Saarland University
Faculty of Natural Sciences and Technology I
Department of Computer Science
Master's Program in Computer Science

Master's Thesis

Completeness Results for Higher-Order Equational Logic

submitted by

Mark Kaminski

on 15.12.2006

Supervisor

Prof. Dr. Gert Smolka

Reviewers

Prof. Dr. Gert Smolka
Prof. Bernd Finkbeiner, Ph.D.

Statement

Hereby I confirm that this thesis is my own work and that I have documented all sources used.

Saarbrücken, 14.12.2006

Mark Kaminski

Declaration of Consent

Herewith I agree that my thesis will be made available through the library of the Computer Science Department.

Saarbrücken, 14.12.2006

Mark Kaminski

Abstract

We present several results concerning deductive completeness of the simply typed λ-calculus with constants and equational axioms.

First, we prove deductive completeness of the calculus with respect to standard semantics for axioms containing neither free nor bound occurrences of higher-order variables. Using this result, we analyze some fundamental deductive and semantic properties of axiomatic systems without higher-order variables and compare them to those of established logical frameworks like first-order logic and Church's higher-order logic.

Second, we present a finite higher-order equational formulation of Henkin's Propositional Type Theory (PTT) and prove its deductive completeness. We introduce a simple criterion which allows to reduce deductive completeness of systems with axiomatically defined constants to completeness of simpler axiomatic systems, and present an application of this criterion to our formulation of PTT.

Third, we prove the simply typed λ-calculus both with and without η-conversion complete with respect to general semantics. The result holds for systems with arbitrary axioms and constants.

Acknowledgements

I am very much indebted to my supervisor Prof. Gert Smolka for the time and the effort he invested into this thesis. He contributed many important ideas and enlightening insights without which the present work would not have been possible. I have greatly benefited from his scientific experience and his love for mathematical rigor and clarity of presentation.

I am grateful to Chad Brown for his sincere interest in my work and his help in improving my understanding of general semantics.

I also want to thank Prof. Bernd Finkbeiner who kindly agreed to review this thesis.

Contents

1 Introduction

1.1 Higher-Order Equational Logic

The simply typed λ-calculus, introduced by Church [13], is nowadays considered one of the most important formal frameworks in mathematical logic and computer science, both in its own right and as a basis for more expressive calculi. Fundamental properties of the pure simply typed λ-calculus, the simplest version of the calculus without constants or axioms, include decidability of deductive equality, essentially proven by Turing [17], and deductive completeness with respect to general and standard set-theoretic semantics, first shown by Friedman [16].

The equational proof system of the simply typed λ-calculus with constants can be used to investigate the logical consequences of arbitrary sets of equational axioms in the same way as first-order equational reasoning is used to study algebraic theories (see [33]). Therefore, in the same sense as first-order equational reasoning from algebraic axioms is called first-order equational logic, equational reasoning in the simply typed λ-calculus from higher-order axioms may be called higher-order equational logic.

The theory of $\beta\eta$-conversion in the pure simply typed λ-calculus, i.e. the set of all constant-free equations derivable by a finite number of $\beta\eta$-conversion steps, can then be seen as a particular theory of higher-order equational logic (λ-theory), namely the one generated by the empty set of axioms. Of course, higher-order equational logic allows us to specify many more interesting theories, like equational formulations of Church's higher-order logic (HOL) [13] or fragments thereof, Gödel's T [21] or Scott's PCF [41] (see also [40, 38, 42, 30, 45]).

Our interests for the thesis lie, on the one hand, in strengthening the existing completeness results for higher-order equational theories. Besides the standard completeness of $\beta\eta$-conversion, it is known that, in the absence of empty types, higher-order equational deduction from any set A of axioms is complete with respect to extensional general semantics as introduced by Henkin [23, 24, 33]. We are able to strengthen both of these results. As for standard completeness, we introduce a specific class of syntactically restricted equations, called plain equations, and show that Plotkin's version of the completeness proof [39] for the λ-theory of $\varnothing$ can be extended to axiomatic systems consisting of plain equations. The completeness result for general semantics is extended to non-extensional theories generated by the simply typed λ-calculus without η-conversion.

On the other hand, we are interested in higher-order equational logic as a tool for studying the algebraic theory of classical logic. We present and analyze several equational formulations of Henkin's Propositional Type Theory (PTT) [25]. In particular we show that PTT can be finitely axiomatized preserving its deductive completeness.

1.2 Overview

We begin our discussion by introducing our main formal framework for the thesis, namely the simply typed λ-calculus with equational deduction. We use the name S to refer to our formulation of the calculus. In Chapter 2 we present the syntax, deduction and standard semantics of S. Along the lines, we introduce the necessary terminology and notational conventions. We also state several fundamental theorems about the syntax and semantics of S. Some of the theorems will be presented together with a short proof or a proof sketch. As for the rest of the theorems, we rely on the reader's ability to verify their validity on his own. The necessary methodology and analogous proofs can be found in [33] and [48].

Chapter 3 presents a completeness result with respect to general semantics for the simply typed λ-calculus with arbitrary equational axioms. In contrast to known results [33], our considerations are not restricted to functionally extensional semantics, which allows us to establish, in addition to a completeness result for S, deductive completeness of a functionally non-extensional version of the calculus obtained from S by dropping the rule η. The overall structure of our proof is strongly inspired by the corresponding constructions by Friedman [16] and Mitchell [33].

Chapters 4 and 5 focus on completeness with respect to the standard semantics of S as introduced in Chapter 2, and are largely independent of the notions and constructions introduced in Chapter 3. Considerations involving general semantics are presented at the end of the two chapters, separately from the main results. A reader who is interested only in the standard semantics of S may skip them, together with Chapter 3.

In Chapter 4 we analyze the relation between semantic and deductive entailment from syntactically restricted sets of axioms. The axioms under consideration are called plain, and are restricted to contain neither free nor bound occurrences of higher-order variables. We prove that for every set A of plain axioms, equational deduction is complete with respect to the standard semantics of S by constructing a standard model of A that satisfies precisely the equations derivable from A. The model construction can be seen as an extension of Plotkin's construction for the pure simply typed λ-calculus [39]. Since plain terms are a proper superset of traditional first-order formulas, plain equations seem well-suited to give an intuitive (as compared to [31] or [34]) equational axiomatization of first-order logic. We analyze basic deductive and semantic properties of plain axiomatizations and conclude that they are indeed sufficient to encode traditional first-order logic with adequate precision.

In Chapter 5 we present an equational formulation of Henkin's Propositional Type Theory [25]. We give an axiomatization of PTT in S, called MT. In contrast to Henkin's set of axioms for PTT, MT is finite. We prove completeness of MT with respect to both Henkin's original semantics and the semantics of S. The completeness proof follows the basic ideas of Henkin's original proof. We show further that completeness of deduction for a given set of axioms is invariant under axiomatic introduction of new constants, provided that the corresponding values are expressible in the initial theory. Using this insight we prove completeness of two alternative formulations of PTT by reduction to completeness of MT.

1.3 Contributions

The contributions of this thesis are as follows:

1. We prove deductive completeness of the simply typed λ-calculus with respect to standard semantics for axioms that contain neither free nor bound occurrences of higher-order variables (in which case they are called plain).

   Using the above result, plain equational specifications are shown to have the first-order Löwenheim-Skolem property and the strong compactness property.

   We conclude that, with respect to both deduction and semantics, λ-theories induced by plain axioms are more expressive than traditional formulations of FOL and strictly less expressive than Church's HOL.

2. We present, for the first time, a finite axiomatization of Henkin's Propositional Type Theory, called MT. We show deductive completeness of MT by introducing and proving a necessary and sufficient deductive criterion.

   Additionally using results by Henkin [24] and Andrews [5], we show that our formulation of PTT requires all its Henkin models to be standard at the relevant types. From this we derive that Henkin semantics of PTT is uniquely determined by the standard higher-order extension of the two-element Boolean algebra (see [12, 14]). Finally, we prove strong compactness of MT.

   We prove that completeness of deduction in S is invariant under axiomatic introduction of new constants, provided that the corresponding values are expressible in the initial axiomatic system without the new constants.

   We apply the above result to prove deductive completeness of two additional formulations of PTT which, unlike MT, have primitive equality by reduction to completeness of MT.

3. We present a proof of deductive completeness with respect to general semantics for arbitrary equational axiomatic systems in the simply typed λ-calculus with or without η-conversion.

   Using the above result we prove the higher-order Löwenheim-Skolem theorem and the weak compactness theorem for non-extensional λ-theories.

2 Basics

The main goal of this chapter is to introduce S, our formulation of the simply typed λ-calculus with equational deduction. We briefly discuss the syntax, deduction and standard semantics of S. For a more detailed presentation see [45].

After introducing the basic concepts and notation, we demonstrate the semantic expressiveness of S by giving an equational axiomatization of number theory which satisfies the requirements of Gödel's first incompleteness theorem [20].

2.1 Syntax and Semantics

2.1.1 Terms

Let Par and Var be disjoint, countably infinite sets, and let Nam denote the set $Par \cup Var$. Elements of Par and Var are called **parameters** and **variables**, respectively. Elements of Nam, that is both parameters and variables, are also called **names**. We refer to elements of each of these sets by the following meta variables:

$$a, b, c \in Par \quad x, y, z, f, g \in Var \quad u \in Nam$$

The set Ter of **terms** is defined as usual (we write $s, t \in Ter$), and consists of names, applications, written st, and abstractions, written $\lambda x.t$. The definition is assumed to satisfy the equation $\lambda x.x = \lambda y.y$.

Terms that contain no abstractions are called **combinatory**.

The **variables occurring** in a term are given by a function $\mathcal{V} \in Ter \to \mathcal{P}(Var)$, characterized by the following recursive equations:

$$\begin{aligned} \mathcal{V}x &= \{x\} \\ \mathcal{V}c &= \varnothing \\ \mathcal{V}(st) &= \mathcal{V}s \cup \mathcal{V}t \\ \mathcal{V}(\lambda x.t) &= \mathcal{V}t - \{x\} \end{aligned}$$

A term t is called **closed** if $\mathcal{V}t = \varnothing$.

The **parameters occurring** in a term are given by a function $\mathcal{P} \in Ter \to \mathcal{P}(Par)$, characterized by the following recursive equations:

$$\begin{aligned} \mathcal{P}x &= \varnothing \\ \mathcal{P}c &= \{c\} \\ \mathcal{P}(st) &= \mathcal{P}s \cup \mathcal{P}t \\ \mathcal{P}(\lambda x.t) &= \mathcal{P}t \end{aligned}$$

The **names occurring** in a term are given by a function $\mathcal{N} \in Ter \to \mathcal{P}(Nam)$ such that, for all terms t, $\mathcal{N}t = \mathcal{V}t \cup \mathcal{P}t$.

