Fachrichtung 6.2 – Informatik
Naturwissenschaftlich-Technische Fakultät I
– Mathematik und Informatik –
Universität des Saarlandes

Studies in Higher-Order Equational Logic

Bachelorarbeit

Angefertigt unter der Leitung von
Prof. Dr. Gert Smolka

Mark Kaminski

Mai 2005

Verfasser:	Mark Kaminski
Erstgutachter:	Prof. Dr. Gert Smolka
Zweitgutachter:	Prof. Bernd Finkbeiner, Ph.D.

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Saarbrücken, den 9. Mai 2005

Mark Kaminski

Abstract

We show that higher-order logic (HOL) can be axiomatized in S, the simply typed λ-calculus with equational deduction. Unlike traditional formulations of HOL, S does not rely on pre-defined semantics of logical constants.

First we show how deduction in traditional HOL can be simulated within S, thus proving S to be a general-purpose higher-order logical system. Afterwards we prove the completeness of S for first-order axioms.

An important task of the thesis is to investigate in how far the usual logical constants and semantic structures can be axiomatized within S. We start by considering Boolean algebras, i.e. systems generated by Boolean axioms and show how they can be axiomatically extended by quantification. We define the identity test and show some important properties of identity in S. We axiomatize in S the usual semantic structure of HOL, thus showing that the semantic expressiveness of S matches that of traditional higher-order formalisms.

Finally we analyze the deductive power of S in more detail and obtain interesting incompleteness results for specific instances of the system.

Acknowledgements

I would like to express my deepest gratitude to Prof. Dr. Gert Smolka for his guidance. Our extensive discussions have made me see a lot of things in a completely new light. His advice on scientific writing has been enormously helpful in improving the readability of this thesis.

Contents

Introduction

Overview

Higher-order logic, also known as type theory, has been introduced in 1908 by Bertrand Russell [33] as a formal basis for mathematical reasoning, based on a functional view of logic originally developed by Gottlob Frege [13]. In its modern form, type theory is based on Alonzo Church's simply typed λ-calculus [8] and the formulations by Leon Henkin [22] and Peter Andrews [4]. Over the years type theory has become an integral part of every subject of study that is in some way concerned with the relationship between computation and logical reasoning. In computer science, higher-order logic has lots of applications, including proof assistant systems like e.g. Isabelle [28] or PVS [29].

Classical formulations of type theory employ rules of inference depending on some dedicated logical constants. Consider, for instance, the well-known rule "Modus ponens", commonly formulated as:

From A *and* $A \to B$ *infer* B.

The rule involves the constant $\to$ and is therefore specific to logical systems where such a constant is built in.

This thesis studies in how far semantic and deductive strengths of higher-order logic can be achieved without building in logical constants and without using custom rules of inference. We consider a simple higher-order system S, which is the simply typed λ-calculus with equational deduction (compare to [6, 44]). In particular, S introduces no logical constants with pre-defined semantics.

We evaluate S with respect to two important properties: deductive power and semantic expressiveness. In both cases we need a reference formalism to which S can be compared. This role will be played by Andrews' higher-order logic (AHOL) as described in his textbook [4]. To keep our considerations more compact, most of the time when talking about AHOL, we will ignore the description operator and the corresponding Axiom of Descriptions, both of which are parts of the full system $\mathcal{Q}_0$ by Andrews. Descriptions are largely independent from most of the other constants and, as it turns out, can be easily axiomatized in S. Andrews' full system will be treated briefly in Chapter 2.

Chapter 1 introduces some basic terms, propositions and notational conventions which will be used by us when we consider S in detail.

After a brief overview of AHOL, in Chapter 2 we show how deduction in Andrews' logic can be simulated in S. We present a set of axioms HOL and prove $\mathsf{S}(HOL)$ having at least the deductive power of AHOL. We also discuss an alternative approach to simulating the deduction in AHOL, namely to introduce an additional rule of inference reflecting the special semantics of the identity constant. We discuss two possible extensions of S that integrate this rule of inference into the initial system.

In Chapter 3 we prove that every term in S can be rewritten to a $\beta\overline{\eta}$-normal form, which is exploited by us in Chapter 4 when we prove a practically useful property of S, namely its completeness for first-order axioms.

In Chapter 5 we explore the semantic expressiveness of S with respect to standard interpretations. Starting with higher-order Boolean algebras, which can easily be axiomatized in S, we axiomatically extend Boolean logic by quantification and study some semantic consequences of this extension. We observe that Boolean algebras satisfying the additional quantifier axioms are complete. Since we can axiomatize quantifiers, we follow the approach used by Russell and Church and define the identity test in terms of universal quantification according to Leibniz' criterion for equality.

Next, we ask ourselves how to axiomatize the semantic structure of AHOL within S. We observe that in order to represent the set $\{0,1\}$ of truth values we first need to exclude from consideration the trivial Boolean algebra. After doing so, we can easily make the interpretations of S isomorphic to those of AHOL with the help of an additional axiom.

Furthermore, we study the expressiveness of the logic we obtain without the restriction of semantic isomorphism to AHOL. We observe that the set $\{0,1\}$ still can be represented as the range of the identity test, which eventually leads us to the conclusion that with respect to semantic expressiveness S is not inferior to AHOL, even if we do not enforce a two-valued interpretation of the truth values. We demonstrate this by providing a finite axiomatization of the natural numbers within S, thus showing the incompleteness of deduction.

In Chapter 6 we investigate some deductive properties of S parameterized by a specific set of axioms $LAx2$ that, just like HOL, was shown in Chapter 5 to be sufficient in order to axiomatize traditional HOL. Comparing $\mathsf{S}(LAx2)$ and AHOL, we discover that, unlike $\mathsf{S}(HOL)$, $\mathsf{S}(LAx2)$ is in fact less powerful than Andrews' logic in so far as deduction is concerned.

Finally we briefly summarize our results and outline several issues that can be addressed within further investigation of S and related systems.

Contributions

This thesis makes the following contributions:

1. Investigation of the semantic expressiveness of S with respect to standard interpretations (Chapter 5).
   (a) Axiomatization of universal and existential quantification in Boolean algebras, based on an axiomatization of (higher-order) Boolean logic in S. Observation and proof that the the quantifier axioms enforce the completeness of underlying Boolean lattices.
   (b) Observation and proof that in a Boolean algebra with quantifiers, the identity test defined with the help of Leibniz' criterion for equality has the range $\{0, 1\}$, independent of whether the algebra contains further Boolean values.
   (c) Axiomatization of the two-valued Boolean algebra $\mathcal{T}_2$. (Resulting system: $\mathsf{S}(LAx2)$.) Observation that the usual semantic structure of HOL can be axiomatized in S if we restrict ourselves to considering non-trivial Boolean algebras.
   (d) Investigation of the semantic expressiveness of general (not necessarily two-valued) Boolean algebras. Encoding of the usual predicate semantics within Boolean logic with quantifiers.
   (e) Finite axiomatization of the natural numbers within Boolean logic with quantifiers. Proof that the semantic closure of finite sets of axioms may be not semi-decidable.
2. Investigation of the deductive power of S and its comparison to deduction in AHOL (Chapters 2, 4, 5 and 6).
   (a) Presentation of the axiom system HOL and proof that $\mathsf{S}(HOL)$ has exactly the deductive power of AHOL.
   (b) Presentation of two alternative systems based on S, which have at least the deductive power of AHOL, with corresponding proofs.
   (c) Proof that S is complete for first-order axioms.
   (d) Proof that the predicate encoding for Boolean logic with quantifiers has no influence on deduction in $\mathcal{T}_2$.
   (e) Observation and proof that, when appropriately instantiated, S allows finite non-standard models.
   (f) Independent proof that Boolean logic with quantifiers (represented by $\mathsf{S}(LAx2)$) admits non-extensional models. (Originally proved by Andrews [2].)
   (g) Observation and proof that $\mathsf{S}(LAx2)$ is deductively strictly less powerful than AHOL.
   (h) Observation and proof that in $\mathsf{S}(LAx2)$ the internal and the external identity are not equivalent with respect to deduction.

Chapter 1

Basics

The definitions below are based on notation and terminology introduced in lecture notes by Gert Smolka [37].

1.1 Types and Terms

Definition Let (TC, VC, ty) be a signature and *Var* the set of all variables. A **context** Γ is a partial function that maps variables to types ($\Gamma \in Var \rightharpoonup Ty(TC)$).

Notation $\Gamma[x := T] \stackrel{\text{def}}{=} \lambda y \in Var.\text{if } y = x \text{ then } T \text{ else } \Gamma y$

Definition Let

$$\begin{array}{rcll} B \in TC & & & \\ x \in Var & & & \\ c \in VC & & & \\ T \in Ty & = & B & \text{base type} \\ & | & T \to T & \text{function type} \end{array}$$

The set of **pre-terms** PT is defined by

$$\begin{array}{rcll} t \in PT & = & x & \text{variable} \\ & | & c & \text{constant} \\ & | & (t\,t) & \text{application} \\ & | & \lambda x : T.t & \text{abstraction} \end{array}$$

Since they have no proper components, variables and constants are called **primitive**. Applications and abstractions are called **compound**. We write $t_1\,t_2\,t_3$ as an abbreviation for $((t_1\,t_2)\,t_3)$. We use infix notation whenever appropriate, e.g. $x \vee y$ for $((\vee x)y)$.

Definition Let Γ be a context. A pre-term t is called a **Γ-term** iff there exists a type T such that $\Gamma \vdash t : T$.

A pre-term t is called a **term** iff there exists a context Γ such that t is a Γ-term.

$$\begin{array}{rcl} Ter^T\,\Gamma & \stackrel{\text{def}}{=} & \{t \in PT \mid \Gamma \vdash t : T\} \\ Ter\,\Gamma & \stackrel{\text{def}}{=} & \bigcup_{T \in Ty} Ter^T\,\Gamma \end{array}$$

Convention When considering terms relative to a signature, we always assume the existence of a **global context** Γ, which is defined together with the particular signature. When it leads to no confusion, we may write

$$\begin{array}{lcl} t : T & \text{for} & \Gamma \vdash t : T \\ x : T & \text{for} & \Gamma x = T \end{array}$$

Notational Convention Unless otherwise stated, index variables like m, n, p etc. are always assumed ≥ 0.

Definition The **order** of a type T ($\mathrm{ord}(T)$) is defined as follows:

$$\begin{array}{rcl} \mathrm{ord}(B) & = & 1 \\ \mathrm{ord}(T_1 \to T_2) & = & \max\{\mathrm{ord}(T_1) + 1, \mathrm{ord}(T_2)\} \end{array}$$

Remark $\mathrm{ord}(T_1 \to \ldots \to T_n \to B) \;=\; \max\{\mathrm{ord}(T_i) \mid 1 \leq i \leq n\} + 1$

Definition Let the function ran be defined as follows:

$$\begin{array}{rcl} \mathrm{ran}(B) & = & B \\ \mathrm{ran}(T_1 \to T_2) & = & \mathrm{ran}(T_2) \end{array}$$

For a term t with $\Gamma \vdash t : T$, let $\mathrm{ran}\, t = \mathrm{ran}(T)$.

Remark $\mathrm{ran}(T_1 \to \ldots \to T_n \to B) \;=\; B$

$$\textbf{Ref}\ \frac{}{t = t} \qquad \textbf{Sym}\ \frac{s = t}{t = s} \qquad \textbf{Trans}\ \frac{s = s' \quad s' = t}{s = t}$$

$$\textbf{Rep}\ \frac{s' = t'}{t[s'] = t[t']} \qquad \textbf{Sub}\ \frac{s = s'}{s[x := t] = s'[x := t]}$$

$$\boldsymbol{\beta}\ \frac{}{(\lambda x : T.t)y = t[x := y]} \qquad \boldsymbol{\eta}\ \frac{}{\lambda x : T.fx = f}$$

Figure 1.1: Equality rules

1.2 Deduction

Definition The rules of inference in S are defined as shown in Figure 1.1.

Notation *Given a set of equations* A *we write* $[t]_A$ *for* $\{s|A \vdash s = t\}$.

Proposition 1.1 *Let* $s, t : T \to T'$, $x : T$. *If* $x \notin FV\, s \cup FV\, t$ *then*

$$sx = tx \ \vdash\ s = t$$

Proof

$$\begin{array}{rll} sx = tx & \vdash\ \lambda x : T.sx = \lambda x : T.tx & \\ & \vdash\ s = t & \eta \end{array}$$

□

1.3 Logical Axioms

In the following, we consider logical systems parameterized with different sets of axioms. By using the notation $L(A)$ we refer to some system L parameterized with the axioms in A. We call A a **parameter** of L.

When A is used to parameterize a logical system, its elements are called (equational) **axioms**.

Definition (Standard Model/Standard Interpretation) Given a signature (TC, VC, ty), a **standard interpretation** $\mathcal{D}$, also called a **standard model**, is a function with the following properties:

1. $\mathcal{D}$ provides denotations for type and value constants:
   $TC \cup VC \subseteq Dom\, \mathcal{D}$
2. Type constants are mapped onto non-empty sets:
   $\forall B \in TC : \mathcal{D}B \neq \varnothing$

3. Function types are mapped onto the corresponding functional spaces: $\mathcal{D}(T_1 \to T_2) = \mathcal{D}T_1 \to \mathcal{D}T_2$
4. Value constants of type T are mapped onto elements of $\mathcal{D}T$: $\forall c \in VC : \mathcal{D}c \in \mathcal{D}(ty\, c)$
5. On the set of pre-terms $\mathcal{D}$ is defined recursively as follows:

$$\begin{array}{rcll} \mathcal{D}c\sigma & = & \mathcal{D}c & \\ \mathcal{D}x\sigma & = & \sigma x & \text{if } x \in Dom\,\sigma \\ \mathcal{D}(st)\sigma & = & \mathcal{D}s\sigma(\mathcal{D}t\sigma) & \text{if } \mathcal{D}t\sigma \in Dom(\mathcal{D}s\sigma) \\ \mathcal{D}(\lambda x : T.t)\sigma & = & \lambda v \in \mathcal{D}T.\mathcal{D}t(\sigma[x := v]) & \end{array}$$

Convention Until we consider non-standard interpretations for the first time in Chapter 6, when talking about interpretations we always mean standard interpretations.

Definition Let (TC, VC, ty) be a signature and $\mathcal{D}, \mathcal{E}$ be interpretations. $\mathcal{D}$ is **isomorphic** to $\mathcal{E}$ ($\mathcal{D} \cong \mathcal{E}$) iff there exists a family of bijections indexed by types

$$\phi_T : \mathcal{D}T \to \mathcal{E}T$$

such that $\phi_{ty\,c}(\mathcal{D}c) = \mathcal{E}c$ for all $c \in VC$.

Definition 1.1 Given a signature (TC, VC, ty) such that

- $0, 1, \neg, \wedge, \vee \in VC$; $\mathsf{B} \in TC$
- it holds

$$\begin{array}{rl} 0, 1 : & \mathsf{B} \\ \neg : & \mathsf{B} \to \mathsf{B} \\ \wedge, \vee : & \mathsf{B} \to \mathsf{B} \to \mathsf{B} \end{array}$$

we define the **Boolean axioms** BAx as depicted in Figure 1.2.

An interpretation $\mathcal{D}$ is called a **Boolean algebra** iff $\mathcal{D} \models BAx$.

Definition Given two Boolean algebras $\mathcal{D}$ and $\mathcal{E}$, $\mathcal{E}$ is called a **subalgebra** of $\mathcal{D}$ if

1. $\mathcal{E}\mathsf{B} \subseteq \mathcal{D}\mathsf{B}$
2. $\mathcal{E}\neg \subseteq \mathcal{D}\neg,\ \mathcal{E}\wedge \subseteq \mathcal{D}\wedge,\ \mathcal{E}\vee \subseteq \mathcal{D}\vee$

Definition 1.2 Assume a signature like in Definition 1.1, with the following additional constraints for every type T:

- $\forall_T \in VC$
- $\forall_T : (T \to \mathsf{B}) \to \mathsf{B}$

Let the set QAx of **quantifier axioms** consist of two axioms for every type T, as defined in Figure 1.2.

We define the set LAx of **logical axioms** to be $BAx \cup QAx$.

Boolean Axioms (*BAx*)
for distinct variables $x, y, z : \mathsf{B}$:

$$\begin{aligned} x \wedge y &= y \wedge x & x \vee y &= y \vee x \\ x \wedge (y \vee z) &= (x \wedge y) \vee (x \wedge z) & x \vee (y \wedge z) &= (x \vee y) \wedge (x \vee z) \\ x \wedge \neg x &= 0 & x \vee \neg x &= 1 \\ x \wedge 1 &= x & x \vee 0 &= x \end{aligned}$$

Quantifier Axioms (*QAx*)
for every type T, distinct $x, u : T$ and $f : T \to \mathsf{B}$:

$$\begin{aligned} \forall_T f &= \forall_T f \wedge fx && (\forall I_T) \\ \forall_T (\lambda x : T.fx \vee u) &= \forall_T f \vee u && (\forall\vee_T) \end{aligned}$$

Figure 1.2: Logical axioms

Notational Convention

- We will omit type annotations from quantifiers when it leads to no confusion.
- All constants are assumed left associative.
- We assume the usual precedence conventions for Boolean constants.
- We introduce the following abbreviations:

$$\begin{aligned} x \to y &\stackrel{\text{def}}{=} \neg x \vee y \\ x \leftrightarrow y &\stackrel{\text{def}}{=} (x \to y) \wedge (y \to x) \\ \forall x.t &\stackrel{\text{def}}{=} \forall(\lambda x : T.t) && \text{if } \Gamma x = T \\ \exists x.t &\stackrel{\text{def}}{=} \neg \forall x.\neg t \end{aligned}$$

Proposition 1.2 ($\forall E$) $LAx \vdash \forall x.u = u$

Proof

$$\begin{aligned} \forall x.u &= \forall x.0 \vee u && BAx \\ &= \forall x.(\lambda x : \mathsf{B}.0)x \vee u && \eta \\ &= \forall(\lambda x : \mathsf{B}.0) \vee u && \forall\vee \\ &= \forall(\lambda x : \mathsf{B}.0) \wedge (\lambda x : \mathsf{B}.0)0 \vee u && \forall I \\ &= \forall(\lambda x : \mathsf{B}.0) \wedge 0 \vee u && \beta \\ &= u && BAx \end{aligned}$$

□

Proposition 1.3 ($\forall\wedge$) $LAx \vdash \forall x.fx \wedge u = \forall f \wedge u$

$$
\begin{array}{rcll}
\forall f & = & \forall f \wedge fx & (\forall I) \\
\forall x.fx \vee u & = & \forall f \vee u & (\forall\vee) \\
\forall x.u & = & u & (\forall E) \\
\forall x.fx \wedge u & = & \forall f \wedge u & (\forall\wedge)
\end{array}
\qquad
\begin{array}{rcll}
\exists f & = & \exists f \vee fx & (\exists I) \\
\exists x.fx \wedge u & = & \exists f \wedge u & (\exists\wedge) \\
\exists x.u & = & u & (\exists E) \\
\exists x.fx \vee u & = & \exists f \vee u & (\exists\vee)
\end{array}
$$

Figure 1.3: Quantifier Theorems

Proof

$$
\begin{array}{rcll}
\forall x.fx \wedge u & = & (\forall x.fx \wedge u) \vee 0 & BAx \\
& = & (\forall x.fx \wedge u) \vee (u \wedge \neg u) & BAx \\
& = & ((\forall x.fx \wedge u) \vee u) \wedge ((\forall x.fx \wedge u) \vee \neg u) & BAx \\
& = & ((\forall x.fx \wedge u) \vee u) \wedge (\forall x.fx \wedge u \vee \neg u) & \forall\vee \\
& = & ((\forall x.fx \wedge u) \vee u) \wedge (\forall x.fx \vee \neg u) & BAx \\
& = & ((\forall x.fx \wedge u) \vee u) \wedge (\forall f \vee \neg u) & \forall\vee \\
& = & \forall f \wedge (\forall x.fx \wedge u) \vee \forall f \wedge u & BAx \\
& = & \forall f \wedge ((\forall x.fx \wedge u) \vee u) & BAx \\
& = & \forall f \wedge \forall x.fx \wedge u \vee u & \forall\vee \\
& = & \forall f \wedge \forall x.u & BAx \\
& = & \forall f \wedge u & \forall E
\end{array}
$$

□

Proposition 1.4 *$DC(LAx)$ contains the equations in Figure 1.3.*

Proof So far, we have proved the theorems for $\forall$. We proceed by straightforward application of BAx and the definition of $\exists$. $\exists I$ can be deduced from $\forall I$, $\exists E$ from $\forall E$, $\exists\wedge$ from $\forall\vee$ and $\exists\vee$ from $\forall\wedge$. □

Chapter 2

$\mathsf{S}(HOL)$

Unlike in AHOL, deduction in S relies entirely on the equality rules, without committing to any constants. In this chapter we show how deduction in AHOL can be simulated in S. We present a set of axioms HOL and show that this set suffices to derive all the axioms of AHOL as well as to simulate Andrews' only rule of inference **R**. By doing so, we prove $\mathsf{S}(HOL)$ being a general-purpose higher-order logical system with at least the deductive power of AHOL.

