathcal{LK}$, we can expect $\mathcal{LK}_\omega^f$ to be too weak to capture the standard semantics validity. Thus we can expect the following to hold:

Theorem 16. *$\mathcal{LK}_\omega^f$ is incomplete wrt standard semantics.*

Proof. Consider a sentence $\forall_i x \exists_{i\to i} f.\ fx = x$. Clearly, it is true in any standard models, since it will contain identity function. However, one cannot prove this sentence in $\mathcal{LK}_\omega^f$. The reason is that by cut-elimination we know that the a proof of this formula will contain a mid-sequent of the form $\vdash \mathbf{T}\alpha = \alpha$ for $\mathbf{T}$ term and α variable. This, however cannot be proven for any $\mathcal{L}_\omega^f$-term, since only syntactically equal terms can be proven equal in $\mathcal{LK}_\omega^f$. That is, $\beta\eta$ equality coincides with syntactic equality in this logic as all terms of $\mathcal{L}_\omega^f$ are in the normal form.

□

Alternatively one can see this using the @-translation: $\vdash @_{i\to i}(\mathbf{T}, \alpha) = \alpha$ is not not $\mathcal{LK}_=$-provable, since not first-order unifiable. Thus we can see the most basic case of the relation of provability in simple type theory to unification established by [Miller 1992]. One cannot unify $[\alpha^{i\to i}c^i = c^i]$ where c is a constant and α a variable, both being $\mathcal{L}_\omega^f$-terms. However, this unification problem has solutions for $\mathcal{L}_\omega^f(\mathbf{C})$-terms and $=$ as $=_{\beta\eta}$-equality.

We can approximate the incompleteness result for the full higher-order logic in the following way: given at least to elements in the domain and a designated constant we can simulate predicate quantification with functional quantification.

Theorem 17. *The set* $\{\phi \in \mathcal{L}_\omega^f + c\ ;\ \exists x^i \exists y^i.\ x \neq y \models^s_{\mathcal{LK}_\omega^f} \phi\}$ *is not recursively enumerable.*

Proof. Express predicate quantification using function quantification (for this one needs the c constant and at least to objects in the domain to represent true and false), then we have the Comprehension principle in the form:

$$\exists f \forall \overrightarrow{x}.\ [f\overrightarrow{x} = c \leftrightarrow A\overrightarrow{x}]$$

where A is an $\mathcal{L}_\omega^f$-formula. The one can express second-order arithmetic truth and make the same argument as for full higher-order logic.

□

3.4 Completeness of $\mathcal{L}_\omega^f$ wrt general semantics

In general semantics D_τ is defined as some non-empty subset of functions of the corresponding type:

Definition 46 (General semantics)**.** *Let $\tau \in \mathbb{F}$ and let D be a set.*

1. $D_\iota = D$
2. $D_{\tau\to\sigma} \subseteq \{f\ ;\ f : D_\tau \to D_\sigma$ *is a set-theoretic function*$\}$

Then (D, I) with the interpretation as above is a general structure if each D_τ is non-empty. We could further restrict the subsets to the functions definable in $\mathcal{L}_\omega^f$, however, then we would get the general semantics with respect to which $\mathcal{LK}_\omega^f$ is not complete, due to an example mentioned above. Let $\models^g_{\mathcal{L}_\omega^f}$ denote the general semantics consequence.

In the case of the general semantics we have more structures, since for each subset of functions we have a corresponding structure. Since with general semantics there are more models available, less sentences are valid, and in fact we end up with such a “small" set that it becomes recursively enumerable. In the case of full higher-order logic this follows from the fact that we can reduce $\models^g_{L_\omega}$-validity to first-order consequence from a recursive set of axioms that describe the type restrictions and translations of comprehension axioms. Thus we can use the first-order procedure to enumerate the general validities of $\mathcal{L}\omega$ or equivalently $\mathcal{L}^f_\omega$+c. Since the comprehension axioms are not required for $\models^g_{\mathcal{L}^f_\omega}$-validity, it is much weaker than $\models^g_{\mathcal{L}_\omega}$-validity, and even $\models^g_{\mathcal{L}^f_\omega(\mathbf{C})}$-validity, thus it is better described as an extension/variant of first-order logic.

Remark 2. *A similar result holds for a logic with quantification over predicates, but no function symbols and no comprehension axioms – the cut-elimination goes through there, and the expressive power is not much higher than that of first-order logic. When one adds just predicative Comprehension, one also stays within reasonable logic complexity and weaker expressive power.*

3.4.1 Completeness

Theorem 18. *$\mathcal{LK}^f_\omega$ is complete wrt general semantics as defined above.*

Proof. Similar to first-order logic. Below we provide the full proof.

□

We use Schütte’s method from [Takeuti, 1987]: given an $\mathcal{L}^f_\omega$ sequent S, construct a “derivation tree” $T(S)$ such that either one can easily transform it into a cut-free proof of S or read off it an interpretation that falsifies S. First we extend the completeness proof for first-order logic without function symbols to first-order logic with function symbols and then extend this proof to $\mathcal{LK}^f_\omega$. Then to extend the same proof to $\mathcal{LK}^f_\omega$ where we will have to modify ∀-L and ∃-R steps in the construction of the reduction tree and the canonical interpretation.

We will need this definition below:

Definition 47 (Pure pre-terms and pure terms)**.** *A (pre-)term of $\mathcal{L}^f_\omega$ is said to be pure if it doesn’t contain any bound variables. Remember that pre-terms are just usual “terms" of any type, whereas terms are pre-terms of the individual type i.*

Theorem 19. *Given a first-order sequent S, there is either a cut-free $\mathcal{LK}$ proof of it, or an interpretation $\mathfrak{M}^{\mathfrak{C}}$ that falsifies it.*

Proof. We adapt the proof from [Takeuti, 1987] to the sequents that may contain function symbols. We need to modify the cases for quantifier rules in both construction of the reduction tree and construction of the canonical interpretation.

Idea: Given a sequent S construct a reduction tree, each node of which is a sequent obtained by reversely applying all possible rules to the previous node, until the axiom sequent is reached. If all the leaves are axioms, then one can easily construct a cut-free proof of S. If, however, it has an infinite branch, that is, the axiom is not reached, then one can construct a canonical interpretation from this tree that falsifies S by taking as domain the terms appearing in the branch.

The reduction tree $T(S)$ for S is constructed from the root S by looping through the following stages:

(Stage 0) If all leaves of $T(S)$ are of the form $A, \Delta \vdash \Lambda, A$, then stop.

(Stage 1) If the topmost sequent obtained at the previous stage $\Pi \vdash \Lambda$ is such that $\neg A_1, \dots, \neg A_n \in \Pi$ not reduced, then the current sequent is $\Pi \vdash \Lambda, A_1, \dots, A_n$ and we say that $\neg A_1, \dots, \neg A_n$ are already reduced.

[For the details of the construction for other connectives see [Takeuti, 1987]. Here we discuss in detail the quantifiers since this is the only stage that changes in the proof.]