Along with terms we introduce the notion of their **size**. The size of a term is given by a function $|\_| \in Ter \to \mathbb{N}$, satisfying the following equations:

$$\begin{aligned} |u| &= 1 \\ |st| &= |s| + |t| \\ |\lambda x.t| &= |t| + 1 \end{aligned}$$

The **subterms** of a term are given by a function $\mathcal{S} \in Ter \to \mathcal{P}(Ter)$ which is characterized as follows:

$$\begin{aligned} \mathcal{S}u &= \{u\} \\ \mathcal{S}(st) &= \{st\} \cup \mathcal{S}\,s \cup \mathcal{S}\,t \\ \mathcal{S}(\lambda x.t) &= \{\lambda x.t\} \cup \mathcal{S}\,t \end{aligned}$$

Note that according to our characterization, abstractions typically have infinitely many subterms. For instance, $\mathcal{S}(\lambda x.x) \supseteq Var$. Subterms of a term t which are not identical to t are called **proper**. Since they have no proper subterms, variables and parameters are called **atomic**.

The **substitution** of a term s for a name u in a term t is denoted by $t[u := s]$. Substitution is assumed capture free. When used as a function $Nam \to Ter$ or $Ter \to Ter$, it will be referred to by the meta variable θ. Since the behaviour of a substitution on terms is uniquely determined by its behaviour on names, we will usually define substitutions as functions $Nam \to Ter$, but apply them as functions $Ter \to Ter$. The **domain** of a substitution θ, written $\operatorname{dom}\theta$, is the set $\{u \in Nam \mid \theta u \neq u\}$. Unless explicitly mentioned otherwise, in the following we will always consider substitutions θ such that $|\operatorname{dom}\theta \cap Var| < \aleph_0$.

Given a substitution θ, we write $\theta[u := t]$ to denote the substitution (uniquely determined by) $\lambda u' \in Nam.\text{if } u' = u \text{ then } t \text{ else } \theta u'$. If u, u' are two distinct names, we write $\theta[uu' := st]$ for $(\theta[u' := t])[u := s]$. Note that $\theta[uu' := st] = \theta[u'u := ts]$. For convenience, we also allow the notation $\theta[uu := tt]$, which is assumed equivalent to $\theta[u := t]$. The notation is extended analogously to arbitrarily long finite sequences of names.

A substitution θ with $\operatorname{dom}\theta \subseteq Par$ is called **stable** for a term t if:
$\forall c \in \mathcal{P}t\ \forall x \in \mathcal{V}(\theta c) :\ x \notin \mathcal{V}t$.

2.1.2 Types and Signatures

Let Sor be a countably infinite set disjoint to Nam. Elements of Sor are called **sorts**. We refer to sorts by the meta variables B and C.

The set Ty of **types** consists of sorts, also called **base types**, and **functional types**. We use the meta variable T to refer to types. The notation T_1T_2 denotes the functional type from T_1 to T_2. It is assumed right associative.

A function $\boldsymbol{\tau} \in Ter \rightharpoonup Ty$ is called a **typing** if

1. $\boldsymbol{\tau} u \in Ty$,
2. $\boldsymbol{\tau}(st) = T$ if and only if for some T': $\boldsymbol{\tau} s = T'T$ and $\boldsymbol{\tau} t = T'$,
3. $\boldsymbol{\tau}(\lambda x.t) = (\boldsymbol{\tau} x)(\boldsymbol{\tau} t)$,
4. for every $T \in Ty$ there exist infinitely many variables x such that $\boldsymbol{\tau} x = T$,
5. for every $T \in Ty$ there exist infinitely many parameters a such that $\boldsymbol{\tau} a = T$.

We say that a term t has type T with respect to a typing $\boldsymbol{\tau}$ if $\boldsymbol{\tau} t = T$. A term is called **well typed** with respect to a typing $\boldsymbol{\tau}$ if it has a type with respect to $\boldsymbol{\tau}$.

In the following we fix a typing $\boldsymbol{\tau}$ and consider only terms which are well typed with respect to $\boldsymbol{\tau}$. We write $t : T$ to indicate that the term t is supposed to have the type T, i.e. $\tau t = T$. Moreover, we require substitution to be well-typed, i.e. we consider only those substitutions θ which satisfy $\boldsymbol{\tau} x = \boldsymbol{\tau}(\theta x)$ for all variables x.

Terms which have a base type are called **basic**.

It is sometimes useful to be able to refer to the set of terms, names, parameters or variables of a given type T. For this purpose we will use the notation Ter_T, Nam_T, Par_T, and Var_T, respectively.

An **equation** e of type T is a pair of terms (s, t) such that $\boldsymbol{\tau} s = \boldsymbol{\tau} t = T$. When there is no danger of confusion, an equation (s, t) will be written as $s = t$.

A **signature** is a set $\Sigma \subseteq Sor \cup Par$ such that for every $c \in \Sigma$, all sorts occurring in $\boldsymbol{\tau} c$ are contained in Σ.

A type T is said to be **licensed** by a signature Σ if all the sorts occurring in T are contained in Σ. A term t is **type-licensed** by a signature Σ if the type of every subterm of t is licensed by Σ. A term t is licensed by Σ if

1. t is type-licensed by Σ,
2. $\mathcal{P}t \subseteq \Sigma$.

We write $Ty(\Sigma)$ and $Ter(\Sigma)$ to denote the set of types and the set of terms licensed by Σ, respectively.

A basic combinatory term is called **algebraic** if it contains only basic variables and only algebraic parameters. A parameter is called algebraic if its type is of the form $C_0 C_1 \dots C_n$, for some $n \in \mathbb{N}$. In other words, our notion of an algebraic term describes precisely the same class of syntactic objects as the notion of a term in first-order predicate logic [5, 15].

Often we want to extend properties which we have defined on terms to equations or sets of equations. In such a case we always assume that a property which is defined on terms holds for an equation if it holds for both sides of the equation. Similarly, a property holds for a set of equations if it holds for every element of the set. For instance, an equation $s = t$ is licensed by a signature Σ if s and t are licensed by Σ.

2.1.3 Deduction

Deduction in S is based on the equational proof system of the simply typed λ-calculus, as presented in Figure 2.1. Similar deduction formalisms are treated in more detail in [33] and [48]. This formalism allows us to specify the desired logical properties of

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

$$\textbf{Cong}\ \frac{s = s' \quad t = t'}{st = s't'} \qquad \boldsymbol{\xi}\ \frac{s = t}{\lambda x.s = \lambda x.t}$$

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

Figure 2.1: S: Rules of Inference

mathematical objects we want to consider as equational axioms and then to derive further properties of these objects, again in the form of equations.

The notions of a formal derivation and derivability are defined as usual. We use the notation $A \vdash e$ to express that an equation e is formally derivable from a set of premises A. In this case we also say that e is a **theorem** of A, or that A **deductively entails** e. The set of all theorems derivable from a set A is called the **theory** (**λ-theory**) of A. Usually we fix some set A and study its theory. In this case elements of A are also called **axioms** of the theory.

It is convenient to extend the notion of deductive entailment to sets of equations. We write $A \vdash A'$ if $\forall e \in A' : A \vdash e$. Two sets of equations A, A' are called **deductively equivalent** (notation $A \dashv\vdash A'$) if $A \vdash A'$ and $A' \vdash A$. When there is no danger of confusion, we will often write e, e,e', A,e and A,A' for the sets $\{e\}$, $\{e\} \cup \{e'\}$, $A \cup \{e\}$ and $A \cup A'$, respectively.

We say two terms s, t are **convertible** or deductively equivalent with respect to a set A of axioms if $A \vdash s = t$. We introduce a special notation for equivalence classes induced by the convertibility relation:

$$[t]_A \stackrel{\text{def}}{=} \{s \mid A \vdash s = t\}$$

When the set A is fixed we may abbreviate $[t]_A$ to $[t]$.

When working with convertibility classes, or, more generally, with sets of terms, we sometimes need to refer to their elements. For this purpose we assume that, for every type $T \in \mathit{Ty}$, we have a choice function $\rho_T \in \mathcal{P}(\mathit{Ter}_T) \to \mathit{Ter}_T$ such that

$$\forall M \in \mathcal{P}(\mathit{Ter}_T) : M \neq \varnothing \Longrightarrow \rho_T M \in M$$

We want to conclude the discussion of our deductive framework by introducing two important derived rules of deduction, the rule of substitution and the rule of replacement. In Chapter 3 we also consider a non-extensional variant of our calculus, called S_N, which

$$
\begin{array}{ll}
t : T \implies \theta t : T & \text{preservation of types under substitution} \\
s : T \text{ and } A \vdash s = t \implies t : T & \text{preservation of types under deduction} \\
A \vdash e \text{ and } \theta \text{ stable for } A \implies \theta A \vdash \theta e & \text{stability of deduction under substitution}
\end{array}
$$

Figure 2.2: Selected Deductive Properties of S

is obtained by dropping the rule η. Replacement and substitution stay derivable in the absence of η. The two rules play a particularly important role in the context of equational reasoning by term rewriting or conversion. After introducing the rules, we will briefly discuss conversion proofs and their relation to formal derivations as presented in Figure 2.1.

Substitution Rule

The rule of substitution allows us to construct substitution instances of equational theorems. It can be formalized as follows:

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

It is not hard to see that **Sub** is derivable in S.

Proposition 2.1.1 *The rule* **Sub** *is derivable.*

Proof

$$
\textbf{Trans}\ \dfrac{\textbf{Sym}\ \dfrac{\beta\ \dfrac{}{(\lambda x.s)t = s[x{:=}t]}}{s[x{:=}t] = (\lambda x.s)t} \qquad \textbf{Trans}\ \dfrac{\textbf{Cong}\ \dfrac{\xi\ \dfrac{s = s'}{\lambda x.s = \lambda x.s'} \quad \textbf{Ref}\ \dfrac{}{t = t}}{(\lambda x.s)t = (\lambda x.s')t} \quad \beta\ \dfrac{}{(\lambda x.s')t = s'[x{:=}t]}}{(\lambda x.s)t = s'[x{:=}t]}}{s[x{:=}t] = s'[x{:=}t]}
$$

□

Replacement Rule

The rule of replacement formalizes the intuition that terms which are provably equal can be used interchangeably in the context of larger terms, preserving deductive equivalence. To formulate the rule, we first need to formalize the notion of a "context".