Afterwards, we discuss some alternatives to $\mathsf{S}(HOL)$, which extend the equality rules by an additional rule **Id**. Although **Id** does not share the general nature of the equality rules, it allows us to achieve the deductive power of $\mathsf{S}(HOL)$ with a reduced set of axioms.

Finally we consider Andrews' system with the Axiom of Descriptions and propose an adequate axiomatization of this system in S.

2.1 AHOL

Before we compare the deductive power of S with that of AHOL, we should learn a little bit more about the latter system. First, let us consider Andrews' axioms in so far as they are relevant to our system. We formulate them in our formalism as axiom schemata relatively to arbitrary types T and T' such that

$$\begin{aligned} p &: \mathsf{B} \to \mathsf{B} \\ q &: T \to \mathsf{B} \\ x, y &: T \\ f, g &: T \to T' \\ \doteq_T &: T \to T \to \mathsf{B} \end{aligned}$$

We assume the usual typing for the Boolean and the quantifier constants, which is given in Definition 1.2. The value constant $\doteq_T$ is assumed to denote the identity test on $\mathcal{D}T$. The identity test is known to be sufficient in order to define all constants of traditional higher-order logic apart from the description operator. Boolean constants and quantifiers can be seen as abbreviations of terms where the only constants being used are those denoting identity relations on different type domains. Henkin [22, 23] was the first to use identity as the only logical primitive. Andrews' definition of higher-order logic follows Henkin's idea. In AHOL identity is introduced as a family of logical constants.

When using $\doteq$ we assume the identity test to take precedence over the Boolean operators. As usual, we omit type annotations when it leads to no confusion.

The set AAx (to stand for "Andrews' axioms") looks as follows:

$$
\begin{array}{rcll}
(p1 \wedge p0) \doteq \forall p & = & 1 & (A1) \\
x \doteq y \rightarrow qx \doteq qy & = & 1 & (A2) \\
(f \doteq g) \doteq (\forall x.fx \doteq gx) & = & 1 & (A3)
\end{array}
$$

$A1$ expresses the idea that there are only two truth values. $A2$ reflects a fundamental congruence property of identity. $A3$ formulates the principle of extensionality.

Andrews' only rule of inference **R** can be stated as follows:

$$
\frac{s' \doteq t' = 1 \qquad t[s'] = 1}{t[t'] = 1}
$$

Note how the correctness of this formulation of **R** depends on the semantics of $\doteq$ and 1. The key insight needed to simulate AHOL in S is to understand how this relation can be expressed in terms of equational axioms. This is what we do next.

2.2 *HOL* and its Deductive Closure

We use the following notation:

$$
\begin{array}{rcl}
\forall FV\, t'.t & \stackrel{\text{def}}{=} & \forall x_1.\ldots\forall x_n.t \ \text{ where } \ \{x_1, \ldots, x_n\} = FV\, t' \\
\forall FV.t & \stackrel{\text{def}}{=} & \forall FV\, t.t
\end{array}
$$

We introduce HOL as an extension of BAx additionally containing the following axioms schemata:

$$
\begin{array}{rcll}
x \doteq x & = & 1 & (\mathit{Ref}) \\
\forall_T q & = & q \doteq (\lambda x : T.1) & (D\forall) \\
p1 \wedge p0 & = & \forall p & (\mathit{Bin}) \\
f \doteq g & = & \forall x.fx \doteq gx & (\mathit{Ext}) \\
(\forall FV.s' \doteq t') \wedge t[s'] & = & (\forall FV.s' \doteq t') \wedge t[t'] & (\mathit{Rep}) \\
x \doteq y \wedge qx & = & x \doteq y \wedge qy & (\mathit{Rep}')
\end{array}
$$

The schemata are defined for all types T, T' such that:

- for all terms t, s', t' it holds

$$\begin{aligned} t &: \mathsf{B} \\ s', t' &: T \end{aligned}$$

- for variables x, y, p, q, f, g it holds

$$\begin{aligned} x, y &: T \\ p &: \mathsf{B} \to \mathsf{B} \\ q &: T \to \mathsf{B} \\ f, g &: T \to T' \end{aligned}$$

- $\doteq_T : T \to T \to \mathsf{B}$

Ref formalizes the reflexivity of the identity test. $D\forall$ defines the universal quantifier in terms of identity. *Bin* and *Ext* are obvious adaptations of $A1$ and $A3$ respectively. *Rep* and *Rep'* express the intended semantics of $\doteq$ with respect to replacement.

In order to prove that $\mathsf{S}(HOL)$ has the deductive power of AHOL, we have to derive AAx from HOL. Furthermore, we must show that $\mathsf{S}(HOL)$ can simulate Andrews' rule of deduction **R**. But first we need to prove some auxiliary statements.

2.2.1 Logical Axioms

We show that LAx can be derived from HOL. (actually, even from a smaller set of axioms – we need neither *Ext* nor *Rep*).

Lemma 2.1 $BAx, Ref, Rep' \vdash x \doteq y = x \doteq y \wedge fx \doteq fy$

Proof

$$\begin{aligned} x \doteq y &= x \doteq y \wedge 1 && BAx \\ &= x \doteq y \wedge fx \doteq fx && Ref \\ &= x \doteq y \wedge (\lambda y : T.fx \doteq fy)x && \beta \\ &= x \doteq y \wedge (\lambda y : T.fx \doteq fy)y && Rep' \\ &= x \doteq y \wedge fx \doteq fy && \beta \end{aligned}$$

□

Lemma 2.2 $BAx, Ref, D\forall, Rep' \vdash \forall f = \forall f \wedge \forall x.fx \doteq 1$

Proof

$$\begin{aligned} \forall f &= f \doteq \lambda x : T.1 && D\forall \\ &= f \doteq (\lambda x : T.1) \wedge fx \doteq 1 && \text{Lem. 2.1 with } \lambda f : T \to \mathsf{B}.fx \\ &= \forall f \wedge fx \doteq 1 && D\forall \end{aligned}$$

□

Proposition 2.3 $BAx, Ref, D\forall, Rep' \vdash \forall I$

Proof

$$\begin{array}{rcll} \forall f &=& \forall f \wedge fx \doteq 1 & \text{by Lemma 2.2} \\ &=& \forall f \wedge fx \doteq 1 \wedge 1 & BAx \\ &=& \forall f \wedge fx \doteq 1 \wedge fx & Rep' \text{ with } \lambda x : B.x \\ &=& \forall f \wedge fx & \text{by Lemma 2.2} \end{array}$$

□

Proposition 2.4 $BAx, Bin \vdash \forall\vee$

Proof

$$\begin{array}{rcll} \forall x.fx \vee u &=& (f0 \vee u) \wedge (f1 \vee u) & Bin \\ &=& f0 \wedge f1 \vee u & BAx \\ &=& \forall f \vee u & Bin \end{array}$$

□

Corollary 2.5 $BAx, Ref, D\forall, Bin, Rep' \vdash LAx$

2.2.2 Andrews' Axioms

First, we derive $A2$ as follows:

Proposition 2.6 $BAx, Ref, Rep' \vdash A2$

Proof

$$\begin{array}{rcll} x \doteq y \rightarrow qx \doteq qy &=& \neg(x \doteq y \wedge \neg(qx \doteq qy)) & BAx \\ &=& \neg(x \doteq y \wedge \neg(qx \doteq qx)) & Rep' \\ &=& \neg(x \doteq y \wedge \neg 1) & Ref \\ &=& 1 & BAx \end{array}$$

□

To derive $A1$ and $A3$, we observe a notable deductive property of $\doteq$:

Proposition 2.7 *For all terms* s, t : $LAx, s = t \vdash s \doteq t = 1$

Proof

$$\begin{array}{rcll} s \doteq t &=& \forall f.fs \rightarrow ft & \text{def } \doteq \\ &=& \forall f.ft \rightarrow ft & s = t \\ &=& \forall f.1 & BAx \\ &=& 1 & \forall E \end{array}$$

□

Remark The opposite direction

$$LAx, s \doteq t = 1 \vdash s = t$$

does not hold, which follows from a stronger claim we prove later (Theorem 11).

Corollary 2.8 $BAx, Ref, D\forall, Bin, Rep', s = t \vdash s \doteq t = 1$

Corollary 2.9 $BAx, Ref, D\forall, Bin, Rep' \vdash A1$

Corollary 2.10 $BAx, Ref, D\forall, Bin, Ext, Rep' \vdash A3$

2.2.3 Conclusion

The axiom schema *Rep* seems to be crucial if we want to simulate **R**, since **R** assumes the same kind of semantic relation between replacement and internal identity as it is expressed by the axiom. Indeed, once we have *Rep*, we can easily express deduction based on **R** using the equality rules:

Lemma 2.11 $LAx, t = 1 \vdash \forall FV.t = 1$

Proof

$$\begin{array}{rcll} \forall FV.t & = & \forall FV\, t.1 & t = 1 \\ & = & 1 & \forall E \end{array}$$

□

Proposition 2.12 $\mathsf{S}(BAx, Rep)$ *can simulate deduction based on* **R**.

Proof We show $BAx, HOL - \{Ext\}, s' \doteq t' = 1, t[s'] = 1 \vdash t[t'] = 1$.

$$\begin{array}{rcll} t[t'] & = & 1 \wedge t[t'] & BAx \\ & = & (\forall FV.s' \doteq t') \wedge t[t'] & \text{by Lemma 2.11} \\ & = & (\forall FV.s' \doteq t') \wedge t[s'] & Rep \\ & = & \forall FV.s' \doteq t' & t[s'] = 1, BAx \\ & = & 1 & \text{by Lemma 2.11} \end{array}$$

□

From what we have seen so far, we can say:

Theorem 1 $\mathsf{S}(HOL)$ *has exactly the deductive power of* AHOL.

Proof By Proposition 2.12, 2.6, Corollary 2.9 and 2.10, $\mathsf{S}(HOL)$ has at least the deductive power of AHOL.

By Andrews' [4] Propositions 5200–5232, AHOL (with a number of further axioms specifying basic properties of β-reduction) has at least the deductive power of $\mathsf{S}(HOL)$. □

2.3 Alternatives

We have seen how deduction in AHOL can be simulated using *Rep*. Observe that unlike the rest of *HOL*, this axiom schema has infinitely many instances for every type. In this section we present two alternative systems based on S that have at least the deductive power of AHOL without making use of *Rep* or of *Rep'*.

Both systems extend the equality rules by the following rule of inference:

$$\textbf{Id}\ \frac{s \doteq t = 1}{s = t}$$

Obviously, **Id** is consistent with the usual semantics of the identity test.

Now we can easily simulate the rule **R**:

Proposition 2.13 *Deduction using* **R** *can be simulated by deduction using the equality rules and* **Id**.

Proof

$$\textbf{Trans}\ \dfrac{\textbf{Rep}\ \dfrac{\textbf{Id}\ \dfrac{s' \doteq t' = 1}{s' = t'}}{\textbf{Sym}\ \dfrac{t[s'] = t[t']}{t[t'] = t[s']}} \qquad t[s'] = 1}{t[t'] = 1}$$

□

Remark **Id** differs from the equality rules in an important aspect. The rule contains the derived constant $\doteq$, which reflects the special status of the corresponding constant in AHOL. Therefore, **Id** cannot be generally considered sound or useful and does not fit well into general higher-order equational logic. A possible way to avoid this inconsistence is strengthening the expressiveness of axioms by admitting **conditional equations**, i.e. equations of the following form:

$$E_1, \ldots, E_n \Rightarrow E$$

A conditional equation is to hold for an assignment σ if either of the following two statements is true:

1. E holds for σ.
2. There exists $i \in \{1, \ldots, n\}$ such that E_i does not hold for σ.

Note that an ordinary (unconditional) equation is just a conditional one with $n = 0$.

The two systems we want to present, integrate **Id** into S in two different ways.

2.3.1 $\mathsf{S}^{\doteq}$

In the first approach, we define the usual constants in terms of $\doteq$, in the same way as it is done in AHOL. We do not have to define the semantics of $\doteq$ explicitly. Partly, this task is accomplished implicitly by *Id*, the conditional axiom corresponding to **Id**. Further relevant properties of $\doteq$ can be easily axiomatized with the help of appropriate parameters. Let us call the resulting system $\mathsf{S}^{\doteq}$.

Since by Proposition 2.13, $\mathsf{S}^{\doteq}$ can simulate **R**, when parameterized with either AAx or $HOL - \{Rep, Rep'\}$, the system clearly has at least the deductive power of AHOL.

Certainly, $\mathsf{S}^{\doteq}$ is much closer to AHOL than S. Nevertheless, $\mathsf{S}^{\doteq}$ and S share two important properties:

1. Neither S nor $\mathsf{S}^{\doteq}$ rely on any built-in semantics of the identity predicate. S specifies $\doteq$ with the help of LAx, whereas in $\mathsf{S}^{\doteq}$ relevant properties of identity are encoded in the rule **Id** and in $A2$. On the contrary, AHOL provides an informal description of the identity test and explicitly requires its identity constant to satisfy this specification.
2. Neither in S nor in $\mathsf{S}^{\doteq}$ do we explicitly require $\mathcal{D}\mathsf{B}$ to be two-valued since this restriction can be easily axiomatized whenever needed.

2.3.2 $\mathsf{S}^{\mathbf{Id}}$

Another possibility to increase the deductive power of S is integrating the rule **Id** into a system, where the identity test is defined as a derived operation:

$$\doteq_T \stackrel{\text{def}}{=} \lambda x : T.\lambda y : T.\forall_{T\to\mathsf{B}} f.fx \to fy$$

In this case, **Id** must be interpreted as a notational abbreviation of the following rule:

$$\frac{\forall f.\neg fs \vee ft = 1}{s = t}$$

We call the resulting system S^{Id}. Since **Id** uses Boolean operators and quantifiers, we should ensure that the corresponding constants are properly defined by requiring our system to satisfy LAx. Therefore, we will only consider systems $\mathsf{S}^{\mathrm{Id}}(A)$ with $A \supseteq LAx$.

Remark Observe that **Id** implicitly constrains the semantics of $\neg$, $\vee$ and $\forall$. We may ask ourselves in how far this semantics is consistent with the axiomatic definition of the constants. We can show the compatibility of **Id** with LAx by proving the rule correct with respect to interpretations satisfying LAx. We will be able to do so in Chapter 5 (Proposition 5.7).

We can show that $A2$ can be derived from the definition of $\doteq$:

Lemma 2.14 $LAx \vdash x \doteq y \to qx \doteq qy = 1$

Proof

$$\begin{array}{rll}
& x \doteq y \to qx \doteq qy & \\
= & (\neg\forall f.\neg fx \vee fy) \vee \forall g.\neg g(qx) \vee g(qy) & \text{def} \doteq, \to \\
= & (\exists f.fx \wedge \neg fy) \vee \forall g.\neg g(qx) \vee g(qy) & \text{def} \exists,\ BAx \\
= & \forall g.(\exists f.fx \wedge \neg fy) \vee \neg g(qx) \vee g(qy) & \forall\vee \\
= & \forall g.(\exists f.fx \wedge \neg fy) \vee \neg g(qx) \vee g(qy) & \\
& \qquad \vee\, g(qx) \wedge \neg g(qy) & \exists I \\
= & \forall g.1 & BAx \\
= & 1 & \forall E
\end{array}$$

□

By our previous results, we conclude:

Theorem 2 $\mathsf{S}^{\mathrm{Id}}(LAx \cup \{Bin, Ext\})$ *has at least the deductive power of* AHOL.

Proof By Proposition 2.7, $LAx, Bin, Ext \vdash A1, A3$. By Lemma 2.14, $A2$ can be inferred from LAx. Deduction in AHOL can be simulated by Proposition 2.13. □

2.4 Descriptions

So far, we have considered Andrews' higher-order logic without the description operator. However, we can easily extend our system by a family of constants ι_C and a corresponding axiom schema:

$$\iota_C(\lambda x : C.y \doteq x) \;=\; y \qquad (Des)$$

According to Andrews, we need an instance of Des with $x, y : C$ for every base type $C \neq \mathsf{B}$. In $\mathsf{S}^{\doteq}(AAx)$ or in $\mathsf{S}^{\mathrm{Id}}(LAx \cup \{Bin, Ext\})$, Des can be proved deductively equivalent to Andrews' Axiom of Descriptions:

$$\iota_C(\doteq y) \doteq y \;=\; 1 \qquad (A5)$$

In $\mathsf{S}(HOL)$, by Proposition 2.7, $A5$ is still a deductive consequence of Des. We have not examined whether the reverse direction holds as well.

We conclude that S and its derivates can be extended by descriptions without further difficulties.

Chapter 3

Long Normal Forms

In the following chapters we will make use of a special normal form for terms that results from a combination of β-reduction and (restricted) η-expansion. In accordance with Terese [44], we call this form $\beta\overline{\eta}$-normal.

Combinations of β-reduction and η-conversion have been studied by several authors (see bibliographic remarks at the end of the chapter). Nevertheless, in the following we will provide a mostly self-contained proof showing that every term can be rewritten to a $\beta\overline{\eta}$-normal form by some sequence of $\alpha\beta\eta$-conversions. Unlike most of the related results from other sources, it applies directly to our formalism.