$\vdots$

(Stage 8) Assume that we have $\Pi \vdash \Lambda$ at the previous stage, such that $\forall x_1 A_1(x_1), \dots,$ $\forall x_n A_n(x_n) \in \Lambda$ and the $\forall$R was not applied to $\forall x_1 A_1(x_1), \dots, \forall x_n A_n(x_n)$. Then at the current stage add the node $\Pi \vdash A_1(\alpha_1), \dots, A_n(\alpha_n), \Lambda$, where $\alpha_1, \dots, \alpha_n$ are some *new* variables. Then we say that the rule $\forall$R was applied to $\forall x_1 A_1(x_1), \dots, \forall x_n A_n(x_n)$ respectively.

(Stage 9) Assume that we have $\Pi \vdash \Lambda$ at the previous stage, such that $\forall x_1 A_1(x_1), \dots,$ $\forall x_n A_n(x_n) \in \Pi$.

Definition 48 (Terms available at a stage k)**.** *Terms available at a stage k are pure terms contained in any sequent added before the stage k. If there are none, take new variables and let them be available at the stage k.*

The goal is to eventually apply the $\forall$-L reduction to all terms in the branch after the first appearance of universally quantified formulae on the left. We will need to order the available terms at the current stage in order to make sure that none of the available terms is missed by applying the $\forall$-L rule, thus we define the availability sequence Avail during the construction of $T(S)$. Later it will be used as the domain of the canonical interpretation.

At this stage add to Avail all pure terms appearing in the precedent node $\Pi \vdash \Lambda$ and not already in Avail, in arbitrary order. If Avail is empty, add any variable.

Then add the node $A_1(t_1), \dots, A_n(t_n), \Pi \vdash \Lambda$, where $t_1, \dots, t_n$ are the first pure terms in Avail not already used for reducing $\forall x_1 A_1(x_1), \dots, \forall x_n A_n(x_n)$ respectively.

Example 13. *Here is an example of the initial segment of $T(\forall x Qx, \forall x Pgx \vdash \forall x Pfx)$; $(\forall x Pfx)^*$ means that the formula is already reduced.*

$$\begin{array}{c}
\vdots \\ \hline
Qga, Pgga, Qfa, Pgfa, Qa, Pga, \forall x Px, \forall x Pgx \ \vdash \ (\forall x Pfx)^*, Pfa \\ \hline
Qfa, Pgfa, Qa, Pga, \forall x Px, \forall x Pgx \ \vdash \ (\forall x Pfx)^*, Pfa \\ \hline
Qa, Pga, \forall x Px, \forall x Pgx \ \vdash \ (\forall x Pfx)^*, Pfa \\ \hline
\forall x Qx, \forall x Pgx \ \vdash \ (\forall x Pfx)^*, Pfa \\ \hline
\forall x Qx, \forall x Pgx \ \vdash \ \forall x Pfx
\end{array}$$

Availability sequence Avail*:* $a, fa, ga, gfa, gga \ldots$

(Stage 10 and 11) Treat ∃-L and ∃-R in the symmetric manner to ∀-R and ∀-L.

(Stage 12) If $\Pi \vdash \Lambda$ have a formula in common, add nothing; otherwise add $\Pi \vdash \Lambda$.

Canonical interpretation If the reduction tree $T(S)$ is finite, then all the leaves are axioms, and one can easily construct a proof of S from the tree (note that this gives an alternative proof of cut-elimination theorem). Now assume that there is an infinite branch $\mathfrak{B}^{\mathfrak{C}} = S_1, \ldots, S_n, \ldots$ in $T(S)$. Consider $\bigcup_i \Gamma_i$ and $\bigcup_i \Lambda_i$ with $S_i = \Gamma_i \vdash \Lambda_i$. We define a canonical interpretation $\mathfrak{M}^{\mathfrak{C}} = (D, I)$ and variable assignment σ such that $\mathfrak{M}^{\mathfrak{C}}, \sigma \models^g_{\mathcal{L}^f_\omega} \bigcup_i \Gamma_i$ but for all $A \in \bigcup_i \Lambda_i$, $\mathfrak{M}^{\mathfrak{C}}, \sigma \not\models^g_{\mathcal{L}^f_\omega} A$.

- D is the set of all pure terms occurring in the branch $\mathfrak{B}^{\mathfrak{C}}$;
- $\sigma(\alpha) = \alpha$ for $\alpha \in \mathcal{V}$;
- $I(f)\overrightarrow{d} = \begin{cases} f\overrightarrow{d} & \text{if } f\overrightarrow{d} \in D \\ k \in D \text{ arbitrary} & \text{otherwise} \end{cases}$
- $(I(t_1), \ldots, I(t_n)) \in R^I$ if $R(t_1, \ldots, t_n) \in \bigcup_i \Gamma_i$
- $(I(t_1), \ldots, I(t_n)) \notin R^I$ if $R(t_1, \ldots, t_n) \in \bigcup_i \Lambda_i$

Now by structural induction prove that $\mathfrak{M}^{\mathfrak{C}}, \sigma \models^g_{\mathcal{L}^f_\omega} A$ for all $A \in \bigcup_i \Gamma_i$ and $\mathfrak{M}^{\mathfrak{C}}, \sigma \not\models^g_{\mathcal{L}^f_\omega} A$ for all $A \in \bigcup_i \Gamma_i$. Here consider the case where $A \in \bigcup_i \Gamma_i$ is of the form $\forall x F(x)$. By construction of the branch we know that all pure terms t in the branch, $F(t)$ is added after the first occurrence of $\forall x F(x)$, thus all elements of D satisfy $F(x)$ by induction hypothesis and thus $\forall x F(x)$ holds in $\mathfrak{M}^{\mathfrak{C}}$. For $A \in \bigcup_i \Lambda_i$ we know that $F(a)$ above is not satisfied by induction hypothesis, and A is also falsified.

□

Theorem 20. *Given a functional higher-order sequent $S \in \mathcal{L}^f_\omega$, there is either a cut-free $\mathcal{LK}^f_\omega$ proof of it, or a general model $\mathfrak{M}^{\mathfrak{C}}$ that falsifies it.*

Proof. We modify the previous proof. First, in the construction of the reduction tree, we change the stages for quantifiers in the following way:

(Stage 9) Assume that we have $\Pi \vdash \Lambda$ at the previous stage such that $\forall x_1^{\tau_1} A_1(x_1), \ldots, \forall x_n^{\tau_n} A_n(x_n) \in \Pi$. Then add the node $A_1(t_1^{\tau_1}), \ldots, A_n(t_n^{\tau_n}), \Pi \vdash \Lambda$ at the current stage, where $t_1^{\tau_1}, \ldots, t_n^{\tau_n}$ are the first available *pure pre-terms* of corresponding types, not already used for reducing $\forall x_1^{\tau_1} A_1(x_1), \ldots, \forall x_n^{\tau_n} A_n(x_n)$ respectively. That is, analogously to the previous proof, we define availability sequences Avail_τ, for each type τ occurring in S.