Contexts can be seen as partially specified terms. The atomic context is called a **hole** (notation •). Non-atomic contexts are built from names and the atomic context in the same way as terms. In the following we restrict our attention to contexts which contain exactly one hole. Contexts are referred to by the meta variable k. The size of a context k (notation $|k|$) is also defined in the same way as that of a term, however with $|\bullet| = 0$. Substitution also can be easily extended to contexts. Given a context k and a substitution θ, we apply θ to k in the same way as if $\bullet$ was a parameter not in $\operatorname{dom}\theta$. Since we also assume that $\bullet$ does not occur in θu for any other name u in k, substitution does not change the number of holes in k.

A context k can be transformed into a term by replacing $\bullet$ with some term s. The resulting term is called an **instance** of k and is referred to by the notation $k[s]$. Variables occurring in s may be captured by k. More formally:

$$\begin{array}{rcl} \bullet[s] & = & s \\ (k\,t)[s] & = & k[s]\,t \\ (t\,k)[s] & = & t\,k[s] \\ (\lambda x.k)[s] & = & \lambda x.k[s] \end{array}$$

A context k is called **well typed** if there exists a term t such that $k[t]$ is again well typed. In the following we consider only well-typed contexts and instances.

Now we are ready to give a precise formulation of the replacement rule:

$$\mathbf{Rep}\ \frac{s = s'}{k[s] = k[s']}$$

Proposition 2.1.2 *The rule* **Rep** *is derivable.*

Proof By induction on $|k|$.

Case $k = \bullet$. The claim follows trivially, because $\bullet[s] \stackrel{\text{def}}{=} s = s' \stackrel{\text{def}}{=} \bullet[s']$.

Case $k = k'\,t$. By definition of context instantiation it suffices to derive $k'[s]\,t = k'[s']\,t$.

$$\mathbf{Cong}\ \frac{\text{IH}\ \dfrac{s = s'}{k'[s] = k'[s']} \qquad \mathbf{Ref}\ \dfrac{}{t = t}}{k'[s]\,t = k'[s']\,t}$$

Case $k = t\,k'$ proceeds analogously to the preceding case.

Case $k = \lambda x.k'$. Here it suffices to derive $\lambda x.k'[s] = \lambda x.k'[s']$.

$$\xi\ \frac{\text{IH}\ \dfrac{s = s'}{k'[s] = k'[s']}}{\lambda x.k'[s] = \lambda x.k'[s']}$$

□

Conversion Proofs

The following discussion is closely related to the field of term rewriting. For a detailed introduction into term rewriting, see [6] or [48].

A **conversion proof** of an equation $s = t$ from a set A of axioms is a sequence $t_0 = \ldots = t_n$, for some $n \in \mathbb{N}$, such that:

1. $t_0 = s$ and $t_n = t$.
2. For every i such that $0 < i \leq n$, there exists a context k, a substitution θ where $\operatorname{dom}\theta \subseteq \mathit{Var}$, and terms s', t' such that $t_{i-1} = k[\theta s']$, $t_i = k[\theta t']$ and
   a) $(s' = t') \in A$ or $(t' = s') \in A$, or
   b) θ is the identity substitution and $s' = t'$ or $t' = s'$ an instance of β or η (no η-conversion in S_N).

The number n is called the **size** of the proof. We write $s \leftrightarrow^n_A t$ to say that $s = t$ has a conversion proof of size n, and $s \leftrightarrow^*_A t$ to say that $s = t$ is provable by conversion in finitely many steps.

The derived rules of substitution and replacement are particularly useful when establishing a connection between formal derivations and conversion proofs:

Proposition 2.1.3 $A \vdash s = t \iff s \leftrightarrow^*_A t$

Proof

- "$\Leftarrow$": First we show $s \leftrightarrow^1_A t \implies A \vdash s = t$ by **Sym**, **Sub** and **Rep**. Then we show the actual subclaim by induction on the size of the conversion proof, using **Ref** and **Trans**.
- "$\Rightarrow$": By induction on the size of the derivation for $A \vdash s = t$, exploiting the fact that $\leftrightarrow^*_A$ is an equivalence relation. □

By the above proposition, every statement of the form $A \vdash e$ can be proven by conversion. Since conversion proofs are often much more compact than the corresponding derivation trees, in the following conversion often will be used to show formal derivability of equations.

2.1.4 Semantics

Since general semantics of S is mainly relevant for Chapter 3, it will be introduced there. Here we present the standard semantics of S.

An **interpretation** is a function $\mathcal{I}$ such that:

1. $\operatorname{dom}\mathcal{I} = \mathit{Ty} \cup \mathit{Par} \cup \mathit{Var}$
2. $\mathcal{I}(T_1 T_2) = \mathcal{I}T_1 \to \mathcal{I}T_2$
3. $\mathcal{I}u \in \mathcal{I}(\boldsymbol{\tau} u)$

Elements of $\operatorname{ran}(\mathcal{I}|_{\mathit{Ty}})$ are called **domains**. Elements of domains a called **values** and will be denoted by the meta variables v and w.

Given an interpretation $\mathcal{I}$, a name u and a value $v \in \mathcal{I}(\boldsymbol{\tau} u)$, we write $\mathcal{I}_v^u$ to denote the modified interpretation satisfying the following equations:

$$\begin{array}{rcl} \mathcal{I}_v^u T &=& \mathcal{I} T \\ \mathcal{I}_v^u u' &=& \text{if } u' = u \text{ then } v \text{ else } \mathcal{I} u \end{array}$$

An **evaluation** $\hat{\mathcal{I}}$ extends an interpretation $\mathcal{I}$ to non-atomic terms such that:

1. $\hat{\mathcal{I}}(st) = \hat{\mathcal{I}}s(\hat{\mathcal{I}}t)$
2. $\hat{\mathcal{I}}(\lambda x.t) = \lambda v \in \mathcal{I}(\boldsymbol{\tau} x).\hat{\mathcal{I}}_v^x t$

Note that $\hat{\mathcal{I}}$ is uniquely determined by $\mathcal{I}$, and vice versa.

Given an interpretation $\mathcal{I}$ and a substitution θ, we write $\mathcal{I}_\theta$ to denote the interpretation satisfying:

$$\begin{array}{rcl} \mathcal{I}_\theta T &=& \mathcal{I} T \\ \mathcal{I}_\theta u &=& \hat{\mathcal{I}}(\theta u) \end{array}$$

A **structure** is a function $\mathcal{A}$ such that $\operatorname{dom} \mathcal{A}$ is a signature and there exists an interpretation $\mathcal{I}$ such that $\mathcal{A} \subseteq \mathcal{I}$. If $\operatorname{dom} \mathcal{A} = \Sigma$, we say that $\mathcal{A}$ is a **structure over** Σ. Given a structure $\mathcal{A}$, we write $\Sigma_{\mathcal{A}}$ to denote $\operatorname{dom} \mathcal{A}$, the **signature of** $\mathcal{A}$. In the following, structures will be also synonymously called **models**.

Given a structure $\mathcal{A}$, a parameter c is called a **constant of** $\mathcal{A}$ if $c \in \operatorname{dom} \mathcal{A}$. Once a structure $\mathcal{A}$ is fixed, constants of $\mathcal{A}$ are simply referred to as constants.

An interpretation $\mathcal{I}$ is said to **satisfy** an equation $s = t$ (notation $\mathcal{I} \vDash s = t$) if $\hat{\mathcal{I}}s = \hat{\mathcal{I}}t$. A structure $\mathcal{A}$ satisfies an equation e (notation $\mathcal{A} \vDash e$) if for all interpretations $\mathcal{I} \supseteq \mathcal{A}$: $\hat{\mathcal{I}} \vDash e$. In this case we also say that e is **valid** in $\mathcal{A}$ or that $\mathcal{A}$ is a **model of** e. Both notions of satisfaction can be pointwise extended to sets A of equations. So, for instance, $\mathcal{A} \vDash A$ if $\forall e \in A : \mathcal{A} \vDash e$. A set A **semantically entails** an equation e (notation $A \vDash e$) if $\forall \mathcal{A} : \mathcal{A} \vDash A \implies \mathcal{A} \vDash e$. In this case the equation e is also called a **semantic consequence** of A. A model $\mathcal{A}$ of a set A is called **minimal** if, for all equations e, $\mathcal{A} \vDash e \implies A \vdash e$. $\mathcal{A}$ is called minimal because, if we consider, for every model $\mathcal{B}$ of A, the set of equations valid in $\mathcal{B}$, the corresponding set for $\mathcal{A}$ will be minimal with respect to inclusion.

It is not hard to check that the notion of validity as defined above is preserved by deduction. Thus, the presented deduction formalism is **sound**.

Proposition 2.1.4 (Soundness) $A \vdash e \Longrightarrow A \vDash e$

Proof By induction on the size of a formal derivation of e from A. See textbook by Mitchell [33] for a stronger result obtained for a similar formulation of the calculus. □

Further semantic properties of S which are relevant for our considerations are summarized in Figure 2.3.

The reverse implication to soundness, $A \vDash e \Longrightarrow A \vdash e$, is usually called **(deductive) completeness**. It will be our main subject of study throughout the thesis.

Given a set A of axioms, we call the theory of A complete if it contains all the semantic consequences of A. Obviously, this is the case if and only if higher-order equational deduction is complete for A. A itself is called complete if its theory is complete.

$$
\begin{array}{ll}
\mathcal{I}T \neq \varnothing & \text{no empty types} \\
\hat{\mathcal{I}}t \in \mathcal{I}(\boldsymbol{\tau}t) & \text{type soundness} \\
\hat{\mathcal{I}}(\theta t) = \hat{\mathcal{I}}_\theta t & \text{substitution lemma} \\
\mathcal{I}|_{\mathcal{N}t} = \mathcal{I}'|_{\mathcal{N}t} \implies \hat{\mathcal{I}}t = \hat{\mathcal{I}}'t & \text{denotational coherence} \\
A \vdash e \implies A \vDash e & \text{deductive soundness}
\end{array}
$$

Figure 2.3: Selected Semantic Properties of S

2.2 Number Theory

As an example of the semantic expressiveness of S, in Figure 2.4 we give an equational axiomatization of number theory, which we call NT.