Definition 3.1 We define the long η-normal form ($\overline{\eta}$-normal form) recursively as follows:
Let $\Gamma \vdash t : T_1 \to \ldots \to T_n \to B$. If

$$t = \lambda x_1 : T_1. \ldots .\lambda x_n : T_n.t_0 t_1 \ldots t_m$$

and

1. $t_1, \ldots, t_m$ are $\overline{\eta}$-normal terms
2. t_0 is either primitive or a $\overline{\eta}$-normal abstraction

then t is $\overline{\eta}$-normal.

Definition 3.2 A term t is $\beta\overline{\eta}$-normal iff t is $\overline{\eta}$-normal and β-normal.

Remark For every $\beta\overline{\eta}$-normal term t of type $T_1 \to \ldots \to T_n \to B$ there exist

1. $m \geq 0$
2. a primitive term t_0
3. $\beta\overline{\eta}$-normal terms $t_1, \ldots, t_m$

such that

$$t = \lambda x_1 : T_1. \ldots .\lambda x_n : T_n.t_0 t_1 \ldots t_m$$

Proposition 3.1 *Given a term*

$$s = \lambda x_1 : T_1. \ldots .\lambda x_m : T_m.s'$$

such that $\Gamma \vdash s : T_1 \to \ldots \to T_n \to B$ *where* $0 \le m \le n$*, there exists a term*

$$t = \lambda x_1 : T_1. \ldots .\lambda x_n : T_n.s' x_{m+1} \ldots x_n$$

such that $\vdash s = t$*.*

Proof We choose some variables $x_{m+1}, \ldots, x_n \notin FV\ s'$ such that

$$\Gamma x_{m+1} = T_{m+1}, \ldots, \Gamma x_n = T_n$$

and obtain t by repeated application of η-expansion on s. □

Lemma 3.2 *Given a primitive term* s*, there exists a* $\beta\overline{\eta}$*-normal term* t *such that* $\vdash s = t$*.*

Proof Let $\Gamma \vdash s : T$. By induction on the order of T:

1. $\text{ord}(T) = 1 \quad (T = B)$: We are done since s is already $\beta\overline{\eta}$-normal.
2. $\text{ord}(T) = k \quad (T = T_1 \to \ldots \to T_n \to B)$:
   We notice that $\max\{\text{ord}(T_1), \ldots, \text{ord}(T_n)\} < k$. Let $x, x_1, \ldots, x_n$ be different. Then

$$\begin{aligned} \vdash s &= \lambda x_1 : T_1. \ldots .\lambda x_n : T_n.s x_1 \ldots x_n && \eta^n \\ &= \lambda x_1 : T_1. \ldots .\lambda x_n : T_n.s t_1 \ldots t_n && \text{by induction hypothesis} \\ & && t_1, \ldots, t_n\ \beta\overline{\eta}\text{-normal} \end{aligned}$$

□

Lemma 3.3 *For every term* s *there exists an* $\overline{\eta}$*-normal term* t *such that* $\vdash s = t$*.*

Proof By induction on the structure of s: Let $\Gamma \vdash s : T_1 \to \ldots \to T_n \to B$.

1. $s = x$ or $s = c$: The claim holds by Lemma 3.2.
2. $s = (s_1 s_2)$: By Proposition 3.1

$$\vdash s_1 s_2 = \lambda x_1 : T_1. \ldots .\lambda x_n : T_n.s_1 s_2 x_1 \ldots x_n$$

By induction hypothesis

$$\begin{aligned} &\vdash \quad s_1 = t_1 && \text{where } t_1\ \overline{\eta}\text{-normal} \\ &\vdash \quad s_2 = t_2 && \text{where } t_2\ \overline{\eta}\text{-normal} \\ &\vdash \quad x_1 = t_{x,1} && \text{where } t_{x,1}\ \overline{\eta}\text{-normal} \\ & \qquad\quad \vdots && \\ &\vdash \quad x_n = t_{x,n} && \text{where } t_{x,n}\ \overline{\eta}\text{-normal} \\ \Longrightarrow &\vdash s_1 s_2 = \lambda x_1 : T_1. \ldots .\lambda x_n : T_n.t_1 t_2 t_{x,1} \ldots t_{x,n} && \end{aligned}$$

3. $s = \lambda x_1 : T_1.s'$: Obviously $\Gamma[x_1 := T_1] \vdash s' : T_2 \to \ldots \to T_n \to B$. By induction hypothesis

$$\begin{array}{rl} & \vdash s' = \lambda x_2 : T_2. \ldots .\lambda x_n : T_n.t' \\ & \qquad \text{where } \lambda x_2 : T_2. \ldots .\lambda x_n : T_n.t' \ \overline{\eta}\text{-normal,} \\ & \qquad\qquad \text{w.l.o.g. } x_1, \ldots, x_n \text{ different} \\ \Longleftrightarrow & \vdash s = \lambda x_1 : T_1. \ldots .\lambda x_n : T_n.t' \end{array}$$

□

Lemma 3.4 *If $\Gamma \vdash \lambda x : T_1.s' : T_1 \to T_2$, $\Gamma \vdash t : T_1$ and both s and t are $\overline{\eta}$-normal, then $s'[x := t]$ is a $\overline{\eta}$-normal term.*

Proof By induction on the structure of s':

1. $s' = c$ or $s' = y \neq x$: Obviously, s' is $\overline{\eta}$-normal. So is $s'[x := t]$ since $s'[x := t] = s'$.
2. $s' = x$: $s'[x := t] = x[x := t] = t$
3. $s' = s_0 s_1 \ldots s_n$ where s_0 is either primitive or $\overline{\eta}$-normal and $s_1 \ldots s_n$ are $\overline{\eta}$-normal:
   (a) $s_0 = c$ or $s_0 = y \neq x$:

   $$\begin{aligned} s'[x := t] &= s_0[x := t](s_1[x := t]) \ldots (s_n[x := t]) \\ &= s_0(s_1[x := t]) \ldots (s_n[x := t]) \end{aligned}$$

   and our claim holds by induction.

   (b) $s_0 = x$:

   $$\begin{aligned} s'[x := t] &= s_0[x := t](s_1[x := t]) \ldots (x_n[x := t]) \\ &= x[x := t](s_1[x := t]) \ldots (s_n[x := t]) \\ &= t(s_1[x := t]) \ldots (s_n[x := t]) \end{aligned}$$

   and our claim holds by induction.

   (c) s_0 $\overline{\eta}$-normal:

   $$s'[x := t] = s_0[x := t](s_1[x := t]) \ldots (s_n[x := t])$$

   and our claim holds by induction.
4. $s' = \lambda y : T.s''$, w.l.o.g. $x \neq y$:

   $$s'[x := t] = \lambda y : T.s''[x := t]$$

   and our claim holds by induction. □

Lemma 3.5 *If s is a $\overline{\eta}$-normal term, then every term t with $s \to_\beta t$ is again $\overline{\eta}$-normal.*

Proof By induction on the structure of s: Let

$$s = \lambda x_1 : T_1. \dots .\lambda x_n : T_n.s' s_1 \dots s_p$$

where s' is either primitive or $\overline{\eta}$-normal. If s contains a β-redex, then either at top level, i.e. $s's_1 \to_\beta t'$ for some t', or in one of the subterms $s', s_1, \dots, s_p$. Since $s_1, \dots, s_p$ are $\overline{\eta}$-normal, as well as s' if it contains a β-redex, the second case is handled by induction. In the first case s' has to be an $\overline{\eta}$-normal abstraction. Lemma 3.4 states that t' is $\overline{\eta}$-normal, which means that $t = \lambda x_1 : T_1. \dots .\lambda x_n : T_n.t' s_2 \dots s_p$ is $\overline{\eta}$-normal as well. □

Lemma 3.6 *Let s be a term and s' an $\overline{\eta}$-normal form of s. Let t be the β-normal form of s'. Then $\vdash s = t$ and t is a $\beta\overline{\eta}$-normal term.*

Proof $\vdash s = t$ is obvious, since $\vdash s = s'$ and $\vdash s' = t$. We prove the second claim by contradiction. Let $\Gamma \vdash t : T_1 \to \dots \to T_n \to B$. By Lemma 3.5, t is $\overline{\eta}$-normal, i.e.

$$t = \lambda x_1 : T_1. \dots .\lambda x_n : T_n.t' t_1 \dots t_p$$

where $t' \neq (t'_1 t'_2)$. We note that

$$\Gamma[x_1 := T_1, \dots, x_n := T_n] \vdash t' t_1 \dots t_p : B \qquad (*)$$

Now we consider two cases:

1. $t' = x$ or $t' = c$: contradiction since t' primitive
2. $t' = \lambda x : T'.t''$: Because of $(*)$ $p \geq 1$ and $t' t_1$ is a β-redex. Thus, t is not β-normal. $\Longrightarrow$ contradiction □

Theorem 3 *For every term s there exists a $\beta\overline{\eta}$-normal term t such that $\vdash s = t$.*

Proof Follows immediately from Lemmas 3.3 and 3.6. □

Bibliographic Remarks

- Termination of $\to_{\overline{\eta}}$ proved in Akama [1], Lemma 6. There it is attributed also to Mints and Cubric.
- Combination of β-reduction and $\overline{\eta}$-expansion studied by di Cosmo and Kesner [11], shown to be confluent and terminating.
- The fact that $\overline{\eta}$-normal forms are closed under β-reduction is stated in van Oostrom [46], Proposition 3.2.10.
- De Vrijer [10] proves strong normalization of $\lambda\beta\eta^\tau$-calculus by associating with every typed λ-term M an increasing functional. Other independent proofs can be found in Dragalin [12], Gandy [15], Hinata [24], Hanatani [19], Tait et al. [43].

Chapter 4

First-Order Completeness

Skolem [35, 36] shows that it is impossible to characterize the natural numbers by any denumerable system of first-order axioms (i.e. first-order variables and any set of functional constants). By restricting ourselves to first-order axioms we can hope to obtain systems that are semantically weak enough to be complete.

In this chapter we will prove the completeness of a family of higher-order logical systems. These systems are obtained by parameterizing S with first-order axioms. A notable member of this family is $\mathsf{S}(BAx)$, a logical system that may be called higher-order Boolean logic. The precise demands on the form of the axioms will become clear later.

The proof is based on Statman's results for the simply-typed lambda calculus [40, 41], which are in turn based on Friedman [14] and Plotkin [31, 32].

Definition 4.1 (Standard Term Model) Given (TC, VC, ty), a context Γ and a set of axioms A, a **standard term model** $\mathcal{D}_A$ is an interpretation such that

$$\forall B \in TC : \mathcal{D}_A(B) = \{[t]_A | \Gamma \vdash t : B\}$$

Definition 4.2 Let $\Gamma \vdash t : B$. We call t **basic** if

1. t is combinatoric
2. $\forall x \in FV\, t : ord(\Gamma x) = 1$

An equation $s = t$ is called basic if both terms are basic. If A is a set of basic equations, A is called basic as well.

Convention The following considerations always assume a signature of order ≤ 2, i.e.

$$\forall c \in VC : ord(ty\, c) \leq 2$$

Furthermore we assume a fixed set of axioms A defined relatively to a context Γ. Therefore, we can write $\mathcal{T}$ for $\mathcal{T}_A$ and $[t]$ for $[t]_A$. Some results will require A to be basic.

Definition 4.3 ($\mathcal{T}$) Let A be a set of equations. We define the interpretation $\mathcal{T}_A$ to be the unique standard term model such that for any constant $c \in VC$ with $\Gamma \vdash c : B_1 \to ... \to B_n \to B$ it holds

$$\mathcal{T}_A c\sigma[t_1]_A \ldots [t_n]_A = [ct_1 \ldots t_n]_A$$

Lemma 4.1 *Let*

1. t be a basic term,

2. $\sigma \in Sta(\mathcal{T}, \Gamma)$,

3. θ be a substitution such that $\forall x \in FV\, t : \sigma x = [\theta x]$.

Then

$$\mathcal{T} t\sigma = [\theta t]$$

Proof By induction on the structure of t:

1. $t = x$:

$$\mathcal{T} t\sigma = \mathcal{T} x\sigma \stackrel{\text{def}\,\mathcal{T}}{=} \sigma x \stackrel{\text{def}\,\theta}{=} [\theta x] = [\theta t]$$

2. $t = ct_1...t_n$:

$$\begin{array}{rcll} \mathcal{T} t\sigma & = & \mathcal{T}(ct_1...t_n)\sigma & \\ & = & \mathcal{T} c\sigma(\mathcal{T} t_1\sigma)...(\mathcal{T} t_n\sigma) & \text{def } \mathcal{T} \\ & = & \mathcal{T} c\sigma[\theta t_1]...[\theta t_n] & \text{by induction hypothesis} \\ & = & [c(\theta t_1)...(\theta t_n)] & \text{def } \mathcal{T} \\ & = & [\theta t] & \end{array}$$

□

Proposition 4.2 (Soundness) *Let A be basic. Then $\mathcal{T} \vDash A$.*

Proof Let σ be an arbitrary assignment and θ a substitution such that $\forall x \in FV\, t : \sigma x = [\theta x]$. Let s, t be two basic terms such that $A \vdash s = t$ (e.g. $(s, t) \in A$). Then

$$\begin{array}{rrcll} & A \vdash & s & = t & \\ \Longrightarrow & A \vdash & \theta s & = \theta t & \\ \Longleftrightarrow & & [\theta s] & = [\theta t] & \\ \Longleftrightarrow & & \mathcal{T} s\sigma & = \mathcal{T} t\sigma & \text{by Lemma 4.1} \end{array}$$

By the transitivity of $\vdash$ and by congruence axioms for the typed λ-calculus we then obtain $\mathcal{T} s\sigma = \mathcal{T} t\sigma$ for arbitrary terms s, t with $A \vdash s = t$. □

Definition Let us define a function $\rho : \mathcal{P}(Ter\,\Gamma) \to Ter\,\Gamma$ such that

$$\forall v \in \mathcal{P}(Ter\,\Gamma) : |v| \geq 1 \Longrightarrow \rho v \in v$$

Definition Now we define a special assignment σ_0 and a family of functions $(\tau^T)_{T \in Ty}$ where

$$\tau^T \in \mathcal{T}(T) \to \mathcal{P}(Ter^T\,\Gamma)$$

by mutual recursion on the order of the argument type T:

$T = B$:

$$\begin{array}{rcl} \sigma_0 x & = & [x] \\ \tau^T v & = & v \end{array}$$

$T = T_1 \to \ldots \to T_n \to B$:

$$\begin{array}{rcl} \sigma_0 x & = & \lambda v_1 \in \mathcal{T}(T_1). \ldots .\lambda v_n \in \mathcal{T}(T_n).[x(\rho(\tau^{T_1} v_1)) \ldots (\rho(\tau^{T_n} v_n))] \\ \tau^T v & = & \{t \in Ter^T \Gamma \mid \forall x_1, \ldots, x_n \notin FV\, t : \\ & & \quad x_1, \ldots, x_n \text{ different, } \Gamma x_1 = T_1, \ldots, \Gamma x_n = T_n \\ & & \quad \Longrightarrow [t x_1 \ldots x_n] = v(\sigma_0 x_1) \ldots (\sigma_0 x_n)\} \end{array}$$

Proposition 4.3 $s, t \in \tau^T v \Longrightarrow A \vdash t = t'$

Proof Let $T = T_1 \to \ldots \to T_n \to B$. Choose different variables $x_1, \ldots, x_n$ such that $\Gamma x_1 = T_1, \ldots, \Gamma x_n = T_n$ and $x_1, \ldots, x_n \notin FV\, s \cup FV\, t$. Then

$$\begin{array}{rcll} & [s x_1 \ldots x_n] = [t x_1 \ldots x_n] & & \text{def } \tau^T \\ \Longleftrightarrow & A \vdash s x_1 \ldots x_n = t x_1 \ldots x_n & & \\ \Longrightarrow & A \vdash s = t & & \text{by Proposition 1.1} \end{array}$$

□

Corollary 4.4 $t \in \tau^T v \Longrightarrow A \vdash t = \rho(\tau^T v)$

Lemma 4.5 *For any $\beta\overline{\eta}$-normal term t such that $\Gamma \vdash t : B$ it holds*

$$\mathcal{T} t \sigma_0 = [t]$$

Proof By induction on the structure of t:

1. $t = x$:

$$\mathcal{T} t \sigma_0 \overset{\text{def}\,\mathcal{T}}{=} \sigma_0 t \overset{\text{def}\,\sigma_0}{=} [t]$$

2. $t = c$:

$$\mathcal{T} t \sigma_0 \overset{\text{def}\,\mathcal{T}}{=} [c]$$

3. $t = x t_1 \ldots t_n$: Let $\Gamma \vdash t_1 : T_1, \ldots, \Gamma \vdash t_n : T_n$.

$$\begin{array}{rcll} \mathcal{T} t \sigma_0 & = & \mathcal{T} x \sigma_0 (\mathcal{T} t_1 \sigma_0) \ldots (\mathcal{T} t_n \sigma_0) & \text{def } \mathcal{T} \\ & = & [x(\rho(\tau^{T_1}(\mathcal{T} t_1 \sigma_0))) \ldots (\rho(\tau^{T_n}(\mathcal{T} t_n \sigma_0)))] & \text{def } \sigma_0 \\ & = & [x t_1 \ldots t_n] & \text{by Corollary 4.4} \\ & = & [t] & \end{array}$$

4. $t = c t_1 \ldots t_n$:

$$\begin{array}{rcll} \mathcal{T} t \sigma_0 & = & \mathcal{T} c \sigma_0 (\mathcal{T} t_1 \sigma_0) \ldots (\mathcal{T} t_n \sigma_0) & \text{def } \mathcal{T} \\ & = & \mathcal{T} c \sigma_0 [t_1] \ldots [t_n] & \text{by induction hypothesis} \\ & & & \text{since } \forall 1 \le i \le n : \Gamma \vdash t_i : B \\ & = & [c t_1 \ldots t_n] & \text{def } \mathcal{T} \\ & = & [t] & \end{array}$$

□

Lemma 4.6 *Let A be basic. Let $\Gamma \vdash t : B$. Then $\mathcal{T} t\sigma_0 = [t]$.*

Proof Let t' be a $\beta\overline{\eta}$-normal form to t. Then

$$\begin{array}{rcll} \mathcal{T} t\sigma_0 & = & \mathcal{T} t'\sigma_0 & \text{by Proposition 4.2 (soundness)} \\ & = & [t'] & \text{by Lemma 4.5} \\ & = & [t] & \text{since } A \vdash t = t' \end{array}$$

□

Lemma 4.7 *Let A be basic. Then*

$$\forall s, t \in Ter\,\Gamma : \mathcal{T} \models s = t \Longrightarrow A \vdash s = t$$

Proof Let $\Gamma \vdash s : T_1 \to \ldots \to T_n \to B$. Choose distinct variables $x_1, \ldots, x_n$ such that $\Gamma x_1 = T_1, \ldots, \Gamma x_n = T_n$ and $x_1, \ldots, x_n \notin FV\,s \cup FV\,t$. Then

$$\begin{array}{rcl} & \mathcal{T} \models s = t & \\ \Longrightarrow & \mathcal{T} \models sx_1 \ldots x_n = tx_1 \ldots x_n & \\ \Longrightarrow & \mathcal{T}(sx_1 \ldots x_n)\sigma_0 = \mathcal{T}(tx_1 \ldots x_n)\sigma_0 & \\ \Longleftrightarrow & [sx_1 \ldots x_n] = [tx_1 \ldots x_n] & \text{by Lemma 4.6} \\ \Longleftrightarrow & A \vdash sx_1 \ldots x_n = tx_1 \ldots x_n & \\ \Longrightarrow & A \vdash s = t & \text{by Proposition 1.1} \end{array}$$

□

Theorem 4 (Completeness) *Given a signature of order ≤ 2 and a set of equations A which is basic relatively to a context Γ, the following holds*

$$\forall s, t \in Ter\,\Gamma : A \models s = t \Longrightarrow A \vdash s = t$$

Proof

$$\begin{array}{rcl} & A \models s = t & \\ \Longrightarrow & \mathcal{T}_A \models s = t & \text{since } \mathcal{T}_A \models A \\ \Longrightarrow & A \vdash s = t & \text{by Lemma 4.7} \end{array}$$

□

Open Problem 1 The completeness result for higher-order Boolean logic seems not to carry over to the two-valued Boolean algebra $\mathcal{T}_2$ as defined by $BAx \cup \{B2\}$. It seems that though $\mathcal{T}_2 \models fx = f(f(fx))$ obviously holds, which can be verified by checking all possible values for f and x, this equality cannot be proved deductively: $BAx, B2 \not\vdash fx = f(f(fx))$. Although we have a strong intuition supporting our claim, a formal proof has not yet been obtained.