Example 14.

$$\begin{array}{c}
\vdots \\ \hline
Qha, Qgha, Qga, \forall x^i.\ Qlx, \forall x^i.\ Qhx, \forall x^i.\ Qgx, \forall y^{i\to i}\forall x^i.\ Qyx \ \vdash \ \ldots \\ \hline
Qga, \forall x^i.\ Qhx, \forall x^i.\ Qgx, \forall y^{i\to i}\forall x^i.\ Qyx \ \vdash \ (\forall x^i \forall z^{i\to i} Pzx)^*, (\forall z^{i\to i} Pza)^*,\ Pha \\ \hline
\forall x^i.\ Qgx, \forall y^{i\to i}\forall x^i.\ Qyx \ \vdash \ (\forall x^i \forall z^{i\to i} Pzx)^*, (\forall z^{i\to i} Pza)^*,\ Pha \\ \hline
\forall x^i.\ Qgx, \forall y^{i\to i}\forall x^i.\ Qyx \ \vdash \ (\forall x^i \forall z^{i\to i} Pzx)^*, \forall z^{i\to i} Pza \\ \hline
\forall y^{i\to i}\forall x^i.\ Qyx \ \vdash \ (\forall x^i \forall z^{i\to i} Pzx)^*, \forall z^{i\to i} Pfa \\ \hline
\forall y^{i\to i}\forall x^i.\ Qyx \ \vdash \ \forall x^i \forall z^{i\to i}.\ Pzx
\end{array}$$

Availability sequences:

Avail_i: $a^i, (ha)^i, (ga)^i, (gha)^i \ldots$

$\mathsf{Avail}_{i\to i}$: $g^{i\to i}, h^{i\to i}, l^{i\to i} \ldots$

Canonical model $\mathfrak{M}^\mathfrak{C}$ Now from the infinite branch $\mathfrak{B}^\mathfrak{C}$ we read off a canonical general model $\mathfrak{M}^\mathfrak{C} = (\{D_\tau \ ; \ \tau \in \mathbb{F}\}, I)$ and define a variable mapping σ such that $\mathfrak{M}^\mathfrak{C}, \sigma \models \bigcup_i \Gamma_i$ and for all $A \in \bigcup_i \Lambda_i$, $\mathfrak{M}^\mathfrak{C}, \sigma \not\models A$. Challenge: we need to build universes D_τ in a sound way. With this aim we define canonical functionals.

Definition 49 (Canonical functionals $F_{t,\tau}$)**.** *Let $\mathcal{T}$ be the set of all pure pre-terms in the given infinite branch $\mathfrak{B}^\mathfrak{C}$ and $\mathcal{T}_\tau$ be the set of all pure semi-terms of type $\tau \in \mathbb{F}$ occurring in $\mathfrak{B}^\mathfrak{C}$. We define sets D_τ of canonical functionals that will be used below to construct domain of the canonical general model:*

- *$D_i = \{F_{t,i} \ ; \ t^i \in T_i\}$ (that is, we add constants corresponding to the pure terms in $\mathfrak{B}^\mathfrak{C}$);*
- *$D_{\sigma\to\tau} = \{F_{t,\sigma\to\tau} : D_\sigma \to D_\tau \ ; \ F_{d,\sigma} \mapsto F_{td,\tau}, t \in T_{\sigma\to\tau}\}$. If $F_{td,\tau} \notin D_\tau$, map $F_{d,\sigma}$ to a random element of D_τ.*

Example 15. *Assume that $D_i = \{F_{g^n a,i}, F_{f^n a,i} \ ; \ n \in \mathbb{N}\}$ and $D_{i\to i} = \{F_{g,i\to i}, F_{f,i\to i}\}$. Then $F_{g,i\to i}(F_{g^n a,i}) = F_{g^{n+1}a,i} = g^{n+1}a \in T_i$, $F_{f,i\to i}(F_{f^n a,i}) = F_{f^{n+1}a,i} = f^{n+1}a \in T_i$.*

Now define the type domains D_τ, for $\tau \in \mathbb{F}$ of the Henkin model $\mathfrak{M}^\mathfrak{C} = (D^c, I^c)$ in the following way:

- $D^c = \{D_\tau \ ; \ \tau \in \mathbb{F}\}$ with D_τ sets of canonical functionals of corresponding types. Note that only finite amount of D_τ are relevant, which are well-defined.
- $\sigma(\alpha^i) = F_{\alpha,i} = \alpha^i$ for $\alpha^i \in \mathcal{T}_i \cap \mathcal{V}^\omega$
- $\sigma(\alpha^\tau) = F_{\alpha,\tau} \in D_\tau$ for $\alpha^\tau \in \mathcal{T}_\tau \cap \mathcal{V}^\omega$
- $(I(t_1), \ldots, I(t_n)) \in I(R)$ if $R(t_1, \ldots, t_n) \in \bigcup_i \Gamma_i$, where $t^i_1, \ldots, t^i_n \in \mathcal{T}_i$
- $(I(t^i_1), \ldots, I(t^i_n)) \notin I(R)$ if $R(t^i_1, \ldots, t^i_n) \in \bigcup_i \Lambda_i$, where $t^i_1, \ldots, t^i_n \in \mathcal{T}_i$

Note that $I(t^\tau) = F_{s,\tau} \in D_\tau$ for some $s^\tau \in \mathcal{T}_\tau$.

Now we show that our construction actually works, that is:

Proposition 7. *For all $A \in \bigcup_i \Gamma_i$: $\mathfrak{M}^\mathfrak{C} \models A$ and for all $A \in \bigcup_i \Lambda_i$: $\mathfrak{M}^\mathfrak{C} \not\models A$.*

Proof. By structural induction on $A \in \bigcup_i \Gamma_i$ or $A \in \bigcup_i \Lambda_i$ (there is no A in common, since otherwise the branch would be finite, and we are considering an infinite branch by assumption). The only interesting case is the quantifier case, thus we only consider it here:

Case $\forall x^\tau B(x) \in \bigcup_i \Gamma_i$ We want to show that $\mathfrak{M}^\mathfrak{C}, \sigma \models \forall x^\tau B(x)$.

$$\mathfrak{M}^\mathfrak{C} = (D^c, I^c), \sigma \models \forall x^\tau B(x)$$

$$\text{iff}$$

$$\text{for all } d \in D_\tau \ : \ (D^c, I^c), \sigma \cup \{x := d\} \models B(x)$$

$$\text{iff}$$

$$(*) \text{ for all } F_{t,\tau} \in D_\tau \ : \ (D^c, I^c), \sigma \cup \{x := F_{t,\tau}\} \models B(x), \text{ with } t \in T_\tau$$

By the construction of the branch $\mathfrak{B}^\mathfrak{C}$ we know that for all pure semi-terms of type τ, that is, for all $t^\tau \in \mathcal{T}_\tau$, $B(t^\tau)$ is added in some sequent S_j, $j > i$ after the first occurrence of $\forall x^\tau B(x)$ in S_i for some $i \in \mathbb{N}$. Now assume that for some $F_{s,\tau}$, $(D^c, I^c), \sigma \cup \{x := F_{s,\tau}\} \not\models B(x)$. Since $s^\tau \in T_\tau$, we know that at some point $B(s^\tau)$ was added in $\cup_i \Gamma_i$, any by induction hypothesis $(D^c, I^c), \sigma \models B(s^\tau)$. If $I^c(s^\tau) = F_{s,\tau}$, we get a contradiction and thus $(D^c, I^c), \sigma \cup \{x := F_{s,\tau}\} \not\models B(x)$ and $(D^c, I^c), \sigma \models \forall x^\tau B(x)$. If $I^c(s^\tau) \neq F_{s,\tau}$, then $F_{s,\tau} \notin D_\tau$ (by $\circ$ below), which also contradicts the assumption.