It is not hard to see that in every model of NT that does not interpret B by a singleton set, every constant has the expected semantics. The axioms I0 and I1 define the semantics of 0, 1 and $(\rightarrow)$. BCA forces every non-singleton interpretation of B to be isomorphic to a two-element set $\{\mathsf{T}, \mathsf{F}\}$. ∀1 and ∀I define universal quantification. N***oo***, N***os***, N***so***, N***ss*** and N*Ind* encode the usual properties of identity on the natural numbers and the naturals themselves using an adaptation of Peano's postulates [35]. Finally, we axiomatically define the operations of addition and multiplication on the naturals.

The essential semantic consequences of the propositional axioms I0, I1 and BCA are discussed more extensively in Chapter 5. For further details see [45, 5, 30].

Note that models of NT are capable of expressing every arithmetic proposition in the sense of [20]. Hence, by Gödel's first incompleteness theorem we conclude:

Proposition 2.2.1 (Incompleteness of S) *There exist sets A of axioms such that $A \vDash e \not\Rightarrow A \vdash e$.*

2.3 Vector Notation

In the following we will often employ vector notation to abbreviate sequences of terms. This notation will be interpreted differently depending on the context. To avoid confusion, let us give a short summary of the possible cases.

1. When used on the meta-level, vectors abbreviate enumerations. For instance, $\vec{x}$ stands for $x_1, \ldots, x_n$.
2. When used in a term, vectors usually stand for sequences of applications. For instance, $s\vec{t}t$ stands for $st_1 \ldots t_n t$.
3. Vectors can be used in the head of a lambda-binder, with the intuitive meaning: $\lambda\vec{x}.t$ stands for $\lambda x_1 \ldots \lambda x_n.t$.
4. Applications of meta-level functions to vectors are expanded pointwise. Assume, for instance, that ρ represents some meta-level function which yields a formula when

Name	**NT**	
Base Types	B, N	truth values and natural numbers
Constants	$0, 1 : \mathsf{B}$	truth and falsehood
	$(\to) : \mathsf{BBB}$	implication (right associative)
	$\forall : (\mathsf{NB})\mathsf{B}$	universal quantifier
	$(\doteq) : \mathsf{NNB}$	equality
	$\boldsymbol{o} : \mathsf{N}$	zero
	$\boldsymbol{s} : \mathsf{NN}$	successor
	$(+), (\cdot) : \mathsf{NNN}$	addition and multiplication
Variables	$x, y : \mathsf{B}$	
	$m, n : \mathsf{N}$	
	$f : \mathsf{NB}$	
Notation	$\forall n.t \stackrel{\text{def}}{=} \forall(\lambda n.t)$	quantification

Axioms

$0 \to x$	$=$	1	I0
$1 \to x$	$=$	x	I1
$f0 \to f1 \to fx$	$=$	1	BCA
$\forall n.1$	$=$	1	$\forall$1
$\forall f \to fn$	$=$	1	$\forall$I
$\boldsymbol{o} \doteq \boldsymbol{o}$	$=$	1	N$\boldsymbol{oo}$
$\boldsymbol{o} \doteq \boldsymbol{s}n$	$=$	0	N$\boldsymbol{os}$
$\boldsymbol{s}n \doteq \boldsymbol{o}$	$=$	0	N$\boldsymbol{so}$
$\boldsymbol{s}m \doteq \boldsymbol{s}n$	$=$	$m \doteq n$	N$\boldsymbol{ss}$
$f\boldsymbol{o} \to (\forall n.fn \to f(\boldsymbol{s}n)) \to fn$	$=$	1	N*Ind*
$n + \boldsymbol{o}$	$=$	n	A$\boldsymbol{o}$
$n + \boldsymbol{s}m$	$=$	$\boldsymbol{s}n + m$	A$\boldsymbol{s}$
$n \cdot \boldsymbol{o}$	$=$	$\boldsymbol{o}$	M$\boldsymbol{o}$
$n \cdot \boldsymbol{s}m$	$=$	$n + (n \cdot m)$	M$\boldsymbol{s}$

Figure 2.4: Axiomatization of Number Theory

applied to a value. Consider an application like $\rho\vec{v}$. There are basically two ways of interpreting this construction, namely either as the meta-level enumeration $\rho v_1, \ldots, \rho v_n$ or as the object-level term $(\rho v_1) \ldots (\rho v_n)$. In either case it will be clear from the context which interpretation is the correct one.

5. Meta-level relations between vectors are again expanded pointwise: $\vec{a} \in \vec{A}$ stands for $a_1 \in A_1, \ldots, a_n \in A_n$. In particular, this usage implies that $\vec{a}$ and $\vec{A}$ are required to have the same length.
6. In formal context, basic terms involving vectors abbreviate conjunctions: $\vec{s} \doteq \vec{t}$ stands for $\bigwedge_{i=1}^{n} s_i \doteq t_i$.
7. When used in a type specification, vectors abbreviate functional types: $\vec{T}C$ stands for the type $T_1 \ldots T_n C$. This case can easily be distinguished from Case 1 by checking whether the context requires a single type specification or a list of types.

If the context contains no indication of what the length of the abbreviated sequence, i.e. the value of n, should be, the statement is assumed to hold for all $n \in \mathbb{N}$. In most cases though, the length will be fixed by a preceding cardinality or typing assumption.

3 Completeness in General Models

General semantics was first introduced by Henkin [23, 24] as a weakening of the standard semantic construction. With respect to this weaker notion of semantics Church's traditional formulation of higher-order logic [13] could be shown complete. In contrast to standard semantics, general semantics does not require functional types to be mapped to full function spaces. Instead, it imposes a different, weaker requirement on functional domains: They must contain enough values to provide denotations for all possible terms of the corresponding types (see [4] for details).

Completeness of the pure simply typed λ-calculus with respect to general semantics was first shown by Friedman [16] as an intermediate step in proving its completeness with respect to standard semantics. Friedman considered a functionally extensional version of the calculus without any parameters.

Mitchell [33] notes that Friedman's proof can be easily extended to systems with arbitrary equational axioms with parameters. However, he, just like Friedman, restricts his attention to the extensional λ-calculus containing the rule η.

The following soundness and completeness proof for equational λ-theories with respect to general semantics largely follows Friedman's original approach, extending it in the way sketched by Mitchell. However, our proof does not depend on the extensionality of the underlying models. As a consequence, the result holds for formulations of the λ-calculus with or without η.

3.1 General Semantics

In standard semantics, the domain of a functional type T_1T_2 is always required to be the full set of functions from the domain of T_1 to the domain of T_2. This requirement is given up by general semantics. Functional domains are no longer needed to consist of functions, nor to be isomorphic to sets of functions. Therefore, to be applicable within the less restrictive framework, the notions of an interpretation, an evaluation or a model need to be generalized.

A pair $(\mathfrak{D}, @)$ is called an **applicative structure** if

1. $\mathfrak{D}$ is a function mapping types to non-empty sets,
2. $@$ is a family of functions $\langle @_{T_1T_2} \,|\, T_1, T_2 \in \mathit{Ty} \rangle$ such that, for all types T_1, T_2: $@_{T_1T_2} \in \mathfrak{D}(T_1T_2) \to \mathfrak{D}T_1 \to \mathfrak{D}T_2$.

When applied on a type, the function $\mathfrak{D}$ returns the corresponding **domain**. @ defines a notion of application on elements of the domains given by $\mathfrak{D}$.

Given $(\mathfrak{D}, @)$, a **general interpretation** $\mathcal{I}$ over $\mathfrak{D}$ is a function $\mathit{Nam} \to \bigcup_{T \in \mathit{Ty}} \mathfrak{D}T$ such that for all types $T \in \mathit{Ty}$: $u : T$ implies $\mathcal{I}u \in \mathfrak{D}T$.

In other words, a general interpretation is a typed function from names to values. Since the notion of a value is fixed by the first component $\mathfrak{D}$ of an applicative structure, interpretations have to be defined relatively to applicative structures. They do not depend on the second component, so two applicative structures of the form $(\mathfrak{D}, @)$ and $(\mathfrak{D}, @')$ have associated with them the same set of general interpretations.

The modification of a general interpretation $\mathcal{I}$ on a name u with a value $v \in \mathcal{I}(\tau u)$, written $\mathcal{I}^u_v$, is defined in the same way as for standard interpretations.

Now we want to uniquely determine how to evaluate non-atomic terms. Since in practice we usually assume the interpretation of certain parameters to be fixed (constant), we also need a means to capture this sort of assumptions. Both issues are addressed by the notion of a general structure:

Given $(\mathfrak{D}, @)$, a **general structure** $\mathcal{H} = (\mathfrak{I}, \hat{\_})$ is a pair of functions such that:

1. $\operatorname{dom} \mathfrak{I} \subseteq Par$ and there exists a general interpretation $\mathcal{I}$ over $\mathfrak{D}$ such that $\mathfrak{I} \subseteq \mathcal{I}$,
2. for every pair of general interpretations $\mathcal{I}, \mathcal{I}'$ over $\mathfrak{D}$ extending $\mathfrak{I}$, $\hat{\mathcal{I}} \supseteq \mathcal{I}$ and $\hat{\mathcal{I}}' \supseteq \mathcal{I}'$ are partial functions defined on terms such that:
   a) $\hat{\mathcal{I}}(st) = \hat{\mathcal{I}}s @ \hat{\mathcal{I}}t$,
   b) $\hat{\mathcal{I}}(\lambda x.t) @ v = \hat{\mathcal{I}}^x_v t$,
   c) if, for all $v \in \mathfrak{D}(\boldsymbol{\tau} x)$, $\hat{\mathcal{I}}^x_v t = \hat{\mathcal{I}}'^x_v t$, then $\hat{\mathcal{I}}(\lambda x.t) = \hat{\mathcal{I}}'(\lambda x.t)$.

Given $(\mathfrak{D}, @)$, $\mathcal{H} = (\mathfrak{I}, \hat{\_})$ and a general interpretation $\mathcal{I}$ over $\mathfrak{D}$, we call $\hat{\mathcal{I}}$ the **general evaluation** induced by $\mathcal{H}$ and $\mathcal{I}$.

Depending on the underlying applicative structure, it is not always possible to obtain general evaluations that are total on Ter. This motivates the definition of general models as a subset of general structures which ensures totality of the associated evaluations:

Given a signature Σ, a general structure $\mathcal{H} = (\mathfrak{I}, \hat{\_})$ is called a **general model** over Σ if

1. $\operatorname{dom} \mathfrak{I} = \Sigma \cap Par$ and
2. for every term $t : T$ and every general interpretation $\mathcal{I}$ over $\mathfrak{D}$ extending $\mathfrak{I}$: $\hat{\mathcal{I}}t \in \mathfrak{D}T$.