Chapter 5

Standard Models

In this chapter we investigate the semantic expressiveness of S, showing that our system can adequately represent every property that can be expressed in AHOL. When doing so, we make two implicit assumptions about the semantics of S:

1. We investigate the expressiveness of S with respect to standard interpretations. Standard interpretations are the most appropriate context for evaluating the expressiveness of logical systems since they are the type of models implicitly used in mathematics.
2. We require the interpretations of S to be non-trivial. The exact meaning of this restriction will be explained and motivated in Section 5.4.

We do not consider HOL directly, but study first the semantic expressiveness of smaller sets of axioms implied by HOL, like BAx and LAx.

We begin by studying the role of QAx in Boolean algebras. We observe that quantifiers as defined by QAx are closely related to infinite intersections and infinite unions of subsets of $\mathcal{D}\mathsf{B}$. This observation leads us to the conclusion that extending Boolean algebras by quantification enforces their completeness.

Then, we focus our attention on the identity test as the primitive operation in AHOL. We show how identity can be axiomatized in $\mathsf{S}(LAx)$ using Leibniz' criterion for equality and point out several important properties of this axiomatization.

We proceed by considering a special Boolean algebra, the two-valued algebra $\mathcal{T}_2$. $\mathcal{T}_2$ is used for the representation of truth values in AHOL. We show that $\mathcal{T}_2$ and all the essential operations of $\mathcal{T}_2$ can be axiomatized in S. Thus, we prove that S has at least the semantic expressiveness of AHOL.

We show that the semantic expressiveness of AHOL can also be achieved without sticking to $\mathcal{T}_2$. In order to do so, we exploit some semantic properties of the identity test that do not depend on the exact structure of the underlying Boolean algebra.

In the course of our discussion we present a finite axiomatization of the natural numbers in S.

Finally, we show how the results can be carried over to $\mathsf{S}(HOL)$.

5.1 Set Algebras

The following considerations will rely on some fundamental semantic properties of $\mathsf{S}(BAx)$. Interpretations satisfying BAx, also known as Boolean algebras, are well-understood. For a detailed account on the subject, see [9]. Here we just want to state briefly our assumptions concerning the semantics of Boolean algebras.

Let (TC, VC, ty) be a signature like in Definition 1.1, where the Boolean constants are defined relatively to the base type $\mathsf{B} \in TC$.

A typical Boolean algebra is a power set algebra looking as follows:

$$\begin{array}{lcl} \mathcal{D}\,0 & = & \varnothing \\ \mathcal{D}\,1 & = & S \\ \mathcal{D}\,\neg & = & \lambda x \in \mathcal{P}(S).S - x \\ \mathcal{D}\wedge & = & \lambda x \in \mathcal{P}(S).\lambda y \in \mathcal{P}(S).x \cap y \\ \mathcal{D}\vee & = & \lambda x \in \mathcal{P}(S).\lambda y \in \mathcal{P}(S).x \cup y \end{array}$$

This characterization defines a family of power set algebras differing from one another by the choice of the underlying set S. Stone [42] showed that every Boolean algebra is isomorphic to a **set algebra**, i.e. a subalgebra of a power set algebra. Therefore, when talking about Boolean algebras, we lose no generality by considering only the above interpretations for constants.

5.2 Quantification

Let us now extend Boolean algebras by universal quantification. On the one hand we add some quantifier constants to our signature, on the other hand we define their semantics by introducing new axioms. Definition 1.2 specifies both extensions formally.

Before we can use the extended algebras in new settings, we should ask ourselves two questions:

1. What impact do the new axioms have on the the structure of admissible models?

   Basically, three cases are possible:

   (a) Every Boolean algebra can be extended by quantification, i.e. the new axioms describe properties that are shared by all Boolean algebras.

   (b) LAx describes properties shared by some non-trivial Boolean algebras. In this case, we want to characterize these special properties as precisely as possible.

(c) LAx is inconsistent, i.e. the only Boolean algebra satisfying the new axioms is $\mathcal{T}_1$.

We want to show that a Boolean algebra can be extended to satisfy LAx if and only if it is complete.

2. What is the semantics of quantifier constants in interpretations satisfying LAx?

   We want to show that all interpretations satisfying LAx interpret $\forall$ by a function with well-known semantic properties.

   We claim that for every type T, LAx uniquely determines $\mathcal{D}\forall_T$ as follows:

$$\mathcal{D}\forall_T f = \inf\{fv | v \in \mathcal{D}T\}$$

In order to prove the claim we show that

1. $\mathcal{D}\forall f$ is a lower bound of $\{fv|v \in \mathcal{D}T\}$
2. every lower bound of $\{fv|v \in \mathcal{D}T\}$ is smaller or equal $\mathcal{D}\forall f$

Lemma 5.1 *For all $v \in \mathcal{D}T$, $f \in \mathcal{D}(T \to \mathsf{B})$*

$$\mathcal{D}\forall f \subseteq fv$$

Proof Assume a context Γ such that $\Gamma x = \mathsf{B}$ and $\Gamma g = T \to \mathsf{B}$.

$$\begin{array}{llcll} & LAx \vdash \forall g \wedge gx & = & \forall g & \forall I \\ \Longrightarrow & \mathcal{D}(\forall g \wedge gx)\sigma & = & \mathcal{D}(\forall g)\sigma & \text{for every } \sigma \\ \Longleftrightarrow & \mathcal{D}(\forall g)\sigma & \subseteq & \mathcal{D}(gx)\sigma & \text{for every } \sigma \\ \Longleftrightarrow & \mathcal{D}\forall f & \subseteq & fv & \text{for all } f, v \end{array}$$

□

Lemma 5.2 *If there exists some value $u \in \mathcal{D}T$ such that for all $v \in \mathcal{D}T$, $f \in \mathcal{D}(T \to \mathsf{B})$ it holds*

$$u \subseteq fv$$

then

$$u \subseteq \mathcal{D}\forall f$$

Proof Assume a context Γ such that $\Gamma x = \Gamma y = \mathsf{B}$ and $\Gamma g = T \to \mathsf{B}$. Assume further $u \subseteq fv$ for all f and v. Observe that

$$\begin{array}{llcll} & u & \subseteq & fv & \text{for all } f, v \\ \Longleftrightarrow & \sigma y & \subseteq & \mathcal{D}(gx)\sigma & \text{for every } \sigma \text{ with } \sigma y = u \\ \Longleftrightarrow & \mathcal{D}(y \wedge gx)\sigma & = & \sigma y & \text{for every } \sigma \text{ with } \sigma y = u \end{array}$$

Take an arbitrary σ with $\sigma y = u$ and $\sigma g = f$. Then

$$\begin{array}{llcll} & \mathcal{D}(y \wedge \forall g)\sigma & = & \mathcal{D}(\forall x.y \wedge gx)\sigma & \forall\wedge \\ & & = & \mathcal{D}(\forall x.y)\sigma & \text{by assumption} \\ & & = & \sigma y & \forall E \\ \Longleftrightarrow & \sigma y & \subseteq & \mathcal{D}(\forall g)\sigma & \\ \Longleftrightarrow & u & \subseteq & \mathcal{D}\forall f & \end{array}$$

□

Proposition 5.3 *A Boolean algebra $\mathcal{D}$ satisfies LAx if and only if the following equations are satisfied for every type T:*

$$\begin{array}{rcl} \mathcal{D}\forall_T f & = & \inf\{fv | v \in \mathcal{D}T\} \\ \mathcal{D}\exists_T f & = & \sup\{fv | v \in \mathcal{D}T\} \end{array}$$

Proof

- "$\Rightarrow$": The result for $\forall$ is an immediate consequence of Lemmas 5.1 and 5.2. The result for $\exists$ follows by duality.
- "$\Leftarrow$": The result for $\forall$ can be obtained by reverting the direction of the proofs of Lemmas 5.1 and 5.2. Again, the result for $\exists$ follows by duality. □

Theorem 5 *A Boolean algebra can be extended to satisfy LAx if and only if it is complete.*

Proof

- "$\Rightarrow$": By the definition of $\mathcal{D}$, $\mathcal{D}\forall_T$ and $\mathcal{D}\exists_T$ exist for all types T and are interpreted by functions from $\mathcal{D}(T \to \mathsf{B})$ to $\mathcal{D}\mathsf{B}$. Let $T = \mathsf{B} \to \mathsf{B}$. Then $|\mathcal{D}T| \geq |\mathcal{D}\mathsf{B}|$. Consequently, every subset of $\mathcal{D}\mathsf{B}$ can be described as $\{fv | v \in \mathcal{D}T\}$ for some $f \in \mathcal{D}(T \to \mathsf{B})$. Therefore, the infimum $\mathcal{D}\forall_T f$ and the supremum $\mathcal{D}\exists_T f$ exist for every subset of $\mathcal{D}\mathsf{B}$.
- "$\Leftarrow$": Whenever we have a complete Boolean algebra, we can give quantifier constants the denotations required by Proposition 5.3. □

Corollary 5.4 *Every Boolen algebra satisfying LAx is complete.*

Remark According to our representation of Boolean algebras, the interpretation of the universal quantifier over $\mathcal{D}T$ is uniquely determined by

$$\mathcal{D}\forall_T\sigma = \lambda f \in \mathcal{D}(T \to \mathsf{B}). \bigcap_{v \in \mathcal{D}T} fv$$

Remark The above results were obtained by Gert Smolka in September 2004.

5.3 Identity

We know so far that the Boolean constants need not be introduced in terms of logical constants. Instead, their semantics can be defined by means of Boolean axioms. Now we want to show that if we restrict ourselves to considering standard interpretations for Boolean algebras, we can define the identity test in terms of Boolean constants and quantifiers, in the same way it can be done in Church's formulation of higher-order logic [8] (and in the same way we did it in S^{Id}).

We define a family of constants $\doteq_T$ indexed by a type T as follows:

$$\doteq_T \stackrel{\text{def}}{=} \lambda x : T.\lambda y : T.\forall_{T \to \mathsf{B}} f.fx \to fy$$

The definition formalizes the characterization of equality by Leibniz, who observed that two values should be considered equal if and only if they have the same properties. We claim that two values of any domain are identical if and only if they are equal with respect to Leibniz' criterion for equality, i.e. if u and v are two values from the same domain, exactly one of the following statements holds

- u and v are identical, in which case $u \doteq v$ denotes $\mathcal{D}1$
- u and v differ and $u \doteq v$ denotes $\mathcal{D}0$

We prove the two cases separately.

Lemma 5.5 *If $s, t : \mathsf{B}$, then for any assignment σ it holds*

$$\mathcal{D}s\sigma = \mathcal{D}t\sigma \Longrightarrow \mathcal{D}(s \to t)\sigma = \mathcal{D}1$$

Proof

$$\begin{array}{rcll} \mathcal{D}(s \to t)\sigma & = & \mathcal{D}(\neg s \vee t)\sigma & \text{def} \to \\ & = & (S - \mathcal{D}s\sigma) \cup \mathcal{D}t\sigma & \text{def } \mathcal{D}\neg,\ \mathcal{D}\wedge \\ & = & (S - \mathcal{D}s\sigma) \cup \mathcal{D}s\sigma & \mathcal{D}s\sigma = \mathcal{D}t\sigma \\ & = & S & \text{set theory} \\ & = & \mathcal{D}1 & \end{array}$$

□

Proposition 5.6 $\mathcal{D}s\sigma = \mathcal{D}t\sigma \Longrightarrow \mathcal{D}(s \doteq t)\sigma = \mathcal{D}1$

Proof Let $s, t : T$, $f : T \to \mathsf{B}$ such that $f \notin FV\, s \cup FV\, t$.

$$\begin{array}{rcll} \mathcal{D}s\sigma = \mathcal{D}t\sigma & \Rightarrow & \mathcal{D}(fs)\sigma = \mathcal{D}(ft)\sigma & \text{regardless of } \sigma f \\ & \Rightarrow & \mathcal{D}(fs \to ft)\sigma = \mathcal{D}1 & \text{by Lemma 5.5} \\ & \Rightarrow & \mathcal{D}(\forall f.fs \to ft)\sigma = \mathcal{D}1 & \text{by Lemma 5.2} \\ & \Leftrightarrow & \mathcal{D}(s \doteq t)\sigma = \mathcal{D}1 & \text{def } \doteq \end{array}$$

□

Proposition 5.7 $\mathcal{D}s\sigma \neq \mathcal{D}t\sigma \Longrightarrow \mathcal{D}(s \doteq t)\sigma = \mathcal{D}0$

Proof Let $\mathcal{D}s\sigma = u$ and $\mathcal{D}t\sigma = v$. By assumption $u \neq v$. Consider the function $g = \lambda z \in \mathcal{D}\mathsf{B}.\text{if } z = u \text{ then } S \text{ else } \varnothing$. Then

$$\begin{array}{rcll} \mathcal{D}(s \doteq t)\sigma & = & \mathcal{D}(\forall f.fs \to ft)\sigma & \text{def } \doteq \\ & \subseteq & \mathcal{D}(fs \to ft)\sigma[f := g] & \text{def } \mathcal{D} \text{ and Lemma 5.1} \\ & = & \varnothing & \text{def}\, g,\ \text{def} \to \\ & = & \mathcal{D}0 & \end{array}$$

□

$\doteq$ takes two values from an arbitrary domain and returns a value from $\{\mathcal{D}0, \mathcal{D}1\}$, dependent on whether the two values are identical. Note that Boolean axioms ensure that the two values differ in every non-trivial algebra.

Proposition 5.8 $BAx \cup \{0 = 1\} \vdash x = 0$

Proof

$$\begin{array}{rcll} x & = & x \wedge 1 & BAx \\ & = & x \wedge 0 & 0 = 1 \\ & = & 0 & BAx \end{array}$$

□

5.4 The Two-Valued Boolen Algebra $\mathcal{T}_2$

It is usual practice to impose an additional restriction on Boolean algebras when they are used to represent truth values. They are required to be built on a two-element set. The two values are then interpreted as truth and falsehood. This is the approach used in AHOL. According to our picture of Boolean algebras, this restriction can be seen equivalent to requiring $|S| = 1$. Thus, we obtain a finite Boolean algebra. It is known that every finite Boolean algebra is isomorphic to a power set algebra (see [9] for reference). Let us write $\mathcal{T}_2$ for such a power set algebra with $|S| = 1$. $\mathcal{T}_2$ is unique up to isomorphism.

As it turns out, the requirement that $\mathcal{T}_2$ is the only Boolean algebra can be weakened without compromising the expressiveness of the resulting system.

Setting $S = \varnothing$ results in an algebra $\mathcal{T}_1$ built on an single-valued set $\mathcal{P}(S)$. In $\mathcal{T}_1$, all domains contain exactly one element, which means that all terms of the same type are given the same denotation. We call $\mathcal{T}_1$ the trivial Boolean algebra. Clearly, $\mathcal{T}_1$ is too weak if we want to specify any non-trivial properties.

However, setting S to be an arbitrary non-empty set actually gives us models that are at least as expressive as $\mathcal{T}_2$. We prove this claim by showing that $\mathcal{T}_2$ can be axiomatized within the more general system.

Convention In the following, we will always assume Boolean algebras to be non-trivial.

5.4.1 Axiomatization

In order to obtain $\mathcal{T}_2$, we extend BAx by one additional axiom:

$$f0 \wedge f1 \quad = \quad f0 \wedge f1 \wedge fx \qquad (B2)$$

where $f : \mathsf{B} \rightarrow \mathsf{B}$.

We claim that in conjunction with the Boolean axioms, $B2$ constrains the admissible interpretations to be isomorphic to $\mathcal{T}_2$:

Lemma 5.9 *Every interpretation satisfying $BAx \cup \{B2\}$ is isomorphic to $\mathcal{T}_2$.*

Proof By contradiction: Let $\mathcal{D} \models BAx \cup \{B2\}$ and $|S| \geq 2$.

$$\begin{array}{rcl} |S| \geq 2 & \Longrightarrow & |\mathcal{P}(S)| \geq 4 \\ & \Longrightarrow & \exists v \in \mathcal{P}(S) : \mathcal{D}0 = \varnothing \neq v \neq S = \mathcal{D}1 \end{array}$$

Choose an assignment σ such that

- $\sigma x = v$
- $\sigma f = \lambda v \in \mathcal{P}(S).\text{if } v = \varnothing \vee v = S \text{ then } S \text{ else } \varnothing$

and we obtain

$$\mathcal{D}(f0 \wedge f1)\sigma = S \neq \varnothing = \mathcal{D}(f0 \wedge f1 \wedge fx)\sigma$$

Thus $\mathcal{D} \not\models B2$. $\Longrightarrow$ contradiction □

Of course, $B2$ is not the only way of axiomatizing $\mathcal{T}_2$. Another possibility would have been to use *Bin* as we know it from *HOL*. Let us prove this claim by by showing the deductive, and therefore semantic, equivalence of $B2$ and *Bin*.

Proposition 5.10 $LAx, B2 \vdash Bin$

Proof

$$\begin{array}{rcll} f1 \wedge f0 & = & \forall x. f1 \wedge f0 & \forall E \\ & = & \forall x. f1 \wedge f0 \wedge fx & B2 \\ & = & \forall f \wedge f1 \wedge f0 & \forall\wedge \\ & = & \forall f & \forall I \end{array}$$

□

Proposition 5.11 $LAx, Bin \vdash B2$

Proof

$$\begin{array}{rcll} f0 \wedge f1 & = & \forall f & Bin \\ & = & \forall f \wedge fx & \forall I \\ & = & f0 \wedge f1 \wedge fx & Bin \end{array}$$

□

Let us introduce the set $LAx2$ as an extension of LAx by $B2$. $LAx2$ axiomatizes two-valued Boolean algebras with quantification:

Definition $LAx2 \stackrel{\text{def}}{=} LAx \cup \{B2\}$

5.4.2 Expressiveness

$B2$ ensures that $\{\mathcal{D}0, \mathcal{D}1\}$ are the only values in $\mathcal{D}\mathsf{B}$. Thus, $\doteq$ has exactly the semantics of Andrews' identity constant Q.