($\circ$) If $F_{s,\tau} \in D_\tau$ then $I^c(s, \tau) = F_{s,\tau}$. Prove by induction on term s: If s is a variable, the claim holds by definition. Assume $s = tl$ such that for t, l the claim holds. Assume $F_{tl,\tau} \in D_\tau$. Then $F_{t,\sigma\to\tau} F_{l,\sigma} = F_{tl,\tau}$, by definition. Since $t^{\sigma\to\tau}, l^\sigma \in \mathcal{T}$ (all subterms of a term in $\mathcal{T}$ are also in $\mathcal{T}$), $F_{t,\sigma\to\tau} \in D_{\sigma\to\tau}$ and $F_{l,\sigma} \in D_\sigma$, then by induction hypothesis we get that $I(s) = I(t)I(s) = F_{tl,\tau}$.

Case $\forall x^\tau B(x) \in \bigcup_i \Lambda_i$ If $A \in \bigcup_i \Lambda_i$, $A \in S_i$ for some i, we know that $B(\alpha^\tau)$ in the sequents S_j, $j > i$, is not satisfied and thus $\forall x^\tau B(x)$ is also falsified.

□

This concludes the proof of completeness of $\mathcal{LK}_\omega^f$ wrt to Henkin semantics. □

Corollary 2. *Given an $S \in \mathcal{L}_\omega^f$, there is either a correctly typed cut-free proof of S^*, or a canonical interpretation $\mathfrak{M}^\mathfrak{C}$ that falsifies S^*.*

Proof. In case $S \in \mathcal{L}_1^*$, the first-order procedure can lead to incorrectly typed proofs of S, since all terms have to be substituted in the $\forall$-L and $\exists$-R cases.

Example 16.

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{@_{i\to i}(\alpha,\alpha)=\alpha, \forall f @_{i\to i}(f,\alpha)=\alpha, \forall x \forall f @_{i\to i}(f,x)=x \ \vdash\ @_{i\to i}(\alpha,\alpha)=\alpha}}{\forall f @_{i\to i}(f,\alpha)=\alpha, \forall x \forall f @_{i\to i}(f,x)=x \ \vdash\ @_{i\to i}(\alpha,\alpha)=\alpha}}{\forall x \forall f @_{i\to i}(f,x)=x \ \vdash\ \exists f @_{i\to i}(\alpha,\alpha)=\alpha}}{\forall x \forall f @_{i\to i}(f,x)=x \ \vdash\ \forall x \exists f @_{i\to i}(f,x)=x}$$

If one restricts quantification to certain types, however, then we get the result. Then in the end we get a a correctly typed proof, but instead we have to consider many-sorted models (i.e., models where each type has a separate domain). This is a simpler model than a Henkin model. However, it is easy to see that they are equivalent, thus we don't need to build Henkin universes as previously.

□

3.4.2 Soundness and completeness of $\mathcal{LK}_\omega^f(\mathbf{C})$

$\mathcal{LK}_\omega^f(\mathbf{C})$ is not sound with respect to the general semantics for $\mathcal{L}_\omega^f$. We need to add all the functionals corresponding to simply typed functional semi-terms. Then to see that the $\mathcal{LK}_\omega^f(\mathbf{C})$ is sound one would need to check if all axioms hold in all Henkin models defined this way, which is true by definition, since all such axioms correspond to β-reductions (applications).

Theorem 21 (Completeness of $\mathcal{LK}_\omega^f(\mathbf{C})$)**.** *$\mathcal{LK}_\omega^f(\boldsymbol{C})$ is complete wrt general semantics for $\mathcal{L}_\omega^f(\boldsymbol{C})$.*

Proof sketch. This can be proven in a similar way as the Henkin completeness for second-order logic by [Prawitz, 1967]. The only difference with $\mathcal{L}_\omega^f$ is that we have to make sure that the structure we get from the reduction tree is indeed a Henkin model: that it includes all the functionals corresponding to $\mathcal{L}_\omega^f(\mathbf{C})$ pre-terms to the domains D_τ. The idea is to gradually extend $\mathfrak{M}^\mathfrak{C}$ to obtain the correct structure. One introduces new functional variables corresponding to each pre-term of $\mathcal{L}^f_\omega(\mathbf{C})$ at each point in the construction, such that all the needed functionals are added when a fixed-point is reached.

$\square$

CHAPTER 4

Interpreting $\mathcal{LK}^f_\omega$ proofs as $\mathcal{LK}$ proofs

Below I show the correspondence between $\mathcal{LK}^f_\omega$ and $\mathcal{LK}$.

4.1 From $\mathcal{LK}^f_\omega$ proofs to $\mathcal{LK}$ proofs

First I show that proofs of $\mathcal{L}^f_\omega$ formulae in $\mathcal{LK}^f_\omega$ can be directly translated into $\mathcal{LK}$ proofs of corresponding first-order formulae.

Let L_1 be a first-order language with $\mathcal{FV} = \mathcal{FV}_\omega$, $\mathcal{BV} = \mathcal{BV}_\omega$, $\mathcal{C} = \mathcal{C}_\omega$ and function symbols $\mathcal{F} = \{@_\tau; \tau \in \mathbb{F}\}$ of the arity corresponding to the type arity. We want to define a translation function $* : \mathcal{L}^f_\omega \to \mathcal{L}_1$ such that a sequent S is provable in $\mathcal{LK}^f_\omega$ iff S^* is provable in $\mathcal{LK}$.

Definition 50 (Translation function *)**.** *On pre-terms:*

- $[\alpha^\tau]^* = \alpha^\tau$;
- $[x^\tau]^* = x^\tau$;
- $[c^\tau]^* = c^\tau$;
- $[t^\tau(t_1^{\tau_1}, \ldots, t_n^{\tau_n})^{\tau_{n+1}}]^* = @_\tau((t^\tau)^*, (t_1^{\tau_1})^*, \ldots, (t_n^{\tau_n})^*)$

On formulae:

- $P(t_1, \ldots, t_n)^* = P(t_1^*, \ldots, t_n^*)$

- $\bot^* = \bot$
- $(A \to B)^* = A^* \to B^*$
- $(\forall x A)* = \forall x A^*$

Extend to sequents S^* and proofs π^* in the obvious manner.