Given an applicative structure $(\mathfrak{D}, @)$, a general model $\mathcal{H} = (\mathfrak{I}, \hat{\_})$ is said to **satisfy** an equation e of the form $s = t$ (notation $\mathcal{H} \vDash e$) if for all general interpretations $\mathcal{I}$ over $\mathfrak{D}$ extending $\mathfrak{I}$: $\hat{\mathcal{I}}s = \hat{\mathcal{I}}t$. The concepts of semantic entailment and minimal models are defined correspondingly.

General models share many properties with standard models. Among these common properties are the validity of the substitution lemma, the absence of empty types, type and deductive soundness. Denotational coherence for interpretations which agree on the variables occurring in a term can be formulated as follows:

Proposition 3.1.1 *Let $\mathcal{H} = (\mathfrak{I}, \hat{\_})$ be a model over a signature Σ. Then, for every term t licensed by Σ and all $\mathcal{I}, \mathcal{I}'$ extending $\mathfrak{I}$: $\mathcal{I}|_{\mathcal{V}t} = \mathcal{I}'|_{\mathcal{V}t} \Longrightarrow \hat{\mathcal{I}}t = \hat{\mathcal{I}}'t$.*

Proof By induction on $|t|$. □

3.2 Model Construction

Given a set A of axioms, the **applicative term structure** $(\mathfrak{T}, @)$ is an applicative structure such that:

1. $\mathfrak{T}T = \{[t]_A \mid t : T\}$,
2. $v@w = [(\rho v)(\rho w)]_A$.

Is is easy to check that $(\mathfrak{T}, @)$ is indeed an applicative structure.

For every general interpretation $\mathcal{I}$ over $\mathfrak{T}$ we define a **corresponding substitution** $\theta_\mathcal{I}$:

$$\theta_\mathcal{I} \stackrel{\text{def}}{=} \lambda u \in Nam.\rho(\mathcal{I}u)$$

Note that this definition implies that, for some $\mathcal{I}$, $\operatorname{dom}\theta_\mathcal{I} \cap Var$ is infinite. This means, we have to be particularly careful in cases where $\theta_\mathcal{I}$ is applied to abstractions $\lambda x.t$ since, in general, we can no longer assume $x \notin \operatorname{dom}\theta_\mathcal{I}$ for any variable x.

The substitution $\theta_\mathcal{I}$ is constructed to satisfy the following property:

Proposition 3.2.1 $[\theta_\mathcal{I} u]_A = \mathcal{I}u$

The next property of $\theta_\mathcal{I}$ (Proposition 3.2.3) is crucial for the soundness of the following model construction. Its validity can be established by proving a more general claim:

Lemma 3.2.2 *If u does not occur in $\vec{u}$, then:* $\theta_{\mathcal{I}^u_v}[\vec{u} := \vec{u}]t = \theta_\mathcal{I}[u\vec{u} := (\rho v)\vec{u}]t.$

Proof Let u and $\vec{u}$ be names such that u does not occur in $\vec{u}$. We proceed by induction on $|t|$. Where necessary, we apply Proposition 3.2.1.

Case $t = u$. Then: $\theta_{\mathcal{I}^u_v}[\vec{u} := \vec{u}]u = \theta_{\mathcal{I}^u_v}u = \rho(\mathcal{I}^u_v u) = \rho v = \theta_\mathcal{I}[u\vec{u} := (\rho v)\vec{u}]u$

Case $t = u' \neq u$, $u' \notin \vec{u}$.

$$\theta_{\mathcal{I}^u_v}[\vec{u} := \vec{u}]u' = \theta_{\mathcal{I}^u_v}u' = \rho(\mathcal{I}^u_v u') = \rho(\mathcal{I}u') = \theta_\mathcal{I}u' = \theta_\mathcal{I}[u\vec{u} := (\rho v)\vec{u}]u'$$

Case $t = u_i$ in $\vec{u}$. Then: $\theta_{\mathcal{I}^u_v}[\vec{u} := \vec{u}]u_i = u_i = \theta_\mathcal{I}[u\vec{u} := (\rho v)\vec{u}]u_i$

Case $t = t_1t_2$.

$$\begin{array}{rcll} \theta_{\mathcal{I}^u_v}[\vec{u} := \vec{u}](t_1t_2) & = & (\theta_{\mathcal{I}^u_v}[\vec{u} := \vec{u}]t_1)(\theta_{\mathcal{I}^u_v}[\vec{u} := \vec{u}]t_2) & \\ & = & (\theta_\mathcal{I}[u\vec{u} := (\rho v)\vec{u}]t_1)(\theta_\mathcal{I}[u\vec{u} := (\rho v)\vec{u}]t_2) & \text{IH} \\ & = & \theta_\mathcal{I}[u\vec{u} := (\rho v)\vec{u}](t_1t_2) & \end{array}$$

Case $t = \lambda x.s$ where $x \notin \{u\} \cup \mathcal{V}(\rho v) \cup \bigcup_{u' \in \mathcal{N}t} \mathcal{V}(\theta_{\mathcal{I}^u_v}u')$.

$$\begin{array}{rcll} \theta_{\mathcal{I}^u_v}[\vec{u} := \vec{u}](\lambda x.s) & = & \lambda x.\theta_{\mathcal{I}^u_v}[x\vec{u} := x\vec{u}]s & \\ & = & \lambda x.\theta_\mathcal{I}[ux\vec{u} := (\rho v)x\vec{u}]s & \text{IH} \\ & = & \theta_\mathcal{I}[u\vec{u} := (\rho v)\vec{u}](\lambda x.s) & \end{array}$$

□

Proposition 3.2.3 $\theta_{\mathcal{I}^u_v} t = \theta_{\mathcal{I}}[u := \rho v]t$

Proof Special case of Lemma 3.2.2. □

Corollary 3.2.4 $\theta_{\mathcal{I}^u_v} = \theta_{\mathcal{I}}[u := \rho v]$

Given a signature Σ and a set A of axioms, the **term model** $\mathcal{T}$ is the unique pair $(\mathfrak{I}, \hat{\ })$ such that:

1. $\operatorname{dom}\mathfrak{I} = \Sigma \cap \mathit{Par}$,
2. if $c \in \Sigma$, then $\mathfrak{I}c = [c]_A$,
3. for every $\mathcal{I}$ over $\mathfrak{T}$, $\hat{\mathcal{I}}t = [\theta_{\mathcal{I}} t]_A$,

For the rest of the chapter, we fix a signature Σ and a set A of equations licensed by Σ. Our first objective is to show that $\mathcal{T}$ is a general model over Σ.

First we show that substitution is coherent modulo conversion. Compare Proposition 3.2.6 with Proposition 3.1.1 to see that this property is a necessary criterion for $\mathcal{T}$ to be a general model.

Lemma 3.2.5 *If, for all names u that occur in t,* $[\theta[\vec{u} := \vec{u}]u] = [\phi[\vec{u} := \vec{u}]u]$*, then* $[\theta[\vec{u} := \vec{u}]t] = [\phi[\vec{u} := \vec{u}]t]$.

Proof Let t be a term and, for all $u \in \mathcal{V}t$, $[\theta[\vec{u} := \vec{u}]u] = [\phi[\vec{u} := \vec{u}]u]$. We proceed by induction on $|t|$.

Case $t = u'$. The claim immediately follows by assumption.

Case $t = t_1 t_2$. Let $\theta' = \theta[\vec{u} := \vec{u}]$, $\phi' = \phi[\vec{u} := \vec{u}]$. Then:

$$\begin{array}{rcll} [\theta'(t_1 t_2)] & = & [(\theta' t_1)(\theta' t_2)] & \\ & = & [(\rho[\theta' t_1])(\rho[\theta' t_2])] & \\ & = & [(\rho[\phi' t_1])(\rho[\phi' t_2])] & \text{IH} \\ & = & [(\phi' t_1)(\phi' t_2)] & \\ & = & [\phi'(t_1 t_2)] & \end{array}$$

Case $t = \lambda x.s$ such that $x \notin \bigcup_{u' \in \mathcal{N}(\lambda x.s)} \mathcal{V}(\theta[\vec{u} := \vec{u}]u')$. Then:

$$\begin{array}{rcll} [\theta[\vec{u} := \vec{u}](\lambda x.s)] & = & [\lambda x.\theta[x\vec{u} := x\vec{u}]s] & \\ & = & [\lambda x.\rho[\theta[x\vec{u} := x\vec{u}]s]] & \\ & = & [\lambda x.\rho[\phi[x\vec{u} := x\vec{u}]s]] & \text{IH} \\ & = & [\lambda x.\phi[x\vec{u} := x\vec{u}]s] & \\ & = & [\phi[\vec{u} := \vec{u}](\lambda x.s)] & \end{array}$$

□

Proposition 3.2.6 *If, for all* $u \in \mathcal{N}t$, $[\theta u] = [\phi u]$*, then* $[\theta t] = [\phi t]$.

Proof Special case of Lemma 3.2.5. □

Proposition 3.2.7 *$\mathcal{T}$ is a general structure.*

Proof First of all, observe that $\operatorname{dom}\mathfrak{I} \subseteq Par$ and, for every constant $c : T$ we have $\mathcal{T}c = [c] \in \mathfrak{T}T$, i.e. $\mathfrak{I}$ can be extended to a general interpretation. It remains to check that, for all interpretations $\mathcal{I}$ extending $\mathfrak{I}$, $\hat{\mathcal{I}}$ is the general evaluation induced by $\mathcal{T}$ and $\mathcal{I}$. So, let $\mathcal{I}$, $\mathcal{I}'$ be two arbitrary but fixed assignments over $\mathfrak{T}$ (we don't actually need them to extend $\mathfrak{I}$ in order to prove the required properties). We have to show four subclaims:

1. $\hat{\mathcal{I}}u = \mathcal{I}u$. By Proposition 3.2.1: $\hat{\mathcal{I}}u = [\theta_{\mathcal{I}}u] = \mathcal{I}u$.
2. $\hat{\mathcal{I}}(st) = \hat{\mathcal{I}}s@\hat{\mathcal{I}}t$.

$$\begin{aligned}
\hat{\mathcal{I}}(st) &= [\theta_{\mathcal{I}}(st)] \\
&= [(\theta_{\mathcal{I}}s)(\theta_{\mathcal{I}}t)] \\
&= [(\rho[\theta_{\mathcal{I}}s])(\rho[\theta_{\mathcal{I}}t])] \\
&= [\theta_{\mathcal{I}}s]@[\theta_{\mathcal{I}}t] \\
&= \hat{\mathcal{I}}s@\hat{\mathcal{I}}t
\end{aligned}$$