Since the identity constant is the only logical constant needed to define the semantics of higher-order logic as defined by Andrews, we have shown that $\mathsf{S}(LAx2)$ has at least the expressiveness of traditional higher-order logic. Every property specified in the traditional system can be translated to our system by using $\doteq$ and operations derived from the identity test.

Remark Of course, the validity of this translation depends on $\doteq$ having the intended semantics. We have shown this for standard models, but as soon as we allow non-standard models, which will be introduced in Chapter 6, the semantics of $\doteq$ may change. In particular, if we drop the extensionality requirement, $\doteq$ obviously no longer denotes identity.

Defining Boolean constants and quantifiers in terms of $\doteq$ would introduce a second version of these operators. We can easily check that in $\mathcal{T}_2$ the derived operators behave in exactly the same way as the original ones. Therefore, we can continue using the original constants without losing expressiveness.

We conclude that with $LAx2$ we have successfully axiomatized the semantics of traditional mathematical logic.

5.5 Beyond $\mathcal{T}_2$

5.5.1 Binary Values

Definition A value $v \in M_1 \to \ldots \to M_n \to \mathcal{D}\mathsf{B}$ is called **binary** if for all $v_1 \in M_1, \ldots, v_n \in M_n$ it holds

$$vv_1 \ldots v_n \in \{\mathcal{D}0, \mathcal{D}1\}$$

If $n \geq 1$, we call v a binary function. **Binary terms** are terms that are always interpreted as binary values. **Binary equations** are equations where both terms are binary.

When applied to binary values, the Boolean operators as well as quantifiers are guaranteed to yield binary values as results. Indeed, it is not difficult to see that in AHOL all the typical constants behave on binary arguments in exactly the same way as they do in $\mathcal{T}_2$, which can easily be checked by using truth tables or some other technique for analysing finite functions. Thus, when dealing with binary terms, their semantics correspond precisely to our intuition for two-valued Boolean logic. In order to keep our subsequent proofs simple, we will rely on this intuition as often as possible.

What does it mean to "be equal according to $\doteq$" if $\mathcal{D}\mathsf{B}$ contains more than two values? The answer to this question is quite obvious, since actually we have already seen that the equality test is binary:

Proposition 5.12 $\doteq$ *is binary.*

Proof Follows immediately from Propositions 5.6 and 5.7. □

We observe that $\doteq$ has the semantics of Andrews' identity constant regardless of the cardinality of $\mathcal{D}\mathsf{B}$.

We have shown that $\doteq$ has proper semantics in $\mathsf{S}(LAx)$. Now we can use $\doteq$ to show that $\mathsf{S}(LAx)$ is as expressive as $\mathsf{S}(LAx2)$.

Again, we can introduce a second version of the common logical operators in terms of $\doteq$. Unlike in $\mathcal{T}_2$, the derived operators behave in a way differing from the semantics of the constants used to axiomatize $\doteq$. Like $\doteq$, they are binary regardless of the cardinality of $\mathcal{D}\mathsf{B}$, whereas the original constants display the typical behaviour of Boolean operations in complete set algebras. However, when dealing with binary values, we can use on the original constants without having to worry about a possible loss of expressiveness. The reason has already been stated before. The behaviour of the original constants on binary values matches that of the derived ones.

But first, we have to find a way to transform ordinary terms with $\operatorname{ran} t = \mathsf{B}$ to binary terms. This is what we will do next.

5.5.2 Predicates

The set $\{\mathcal{D}0, \mathcal{D}1\}$ obviously corresponds to the set of truth values in Andrews' formulation of higher-order logic, while the set of individuals can be chosen arbitrarily from $\{\mathcal{D}T | T \in Ty\}$.

In AHOL it is possible to represent functions whose range is the set of truth values. Such functions are commonly called predicates. By what we have seen so far, in our system predicates correspond to binary values. In higher-order predicate logic, predicates are considered first-class, just like ordinary values. Thus, we need a way to represent variables over predicates. In order to enforce the binarity of a target function, we can use the identity test:

Definition 5.1 Let $\Gamma f = T_1 \to \ldots \to T_n \to \mathsf{B}$. Then

$$\hat{f} \stackrel{\text{def}}{=} \lambda x_1 : T_1. \ldots \lambda x_n : T_n . f x_1 \ldots x_n \doteq 1$$

It is not hard to see that the denotation of $\hat{f}$ is always binary. Moreover, $\mathcal{D}\hat{f}\sigma$ depends solely on the value of σ for its only free variable f. Indeed, by properly instantiating f, the denotation of $\hat{f}$ can represent every binary function in $\mathcal{D}(T_1 \to \ldots \to T_n \to \mathsf{B})$.

Let us extend our notation such that we can represent predicates over truth values:

Definition 5.2 Let $\Gamma f = \mathsf{B} \to \ldots \to \mathsf{B} \to \mathsf{B}$. Then

$$\check{f} \stackrel{\text{def}}{=} \lambda x_1 : \mathsf{B}. \ldots \lambda x_n : \mathsf{B}. f(x_1 \doteq 1) \ldots (x_n \doteq 1) \doteq 1$$

We can prove that in $\mathsf{S}(LAx2)$ the encoded terms are deductively equivalent to the original terms, i.e. $LAx2 \vdash \hat{t} = t$ and $LAx2 \vdash \check{t} = t$ hold for all terms t of appropriate types. This means, when considering $\mathsf{S}(LAx2)$, we can just substitute terms using predicate encoding by their non-accented versions.

Let us proceed by proving the claimed deductive equivalence. First, we observe that $\doteq$ has some notable deductive properties:

Proposition 5.13 *For all* $s, t : \mathsf{B}$ *it holds* $LAx, s \doteq t = 1 \vdash s = t$

Proof

$$\begin{array}{rcll}
s & = & 1 \wedge s & BAx \\
 & = & s \doteq t \wedge s & s \doteq t = 1 \\
 & = & (\forall f. fs \to ft) \wedge s & \text{def } \doteq \\
 & = & (\forall f. fs \to ft) \wedge s \wedge (s \to t) \wedge (t \to s) & \forall I \text{ with } \lambda x : \mathsf{B}.x, \neg \\
 & = & (\forall f. fs \to ft) \wedge s \wedge t & BAx \\
 & = & (\forall f. fs \to ft) \wedge t \wedge (s \to t) \wedge (t \to s) & BAx \\
 & = & (\forall f. fs \to ft) \wedge t & \forall I \\
 & = & 1 \wedge t & \text{def } \doteq,\ s \doteq t = 1 \\
 & = & t &
\end{array}$$

□

Proposition 5.14 $LAx \vdash x \doteq x = 1$

Proof

$$\begin{array}{rcll}
x \doteq x & = & \forall f. fx \to fx & \text{def } \doteq \\
 & = & \forall f. 1 & BAx \\
 & = & 1 & \forall E
\end{array}$$

□

From this we can conclude the following:

Lemma 5.15 $LAx2 \vdash 0 \doteq 1 = 0$

Proof

$$\begin{array}{lll} 0 \doteq 1 & & \\ = & 0 \doteq 1 \wedge 1 & BAx \\ = & 0 \doteq 1 \wedge 1 \doteq 1 & \text{Prop. 5.14} \\ = & \forall x.x \doteq 1 & \text{Prop. 5.10} \\ = & \forall x.\forall f.fx \rightarrow f1 & \text{def } \doteq \\ = & (\forall x.\forall f.fx \rightarrow f1) \wedge \forall f.f0 \rightarrow f1 & \forall I \text{ with } 0 \\ = & (\forall x.\forall f.fx \rightarrow f1) \wedge (\forall f.f0 \rightarrow f1) \wedge (\neg 0 \rightarrow \neg 1) & \forall I \text{ with } \neg \\ = & 0 & BAx \end{array}$$

□

Lemma 5.16 $LAx2 \vdash \forall x.(x \doteq 1) \doteq x = 1$

Proof

$$\begin{array}{lll} \forall x.(x \doteq 1) \doteq x & & \\ = & (0 \doteq 1) \doteq 0 \wedge (1 \doteq 1) \doteq 1 & \text{by Proposition 5.10} \\ = & (1 \wedge 1) \doteq 1 \doteq 1 & \text{by Lemma 5.15, Proposition 2.7} \\ = & 1 \wedge 1 & \text{by Propositions 5.14 and 2.7} \\ = & 1 & BAx \end{array}$$

□

Lemma 5.17 $LAx2 \vdash (x \doteq 1) \doteq x = 1$

Proof

$$\begin{array}{rcll} (x \doteq 1) \doteq x & = & 1 \wedge (x \doteq 1) \doteq x & BAx \\ & = & (\forall x.(x \doteq 1) \doteq x) \wedge (x \doteq 1) \doteq x & \text{by Lemma 5.16} \\ & = & \forall x.(x \doteq 1) \doteq x & \forall I \\ & = & 1 & \text{by Lemma 5.16} \end{array}$$

□

Lemma 5.18 $LAx2 \vdash x \doteq 1 = x$

Proof By Lemma 5.17 and Proposition 5.13. □

Theorem 6 *Let* $\Gamma f = \mathsf{B} \rightarrow \ldots \rightarrow \mathsf{B} \rightarrow \mathsf{B}$. *Then*

$$LAx2 \vdash \check{f} = f$$

Proof

$$\begin{array}{rcll} \check{f} & = & \lambda x_1 : \mathsf{B}. \ldots \lambda x_n : \mathsf{B}.f(x_1 \doteq 1) \ldots (x_n \doteq 1) \doteq 1 & \text{def } \check{} \\ & = & \lambda x_1 : \mathsf{B}. \ldots \lambda x_n : \mathsf{B}.fx_1 \ldots x_n \doteq 1 & \text{Lem. 5.18} \\ & = & \lambda x_1 : \mathsf{B}. \ldots \lambda x_n : \mathsf{B}.fx_1 \ldots x_n & \text{Lem. 5.18} \\ & = & f & \eta \end{array}$$

□

Corollary 5.19 *Let* $\Gamma f = T_1 \to \ldots \to T_n \to \mathsf{B}$. *Then*

$$LAx2 \vdash \hat{f} = f$$

Remark Proposition 5.14, Lemma 5.15 and Lemma 5.17 correspond to Andrews' [4] Propositions 5210, 5217 and 5218 respectively. While the proofs of the two lemmas basically resemble the corresponding proofs by Andrews, note that in order to prove 5210 and 5217 he makes use of his extensionality axiom, whereas our proofs work without any extensionality assumptions.

Let us demonstrate with two examples how we can adapt classical results to our system without requiring $\mathcal{D}\mathsf{B}$ to be two-valued.

5.5.3 Finite Domains

Axiomatization of $\mathcal{T}_2$ we did before is just a special case of a more general setting, which is axiomatization of finite domains.

Let us assume, we are working with a signature (TC, VC, ty) such that $c_1, \ldots, c_m \in VC$ and $ty\, c_1 = \ldots = ty\, c_m = T$ for some type T. Let $\mathcal{D}$ be a non-trivial Boolean algebra based on a type B. We want to restrict $\mathcal{D}T$ to contain exactly n elements. This can be done with the help of the following axiom:

$$\exists x_1. \ldots \exists x_n. \bigwedge_{1 \le i < j \le n} \neg x_i \doteq x_j \wedge \forall x. \bigvee_{1 \le i \le n} x \doteq x_i = 1 \qquad (FDom\, n)$$

where we assume $\Gamma x = \Gamma x_{m+1} = \ldots = \Gamma x_n = T$.

$FDom\, n$ states that $\mathcal{D}T$ must contain n distinct values and that $\mathcal{D}T$ has no other elements apart from these n values.

If we further want $c_1, \ldots, c_m$ to denote pairwise distinct values, we can easily achieve this goal as follows:

$$\bigwedge_{1 \le i < j \le m} \neg c_i \doteq c_j = 1$$

As we see, both axioms are binary, which means, that the classical approaches for showing the appropriateness of the axioms can be used without restrictions.

5.5.4 The Natural Numbers

We have seen how to axiomatically represent arbitrary finite domains. How can we generalize the approach to handle infinite domains? Let our next task be the axiomatization of the natural numbers. We extend the Boolean signature (TC, VC, ty) as follows:

- $TC \supseteq \{\mathsf{N}, \mathsf{B}\}$
- $VC \supseteq \{0, 1, \neg, \wedge, \vee, \dot{o}, \dot{s}\} \cup \{\forall_T | T \in Ty\}$
- ty is defined by the following table:

$$\begin{aligned} 0,1 &: \mathsf{B} \\ \neg &: \mathsf{B} \to \mathsf{B} \\ \wedge, \vee &: \mathsf{B} \to \mathsf{B} \to \mathsf{B} \\ \dot{o} &: \mathsf{N} \\ \dot{s} &: \mathsf{N} \to \mathsf{N} \\ \forall_T &: (T \to \mathsf{B}) \to \mathsf{B} \text{ for all } T \in Ty \end{aligned}$$

Let the set $\dot{\mathbb{N}}$ be the domain of our new type constant N. We want the set $\dot{\mathbb{N}}$ to contain exactly the natural numbers. It is widely known that the structure of $\dot{\mathbb{N}}$ can be axiomatized by means of Peano's postulates. Informally they can be stated as follows:

Let M be a set such that

($N0$) M contains a dedicated element 0.

($N\mathbf{S}$) For every element m in M there exists a successor element Sm. m is called the predecessor of Sm.

($N1$) 0 has no predecessor.

($N2$) The mapping S is injective.

($N3$) The principle of mathematical induction holds on M ordered by the successor relation.

Then M is isomorphic to the natural numbers.

The following formalization of the last three postulates (NAx) assumes $x, y : \mathsf{N}$.

$$\begin{aligned} \neg \dot{s}x \doteq \dot{o} &= 1 && (N1) \\ (\dot{s}x \doteq \dot{s}y) \to (x \doteq y) &= 1 && (N2) \\ \hat{f}\dot{o} \wedge (\forall x.\hat{f}x \to \hat{f}(\dot{s}x)) \to \hat{f}y &= 1 && (N3) \end{aligned}$$

Notice that $N0$ and NS are automatically satisfied by every standard interpretation of our system and hence need not be stated formally.

Let us show that NAx in conjunction with BAx is indeed an axiomatization of the natural numbers, i.e. $\dot{\mathbb{N}} \cong \mathbb{N}$. In order to prove this claim we need to take a closer look at interpretations $\mathcal{D} \vDash LAx \cup NAx$.

Lemma 5.20 *It holds:*

1. $\mathcal{D}\dot{o} \notin Ran(\mathcal{D}\dot{s})$
2. $\mathcal{D}\dot{s}$ *is injective*

Proof Both claims are easy to prove because of the binarity of $N1$ and $N2$. $\square$

Lemma 5.21 *Let $f \in M \to M$ be injective and let $x \in M - Ran\, f$. Then it holds for all $m, n \geq 0$*

$$f^m x = f^n x \iff m = n$$

Proof We prove "$\Rightarrow$". The inverse direction is obvious. Assume f injective, $x \notin Ran\, f$ and $m \neq n$. Let w.l.o.g. $m > n$. We show $f^m x \neq f^n x$ by induction on $n \in \mathbb{N}$:

- $n = 0$: By assumption $x \notin Ran\, f$. Since $m \geq 1$, $f^m x \in Ran\, f$. Therefore $f^n x = x \neq f^m x$.
- $n - 1 \to n$: By induction hypothesis $f^{m-1}x \neq f^{n-1}x$. Since f injective, we conclude $f^m x \neq f^n x$. $\square$

Lemma 5.22 *For any assignment σ and for any $m, n \geq 0$ it holds*

$$\mathcal{D}(\dot{s}^m \dot{o})\sigma = \mathcal{D}(\dot{s}^n \dot{o})\sigma \iff m = n$$

Proof By Lemma 5.20, $\mathcal{D}\dot{o}$ and $\mathcal{D}\dot{s}$ have all the properties needed to derive the claim from Lemma 5.21. $\square$

Lemma 5.22 shows that syntactically distinct terms built up from the constants $\dot{o}$ and $\dot{s}$ have distinct denotations. Since there exist countably infinitely many such terms, it is not difficult to see that with respect to set isomorphism it holds

$$\mathbb{N} \cong \bigcup_{n=0}^{\infty} (\mathcal{D}\dot{s})^n (\mathcal{D}\dot{o})$$

Obviously, $\bigcup_{n=0}^{\infty} (\mathcal{D}\dot{s})^n (\mathcal{D}\dot{o}) \subseteq \dot{\mathbb{N}}$. Thus, we have shown $\mathbb{N}$ isomorphic to a subset of $\dot{\mathbb{N}}$. It remains to show this subset to be the whole set $\dot{\mathbb{N}}$.

This is intended to be accomplished by the binary axiom $N3$. $N3$ is equivalent to the usual formalization of induction in higher-order logic, which is known to be sufficient in order to enforce the desired property.

Lemma 5.23

$$\dot{\mathbb{N}} \subseteq \bigcup_{n=0}^{\infty} (\mathcal{D}\dot{s})^n (\mathcal{D}\dot{o})$$

Proof By contradiction: Let $v \in \dot{\mathbb{N}} - \bigcup_{n=0}^{\infty} (\mathcal{D}\dot{s})^n (\mathcal{D}\dot{o})$. Let

- $\sigma y = v$
- $\sigma f = \lambda x \in \dot{\mathbb{N}}.\text{if } \exists n \geq 0 : x = (\mathcal{D}\dot{s})^n (\mathcal{D}\dot{o}) \text{ then } \mathcal{D}1 \text{ else } \mathcal{D}0$

Clearly, σf is a binary function, i.e. we have $\mathcal{D}\hat{f}\sigma = \sigma f$. We observe:

- $\mathcal{D}(\hat{f}\dot{o} \wedge (\forall x.\hat{f}x \rightarrow \hat{f}(\dot{s}x)))\sigma = \mathcal{D}1$
- $\mathcal{D}(\hat{f}y)\sigma = \mathcal{D}0$

Therefore $\mathcal{D}(\hat{f}\dot{o} \wedge (\forall x.\hat{f}x \rightarrow \hat{f}(\dot{s}x)) \rightarrow \hat{f}y)\sigma = \mathcal{D}0 \neq \mathcal{D}1$, which is a contradiction to $N3$. □

Proposition 5.24 $\dot{\mathbb{N}} \cong \mathbb{N}$

Proof Follows immediately from Lemmas 5.22 and 5.23. □

Let us interpret the constants $0_{\mathbb{N}}, \mathrm{S}_{\mathbb{N}}, +_{\mathbb{N}}, *_{\mathbb{N}}$ as the natural zero, the function $(\lambda x \in \mathbb{N}.x + 1)$, the addition and the multiplication over naturals respectively.

We complete our axiomatization of the natural numbers by establishing a structural isomorphism between our formalization of the naturals and the algebra $\langle \mathbb{N}, 0_{\mathbb{N}}, \mathrm{S}_{\mathbb{N}}, +_{\mathbb{N}}, *_{\mathbb{N}} \rangle$.