Example 17 (Translation of a Skolemized Choice instance)**.**

$$\text{Let } S = \forall x.x \neq g(x) \vdash \exists f.hf \neq f(hf).$$

$$\text{Then } S^* = \forall x.x \neq @_{i\to i}(g,x) \vdash \exists f.@_{(i\to i)\to i}(h,f) \neq @_{i\to i}(f, @_{(i\to i)\to i}(h,f))$$

$\mathcal{LK}^f_\omega$ *proof* π *of* S*:*

$$\dfrac{\dfrac{hg \neq g(hg) \ \vdash \ hg \neq g(hg)}{\forall x.x \neq gx \ \vdash \ hg \neq g(hg)}\ \forall\text{-}l}{\forall x.x \neq gx \ \vdash \ \exists f.hf \neq f(hf)}\ \exists_f\text{-}r$$

$\mathcal{LK}$ *proof* π^* *of* S^**:*

$$\dfrac{\dfrac{@_{(i\to i)\to i}(h,g) \neq @_{i\to i}(g, @_{(i\to i)\to i}(h,g)) \ \vdash \ @_{(i\to i)\to i}(h,g) \neq @_{i\to i}(f, @_{(i\to i)\to i}(h,g))}{\forall x \neq @_{i\to i}(g,x) \ \vdash \ @_{(i\to i)\to i}(h,g) \neq @_{i\to i}(f, @_{(i\to i)\to i}(h,g))}\ \forall\text{-}l}{\forall x \neq @_{i\to i}(g,x) \ \vdash \ \exists f @_{(i\to i)\to i}(h,f) \neq @_{i\to i}(f, @_{(i\to i)\to i}(h,f))}\ \exists\text{-}r$$

Lemma 8. *If S is $\mathcal{LK}^f_\omega$-provable then S^* is $\mathcal{LK}$-provable. Moreover, if π is the $\mathcal{LK}_\omega$ proof of S, then π^* is the corresponding $\mathcal{LK}$ proof of S^*.*

Proof. By induction on the length of the proof π. Consider the case when S is obtained by $\exists$-r application:

$$\dfrac{\Gamma \ \vdash \ \Delta, A(\mathbf{T})}{\Gamma \ \vdash \ \Delta, \exists f A(f)}\ \exists_f\text{-r}$$

$[\Gamma \vdash \Delta, \exists f A(f)]^* = \Gamma^* \vdash \Delta^*, \exists f A^*(f)$. Then π^* will be the proof γ of $\Gamma^* \vdash \Delta^*, A(\mathbf{T})^*$ available by induction hypothesis combined with the $\exists$-r application, since $\mathbf{T}^*$ is a first-order term and $A(\mathbf{T})^* = A^*(\mathbf{T}^*)$:

$$\dfrac{\Gamma^* \ \vdash \ \Delta^*, A^*(\mathbf{T}^*)}{\Gamma^* \ \vdash \ \Delta^*, \exists x A^*(x)}\ \exists\text{-r}$$

□

In order to characterize $\mathcal{LK}_\omega$ in $\mathcal{LK}$ we need the converse as well. We will show the following:

Lemma 9. *For $S \in \mathcal{L}^f_\omega$: if S^* is $\mathcal{LK}$-provable and its proof γ is correctly typed, then $\gamma = \pi^*$ for some $\mathcal{LK}_\omega$ proof π of S.*

Moreover, any correctly typed proof in the language of $\mathcal{L}^*_1$ in $\mathcal{LK}$ corresponds to a proof in $\mathcal{LK}^f_\omega$.

Definition 51 (Correctly typed terms, formulae, proofs)**.** *For first-order terms in the language $\mathcal{L}^*_1$ define the set of correctly typed terms CT:*

- $\mathcal{FV}, \mathcal{BV}, \mathcal{C} \subseteq CT$
- $@_\tau(t, t_1, \ldots, t_n) \in CT$ *if* $\tau = \tau_1 \to \ldots \to \tau_n \to \tau_{n+1}$ *and* $type(t) = \tau, type(t_1) = \tau_1, \ldots, type(t_n) = \tau_n$ *and* $t, \overrightarrow{t} \in CT$

Where $type(x^\tau) = \tau$, $type(@_{\tau_1 \to \ldots \to \tau_n}(\bar{t})) = \tau_n$. A formula is correctly typed if all terms occurring in it are correctly typed. The definition extends to sequents and proofs in an obvious way.

Lemma 10. *If a term t of $\mathcal{L}^*_1$ is correctly typed, then there is a term s in the language L_ω such that $t = s^*$. Same holds for formulae.*

Proof. Follows from the definitions. □

Proof of lemma 9. By induction on the length of the proof γ. Consider the case when S^* is obtained by $\exists$-r application:

$$\frac{\Gamma^* \ \vdash \ \Delta^*, A^*(t)}{\Gamma^* \ \vdash \ \Delta^*, \exists x A^*(x)} \ \exists\text{-r}$$

Since by assumption the proof is correctly typed, by lemma 10 we have that $t = s^*$ for some $\mathcal{L}_\omega$ term s. Thus we can apply induction hypothesis and thus we have a proof of $\Gamma \vdash \Delta, A(s)$, from which by $\exists_f$-r we get $\Gamma \vdash \Delta, \exists x A(x)$ which concludes the proof π. (In the case of cut use the second part of the lemma 10.)

□

4.2 Translation back: from $\mathcal{LK}^*$ to $\mathcal{LK}^f_\omega$

Now we want from a first-order proof go to higher-order proof.

Given a sequent $S \in \mathcal{L}^*_1$, define *folding* of S to $S^f \in \mathcal{L}_\omega$:

- $[\alpha^\tau]^f = \alpha^\tau$;
- $[x^\tau]^f = x^\tau$;
- $[c^\tau]^f = c^\tau$;
- $[@_\tau(t, t_1, \ldots, t_n)]^f = t^f(t_1^f, \ldots, t_n^f)$

The definition naturally extends to formulae, sequents, proofs.

Proposition 8. *If S is correctly typed, then $S^f \in \mathcal{L}_\omega$*

Proof. By definitions. □

Due to subformula property of cut-free proofs, if S is correctly typed, but its proof is incorrectly typed, then the incorrectly typed terms can only occur in the ancestor formula of a weak quantifier inference. Thus:

Proposition 9. *For $S \in \mathcal{L}^*_1$: if S is correctly typed, doesn't contain any weak quantifiers and $\mathcal{LK}$-provable by π, then π^f proves S^f.*

Clearly, if we restrict axiom and weakening to correctly typed formulae, then we are getting $\mathcal{LK}^f_\omega$-proofs immediately.

Proposition 10. *If $S \in \mathcal{L}^*_1$ is correctly typed and S is $\mathcal{LK}$-provable from correctly typed axioms by proof π with only correctly typed weakening formulae, then S^f is $\mathcal{LK}^f_\omega$-provable by the proof π^f.*

Proof. By induction on the number of inferences. □

Thus it is enough to restrict the proofs to the ones using only correctly typed proofs. We give a semantic argument:

Lemma 11. *For $S \in \mathcal{L}^f_\omega$: if S^* is $\mathcal{LK}$-provable, then S is $\mathcal{LK}^f_\omega$-provable.*

Proof. If S is not $\mathcal{LK}^f_\omega$-provable, by completeness of $\mathcal{LK}^f_\omega$ wrt Henkin semantics, there is a Henkin model $\mathfrak{M} = (\{D_\tau; \tau \in \mathbb{F}\}), I)$ such that $\mathfrak{M} \not\models S$. From $\mathfrak{M}$ obtain a first-order model $\mathfrak{N} = (D, J)$ such that $\mathfrak{N} \not\models S^*$ in the following way:

- $D = \bigcup_{\tau \in \mathbb{F}} D_\tau$
- Define interpretation J :
  - $J(x^\tau) = I(x^\tau)$ for $x^\tau \in \mathcal{V} \cup \mathcal{C}$ (variables and constants)
  - $J(P) = I(P)$ for predicate constants
  - $J(@_\tau(t, t_1, \dots, t_n)) = I(t(t_1, \dots, t_n))$ – on the object of not a corresponding type, extend randomly.