3. $\hat{\mathcal{I}}(\lambda x.t)@v = \hat{\mathcal{I}}^x_v t$. Remember that substitution is generally assumed to be capture free. Moreover, w.l.o.g., $x \notin \bigcup_{y \in \mathcal{V}(\lambda x.t)} \mathcal{V}(\theta_{\mathcal{I}}y)$. Then:

$$\begin{aligned}
\hat{\mathcal{I}}(\lambda x.t)@v &= [(\rho(\hat{\mathcal{I}}(\lambda x.t)))(\rho v)] \\
&= [(\rho[\theta_{\mathcal{I}}(\lambda x.t)])(\rho v)] \\
&= [(\theta_{\mathcal{I}}(\lambda x.t))(\rho v)] \\
&= [(\lambda x.\theta_{\mathcal{I}}[x := x]t)(\rho v)] \\
&= [\theta_{\mathcal{I}}[x := \rho v]t] && \beta \\
&= [\theta_{\mathcal{I}^x_v}t] && \text{Prop. 3.2.3} \\
&= \hat{\mathcal{I}}^x_v t
\end{aligned}$$

4. If, for all values $v \in \mathfrak{T}(\boldsymbol{\tau}x)$, $\hat{\mathcal{I}}^x_v t = \hat{\mathcal{I}}'^x_v t$, then $\hat{\mathcal{I}}(\lambda x.t) = \hat{\mathcal{I}}'(\lambda x.t)$. Let, w.l.o.g., $x \notin \bigcup_{y \in \mathcal{V}(\lambda x.t)} \mathcal{V}(\theta_{\mathcal{I}}y) \cup \mathcal{V}(\theta_{\mathcal{I}'}y)$, and assume, for all $v \in \mathfrak{T}(\boldsymbol{\tau}x)$: $\hat{\mathcal{I}}^x_v t = \hat{\mathcal{I}}'^x_v t$, i.e. $[\theta_{\mathcal{I}^x_v}t] = [\theta_{\mathcal{I}'^x_v}t]$.

$$\begin{aligned}
\hat{\mathcal{I}}(\lambda x.t) &= [\theta_{\mathcal{I}}(\lambda x.t)] \\
&= [\lambda x.\theta_{\mathcal{I}}[x := x]t] \\
&= [\lambda x.\rho[\theta_{\mathcal{I}}[x := x]t]] \\
&= [\lambda x.\rho[\theta_{\mathcal{I}}[x := \rho[x]]t]] && \text{Prop. 3.2.6} \\
&= [\lambda x.\rho[\theta_{\mathcal{I}^x_{[x]}}t]] && \text{Prop. 3.2.3} \\
&= [\lambda x.\rho[\theta_{\mathcal{I}'^x_{[x]}}t]] && \text{assumption} \\
&= [\lambda x.\rho[\theta'_{\mathcal{I}}[x := \rho[x]]t]] && \text{Prop. 3.2.3} \\
&= [\lambda x.\rho[\theta'_{\mathcal{I}}[x := x]t]] && \text{Prop. 3.2.6} \\
&= [\lambda x.\theta'_{\mathcal{I}}[x := x]t] \\
&= [\theta'_{\mathcal{I}}(\lambda x.t)] \\
&= \hat{\mathcal{I}}'(\lambda x.t)
\end{aligned}$$

□

Proposition 3.2.8 *$\mathcal{T}$ is a general model.*

Proof By definition, $\mathcal{T}$ satisfies $\operatorname{dom} \mathfrak{I} = \Sigma \cap Par$. By Proposition 3.2.7 it suffices to show that for every interpretation $\mathcal{I}$ over $\mathfrak{T}$ and every $t : T$: $\hat{\mathcal{I}}t \in \mathfrak{T}T$. Since $\hat{\mathcal{I}}t = [\theta_{\mathcal{I}}t]$, all that remains to show is $\theta_{\mathcal{T}}t : T$. This is the case because $\theta_{\mathcal{I}}$ is assumed to be well-typed and hence type preserving. □

Knowing that $\mathcal{T}$ is indeed a general model, we now want to check that $\mathcal{T}$ satisfies A.

Theorem 1 (Soundness) $A \vdash e \Longrightarrow \mathcal{T} \vDash e$

Proof Let $A \vdash e$ where $e = (s = t)$, i.e. $[s] = [t]$. Then, for every assignment $\mathcal{I} \supseteq \mathfrak{I}$:

$$\hat{\mathcal{I}}s = [\theta_{\mathcal{I}}s] = [\theta_{\mathcal{I}}t] = \hat{\mathcal{I}}t$$

The fact that $[s] = [t]$ implies $[\theta_{\mathcal{I}}s] = [\theta_{\mathcal{I}}t]$ requires some explanation. Let ϕ be a substitution such that $\operatorname{dom} \phi = \mathcal{P}e - \Sigma$ and, for every $a, b \in \operatorname{dom} \phi$:

1. $\phi a \in Var - \mathcal{V}e$,
2. $a \neq b \implies \phi a \neq \phi b$. □

Clearly, ϕ is stable for e, and, since A is licensed by Σ, $\phi A = A$. Hence, by stability of deduction under substitution, $A \vdash \phi e$. Note that ϕe is licensed by Σ.

Let $\theta = \lambda u \in Nam.$if $u \in \phi(\mathcal{P}e)$ then $\theta_{\mathcal{I}}(\phi|_{\mathcal{P}e}^{-1}u)$ else $\theta_{\mathcal{I}}u$. Note that, since $\mathcal{I} \supseteq \mathfrak{I}$, for all parameters $c \in \Sigma$ it holds: $[\theta c] = [\theta_{\mathcal{I}}] = [c]$. Therefore, by repeated application of the substitution rule (starting from ϕe) we can derive $\theta'(\phi e)$, for some substitution θ' that is equivalent to θ in the sense of Proposition 3.2.6. Hence $A \vdash \theta(\phi e)$, i.e. $A \vdash (\theta \circ \phi)e$. Since, again by Proposition 3.2.6, $[\theta_{\mathcal{I}}s] = [(\theta \circ \phi)s]$ and $[\theta_{\mathcal{I}}t] = [(\theta \circ \phi)t]$, the claim follows.

3.3 Minimal Models and Completeness

If $\mathcal{H}$ is a model of A, i.e., for all e, $A \vDash e \Longrightarrow \mathcal{H} \vDash e$, minimality of $\mathcal{H}$ immediately implies deductive completeness of A. Since we already know $\mathcal{T}$ to be a model of A, it suffices to show $\mathcal{T}$ minimal.

To do so, we consider a particular evaluation, the one induced by $\mathcal{T}$ and $\lambda u \in Nam.[u]$. Note that $(\lambda u \in Nam.[u]) \supseteq \mathfrak{I}$.

First we extend Proposition 3.2.3 to substitutions which a modified at more than one point:

Lemma 3.3.1 *If $\vec{u}$ contains no multiple occurrences of names, then* $\theta_{\mathcal{I}_{\vec{v}}^{\vec{u}}}t = \theta_{\mathcal{I}}[\vec{u} := \rho\vec{v}]t$.

Proof Let $\vec{u}$ be as required. We proceed by induction on $|\vec{u}|$.

Case $|\vec{u}| = 0$. The claim follows trivially.

Case $\vec{u} = u\vec{u_1}$. Then:

$$\begin{array}{rcll} \theta_{\mathcal{I}^{u\vec{u_1}}_{v\vec{v_1}}} t & = & \theta_{\mathcal{I}^u_v}[\vec{u_1} := \rho\vec{v_1}]t & \text{IH} \\ & = & (\theta_{\mathcal{I}}[u := \rho v])[\vec{u_1} := \rho\vec{v_1}]t & \text{Cor. 3.2.4} \\ & = & \theta_{\mathcal{I}}[\vec{u_1}u := (\rho\vec{v_1})(\rho v)]t & \\ & = & \theta_{\mathcal{I}}[u\vec{u_1} := (\rho v)(\rho\vec{v_1})]t & \end{array}$$

□

Next we show that the substitution corresponding to the "identity" interpretation $\lambda u \in Nam.[u]$ is the identity substitution:

Proposition 3.3.2 $[\theta_{\lambda u \in Nam.[u]} t] = [t]$

Proof Let $\mathcal{I}$ be an arbitrary but fixed interpretation over $\mathfrak{T}$ and let $\vec{u}$ contain, without multiple occurrences, precisely the names in $\mathcal{N}t$. Then:

$$\begin{array}{rcll} [\theta_{\lambda u \in Nam.[u]} t] & = & [\theta_{\mathcal{I}^{\vec{u}}_{[\vec{u}]}} t] & \text{Prop. 3.2.6} \\ & = & [\theta_{\mathcal{I}}[\vec{u} := \rho[\vec{u}]]t] & \text{Lemma 3.3.1} \\ & = & [[\vec{u} := \vec{u}]t] & \text{Prop. 3.2.6} \\ & = & [t] & \end{array}$$

□

With this, we are ready to prove minimality of $\mathcal{T}$:

Theorem 2 (Minimality) $\mathcal{T} \vDash e \Longrightarrow A \vdash e$

Proof Let $e = (s = t)$, $\mathcal{T} \vDash s = t$, $\mathcal{I} = \lambda u \in Nam.[u]$. By Proposition 3.3.2, it holds

$$[s] = [\theta_{\mathcal{I}} s] = \hat{\mathcal{I}}s = \hat{\mathcal{I}}t = [\theta_{\mathcal{I}} t] = [t]$$

i.e. $A \vdash s = t$. □

Theorem 3 (Completeness) $A \vDash e \Longrightarrow A \vdash e$

Proof Immediately follows by Theorem 1 and 2. □

3.4 Consequences

Due to the crucial role of Proposition 3.2.6, the completeness proof certainly relies on congruence of deduction, guaranteed by the rules **Cong** and ξ.

In contrast to ξ, η turns out to be inessential for the completeness proof. Nowhere in our argument do we have to apply η. Neither do we ever assume that $\mathcal{T}$ is functionally extensional, which is known to be the case if and only if η is admissible the presence of ξ (see [9] and [11] for details).

Therefore, the above completeness result applies to both S, our main deductive framework for the thesis, and S_N, the non-extensional version of S without η.