Our system can easily be extended by formal equivalents of addition and multiplication. We add the constants $+$ and $*$ to VC and extend ty such that $ty\,+ = ty\,* = \mathsf{N} \rightarrow \mathsf{N} \rightarrow \mathsf{N}$. Also, we have to provide an axiomatic definition ($NOpAx$) of the new constants. The definition is based on the theory of primitive recursive arithmetic, originating in Skolem's work [34]. Relatively to a type environment Γ with $\Gamma x = \Gamma y = \mathsf{N}$, $NOpAx$ can be stated as follows:

$$\begin{array}{rclrcl} x + \dot{o} & = & x & x * \dot{o} & = & \dot{o} \\ x + (\dot{s}y) & = & (\dot{s}x) + y & x * (\dot{s}y) & = & x + (x * y) \end{array}$$

Lemma 5.25 $\langle \dot{\mathbb{N}}, \dot{o}, \dot{s}, +, * \rangle \cong \langle \mathbb{N}, 0_{\mathbb{N}}, \mathrm{S}_{\mathbb{N}}, +_{\mathbb{N}}, *_{\mathbb{N}} \rangle$

Proof We define a mapping $\phi : \langle \dot{\mathbb{N}}, \dot{o}, \dot{s}, +, * \rangle \rightarrow \langle \mathbb{N}, 0_{\mathbb{N}}, \mathrm{S}_{\mathbb{N}}, +_{\mathbb{N}}, *_{\mathbb{N}} \rangle$ such that:

c	$\dot{o}$	$\dot{s}$	$+$	$*$
ϕc	$0_{\mathbb{N}}$	$\mathrm{S}_{\mathbb{N}}$	$+_{\mathbb{N}}$	$*_{\mathbb{N}}$

By straightforward inductive reasoning we can verify that ϕ is a homomorphism. By Proposition 5.24, ϕ is bijective. □

Definition We say an interpretation $\mathcal{D}$ **contains the natural numbers** if for some $T \in Ty$ and for some value constants $\dot{o}, \dot{s}, +, * \in VC$ with

$$\begin{array}{c} \dot{o}, \dot{s} : T \\ +, * : T \rightarrow T \rightarrow T \end{array}$$

it holds $\langle \mathcal{D}T, \dot{o}, \dot{s}, +, * \rangle \cong \langle \mathbb{N}, 0_{\mathbb{N}}, \mathrm{S}_{\mathbb{N}}, +_{\mathbb{N}}, *_{\mathbb{N}} \rangle$

Definition Let A be a set of equations. We call A an **axiomatization** of an interpretation $\mathcal{D}$ if for every interpretation $\mathcal{E}$ it holds:

$$\mathcal{E} \models A \iff \mathcal{E} \cong \mathcal{D}$$

A axiomatizes a family of interpretations $\mathbf{F}$ if

$$\mathcal{E} \models A \iff \exists \mathcal{D} \in \mathbf{F} : \mathcal{E} \cong \mathcal{D}$$

Theorem 7 *Interpretations containing the natural numbers can be finitely axiomatized.*

Proof To apply Lemma 5.25, we need BAx, the axioms $\forall I_T$ and $\forall \vee_T$ for $T \in \{\mathsf{B} \to \mathsf{B}, \mathsf{N}, \mathsf{N} \to \mathsf{B}\}$, NAx and $NOpAx$. □

Remark If we extend the above axiomatization by $B2$, we can replace $N3$ by its non-accented version

$$f\dot{o} \wedge (\forall x.fx \to f(\dot{s}x)) \to fy \quad = \quad 1$$

thus making redundant the two quantifier axioms for $\mathsf{B} \to \mathsf{B}$. By doing so, we obtain an axiomatization of $\mathcal{T}_2$ containing the natural numbers.

We have shown that we can encode the natural numbers within S. By Gödel's first incompleteness theorem [16], this means that, when parameterized with the above axioms, S becomes essentially incomplete, i.e. the semantic closure of the axioms is no longer recursively enumerable.

Corollary 5.26 *There exist finite sets of axioms A such that $SC(A)$ is not recursively enumerable.*

5.6 *HOL* and its Semantic Closure

We finish our investigations of the semantic expressiveness of S by considering $SC(HOL)$. We have already seen that $LAx2$ is at least as expressive as AHOL. By showing $SC(HOL) = SC(LAx2)$, are able to make an equivalent statement for HOL.

Remark We take the consistency of $LAx2$ for granted. We do this relying on the consistency of AHOL, since every axiom from $LAx2$ can be proved a theorem of Andrews' logic.

We begin by proving $SC(HOL) \supseteq SC(LAx2)$. Since we already know that $HOL \vdash LAx2$ (by Corollary 2.5, Proposition 5.11), it suffices to verify that the constants of $\mathsf{S}(LAx2)$ really correspond to that of $\mathsf{S}(HOL)$. This is clearly the case for the Boolean constants and for quantifiers. The situation is different for $\doteq$. In $\mathsf{S}(HOL)$, $\doteq$ is a primitive constant, whose semantics is mainly defined by *Ref* and *Rep*. In $\mathsf{S}(LAx2)$, $\doteq$ is a notational abbreviation derived from quantification.

Thus, we have no formal correspondence between $\doteq$ in $\mathsf{S}(HOL)$ and the identity test based on Leibniz' characterization, as it was studied in this chapter. So, let us establish the missing correspondence.

Proposition 5.27 $HOL \vdash \doteq\ = \lambda x : T.\lambda y : T.\forall f.fx \to fy$

Proof Note that, again by Corollary 2.5 and Proposition 5.11, we may use $LAx2$. Then

$$\begin{array}{rcll} x \doteq y & = & x \doteq y \wedge 1 & BAx \\ & = & x \doteq y \wedge (x \doteq y \to (\forall f.fx \to fx) \doteq \forall f.fx \to fy) & Con \\ & = & x \doteq y \wedge (\forall f.fx \to fx) \doteq \forall f.fx \to fy & BAx \\ & = & x \doteq y \wedge (\forall f.1) \doteq \forall f.fx \to fy & BAx \\ & = & x \doteq y \wedge 1 \doteq \forall f.fx \to fy & \forall E \\ & = & x \doteq y \wedge \forall f.fx \to fy & \text{Lem. 5.18} \\ & = & (x \doteq x \to x \doteq y) \wedge \forall f.fx \to fy & BAx \\ & = & \forall f.fx \to fy & \forall I \end{array}$$

$$\iff \qquad \doteq\ = \lambda x : T.\lambda y : T.\forall f.fx \to fy \qquad \square$$

In order to show $SC(HOL) = SC(LAx2)$ we still need to prove the inclusion $SC(HOL) \subseteq SC(LAx2)$. To do so, it suffices to show $HOL \subseteq SC(LAx2)$. Clearly, we have $Ext \in SC(\varnothing)$, since standard interpretations are extensional by definition. $Ref \in SC(LAx)$ holds by Propositions 5.6 and 5.7. $Bin \in SC(LAx2)$ follows immediately from Proposition 5.10. It remains to check the validity of $D\forall$, *Rep* and *Rep*$'$.

Lemma 5.28 $LAx2, Ext \vdash D\forall$

Proof

$$\begin{array}{rcll} \forall & = & \lambda f : T \to \mathsf{B}.\forall x.fx & \eta \\ & = & \lambda f : T \to \mathsf{B}.\forall x.fx \doteq 1 & \text{by Lemma 5.18} \\ & = & \lambda f : T \to \mathsf{B}.\forall x.fx \doteq (\lambda x : T.1)x & \eta \\ & = & \lambda f : T \to \mathsf{B}.f \doteq (\lambda x : T.1) & Ext \end{array} \qquad \square$$

Lemma 5.29 *For all interpretations $\mathcal{D}$ satisfying LAx, for all terms t, s', t' such that $t : B$ and for every assignment σ:*

$$\mathcal{D}((\forall FV.s' \doteq t') \wedge t[s'])\sigma = \mathcal{D}((\forall FV.s' \doteq t') \wedge t[t'])\sigma$$

Proof By Propositions 5.6, 5.7 and Lemma 5.1, we need to distinguish two cases:

1. $\mathcal{D}(\forall FV.s' \doteq t')\sigma = \mathcal{D}0$: Then

$$\begin{array}{rcll} \mathcal{D}((\forall FV.s' \doteq t') \wedge t[s'])\sigma & = & \mathcal{D}0 \cap \mathcal{D}(t[s'])\sigma & \text{def}\,\mathcal{D} \\ & = & \mathcal{D}0 & \text{def}\,\mathcal{D}0 \\ & = & \mathcal{D}0 \cap \mathcal{D}(t[t'])\sigma & \text{def}\,\mathcal{D}0 \\ & = & \mathcal{D}((\forall FV.s' \doteq t') \wedge t[t'])\sigma & \text{def}\,\mathcal{D} \end{array}$$

2. $\mathcal{D}(\forall FV.s' \doteq t')\sigma = \mathcal{D}1$: By Proposition 5.7 and Lemma 5.1, we obtain

$$\mathcal{D}s'\sigma' = \mathcal{D}t'\sigma' \qquad (*)$$

for every assignment σ'. Then

$$\begin{array}{rll} \multicolumn{3}{l}{\mathcal{D}((\forall FV.s' \doteq t') \wedge t[s'])\sigma} \\ = & \mathcal{D}1 \cap \mathcal{D}(t[s'])\sigma & \text{def}\,\mathcal{D} \\ = & \mathcal{D}(t[s'])\sigma & \text{def}\,\mathcal{D}1 \\ = & \mathcal{D}(t[t'])\sigma & \text{by } (*) \text{ and congruence} \\ = & \mathcal{D}1 \cap \mathcal{D}(t[t'])\sigma & \text{def}\,\mathcal{D}1 \\ = & \mathcal{D}((\forall FV.s' \doteq t') \wedge t[t'])\sigma & \text{def}\,\mathcal{D} \end{array}$$

□

Lemma 5.30 *For all interpretations $\mathcal{D}$ satisfying LAx and for every σ:*

$$\mathcal{D}(x \doteq y \wedge fx)\sigma = \mathcal{D}(x \doteq y \wedge fy)\sigma$$

Proof Again, we need to distinguish two cases:

1. $\mathcal{D}(x \doteq y)\sigma = \mathcal{D}0$: Then

$$\begin{array}{rcll} \mathcal{D}(x \doteq y \wedge fx)\sigma & = & \mathcal{D}0 \cap \mathcal{D}(fx)\sigma & \text{def}\,\mathcal{D} \\ & = & \mathcal{D}0 & \text{def}\,\mathcal{D}0 \\ & = & \mathcal{D}0 \cap \mathcal{D}(fy)\sigma & \text{def}\,\mathcal{D}0 \\ & = & \mathcal{D}(x \doteq y \wedge fy)\sigma & \text{def}\,\mathcal{D} \end{array}$$

2. $\mathcal{D}(x \doteq y)\sigma = \mathcal{D}1$: By Proposition 5.7, we obtain

$$\sigma x = \sigma y \qquad (*)$$

Then

$$\begin{array}{rcll} \mathcal{D}(x \doteq y \wedge fx)\sigma & = & \mathcal{D}1 \cap \sigma f(\sigma x) & \text{def}\,\mathcal{D} \\ & = & \sigma f(\sigma x) & \text{def}\,\mathcal{D}1 \\ & = & \sigma f(\sigma y) & \text{by } (*) \\ & = & \mathcal{D}1 \cap \sigma f(\sigma y) & \text{def}\,\mathcal{D}1 \\ & = & \mathcal{D}(x \doteq y \wedge fx) & \text{def}\,\mathcal{D} \end{array}$$

□

Proposition 5.31 $SC(HOL) = SC(LAx2)$

Proof Follows from $SC(HOL) \supseteq SC(LAx2)$ in conjunction with Propositions 5.6, 5.7, 5.10, Lemmas 5.28, 5.29 and 5.30. □

Theorem 8 *Every semantic property representable in* AHOL *can be expressed in* $\mathsf{S}(HOL)$.

Proof Follows from Proposition 5.31. □

Chapter 6

General Models

We continue our studies of S by introducing a type of non-standard models known as Henkin models or general models. These models allow us to analyse the process of formal deduction by means of semantic reasoning.

We use general models to study the deductive power of $\mathsf{S}(LAx2)$. Although $\mathsf{S}(LAx2)$ is as powerful as AHOL with respect to semantic expressiveness, we find out that the deductive closure of $LAx2$ is strictly smaller than that of *HOL*. This result motivates the choice of the latter set of axioms for general-purpose applications of S as a logical system.

Finally we introduce a special kind of general models and use them to obtain an important incompleteness result for $\mathsf{S}(LAx2)$ unrelated to AHOL.

6.1 Henkin's Theorem

In his doctoral thesis, Henkin [20] (also in [21]) introduces general models as a new interpretation for the higher-order calculus. He observes that with respect to general models, higher-order axiom systems are complete. Although Henkin only consideres a restricted set of deduction systems in detail, with a more or less fixed set of axioms and with custom-built rules of inference, it is easy to use his results within more general settings, including S.

Definition 6.1 (General Interpretation) Given (TC, VC, ty), a **general interpretation** $\mathcal{H}$ is a function with the following properties:

1. $\mathcal{H}$ provides denotations for type and value constants:
   $TC \cup VC \subseteq Dom\,\mathcal{H}$
2. Types are mapped onto non-empty sets:
   $\forall T \in Ty : \mathcal{H}T \neq \varnothing$

3. On the set of pre-terms $\mathcal{H}$ is defined recursively as follows:

$$\begin{array}{rcll} \mathcal{H}c\sigma & = & \mathcal{H}c & \\ \mathcal{H}x\sigma & = & \sigma x & \text{if } x \in Dom\,\sigma \\ \mathcal{H}(st)\sigma & = & \mathcal{H}s\sigma(\mathcal{H}t\sigma) & \text{if } \mathcal{H}t\sigma \in Dom(\mathcal{H}s\sigma) \\ \mathcal{H}(\lambda x : T.t)\sigma v & = & \mathcal{H}t(\sigma[x := v]) & \text{for all } v \in \mathcal{H}T \end{array}$$

Definition 6.2 (General Model) A general interpretation $\mathcal{H}$ is a **general model** if it provides denotations for all terms:

$$\forall T \in Ty \forall t \in Ter^T \Gamma \forall \sigma \in Sta(\mathcal{H}, \Gamma) : \mathcal{H}t\sigma \in \mathcal{H}T$$

Remark In standard interpretations, for all types T_1 and T_2, $\mathcal{D}(T_1 \to T_2)$ is the set of all functions from $\mathcal{D}T_1$ to $\mathcal{D}T_2$. Therefore, every standard interpretation is a model.

Let us formulate one of Henkin's most important results on general models in a form that will be useful to us later:

Theorem 9 (Henkin's Completeness and Soundness Theorem) *For every set of axioms A and for every equation E it holds*

$$A \vdash E \iff \text{for every general model } \mathcal{H} : \mathcal{H} \models A \Rightarrow \mathcal{H} \models E$$

Proof Essentially obtained by Friedman [14]. Friedman's proof in its original form applies to the simply typed λ-calculus with no value constants, parameterized with an empty set of axioms, but his approach can easily be generalized to S. □

We will use general models in conjunction with Henkin's Soundness Theorem to show non-provability of equations and incompleteness of logical systems. Whenever we want to show that an equation E is not provable from a set of axioms A we can achieve this by making use of Henkin's theorem. All we need to do is to find a general model satisfying A but not E. By the soundness result, we conclude $A \not\vdash E$.

6.2 Deductive Power of $LAx2$

We know now that every semantic property we can represent in AHOL can also be formalized using $\mathsf{S}(LAx2)$. Of course, this does not automatically mean that every theorem in Andrews' logic can also be derived from $LAx2$. Indeed, $LAx2$ alone turns out to have a deductive closure that does not even contain all of Andrews' axioms.

If we wanted to show that the deductive closure of $LAx2$ contained all the theorems of AHOL, we had to prove AAx being theorems of our system.

Furthermore, we needed to show that our axioms and rules of inference suffice in order to simulate inference in Andrews' system.

$A1$ can be derived from $LAx2$ by Proposition 5.10 and Proposition 2.7. By Lemma 2.14, the same holds for $A2$. However, as we show in the following, general models satisfying $LAx2$ need not be extensional. Therefore, $LAx2 \not\vdash A3$.

We prove our claim by constructing a non-extensional general model satisfying $LAx2$.

Definition 6.3 Let $\mathcal{N}$ be the general interpretation defined as follows:

- $\mathcal{N}\mathsf{B} = \{0, 1\}$
- $\mathcal{N}(T_1 \to T_2)$ contains every function from $\mathcal{N}T_1 \to \mathcal{N}T_2$ twice, i.e. for every $f \in \mathcal{N}T_1 \to \mathcal{N}T_2$, $\mathcal{N}(T_1 \to T_2)$ contains two distinct objects f_1 and f_2 such that
$$\forall x \in \mathcal{N}T_1 : f_1x = f_2x = fx$$
$\mathcal{N}(T_1 \to T_2)$ contains no further objects.
- $\mathcal{N}0 = 0$, $\mathcal{N}1 = 1$
- $\mathcal{N}(\mathsf{B} \to \mathsf{B})$ contains two appropriate denotations for Boolean negation. Let $\mathcal{N}\neg$ be either of them.
- Let $\wedge$ and $\vee$ denote an arbitrary function representing conjunction and disjunction respectively. In both cases we can choose from four denotations.
- Let $f \in \mathcal{N}(T \to \mathsf{B})$ for some type T. Since $\mathcal{N}T$ is finite, we know that $\inf\{fx | x \in \mathcal{N}T\}$ exists.
Therefore, $\mathcal{N}((T \to \mathsf{B}) \to \mathsf{B})$ contains two functions g_1 and g_2 satisfying
$$\forall f \in \mathcal{N}(T \to \mathsf{B}) : g_1 f = g_2 f = \inf\{fx | x \in \mathcal{N}T\}$$
Let $\mathcal{N}\forall_T$ be either of them.
- Whenever there is a choice for $\mathcal{N}t\sigma$ between f_1 and f_2, choose f_1.