  That is, interpret @ as application function, and everything else stays the same.

Clearly then $\mathfrak{N} \not\models S^*$.

□

4.3 Skolemization in $\mathcal{L}_1$, $\mathcal{L}_\omega$ and $\mathcal{L}_\omega^f$

During proof transformation and analysis keeping track of eigenvariable conditions can be tricky; for this reason one often considers Skolemized proofs, proofs without strong quantifier introductions (for definitions see below). In first-order logic, every sentence has a Skolemized form: an equi-satisfiable sentence that contains no strong quantifiers; moreover one can obtain such a sentence by a simple algorithm. In higher-order logic this is no longer the case unless one imposes some restrictions on Skolem terms. The defined translation of $\mathcal{L}_\omega^f$ into first-order logic provides an explanation for imposing such restrictions.

4.3.1 First-order Skolemization

In first-order logic we can define a function *sk* on formulae A such that $\vdash_{\mathcal{LK}} A$ iff $\vdash_{\mathcal{LK}} sk(A)$ and $sk(A)$ contains no strong quantifiers.

Definition 52 (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.

Definition 53 (Positive and negative occurrences)**.** *Suppose B contains an occurrence of A.*

- *If A is B then A occurs positively in B.*
- *If B is $(C \wedge D), (C \vee D), \forall x.C$ or $\exists x.C$ and A occurs positively (negatively) in C (or in D respectively) then A occurs in B positively (negatively).*
- *If B is $(C \to D)$ and A occurs positively (negatively) in D then the corresponding occurrence of A in B is positive (negative); if A occurs in B positively (negatively) in C then the corresponding occurrence of A in B is negative (positive).*

- *If B is $\neg C$ and A occur positively (negatively) in C then the corresponding occurrence of A in B is negative (positive).*

Example 18. $\forall y.P(y) \to \forall x.P(x)$*: $\forall y$ is a weak quantifier, $\forall x$ is a strong quantifier.*

Note that the strong quantifier introduction rule requires eigenvariable conditions. Below we define the structural Skolemization operator *sk*:

Definition 54 (Skolemization)**.** *sk is a function from closed formulas to closed formulas:*

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

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

$$sk(F) \quad = \quad sk(F_{(Qy)}\{y := f(x_1, \ldots, x_n)\}).$$

where $F_{(Qy)}$ is F after omission of (Qy). $sk(F)$ is called the (structural) Skolemization *of F.*

Definition 55 (Skolemization of sequents)**.** *Let S be the sequent $A_1, \ldots, A_n \vdash B_1, \ldots, B_m$ consisting of closed formulas only and*

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

Then the sequent

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

is called the Skolemization *of S.*

Example 19. *Let S be the sequent $(\forall x)(\exists y)P(x,y) \vdash (\forall x)(\exists y)P(x,y)$. Then the Skolemization of S is S' : $(\forall x)P(x, f(x)) \vdash (\exists y)P(c,y)$ for a one-place function symbol f and a constant symbol c.*

By a *Skolemized proof* we mean a proof of the Skolemized end-sequent. The Skolemized proof can be obtained by Skolemizing end-sequent and then propagating the changes upwards in the proof.

4.3.2 Problems with Skolemization in $\mathcal{L}_\omega$ and $\mathcal{L}^f_\omega$

We would want to obtain similar function sk^ω for $\mathcal{L}_\omega$ and consequently $\mathcal{L}^f_\omega$. If we simply generalize the first-order function *sk* to include higher types, we don't obtain the needed result. First of all, it becomes impossible to Skolemize proofs, as the following simple example shows:

$$\cfrac{\begin{array}{c}\vdots\\ \forall x.P(x,\alpha) \to \forall x.P(x,\alpha) \vdash \forall x^i P(x,\alpha) \to P(s,\beta)\end{array}}{\forall x^{i\to o}.x(\alpha) \to x(\alpha) \vdash \forall x^i P(x,\alpha) \to P(s,\beta)}\ \forall\text{L},\ \mathbf{T} = \lambda y.\forall x P(x,y)$$

The end-sequent is Skolemized, however, the proof is not and cannot be. If one Skolemizes $\forall x.P(x,a) \to \forall x.P(x,a), \Delta \vdash \Gamma$, one ends up with $P(c,a) \to \forall x.P(x,a), \Delta \vdash \Gamma$ and then the $\forall$L cannot be applied as before.

A related feature of Skolemization that can be used in $\mathcal{LK}$ proof transformation is the following:

Proposition 11. *Skolemized $\mathcal{LK}$-proofs do not contain strong quantifier inferences on the ancestors of the end-sequent.*

For $\mathcal{LK}_\omega$ this proposition fails as shown by this simple example:

Example 20.
$$\cfrac{\cfrac{\cfrac{P(\alpha)\ \vdash\ P(\alpha)}{\forall x.P(x)\ \vdash\ P(\alpha)}\ \forall\text{-}l}{\forall x.P(x)\ \vdash\ \forall x.P(x)}\ \exists\text{-}r}{\forall x.P(x)\ \vdash\ \exists Z.Z}\ \exists\text{-}r;\ \mathbf{T} := \lambda y.\forall x.P(x)$$

However, this holds for $\mathcal{LK}_\omega^f$:

Proposition 12. *Skolemized $\mathcal{LK}_\omega^f$-proofs do not contain strong quantifier inferences on the ancestors of the end-sequent.*

Proof. Same as for $\mathcal{LK}$. By induction on the number of inferences in the proof of the Skolemized sequent S. In the case $\exists$R, which was problematic in $\mathcal{LK}_\omega$ the induction goes through since the weak quantifier-introduction can't "hide" the strong quantifier inferences as in the above example.

□

However, sk^ω is not validity-preserving neither in $\mathcal{LK}_\omega^f$ nor in $\mathcal{LK}_\omega$:

Proposition 13. *There is an $A \in \mathcal{L}_\omega^f$ such that $\vdash_{\mathcal{LK}_\omega^f} sk^\omega(A)$, but $\not\vdash_{\mathcal{LK}_\omega^f} A$.*

Proof. In particular, the Axiom of Choice is not provable in $\mathcal{LK}_\omega^f$ (cf. [**?**]). The following instance of Axiom of Choice stated as a functional second-order formula is not provable in $\mathcal{LK}_\omega$ and thus not in $\mathcal{LK}_\omega^f$:

$$\forall x \exists y.x \neq y \to \exists f \forall z.z \neq fz.$$

This can be shown using the Henkin models: consider a general structure $(\{a,b,c\}, I)$ with any I and $\sigma = \emptyset$ with $D^{i\to i}$ consisting of all functions definable by second-order formulae.