The definition of general models in Section 3.1 can be further weakened such that a general structure $\mathcal{H} = (\mathfrak{I}, \hat{\ })$ is still considered a model over Σ if

1. $\operatorname{dom} \mathfrak{I} \supseteq \Sigma \cap Par$ and
2. for every term $t : T$ type-licensed by Σ and every general interpretation $\mathcal{I}$ over $\mathfrak{D}$ extending $\mathfrak{I}|_{\Sigma \cap Par}$: $\hat{\mathcal{I}}t \in \mathfrak{D}T$.

This definition allows a single general structure to be simultaneously considered a model over several distinct signatures. Moreover, we allow a model to be partial on terms as long as it can evaluate all terms that are type-licensed by Σ. This definition is slightly more complicated than the one used for the completeness proof, but might be useful for more practical model constructions. As for the completeness proof, both definitions are equally acceptable. Since $\mathcal{T}$ satisfies the stronger one from Section 3.1, it also satisfies the latter one.

For further discussion we will need some definitions:

An applicative structure $(\mathfrak{D}, @)$ is called **trivial** at type T if $|\mathfrak{D}T| = 1$. A general structure is called trivial at T if the corresponding applicative structure is trivial at T. Analogously, a general structure is called countable at T if the domain of T is countable.

A set of equations is said to be **satisfiable** at type T with respect to general semantics if it has a general model which is non-trivial at T.

An important property of applicative term structures is the fact that all of their domains which are non-trivial have the same cardinality, namely $\aleph_0$:

Proposition 3.4.1 *An applicative term structure* $(\mathfrak{T}, @)$ *is non-trivial at type* T *if and only if* $|\mathfrak{T}T| = \aleph_0$.

Proof Let $(\mathfrak{T}, @)$ be an applicative term structure and T a type. Non-triviality of $\mathfrak{T}T$ requires that there exist at least two terms of type T which are not deductively equivalent. By the rule of substitution, this implies that no two distinct variables of type T may be deductively equivalent, i.e. contained in the same convertibility class. Therefore, there must exist at least as many convertibility classes in $\mathfrak{T}T$ as there are distinct variables of type T. So, the definitions of variables and typings give us $\aleph_0$ as a lower bound on the cardinality of $\mathfrak{T}T$. The upper bound is provided by the fact that we have only countably many terms (see [29]). The converse is immediate. □

Corollary 3.4.2 (Löwenheim-Skolem) *A set* A *of equations is satisfiable at type* T *if and only if it has a general model which is countable at* T.

Finally, let us prove compactness of S and S_N with respect to general semantics:

Proposition 3.4.3 (Compactness) *A set of equations* A *is satisfiable at a type* T *if and only if every finite subset of* A *is satisfiable at* T.

Proof Let A be a set of equations and T a type such that every finite subset of A is satisfiable at T. Let $x, y : T$ be two distinct variables. By soundness, the equation $x = y$ is not provable from any finite subset of A. Consequently, since all proofs from A are finite, $A \not\vdash x = y$, i.e. $[x]_A \neq [y]_A$. Hence $|\mathcal{T}T| > 1$. Then A is satisfiable by Theorem 1. The converse is immediate. □

3.5 Further Work

Now that we have shown completeness of deduction in the absence of η, it is natural to ask whether a similar result can be obtained in the absence of ξ, with or without η. Unfortunately, the λ-calculus without ξ is not compatible with equational reasoning in the sense it was presented so far. Without ξ, the rule of replacement is obviously no longer admissible, which in particular invalidates the fundamental Proposition 2.1.3. Therefore the discussion of equational deduction in higher-order calculi without ξ requires a more general approach than the one taken in this thesis. A detailed analysis of such calculi is an interesting subject for further investigations.

4 Completeness for Plain Axioms

As we pointed out in Section 2.2, equational systems generated by syntactically unrestricted axiom systems are, in general, incomplete with respect to standard semantics. However, generic completeness results can be established for several classes of syntactically restricted axiom systems.

Algebraic equations are probably the conceptually simplest non-trivial class of axioms for which S can be proven complete. S with algebraic axioms combines the expressiveness of first-order equational logic with that of the pure simply typed λ-calculus.

The completeness proof for algebraic axioms [30] extends Plotkin's model construction [39] for the simply typed lambda calculus to signatures with algebraic parameters, which are treated as in Birkhoff's classical construction for first-order equational logic [10].

In the following we prove deductive completeness of S for a richer class of axioms. These axioms will be called plain.

4.1 Plain Axioms and First-Order Logic

A term is called **plain** if it contains no functional variables as subterms. Note that a term is plain if and only if all its subterms are plain. As usual, an equation is called plain if both sides are plain, and a set is called plain if all of its elements are plain.

In contrast to algebraic equations, plain equations allow parameters of arbitrary type and abstractions over basic variables. To demonstrate the expressiveness of plain specifications, let us give an intuitive plain axiomatization of first-order predicate logic (Figure 4.1).

The first four axioms, $\mathrm{I0}, \mathrm{I1}, \mathrm{BCA}$ and Comm describe the propositional part of FOL, while $\forall 1$ and $\forall \mathrm{I}$ specify the deductive properties of quantifiers.

First, observe that the rule of **modus ponens** (MP) is easily derivable from FOL:

Proposition 4.1.1 (MP) *For all terms s and t:* $\mathrm{FOL}, s \to t = 1, s = 1 \vdash t = 1.$

Proof Let $\mathrm{FOL}, s \to t = 1, s = 1$ be derivable. Then $t = 1 \to t = s \to t = 1.$ □

In a similar way, the axiom $\forall 1$ can be shown to entail deductively the rule of universal generalization. And, together with MP, $\forall \mathrm{I}$ can be used to mimic universal instantiation. For more details, see [45, 30, 5].

While the axioms I0, I1 and BCA are sufficient to describe the propositional part of FOL semantically, Comm seems to be necessary to achieve the full deductive expressiveness of propositional logic (see Chapter 5 and [45] for details). We do not yet know whether it is truly independent from the other three axioms.

Name	**FOL**	
Base Types	B, I	truth values and individuals
Constants	$0, 1 : \mathsf{B}$	truth and falsehood
	$(\rightarrow) : \mathsf{BBB}$	implication (right associative)
	$\forall : (\mathsf{IB})\mathsf{B}$	universal quantifier
Meta Variables	$x, y : \mathsf{B}$	Boolean variables
	$z : \mathsf{I}$	individual variables
	$\mathsf{A} : \mathsf{B}$	(Boolean) plain terms

Notation

$x \vee y$	$\stackrel{\text{def}}{=}$	$(x \rightarrow y) \rightarrow y$	disjunction
$\forall z.t$	$\stackrel{\text{def}}{=}$	$\forall(\lambda z.t)$	quantification

Axioms

$0 \rightarrow x$	$=$	1	I0
$1 \rightarrow x$	$=$	x	I1
$(\lambda x.\mathsf{A})0 \rightarrow (\lambda x.\mathsf{A})1 \rightarrow \mathsf{A}$	$=$	1	BCA
$x \vee y$	$=$	$y \vee x$	Comm
$\forall z.1$	$=$	1	∀1
$(\forall z.\mathsf{A}) \rightarrow \mathsf{A}$	$=$	1	∀I

Figure 4.1: Axiomatization of FOL

By the above, it is not very hard to show that

1. our axiomatization is sound with respect the usual semantics of FOL,
2. deduction in traditional formulations of FOL can be simulated in S using the given axioms.

For reference on syntax and semantics of FOL, see for instance introductory textbooks by Andrews [5] or Fitting [15]. Further properties of plain equational specifications will be discussed in Section 4.6.

4.2 Extensionality of External Equality

Given a model $\mathcal{A}$, a family $\langle \sim_T \mid T \in Ty \rangle$ of equivalence relations such that $\sim_T \subseteq (\mathcal{A}T)^2$ is called **(functionally) extensional** if, for all types $T = T_1 T_2$ and all values $v, v' \in \mathcal{A}T$: $(\forall w \in \mathcal{A}T_1 : vw \sim_{T_2} v'w) \implies v \sim_T v'$.

Obviously, the identity relation induced by the definition of equational validity, which we also call **external equality**, is extensional. More interesting is the fact that extensionality of external equality can be exploited deductively (see also [9, 11, 33] for a semantic justification of this fact):

Proposition 4.2.1 *Let $s, t : T \to T'$, $x : T$. If $x \notin \mathcal{V}s \cup \mathcal{V}t$ then*

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

Proof

$$\begin{array}{rcll} sx = tx & \vdash & \lambda x.sx = \lambda x.tx & \xi \\ & \vdash & s = t & \eta \end{array}$$

□

Corollary 4.2.2 *Let s, t be terms. Let $\vec{x} \notin \mathcal{V}s \cup \mathcal{V}t$ be pairwise distinct. Then*

$$s\vec{x} = t\vec{x} \;\vdash\; s = t$$

The reverse direction, $s = t \;\vdash\; s\vec{x} = t\vec{x}$, obviously follows by reflexivity and congruence of our deduction formalism, which allows us to strengthen the above corollary to:

Corollary 4.2.3 *Let s, t be terms. Let $\vec{x} \notin \mathcal{V}s \cup \mathcal{V}t$ be pairwise distinct. Then*

$$s\vec{x} = t\vec{x} \dashv\vdash s = t$$

The latter corollary implies that every equation between higher-order terms can be transformed into a deductively equivalent equation between basic terms, and vice versa.

4.3 Model Construction

Given a set A of axioms, a **term structure** $\mathcal{D}$ is a structure satisfying

$$\forall C \in Sor : \mathcal{D}C = \{[t]_A \mid t : C\}$$

For the rest of the chapter we fix some signature Σ and consider only structures $\mathcal{D}$ over Σ. Moreover, we fix a set A of axioms which is licensed by Σ. Some results will require A to be plain. In each such case the additional requirement will be mentioned explicitly.

Note that once A has been fixed, term structures agree on all types:

$$\forall \mathcal{D}, \mathcal{D}' \; \forall T \in Ty : \mathcal{D}T = \mathcal{D}'T$$

Hence, whenever we need to refer to the domain of a type T in a term structure, we will usually simply write $\mathcal{D}T$ without first introducing any particular structure $\mathcal{D}$; $\mathcal{D}T$ does not depend on how we choose $\mathcal{D}$.