Proposition 6.1 *$\mathcal{N}$ is a general model.*

Proof Let $\Gamma \vdash t : T$ where $T = T_1 \to \ldots \to T_n$. Let $\sigma \in \mathit{Sta}(\mathcal{N}, \Gamma)$. We show that $\mathcal{N}$ provides a denotation for t relatively to σ by induction on the structure of t:

1. $t = x$: $\mathcal{N}t\sigma = \sigma x \in \mathcal{N}T$ by the definition of $\mathit{Sta}(\mathcal{N}, \Gamma)$.
2. $t = c$: $\mathcal{N}t\sigma = \mathcal{N}c \in \mathcal{N}T$ by Definition 6.1.
3. $t = (t_1t_2)$: Let $\Gamma \vdash t_1 : T' \to T$. By induction, $\mathcal{N}t_1\sigma \in \mathcal{N}(T' \to T)$ and $\mathcal{N}t_2\sigma \in \mathcal{N}T'$. By Definition 6.3, there exists a function $f \in \mathcal{N}T' \to \mathcal{N}T$ such that
$$\forall x \in \mathcal{N}T' : \mathcal{N}t_1\sigma x = fx$$
Then
$$\mathcal{N}t\sigma = \mathcal{N}t_1\sigma(\mathcal{N}t_2\sigma) = f(\mathcal{N}t_2\sigma) \in \mathcal{N}T$$

4. $t = \lambda x : T_1.t'$: By induction hypothesis, $\mathcal{N}t'\sigma \in \mathcal{N}(T_2 \to \ldots \to T_n)$. Therefore, $\mathcal{N}t'(\sigma[x := v]) \in \mathcal{N}(T_2 \to \ldots \to T_n)$ for all $v \in \mathcal{N}(\Gamma x)$. By Definition 6.1

$$\mathcal{N}t\sigma v = \mathcal{N}(\lambda x : T_1.t')\sigma v = \mathcal{N}t'(\sigma[x := v]) \in \mathcal{N}(T_2 \to \ldots \to T_n)$$

for all $v \in \mathcal{N}T_1$. By Definition 6.3, $\mathcal{N}T$ contains two objects f_1, f_2 satisfying

$$\forall v \in \mathcal{N}T_1 : f_1 v = f_2 v = \mathcal{N}t\sigma v$$

Then, again by Definition 6.3, $\mathcal{N}t\sigma = f_1$. In particular, $\mathcal{N}t\sigma$ exists. □

Proposition 6.2 $\mathcal{N} \vDash LAx2$

Proof Is an immediate consequence of Definition 6.3. □

Proposition 6.3 $\mathcal{N} \nvDash A3$

Proof Let $f' \in \mathcal{N}T_1 \to \mathcal{N}T_2$ for some types T_1, T_2. Let $f_1, f_2 \in \mathcal{N}(T_1 \to T_2)$ be two distinct values such that

$$\forall x \in \mathcal{N}T_1 : f_1 x = f_2 x = f' x$$

Let σ be an assignment such that

- $\sigma f = f_1$
- $\sigma g = f_2$
- $\sigma h v = \text{if } v = f_1 \text{ then } 0 \text{ else } 1$

Then

$$\begin{array}{rcll} \mathcal{N}(f \doteq g)\sigma & = & \mathcal{N}(\forall h.hf \to hg)\sigma & \text{def} \doteq \\ & = & \mathcal{N}((\forall h.hf \to hg) \wedge (hf \to hg))\sigma & \forall I \\ & = & \mathcal{N}((\forall h.hf \to hg) \wedge (0 \to 1))\sigma & \text{def}\,\sigma \\ & = & \mathcal{N}0 & BAx \end{array}$$

We know

$$\mathcal{N}(fx)\sigma = f_1(\sigma x) = f_2(\sigma x) = \mathcal{N}(gx)\sigma$$

Proposition 5.6 holds for $\mathcal{N}$ as it does for standard models. The generalization of the proof is straightforward. By applying Proposition 5.6 we obtain

$$\begin{array}{rrcll} & \mathcal{N}(fx \doteq gx)\sigma & = & \mathcal{N}1 & \\ \Longrightarrow & \mathcal{N}(\forall x.fx \doteq gx)\sigma & = & \mathcal{N}(\forall x.1)\sigma & \\ & & = & \mathcal{N}1 & \forall E \end{array}$$

Thus, we have

$$\begin{array}{rcll} \mathcal{N}(f \doteq g)\sigma & = & \mathcal{N}0 & (*) \\ \mathcal{N}(\forall x.fx \doteq gx)\sigma & = & \mathcal{N}1 & (**) \end{array}$$

$$
\begin{array}{lll}
\mathcal{N}(f \doteq g \doteq \forall x.fx \doteq gx)\sigma & & \\
\quad = \mathcal{N}(\forall i.i(f \doteq g) \to i(\forall x.fx \doteq gx))\sigma & & \text{def } \doteq \\
\quad = \mathcal{N}(\forall i.i0 \to i1)\sigma & & \text{by } (*), (**) \\
\quad = \mathcal{N}((\forall i.i0 \to i1) \wedge (0 \to 1))\sigma & & \forall I \\
\quad = \mathcal{N}0 & & BAx \\
\quad \neq \mathcal{N}1 & &
\end{array}
$$

□

Proposition 6.4 $LAx2 \nvdash A3$

Proof Follows immediately from Proposition 6.3 and Theorem 9. □

Remark Andrews [2] obtains an analogous result for the system used by Henkin [21] in his proof of Theorem 9. However, our proof is completely independent from Andrews' approach. Moreover, the general model constructed within our proof contradicts Andrews' claim that every general model $\mathcal{H}$ of Henkin's system is extensional if $\mathcal{H}(T \to T \to \mathsf{B})$ contains the identity test for every type T.

Theorem 10 *The deductive closure of $LAx2$ in* S *is strictly smaller than that of Andrews' axioms in* AHOL*.*

Proof The claim is implied by the the following three statements:

1. Every axiom from $LAx2$ can be derived from Andrews' axioms using **R**.
2. **R** can simulate every rule of inference in S.
3. Proposition 6.4.

The first two statements are for the most part proved by Andrews [4]. The remaining proofs are straightforward. □

Corollary 6.5 *The deductive closure of $LAx2$ is strictly smaller than that of HOL.*

Open Problem 2 We are able to deduce $A1$ from $LAx2$, whereas it does not seem possible to obtain the same result from LAx alone by restricting quantification to binary functions. While we have proved that $B2$ contributes nothing to the semantic expressiveness of a logical system, systems including $B2$ seem to have a greater deductive power than those without.

6.3 Dependent Models

Since general interpretations allow smaller functional spaces than standard interpretations, it is often not obvious, whether a general interpretation is a model or not. Andrews [3] discusses certain closure conditions that are satisfied by an interpretation if and only if it is a model. Nevertheless, constructing general models stays a difficult task.

To facilitate this task, we develop a special construction principle for general interpretations. We do not build a general interpretation $\mathcal{H}$ from scratch, but use a well-understood standard interpretation $\mathcal{D}$ as a basis, such that

- $\forall T \in Ty : \mathcal{H}T \subseteq \mathcal{D}T$
- $\forall t \in Ter\,\Gamma, \sigma \in Sta(\mathcal{H}, \Gamma) : \mathcal{H}t\sigma = \mathcal{D}t\sigma$

We call such interpretations **dependent** since their semantic structure depends on the underlying standard interpretation.

The tricky part in the definition of a dependent interpretation $\mathcal{H}$ is the characterization of its domains. These are constructed with the help of $\mathcal{D}$ as follows:

$$\mathcal{H}T = \{\mathcal{D}t\sigma | \sigma \in Sta(\mathcal{D}, \Gamma),\ \Gamma \vdash t : T,\ P(t)\}$$

where $P(t)$ is some constraint imposed on the structure of t.

Why do we need $P(t)$? Suppose, we dropped it. Then t would be allowed to be any term. In particular, we would allow $t = x$ for any variable $x : T$. Clearly,

$$\{\sigma x | \sigma \in Sta(\mathcal{D}, \Gamma),\ \Gamma x = T\} = \mathcal{D}T$$

So, we need $P(t)$ to enforce $\mathcal{H}T \subsetneq \mathcal{D}T$ at least for some T.

The way we construct the domains of $\mathcal{H}$ gives us a relatively convenient way to determine whether $\mathcal{H}$ is a model. What we need to check is whether every term, regardless of its structure, denotes a value that can be denoted by some term satisfying P.

In the following, we want to construct two dependent models. The first one, $\mathcal{K}_0$, serves mainly to explain the construction principle, whereas the second one, $\mathcal{K}$, is later used to prove an interesting incompleteness result concerning $\mathsf{S}(LAx2)$.

6.3.1 $\mathcal{K}_0$ and Finite Models

Henkin [22] proves that in higher-order logic with identity and descriptions every finite model is standard. This restriction does not apply to our system in general. This is not surprising, since we neither require every instance of S to include identity or descriptions, nor do we restrict S to allow only extensional models. Nevertheless, we want to prove that S allows finite non-standard models by constructing such a model for a simple instance of S.

How do we construct a finite non-standard model? Consider the finite domain structure of $\mathcal{T}_2$. Apparently, negation in $\mathcal{T}_2$ cannot be represented by any term containing no value constants if all its free variables are first-order. We construct an interpretation $\mathcal{K}_0$ based on denotations of such terms in a standard finite model and show the constructed interpretation being a model. If we succeed in proving that we cannot represent negation by terms t with $\max\{\mathrm{ord}(\Gamma x) | x \in FV\,t\} = 1$, the constructed model is indeed non-standard.

Definition 6.4 Let (TC, VC, ty) be a signature such that $TC = \{\mathsf{B}\}$ and $VC = \varnothing$. Let $\mathcal{D}_{\mathbb{B}}$ be the standard interpretation built on the set $\{0, 1\}$, i.e. $\mathcal{D}_{\mathbb{B}}\mathsf{B} = \{0, 1\}$. We define the general interpretation $\mathcal{K}_0$ as follows:

- $\mathcal{K}_0 T = \{\mathcal{D}_{\mathbb{B}} t\sigma \mid \sigma \in \mathit{Sta}(\mathcal{D}_{\mathbb{B}}, \Gamma),$
  $\Gamma \vdash t : T,$
  $\max\{\mathrm{ord}(\Gamma x) | x \in FV\, t\} = 1\}$
- $\mathcal{K}_0 t\sigma = \mathcal{D}_{\mathbb{B}} t\sigma$

Remark Observe that since $\mathcal{K}_0 T \subseteq \mathcal{D}_{\mathbb{B}} T$ holds for all types T, it also holds $\mathit{Sta}(\mathcal{K}_0, \Gamma) \subseteq \mathit{Sta}(\mathcal{D}_{\mathbb{B}}, \Gamma)$. Therefore, $\mathcal{D}_{\mathbb{B}} t\sigma$ is well-defined for every term t and every $\sigma \in \mathit{Sta}(\mathcal{K}_0, \Gamma)$.

Remark $\mathcal{K}_0$ is indeed a general interpretation since $\mathcal{D}_{\mathbb{B}}$ is one. The only requirement in the definition of a general interpretation which is not obviously satisfied is the second one. However, if we consider an assignment σ with $\sigma y = 0$, we easily see that for all types $T = T_1 \to \ldots \to T_n \to \mathsf{B}$

$$\mathcal{K}_0 T \supseteq \{\mathcal{D}_{\mathbb{B}}(\lambda x_1 : T_1. \ldots \lambda x_n : T_n.y)\sigma\} \neq \varnothing$$

Proposition 6.6 *$\mathcal{K}_0$ is finite*

Proof Since $\mathcal{K}_0 T \subseteq \mathcal{D}_{\mathbb{B}} T$ for all types T and $\mathcal{D}_{\mathbb{B}}$ is finite, $\mathcal{K}_0$ is finite as well. □

In order to prove $\mathcal{K}_0$ being a model, we will make use of the following substitution lemma:

Lemma 6.7 *Let $\mathcal{H}$ be an arbitrary general model, t be a term with $y \notin FV\, t$, $\Gamma x = \Gamma y$ and $x \in \mathit{Dom}\, \sigma$. Then*

$$\mathcal{H} t\sigma = \mathcal{H}(t[x := y])(\sigma[y := \sigma x])$$

Proof By induction on the structure of t. □

Proposition 6.8 *$\mathcal{K}_0$ is a general model*

Proof We show that if $\Gamma \vdash t : T$, $\mathcal{K}_0 t\sigma \in \mathcal{K}_0 T$ holds for any $\sigma \in \mathit{Sta}(\mathcal{K}_0, \Gamma)$ by induction on the structure of t:

1. t primitive, i.e. $t = x$: Then $\mathcal{K}_0 t\sigma = \sigma x$. By definition of $\mathit{Sta}(\mathcal{K}_0, \Gamma)$, $\sigma x \in \mathcal{K}_0 T$.
2. t compound: Let $FV t = \{x_1, \ldots, x_n\}$. By induction hypothesis, for all $i \in \{1, \ldots, n\}$ it holds $\sigma x_i \in \mathcal{K}_0(\Gamma x_i)$, i.e. there exists an assignment σ_i and a term t_i with $\max\{\mathrm{ord}(\Gamma x) | x \in FV\, t_i\} = 1$ such that $\sigma x_i = \mathcal{K}_0 t_i \sigma_i$. By Lemma 6.7, we can assume without loss of generality that

$$\forall i, j : i \neq j \Rightarrow FV t_i \cap FV t_j = \varnothing$$

Then there exists a single assignment σ' such that for all i it holds $\mathcal{K}_0 t_i \sigma' = \mathcal{K}_0 t_i \sigma_i$. Let $t' = t[x_1 := t_1] \ldots [x_n := t_n]$. Then $\mathcal{K}_0 t\sigma = \mathcal{K}_0 t'\sigma'$. Observe that $\max\{\mathrm{ord}(\Gamma x) | x \in FV\, t'\} = 1$.
It holds $\mathcal{K}_0 t\sigma = \mathcal{K}_0 t'\sigma' = \mathcal{D}_{\mathbb{B}} t'\sigma'$. Therefore, $\mathcal{K}_0 t\sigma \in \mathcal{K}_0 T$. □

Lemma 6.9 *Let $\Gamma \vdash t : \mathsf{B} \to \mathsf{B}$. Let σ be an assignment. Then there exists an assignment σ' and a term t' such that*

- *$t' = \lambda x : \mathsf{B}.y$ with x and y not necessarily distinct*
- *$\mathcal{K}_0 t\sigma = \mathcal{D}_{\mathbb{B}} t'\sigma'$*

Proof By the definition of $\mathcal{K}_0$, there exists an assignment σ' and a term t'' with $\forall x \in FV\, t'' : \mathrm{ord}(\Gamma x) = 1$ such that $\mathcal{K}_0 t\sigma = \mathcal{D}_{\mathbb{B}} t''\sigma'$.

Let $t' = \lambda x : \mathsf{B}.y t_1 \ldots t_n$ be a $\beta\overline{\eta}$-normal form of t''. Since $FV\, t' \subseteq FV\, t''$, it holds $\forall x \in FV\, t' : \mathrm{ord}(\Gamma x) = 1$. Observe that $\mathrm{ord}(\Gamma y) = 1$ since

- either $y = x$ and $\mathrm{ord}(\Gamma x) = 1$,
- or $y \neq x$, but then $y \in FV\, t'$ and consequently $\mathrm{ord}(\Gamma y) = 1$.

Since $\Gamma[x := \mathsf{B}] \vdash y t_1 \ldots t_n : \mathsf{B}$, $n = 0$. □

Proposition 6.10 *Let t be a term and σ an assignment. Then*

$$\mathcal{K}_0 t\sigma \neq \lambda v \in \mathcal{K}_0 \mathsf{B}.1 - v$$

Proof Let w.l.o.g. $\Gamma \vdash t : \mathsf{B} \to \mathsf{B}$. Otherwise, the claim holds trivially. Let us write f for $\lambda v \in \mathcal{K}_0 \mathsf{B}.1 - v$. Notice that since 0 and 1 are distinct, f satisfies the following two inequalities:

- $f0 \neq f1$
- $\forall v \in \mathcal{K}_0 \mathsf{B} : fv \neq v$

By Lemma 6.9, the denotation of t equals to that of a term $t' = \lambda x : \mathsf{B}.y$. Let us write g for the denotation of t'. We need to consider two cases:

1. $x = y$: Then $g = \lambda v \in \mathcal{K}_0 \mathsf{B}.v \Longrightarrow gv = v$
2. $x \neq y$: Then $g = \lambda v \in \mathcal{K}_0 \mathsf{B}.w$ where $w \in \mathcal{K}_0 \mathsf{B} \Longrightarrow g0 = g1 = w$

Therefore, in both cases we have $g \neq f$, which completes the proof. □

Corollary 6.11 *$\mathcal{K}_0$ is non-standard.*

6.3.2 $\mathcal{K}$ and Identity

Leibniz' criterion is certainly appropriate in order to specify the identity test in standard models. However, when we consider non-standard models, we cannot rely on the criterion to be a sufficient characterization of identity. As a consequence, certain propositions that are obviously valid in standard models turn out to be non-provable. We have seen an example for this in Section 2.2, when we stated the non-validity of the reverse direction of Proposition 2.7. Let us now consider why it is the case.

Definition Let the function tc be defined as follows:

$$\begin{aligned} \mathrm{tc}(B) &= \{B\} \\ \mathrm{tc}(T_1 \to T_2) &= \mathrm{tc}(T_1) \cup \mathrm{tc}(T_2) \end{aligned}$$

Remark $\mathrm{tc}(T)$ returns the set of type constants occuring in T.

Definition 6.5 Let (TC, VC, ty) be a signature such that

- $TC = \{\mathsf{B}, \mathsf{C}\}$
- $VC = \{0, 1, \neg, \wedge, \vee\} \cup \{\forall_T | T \in Ty\}$
- ty is defined as follows:

$$\begin{aligned} 0, 1 &: \mathsf{B} \\ \neg &: \mathsf{B} \to \mathsf{B} \\ \wedge, \vee &: \mathsf{B} \to \mathsf{B} \to \mathsf{B} \\ \forall_T &: (T \to \mathsf{B}) \to \mathsf{B} \text{ for all } T \in Ty \end{aligned}$$

We define a standard Boolean algebra $\mathcal{D}_\mathcal{K}$ and a general interpretation $\mathcal{K}$ by mutual recursion on the order of the type T in $\mathcal{K}T$:

- $\mathcal{D}_\mathcal{K}\mathsf{B} = \mathcal{P}(S)$ where $S = \{\varnothing\}$
- $\mathcal{D}_\mathcal{K}\mathsf{C} = \{\bot, \top\}$
- $\mathcal{D}_\mathcal{K}0 = \varnothing$
- $\mathcal{D}_\mathcal{K}1 = S$
- $\mathcal{D}_\mathcal{K}\neg = \lambda x \in \mathcal{D}_\mathcal{K}\mathsf{B}.S - x$
- $\mathcal{D}_\mathcal{K}\wedge = \lambda x \in \mathcal{D}_\mathcal{K}\mathsf{B}.\lambda y \in \mathcal{D}_\mathcal{K}\mathsf{B}.x \cap y$
- $\mathcal{D}_\mathcal{K}\vee = \lambda x \in \mathcal{D}_\mathcal{K}\mathsf{B}.\lambda y \in \mathcal{D}_\mathcal{K}\mathsf{B}.x \cup y$
- $\mathcal{D}_\mathcal{K}\forall_T = \lambda f \in \mathcal{D}_\mathcal{K}(T \to \mathsf{B}).\inf\{fv | v \in \mathcal{K}T\}$
- $\mathcal{K}T = \{\mathcal{D}_\mathcal{K}t\sigma | \sigma \in Sta(\mathcal{D}_\mathcal{K}, \Gamma),$
  $\Gamma \vdash t : T,$
  $\mathrm{ran}\, t = \mathsf{B} \Longrightarrow \forall x \in FV\, t : \mathsf{C} \notin \mathrm{tc}(\Gamma x)\}$
- $\mathcal{K}t\sigma = \mathcal{D}_\mathcal{K}t\sigma$

Remark Note that $\mathcal{D}_\mathcal{K}\forall$ always exists since both $\mathcal{D}_\mathcal{K}$ and $\mathcal{K}$ are finite and therefore complete.

Remark As in the case of Definition 6.4, we can easily verify that $\mathcal{K}T \neq \varnothing$ holds for every type T.