We have that $(\{a,b,c\}, I), \sigma \models \forall x \exists y. x \neq y$, however $(\{a,b,c\}, \emptyset) \not\models \exists f \forall z. z \neq fz$. Assume that this holds: then there is a $\mathcal{L}^f_\omega$-definable function F such that $(\{a,b,c\}, I), \{f := F\} \not\models \forall z. z \neq fz$ holds. That is, $(\{a,b,c\}, I) \models \forall z \exists ! y. A[z,y] \wedge z \neq y$. Then $(\{a,b,c\}, I), \{x := a, y := b\} \models A[z,y] \wedge z \neq y$. But also $(\{a,b,c\}, I), \{x := a, y := c\} \models A[z,y] \wedge z \neq y$, which contradicts $(\{a,b,c\}, I), \{x := a, y := b\} \models \forall x \exists ! y A[z,y] \wedge z \neq y$.

However, the Skolemized version of the Axiom of Choice $sk^\omega(AC) = \forall x. x \neq gx \rightarrow \exists f. hf \neq f(hf)$ is provable in $\mathcal{LK}^f_\omega$:

$$\dfrac{\dfrac{hg \neq g(hg) \ \vdash \ hg \neq g(hg)}{\forall x. x \neq gx \ \vdash \ hg \neq g(hg)}\ \forall \mathrm{L}}{\forall x. x \neq gx \ \vdash \ \exists f. hf \neq f(hf)}\ \exists_f \mathrm{R}$$

$\square$

Note that the Skolem function g is "used" in the proof for introducing an existential quantifier on the right-hand-side.

4.3.3 Restricted Skolemization

In order to obtain sound Skolemization in $\mathcal{L}^f_\omega$ we need to restrict Skolem functions. Based on the counter-example above [Miller, 1987] proposes two restrictions and proves that they suffice for obtaining a sound Skolemization procedure for higher-order formulae.

We assume that we are given Skolem function symbols for each type. Moreover, each such symbol comes with an indicated arity – the number of arguments it can take to form a pre-term. Thus we redefine the notion of pre-term:

Definition 56 (Pre-terms of $\mathcal{L}^{f^{sk}}_\omega$)**.** *Define inductively:*

- *All pre-terms of $\mathcal{L}^f_\omega$ are pre-terms of $\mathcal{L}^{f^{sk}}_\omega$;*
- *If f is a Skolem symbol of type $\tau_1 \rightarrow \ldots \rightarrow \tau_n \rightarrow \tau_{n+1}$ and arity n, and $t_1, \ldots, t_n$ are pre-terms of $\mathcal{L}^{f^{sk}}_\omega$, then $f(t_1, \ldots, t_n)$ is a pre-term (of type τ_{n+1})) of $\mathcal{L}^{f^{sk}}_\omega$.*

Example 21. *$f_2^{\sigma_1 \rightarrow (\sigma_2 \rightarrow \sigma_3)}$ is a Skolem function with arity 2 and the type $\sigma_1 \rightarrow (\sigma_2 \rightarrow \sigma_3)$. $f_1^{\sigma_1 \rightarrow (\sigma_2 \rightarrow \sigma_3)}$ is a Skolem function with arity 1 and type $\sigma_1 \rightarrow (\sigma_2 \rightarrow \sigma_3)$. Then we want to define pre-terms in way that $f_2^{\sigma_1 \rightarrow (\sigma_2 \rightarrow \sigma_3)}(t_1, t_2)$ is a pre-term of type σ_3 iff t_1 has type σ_1 and t_2 type σ_2 and $f_1^{\sigma_1 \rightarrow (\sigma_2 \rightarrow \sigma_3)}(t)$ is a pre-term of type $\sigma_2 \rightarrow \sigma_3$ iff t has type σ_1.*

Thus the first restriction is that Skolem function symbols cannot form pre-terms in the same way as usual higher-order constants and variables; they come with necessary arguments. This way, the above proof of the Skolemized Choice cannot go through: g

cannot be used as a pre-term of type $i \to i$ in the $\exists$R-introduction, since it is not a pre-term without a necessary argument of type i.

Of course in $\mathcal{LK}_\omega^f(\mathbf{C})$ and $\mathcal{LK}_\omega$, one could replace g with $\lambda x.gx$ and obtain another proof of the Skolemized Choice. This leads to the second restriction on Skolem pre-terms: no necessary argument of a Skolem function can be bound by lambda abstraction. Same definition of pre-terms for $\mathcal{LK}_\omega^f(\mathbf{C})$ allows to formalize this second requirement:

Definition 57 (Pre-terms of $\mathcal{L}_\omega^{f^{sk}}(\mathbf{C})$)**.** *Define inductively:*

- *All pre-terms of $\mathcal{L}_\omega^f(\mathbf{C})$ are pre-terms of $\mathcal{L}_\omega^{f^{sk}}(\mathbf{C})$;*
- *If f is a Skolem symbol of type $\tau_1 \to \ldots \to \tau_n \to \tau_{n+1}$ and arity n, and $t_1, \ldots, t_n$ are pre-terms of $\mathcal{L}_\omega^{f^{sk}}$, then $f(t_1, \ldots, t_n)$ is a pre-term (of type τ_{n+1}) of $\mathcal{L}_\omega^{f^{sk}}(\mathbf{C})$.*

Now in the definition of $\mathcal{LK}_\omega^f$ and $\mathcal{LK}_\omega^f(\mathbf{C})$ modify the weak quantifier rules: $\mathbf{T}$ is a $\mathcal{L}_\omega^{f^{sk}}$ or a $\mathcal{L}_\omega^{f^{sk}}(\mathbf{C})$ pre-term.

$$\frac{\Gamma, A\{x^\tau := \mathbf{T}\} \vdash \Delta}{\Gamma, \forall x^\tau.A \vdash \Delta}\ \forall\text{L}$$

$$\frac{\Gamma \vdash \Delta, A\{x^\tau := \mathbf{T}\}}{\Gamma \vdash \Delta, \exists x^\tau.A}\ \exists\text{R}$$

where $\mathbf{T}$ is any pre-term of $\mathcal{L}_\omega^{f^{sk}}$ of type τ replaced with x^τ in the lower sequent.

Using the $*$-translation we can provide explanation of the restrictions proposed by Miller from a more general point of view. Consider the example of the Axiom of Choice

$$AC = \forall x \exists y.Pxy \to \exists f \forall x.Pxfx$$

and its $\mathcal{LK}_\omega^f$ proof π again:

$$\frac{\dfrac{hg \neq g(hg) \vdash hg \neq g(hg)}{\forall x.x \neq gx \vdash hg \neq g(hg)}\ \forall\text{L}}{\forall x.x \neq gx \vdash \exists f.hf \neq f(hf)}\ \exists_f\text{R}$$

We also have a corresponding $\mathcal{LK}$ proof of its $*$-translation, by previous results of this section:

$\mathcal{LK}$ proof π^* of S^*:

$$\dfrac{\dfrac{@_{(i\to i)\to i}(h,g) \neq @_{i\to i}(g, @_{(i\to i)\to i}(h,g)) \ \vdash\ @_{(i\to i)\to i}(h,g) \neq @_{i\to i}(f, @_{(i\to i)\to i}(h,g))}{\forall x \neq @_{i\to i}(g,x) \ \vdash\ @_{(i\to i)\to i}(h,g)\neq @_{i\to i}(f,@_{(i\to i)\to i}(h,g))}\ \forall \mathrm{L}}{\forall x \neq @_{i\to i}(g,x)\ \vdash\ \exists f @_{(i\to i)\to i}(h,f) \neq @_{i\to i}(f, @_{(i\to i)\to i}(h,f))}\ \exists \mathrm{R}$$

Now note that although the end-sequent of π^* is the $*$-translation of Skolemized AC it is not a Skolemization of the $*$-translation of AC. That is, $sk(AC^*) \neq sk^\omega(AC)^*$. $sk^\omega(AC)^* = \forall x \neq @_{i\to i}(g,x) \vdash \exists f @_{(i\to i)\to i}(h,f) \neq @_{i\to i}(f, @_{(i\to i)\to i}(h,f))$ is not the first-order Skolemization of AC^*, since it introduces constants, while by definition of sk it is supposed to introduce function symbols.