Lemma 6.12 *Let t be a $\beta\overline{\eta}$-normal term with* $\mathrm{ran}\, t = \mathsf{B}$*, $S \subseteq Var$. If*

$$\forall x \in FV\, t - S : \mathrm{ran}(\Gamma x) = \mathsf{B} \Longrightarrow \mathsf{C} \notin \mathrm{tc}(\Gamma x)$$

then

$$\forall x \in FV\, t - S : \mathsf{C} \notin \mathrm{tc}(\Gamma x)$$

Proof By induction on the structure of t:

1. $t = x$: Since $\operatorname{ran} t = \mathsf{B}$, $\operatorname{ran}(\Gamma x) = \mathsf{B}$. By assumption, $\mathsf{C} \notin \operatorname{tc}(\Gamma x)$.
2. $t = c$: $FV\,t - S \subseteq FV\,t = \varnothing$. Thus, the claim is trivially true.
3. $t = \lambda x_1 : T_1. \dots \lambda x_n : T_n.t_0 t_1 \dots t_m$: Let $\Gamma' = \Gamma[x_1 := T_1, \dots, x_n := T_n]$, $S' = S \cup \{x_1, \dots, x_n\}$. By assumption and induction hypothesis, for all $i \in \{1, \dots, m\}$ it holds

$$\forall x \in FV\,t_i - S' : \mathsf{C} \notin \operatorname{tc}(\Gamma' x)$$

By the definition of $\beta\overline{\eta}$-normal form, t_0 is primitive. We consider two cases:

(a) $t_0 = c$: Then $FV\,t \subseteq \bigcup_{i=1}^{m} FV\,t_i$ and we are done.

(b) $t_0 = x$: Let w.l.o.g. $x \in FV\,t$, otherwise $FV\,t \subseteq \bigcup_{i=1}^{m} FV\,t_i$ and we are done. Since $\Gamma' \vdash t_0 t_1 \dots t_m : \mathsf{B}$, it holds $\operatorname{ran}(\Gamma' x) = \mathsf{B}$. By assumption, $\mathsf{C} \notin \operatorname{tc}(\Gamma' x)$ and we are done. □

Proposition 6.13 *$\mathcal{K}$ is a general model*

Proof We show that if $\Gamma \vdash t : T$, $\mathcal{K}t\sigma \in \mathcal{K}T$ holds for any $\sigma \in Sta(\mathcal{K}, \Gamma)$. We need to consider two cases:

- $\operatorname{ran}(T) = \mathsf{C}$: Let t' be a $\beta\overline{\eta}$-normal form of t. Then

$$\mathcal{K}t\sigma = \mathcal{K}t'\sigma = \mathcal{D}_{\mathcal{K}}t'\sigma \in \mathcal{K}T$$

- $\operatorname{ran}(T) = \mathsf{B}$: We proceed by induction on the structure of t:
  1. $t = x$: Then $\mathcal{K}t\sigma = \sigma x$. By the definition of $Sta(\mathcal{K}, \Gamma)$, $\sigma x \in \mathcal{K}T$.
  2. $t = c$: By Definition 6.1, $\mathcal{K}t\sigma = \mathcal{K}c \in \mathcal{K}T$.
  3. t compound: Let

$$\{x_1, \dots, x_m\} = \{x | x \in FV\,t,\ \operatorname{ran}(\Gamma x) = \mathsf{B}\}$$

By induction hypothesis, for all $i \in \{1, \dots, m\}$ it holds $\sigma x_i \in \mathcal{K}(\Gamma x_i)$. In particular, there exists an assignment σ_i and a $\beta\overline{\eta}$-normal term t_i with $\forall x \in FV\,t_i : \mathsf{C} \notin \operatorname{tc}(\Gamma x)$ such that $\sigma x_i = \mathcal{K}t_i\sigma_i$.
By Lemma 6.7, we can assume without loss of generality that

$$\forall i, j : i \neq j \Rightarrow FV t_i \cap FV t_j = \varnothing$$

Then there exists a single assignment σ' such that for all i it holds $\mathcal{K}t_i\sigma' = \mathcal{K}t_i\sigma_i$. Let $t' = t[x_1 := t_1] \dots [x_m := t_m]$. Then

$$\mathcal{K}t\sigma = \mathcal{K}t'\sigma'$$

Observe that

$$\forall x \in FV\,t' : \operatorname{ran}(\Gamma x) = \mathsf{B} \Longrightarrow \mathsf{C} \notin \operatorname{tc}(\Gamma x)$$

Let t'' be a $\beta\overline{\eta}$-normal form of t'. Then $FVt'' \subseteq= FVt'$ and the above statement holds for t'' as well. By Lemma 6.12, it holds

$$\forall x \in FV\, t'' : \mathsf{C} \notin \mathrm{tc}(\Gamma x)$$

Then

$$\mathcal{K}t\sigma = \mathcal{K}t''\sigma' = \mathcal{D}_{\mathcal{K}}t''\sigma' \in \mathcal{K}T$$

□

Proposition 6.14 *$\mathcal{K} \vDash LAx2$.*

Proof Like $\mathcal{D}_{\mathcal{K}}$, $\mathcal{K}$ is a finite and therefore complete Boolean algebra built on a two-valued set. It remains to show that $\mathcal{K}$ satisfies QAx. By the definition of $\mathcal{D}_{\mathcal{K}}$, it holds

$$\mathcal{K}\forall_T f = \mathcal{D}_{\mathcal{K}}\forall_T f = \inf\{fv | v \in \mathcal{K}T\}$$

for all $f \in \mathcal{K}(T \to \mathsf{B})$. Since $\mathcal{K}$ is a set algebra, Lemma 5.3 can be generalized to $\mathcal{K}$. □

Lemma 6.15 *For every $f \in \mathcal{K}(\mathsf{C} \to \mathsf{B})$ and for every $v, w \in \mathcal{K}\mathsf{C}$ it holds*

$$fv = fw$$

Proof Let $f \in \mathcal{K}(\mathsf{C} \to \mathsf{B})$. Then there exists some term t and an assignment σ such that $f = \mathcal{K}t\sigma$. Let

$$t'' = \lambda x : \mathsf{C}.t' \text{ where } t' = t_0 t_1 \dots t_n$$

be a $\beta\overline{\eta}$-normal form of t. Then $FV\, t' \subseteq FV\, t'' \cup \{x\} \subseteq FV\, t \cup \{x\}$. By Definition 6.5, we have

$$\forall x \in FV\, t : \mathsf{C} \notin \mathrm{tc}(\Gamma x)$$

Let $\Gamma' = \Gamma[x := \mathsf{C}]$. Since $\mathrm{ran}\,(\Gamma' x) = \mathsf{C}$, we still have

$$\forall x \in FV\, t' : \mathrm{ran}(\Gamma' x) = \mathsf{B} \Longrightarrow \mathsf{C} \notin \mathrm{tc}(\Gamma' x)$$

By Lemma 6.12, we obtain

$$\forall x \in FV\, t' : \mathsf{C} \notin \mathrm{tc}(\Gamma' x)$$

Since $\mathsf{C} \in \mathrm{tc}(\Gamma' x)$, this implies $x \notin FV\, t'$. Therefore, for every $v, w \in \mathcal{K}\mathsf{C}$ it holds:

$$fv = \mathcal{K}t\sigma v = \mathcal{K}t'(\sigma[x := v]) = \mathcal{K}t'(\sigma[x := w]) = \mathcal{K}t\sigma w = fw$$

□

Proposition 6.16 *Let $s, t : \mathsf{C}$. Then for every assignment σ it holds*

$$\mathcal{K}(s \doteq t)\sigma = \mathcal{K}1$$

Proof Let $g = \sigma f$, $v = \mathcal{K}s\sigma$ and $w = \mathcal{K}t\sigma$.

$$\begin{array}{rcll} \mathcal{K}(fs \to ft)\sigma & = & (S - gv) \cup gw & \\ & = & (S - gv) \cup gv & \text{by Lemma 6.15} \\ & = & S & \text{set theory} \\ & = & \mathcal{K}1 & (*) \end{array}$$

Then

$$\begin{array}{rcll} \mathcal{K}(s \doteq t)\sigma & = & \mathcal{K}(\forall f. fs \to ft)\sigma & \\ & = & \mathcal{K}(\forall f.1)\sigma & \text{by } (*) \\ & = & \mathcal{K}1 & \forall I \end{array}$$

□

We now have everything neccesary to prove that the reverse direction of Proposition 2.7 does not hold.

Theorem 11 *There exist terms s, t such that $LAx2, s \doteq t = 1 \not\vdash s = t$*

Proof Let x, y be distinct variables with $\Gamma x = \Gamma y = \mathsf{C}$. By Propositions 6.14 and 6.16, $\mathcal{K} \models LAx2 \cup \{x \doteq y = 1\}$.

Let σ be an assignment with $\sigma x = \bot$ and $\sigma y = \top$. Then

$$\begin{array}{rc} & \mathcal{K}x\sigma = \bot \neq \top = \mathcal{K}y\sigma \\ \Longrightarrow & \mathcal{K} \not\models x = y \end{array}$$

The claim follows by Theorem 9. □

Open Problem 3 So far we do not know whether $HOL, s \doteq t = 1 \vdash s = t$. Our intuition suggests that this is not the case. A corresponding proof could possibly be obtained by showing $\mathcal{K} \models HOL$.

Chapter 7

Conclusion and Further Work

We have presented S as an alternative definition of higher-order logic. As we have seen, S instantiated with different sets of axioms generates logical systems with differing structure and expressivity. In particular, we considered $\mathsf{S}(HOL)$ and showed this system deductively equivalent to AHOL. The equivalence in semantic expressiveness was observed even for a deductively weaker set of the logical axioms LAx. Thus, we have shown S to be an adequate formulation of higher-order logic, well-suited for general-purpose application.

Amongst other things, we considered systems based on S instantiated with first-order axioms and proved them complete with respect to standard models. When using such higher-order systems, like $\mathsf{S}(BAx)$, we can rely on the fact that any valid equality of the logical system can be formally proved within the system, which obviously increases the system's practical applicability.

The investigation of S and related systems is far from complete. Although we now have some understanding of our system's expressiveness with respect to standard models, we have not much knowledge about the semantics of our system for general models. This knowledge is important since it would allow us to draw further conclusions about deduction in S. Of particular interest might be the role of descriptions and their influence on the semantics of the internal identity test for non-standard models.

We have seen that in $\mathsf{S}(LAx2)$ the internal identity test, though semantically equivalent to external identity relative to standard models, is weaker than external identity with respect to deduction. The proof was obtained using a specially constructed general model $\mathcal{K}$. It is not yet clear whether the same difference between internal and external identity exists in $\mathsf{S}(HOL)$. It may be possible to extend $\mathcal{K}$ to satisfy HOL, thus proving this assumption. This would include showing $\mathcal{K}$ being extensional. If $\mathcal{K}$ cannot be extended to satisfy *HOL*, perhaps we can construct a more appropriate model or show the deductional equivalence of the two types of identity.

By extending the equality rules with **Id** we destroy the generic nature of deduction in S. As we have noticed, this generic nature can be regained at a higher level of abstraction by introducing conditional equations [44]. In such an extended system, rules of inference could be derived in the same way we do it in S for ordinary equations. Such a formalism could find useful application in proof assistants and other practically valuable systems. It is certainly an interesting extension of our system that should be explored in detail.

We have shown the completeness of S for first-order axioms, but we did not obtain any results about whether the validity of equations in S is decidable. Finding an algorithm to efficiently decide the validity of propositions in certain subsystems of S would be an important contribution to the system's usefulness. Research in this area could possibly be based on Meinke's results on term rewriting in higher-order equational logic [27] or on Statman's work [39, 41].

Further, we could weaken some restrictions on the form of the axioms in our completeness result, like e.g. admitting constants of order greater than 2, and analyze the consequences, possibly obtaining a stronger version of the completeness theorem. A stronger result might also be obtained by modifying Friedman's completeness proof for the simply typed λ-calculus [14].

Bibliography

[1] Akama, Y. On Mints' reduction for ccc-calculus. In *Proceedings of the International Conference on Typed Lambda Calculi and Applications (TLCA '93)* (1993), M. Bezem and J. Groote, Eds., vol. 664 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 1–12.

[2] Andrews, P. B. General models and extensionality. *Journal of Symbolic Logic 37* (1972), 395–397.

[3] Andrews, P. B. General models, descriptions and choice in type theory. *Journal of Symbolic Logic 37* (1972), 385–394.

[4] Andrews, P. B. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof*, second ed., vol. 27 of *Applied Logic Series*. Kluwer Academic Publishers, 2002.

[5] Andrews, P. B., and Bishop, M. On sets, types, fixed points, and checkerboards. In *Theorem Proving with Analytic Tableaux and Related Methods. 5th International Workshop. (TABLEAUX '96)* (May 1996), P. Miglioli, U. Moscato, D. Mundici, and M. Ornaghi, Eds., vol. 1071 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, pp. 1–15.

[6] Baader, F., and Nipkow, T. *Term Rewriting and All That.* Cambridge University Press, 1998.

[7] Boolos, G. *Logic, logic and logic.* Harvard University Press, 1998.

[8] Church, A. A formulation of the simple theory of types. *Journal of Symbolic Logic 5*, 1 (1940), 56–68.

[9] Davey, B. A., and Priestley, H. A. *Introduction to Lattices and Order*, second ed. Cambridge University Press, 2002.

[10] de Vrijer, R. C. Exactly estimating functionals and strong normalization. *Indagationes Mathematicae 49*, 4 (1987), 479–493.

[11] di Cosmo, R., and Kesner, D. Simulating expansions without expansions. *Mathematical Structures in Computer Science 4* (1994), 315–362.

[12] Dragalin, A. G. The computability of primitive recursion terms of finite type, and primitive recursive realization. *Seminars in Mathematics* (1968), 32–45. In Russian.

[13] Frege, G. Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens. Halle, 1879. Translated in [45], pp. 1–82.

[14] Friedman, H. Equality between functionals. In *Proceedings of the Logic Colloquium 72-73* (1975), R. Parikh, Ed., vol. 453 of *Lecture Notes in Mathematics*, Springer-Verlag, pp. 22–37.

[15] Gandy, R. O. Proofs of strong normalization. In *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, J. R. Hindley and J. P. Seldin, Eds. Academic Press, 1980, pp. 457–477.

[16] Gödel, K. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I. *Monatshefte für Mathematik und Physik 38* (1931), 173–198. Translated in [45], pp. 596–616.

[17] Gödel, K. Über die Länge von Beweisen. *Ergebnisse eines Mathematischen Kolloquiums 7* (1936), 23–24. Translated in [18], pp. 396–399.

[18] Gödel, K. *Collected Works, Volume I.* Oxford University Press, 1986.

[19] Hanatani, Y. Calculabilité des fonctionelles récursives primitives de type fini sur les nombres naturels. *Annals of the Japan Association for Philosophy of Science 3* (1966), 19–30.

[20] Henkin, L. *The completeness of formal systems.* Thesis in candidacy for the degree of doctor of philosophy, Princeton University, 1947.

[21] Henkin, L. Completeness in the theory of types. *Journal of Symbolic Logic 15*, 2 (June 1950), 81–91.

[22] Henkin, L. A theory of propositional types. *Fundamenta Mathematicae 52* (1963), 323–344.

[23] Henkin, L. Identity as a logical primitive. *Philosophia 5* (1975), 31–45.

[24] Hinata, S. Calculability of primitive recursive functionals of finite type. *Science Reports of the Tokyo Kyoiku Daigaku A 9* (1967), 218–235.

[25] Hindley, J. R. *Basic Simple Type Theory*, vol. 42 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 1997.

[26] Kamareddine, F., Laan, T., and Nederpelt, R. *A Modern Perspective on Type Theory. From its Origins until Today*, vol. 29 of *Applied Logic Series.* Kluwer Academic Publishers, 2004.

[27] Meinke, K. Proof theory of higher-order equations: conservativity, normal forms and term rewriting. *Journal of Computer and System Sciences 67* (2003), 127–173.

[28] Nipkow, T., Paulson, L. C., and Wenzel, M. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, vol. 2283 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.

[29] Owre, S., Rushby, J. M., and Shankar, N. PVS: A prototype verification system. In *11th International Conference on Automated Deduction (CADE)* (Saratoga, NY, June 1992), D. Kapur, Ed., vol. 607 of *Lecture Notes in Artificial Intelligence*, Springer-Verlag, pp. 748–752.

[30] Pierce, B. C. *Types and Programming Languages*. The MIT Press, 2002.

[31] Plotkin, G. D. Lambda-definability and logical relations. Memorandum SAI-RM-4, University of Edinburgh, 1973.

[32] Plotkin, G. D. Lambda-definability in the full type theory. In *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, J. R. Hindley and J. P. Seldin, Eds. Academic Press, 1980, pp. 365–373.

[33] Russell, B. Mathematical logic as based on the theory of types. *American Journal of Mathematics 30* (1908), 222–262.

[34] Skolem, T. Begründung der elementaren Arithmetik durch die rekurrierende Denkweise ohne Anwendung scheinbar Veränderlichen mit unendlichem Ausdehnungsbereich. *Videnskapsselskapets skrifter, I. Matematisk-naturvidenskabelig klasse*, 6 (1923), 1–38. Translated in [45], pp. 302–333.

[35] Skolem, T. Über die Unmöglichkeit einer vollständigen Charakterisierung der Zahlenreihe mittels eines endlichen Axiomensystems. *Norskmatematisk forenings skrifter 2*, 10 (1933), 73–82.

[36] Skolem, T. Über die Nicht-charakterisierbarkeit der Zahlenreihe mittels endlich oder abzählbar unendlich vieler Aussagen mit ausschließlich Zahlenvariablen. *Fundamenta mathematicae 23* (1934), 150–161.

[37] Smolka, G. *Introduction to Computational Logic: Lecture Notes*. Universität des Saarlandes, 2004. www.ps.uni-sb.de/courses/cl-ss04/script/.

[38] Statman, R. Bounds for proof search and speed-up in the predicate calculus. *Annals of Mathematical Logic 15* (1978), 225–287.

[39] Statman, R. Completeness, invariance and λ-definability. *Journal of Symbolic Logic 47*, 1 (1982), 17–26.

[40] Statman, R. Equality between functionals revisited. In *Harvey Friedman's Research on the Foundations of Mathematics*, L. A. Harrington et al., Eds. North-Holland, 1985, pp. 331–338.

[41] Statman, R., and Dowek, G. On statman's finite completeness theorem. Tech. Rep. CMU-CS-92-152, Carnegie Mellon University, 1992.

[42] Stone, M. H. The representation theorem for Boolean algebras. *Transactions of the American Mathematical Society 40* (1936), 37–111.

[43] Tait, W. Intensional interpretations of functionals of finite type I. *Journal of Symbolic Logic 32* (1967), 198–212.

[44] Terese. *Term Rewriting Systems*, vol. 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

[45] van Heijenoort, J., Ed. *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931*. Source Books in the History of the Sciences. Harvard University Press, 2002.

[46] van Oostrom, V. *Confluence for Abstract and Higher-Order Rewriting*. Dissertation, Vrije Universiteit, Amsterdam, 1994.

Index