The $*$-translation of AC looks like this:

$$(\forall x \exists y.Pxy \to \exists f \forall x.Pxfx)^* = \forall x^i \exists y^i.Px^iy^i \to \exists f^{i\to i}\forall x^i.Px^i@_{i\to i}(f,x).$$

Its first-order Skolemization $sk(AC^*)$ is the following:

$$\forall x^i.Px^ig^{sk}y^i \to \exists f^{i\to i}.Ph^{sk}f, @_{i\to i}(f, h^{sk}f)$$

where g^{sk}, h^{sk} are Skolem function symbols.

Note that the resulting formula is not a $*$-translation of any $\mathcal{L}^f_\omega$-formula: neither $g^{sk}y$ nor $h^{sk}f$ are (semi-)terms of $\mathcal{L}^*_1$ by our definition, since we only used terms to be of the form $@_\tau(t_1,\ldots,t_n)$.

Now we cannot "use" g^{sk} to prove this statement as we did before: g^{sk} cannot be used for quantification introduction, since it is not a well-formed pre-term by our definition, only when it is applied to the argument – then it is just a well-formed first-order term. This exactly describes the first restriction to higher-order Skolemization by Miller. Thus we can arrive at the sound higher-order Skolemization procedure by generalizing first-order Skolemization using $*$-translation.

CHAPTER 5

Conclusion

In this thesis we studied sub-logics of higher-order logic in which quantification is restricted to objects of functional type only. We saw that such a restriction gives rise to logics that have a manageable proof-theory and at the same time share some interesting properties with full higher-order logic.

First, we defined syntax and semantics of two variants of such a logic and gave the corresponding cut-elimination proofs, which are easy adaptations of the Gentzen's proof for $\mathcal{LK}$: one just has to incorporate $\beta\eta$-equality in the calculus and then repeat the argument for $\mathcal{LK}$ with equality. Thus, the logic defined this way is proof-theoretically closer to $\mathcal{LK}$ than to $\mathcal{LK}_\omega$. This became obvious by looking at a proof-preserving translation from functional higher-order proofs to first-order proofs defined in Chapter 4. In addition, such translation provides a new perspective on Skolemization which is problematic in higher-order logic. Moreover, we studied semantics of this logic: we showed incompleteness wrt standard semantics and gave a completeness proof wrt general semantics based on Schütte's reduction tree method. This is a simpler and more elegant way of proving this result than a more obvious specialization of a Henkin-style completeness proof for full higher-order logic.

In conclusion, one can say that restricting to functional types allows a better understanding of the sources of logical complexity of higher-order logic, both in proof theory and semantics. By looking at restricted forms of comprehension, it is known that proof-theoretic complexity in higher-order logics comes from impredicative comprehension. Similarly, by looking at the functional fragments we can see that some of the complexity of higher-order logic is due to allowing λ-terms and β-reduction. First, functional higher-order unification problems are already undecidable. Second, as we have seen, adding λ-terms already complicates things semantically, since one cannot extract the counter-model for an unprovable sequent from a reduction tree directly. Moreover, Skolemization has to take a different form: problems with Skolemization start when they are treated as objects of the domain of quantification and not as syntactic tools.

Further work could include:

- Cut-elimination and completeness proof for $\mathcal{LK}^f_\omega(\mathbf{C})$ with β-reduction instead of equality. Intuitively, it should be possible to separate the β-reduction steps from the logical steps in the proof.
- Extending the term language to other term languages, such as system $\mathbf{T}$.
- Since the $\mathcal{LK}^f_\omega$ is so close to $\mathcal{LK}$, it should be easy to adapt other first-order cut-elimination procedures. For instance, one could try adapting the Skolemization-free CERES method for first-order logic to $\mathcal{LK}^f_\omega$ and $\mathcal{LK}^f_\omega(\mathbf{C})$.

Bibliography

[Benzmüller and Miller, 2014] Benzmüller, C. and Miller, D. (2014). Automation of higher-order logic. In *Computational Logic*, pages 215–254.

[Danos et al., 1997] Danos, V., Joinet, J., and Schellinx, H. (1997). A new deconstructive logic: Linear logic. *J. Symb. Log.*, 62(3):755–807.

[Gentzen, 1935] Gentzen, G. (1934–1935). Untersuchungen über das logische Schließen I+II. *Mathematische Zeitschrift*, 39:176–210,405–431.

[Girard et al., 1989] Girard, J., Lafont, Y., and Taylor, P. (1989). *Proofs and Types.* Cambridge University Press.

[Girard, 1991] Girard, J. Y. (1991). Proof theory and logical complexity. *Annals of Pure and Applied Logic*, 53(4):197.

[Hetzl et al., 2011] Hetzl, S., Leitsch, A., and Weller, D. (2011). CERES in higher-order logic. *Ann. Pure Appl. Logic*, 162(12):1001–1034.

[Leitsch, 2015] Leitsch, A. (2015). On proof mining by cut-elimination. In *All about proofs, proofs for all*, volume 55 of *Mathematical Logic and Foundations*, pages 173–200.

[Leivant, 1994] Leivant, D. (1994). Higher order logic. In *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume2, Deduction Methodologies*, pages 229–322.

[Miller, 1992] Miller, D. (1992). Unification under a mixed prefix. *J. Symb. Comput.*, 14(4):321–358.

[Miller, 1987] Miller, D. A. (1987). A compact representation of proofs. *Studia Logica*, 46(4):347–370.

[Prawitz, 1967] Prawitz, D. (1967). Completeness and Hauptsatz for second order logic. *Theoria*, 33(3):246–258.

[Prawitz, 1968] Prawitz, D. (1968). Hauptsatz for higher order logic. *J. Symb. Log.*, 33(3):452–457.

[Tait, 1966] Tait, W. W. (1966). A nonconstructive proof of Gentzen's Hauptsatz for second order predicate logic. *Bull. Amer. Math. Soc.*, 72(6):980–983.

[Takeuti, 1987] Takeuti, G. (1987). *Proof Theory*. Dover Books on Mathematics.

[Van Benthem and Doets, 2001] Van Benthem, J. and Doets, K. (2001). *Higher-Order Logic*, pages 189–243. Springer Netherlands, Dordrecht.