For every type T, we define functions

$$\begin{aligned} \delta_T &\in Ter_T \to \mathcal{D}T \\ \tau_T &\in \mathcal{D}T \to \mathcal{P}(Ter_T) \end{aligned}$$

by mutual recursion on T:

$$\begin{aligned} \delta_{\vec{T}C} t &\stackrel{\text{def}}{=} \lambda \vec{v} \in \mathcal{D}\vec{T}.[t(\rho(\tau\vec{v}))] \\ \tau_{\vec{T}C} v &\stackrel{\text{def}}{=} \{t \in Ter_{\vec{T}C} \mid \forall \vec{x} \in Var_{\vec{T}} : v(\delta\vec{x}) = [t\vec{x}]\} \end{aligned}$$

These functions are used to establish a connection between terms and domains. Let us explain the main intuition behind the definitions. We want δ to map terms to their denotations according to a specific "identity" evaluation extending $\mathcal{D}$. More precisely, a term $t : \vec{T}C$ should be mapped to a function which takes values $\vec{v}$ according to the type of t and returns the convertibility class of t applied to some terms $\vec{t}$ from the pre-image of $\vec{v}$ under δ. The construction of this pre-image should be accomplished by the function τ. In order to find a pre-image of some value v, τ applies v to the denotations of all possible variables $\vec{x}$ of appropriate types until it evaluates to a convertibility class of the form $[t\vec{x}]$, and collects those terms which are deductively equivalent to t for every choice of $\vec{x}$.

Due to the uncountability of functional domains there will exist values v which are not in the range of δ. For such values, τ may behave in two different ways. Either it detects that v is not in the range of δ and returns $\varnothing$ or it returns a non-empty class $[t]$ of terms such that $\delta t = w \neq v$. Obviously, in this second case v and w must agree on the denotations of all variables. This is possible because even basic domains contain more than just denotations of variables. Therefore, a functional value certainly cannot be uniquely determined by its behaviour on just these arguments. Fortunately, for our means it suffices to consider only those values which are in the range of δ. These values will turn out to be indeed uniquely determined by their behaviour on denotations of variables, justifying our construction of τ.

Before looking at functional types, let us interpolate our intuition to base types and observe that at least there δ and τ behave exactly as expected:

Proposition 4.3.1

1. $\delta_C t = [t]$
2. $\tau_C v = v$
3. $[\rho_C v] = v$

Proof Obvious from the definitions of $\mathcal{D}C$, δ, τ and ρ. □

Next we want to show that δ is invariant under deduction, i.e. that convertible terms are mapped by δ to the same value. This implies that the following family of functions $\delta^* = \langle {\delta_T}^* \mid T \in \mathit{Ty} \rangle$ is well defined:

$$\begin{array}{l} {\delta_T}^* \in \mathit{Ter}_T/_A \to \mathcal{D}T \\ {\delta_T}^*[t] = \delta_T t \end{array}$$

Although δ^* is not needed for the formal parts of our proof, it is a construction which closely corresponds to our intuitive goal, which is connecting convertibility classes with domains.

Proposition 4.3.2 $[s] = [t] \Longrightarrow \delta s = \delta t$

Proof Let $[s] = [t]$. Then

$$\begin{array}{rcll} \delta s & = & \lambda \vec{v} \in \mathcal{D}\vec{T}.[s(\rho(\tau(\vec{v})))] & \\ & = & \lambda \vec{v} \in \mathcal{D}\vec{T}.[t(\rho(\tau(\vec{v})))] & [s] = [t] \\ & = & \delta t & \end{array}$$

□

Now let us have a closer look at τ. The following proposition states that, given a value v, τ always returns either the empty set or some convertibility class. So, τ_T can also be seen as a function $\mathcal{D}T \to \mathit{Ter}_T/_A \cup \{\varnothing\}$.

Proposition 4.3.3 $t \in \tau v \iff \tau v = [t]$

Proof The direction "⇐" is obvious. In order to show "⇒", assume $t \in \tau_{\vec{T}C} v$. We prove $\forall s \in \mathit{Ter}_{\vec{T}C} :\ s \in \tau v \iff A \vdash s = t$.

- "⇒": Choose $\vec{x} \in \mathit{Var}_{\vec{T}}$ pairwise distinct and disjoint from $\mathcal{V}s \cup \mathcal{V}t$. Since by assumption $s \in \tau v$, we have

$$\begin{array}{rrll} & [s\vec{x}] & = v(\delta\vec{x}) & s \in \tau v \\ & & = [t\vec{x}] & t \in \tau v \\ \iff & A \vdash s\vec{x} = t\vec{x} & & \\ \Longrightarrow & A \vdash s = t & & \text{Corollary 4.2.2} \end{array}$$

- "⇐": It suffices to show that for all $\vec{x} : \vec{T}$: $A \vdash s = t \Longrightarrow v(\delta\vec{x}) = [s\vec{x}]$.

$$\begin{array}{rrll} & A \vdash s = t & & \\ \Longrightarrow & A \vdash s\vec{x} = t\vec{x} & & \\ \iff & [s\vec{x}] = [t\vec{x}] & & \\ & & = v(\delta\vec{x}) & t \in \tau v \end{array}$$

□

Of course, the above statement is rather weak. What we are going to show now is a much more interesting fact, namely that τ is a left inverse of δ^*. This immediately implies that:

1. δ^* is injective,
2. $\tau_T|_{\operatorname{ran}\delta_T}$ is surjective on $\mathit{Ter}_T/_A$,
3. $\operatorname{ran}\delta_T \cong \mathit{Ter}_T/_A$ with $\tau_T|_{\operatorname{ran}\delta_T}$ and δ^* being the corresponding set isomorphism and its inverse, respectively.

Remark The latter two facts would be of major importance to us if we had based our proof on Friedman's approach [16]. There, we would have to show that $\tau_T|_{\operatorname{ran}\delta_T}$ can be extended to a suitable partial homomorphism from $\operatorname{ran}\delta_T$ onto $\mathit{Ter}_T/_A$ (see the discussion in Section 4.8).

Proposition 4.3.4 $\tau(\delta t) = [t]$

Proof We prove $\forall T\,\forall t \in \mathit{Ter}_T : \tau(\delta t) = [t]$ by induction on T. Let $T = \vec{T}C$ and $\vec{x} \in \mathit{Var}_{\vec{T}}$. By Proposition 4.3.3 it suffices to show: $\delta t(\delta\vec{x}) = [t\vec{x}]$.

$$\begin{array}{rcll} \delta t(\delta\vec{x}) & = & [t(\rho(\tau(\delta\vec{x})))] & \\ & = & [t(\rho[\vec{x}])] & \text{IH} \\ & = & [t\vec{x}] & \end{array}$$

□

The second claim of the following proposition states that δ is a homomorphism from terms to domains of $\mathcal{D}$.

The first claim of the proposition states how functional application of values constructed by δ to arguments corresponds to syntactic application of the initial terms. While being somewhat technical it turns out to be of some help not only for the proof of the second claim, but also later for the important Lemma 4.5.6 ("Faithfulness Lemma").

Proposition 4.3.5

1. $\delta t\vec{v} = \delta(t(\rho(\tau\vec{v})))$

2. $\delta s(\delta t) = \delta(st)$

Proof

1. Let $t : \vec{T}\vec{T'}$, $\vec{v} \in \mathcal{D}\vec{T}$. Then

$$\begin{array}{rcl} \delta_{\vec{T}\vec{T'}} t\vec{v} & = & \lambda\vec{u}\vec{w} \in \mathcal{D}(\vec{T}\vec{T'}).[t(\rho(\tau\vec{u}))(\rho(\tau\vec{w}))]\vec{v} \\ & = & \lambda\vec{w} \in \mathcal{D}\vec{T'}.[t(\rho(\tau\vec{v}))(\rho(\tau\vec{w}))] \\ & = & \delta_{\vec{T'}}(t(\rho(\tau\vec{v}))) \end{array}$$

2.

$$\begin{array}{rcll} \delta s(\delta t) & = & \delta(s(\rho(\tau(\delta t)))) & \text{Part 1} \\ & = & \delta(s(\rho[t])) & \text{Prop. 4.3.4} \\ & = & \delta(st) & \text{Prop. 4.3.2} \end{array}$$

□

For every interpretation $\mathcal{I}$ extending a structure $\mathcal{D}$ with $|\{u \,|\, \mathcal{I}u \neq \delta u\}| < \infty$ we define a **corresponding substitution** $\theta_{\mathcal{I}}$:

$$\theta_{\mathcal{I}} \stackrel{\text{def}}{=} \lambda u \in \mathit{Nam}.\text{if } \mathcal{I}u = \delta u \text{ then } u \text{ else } \rho(\tau(\mathcal{I}x))$$

The substitution corresponding to an interpretation $\mathcal{I}$, $\theta_{\mathcal{I}}$, is used to reflect the behaviour of $\mathcal{I}$ on term level. Given a name u, $\theta_{\mathcal{I}}u$ returns a term from the image of $\mathcal{I}u$ under τ, provided the image is non-empty. The essential properties $\theta_{\mathcal{I}}$ are summarized by the following two propositions.

Proposition 4.3.6 *For all* $\mathcal{I} \supseteq \mathcal{D}$: $[\theta_{\mathcal{I}}u] = [\rho(\tau(\mathcal{I}u))]$.

Proof We distinguish two cases:

Case $\mathcal{I}u \neq \delta u$. The claim follows immediately by the definition of $\theta_{\mathcal{I}}$.

Case $\mathcal{I}u = \delta u$.

$$\begin{array}{rcll} [\theta_{\mathcal{I}}u] & = & [u] & \\ & = & [\rho[u]] & \\ & = & [\rho(\tau(\delta u))] & \text{Prop. 4.3.4} \\ & = & [\rho(\tau(\mathcal{I}u))] & \end{array}$$

□

Proposition 4.3.7 *For all* $\mathcal{I} \supseteq \mathcal{D}$: $[\theta_{\mathcal{I}^u_v}t] = [\theta_{\mathcal{I}}[u := \rho(\tau v)]t]$

Proof It suffices to show: $\forall u' \in \mathit{Nam} : [\theta_{\mathcal{I}^u_v}u'] = [\theta_{\mathcal{I}}[u := \rho(\tau v)]u']$

Case $u' = u$.

$$\begin{array}{rcll} [\theta_{\mathcal{I}^u_v}u] & = & [\rho(\tau(\mathcal{I}^u_v u))] & \text{Prop. 4.3.6} \\ & = & [\rho(\tau v)] & \\ & = & [\theta_{\mathcal{I}}[u := \rho(\tau v)]u] & \end{array}$$

Case $u' \neq u$.

$$\begin{array}{rcll} [\theta_{\mathcal{I}^u_v}u'] & = & [\rho(\tau(\mathcal{I}^u_v u'))] & \text{Prop. 4.3.6} \\ & = & [\rho(\tau(\mathcal{I}u'))] & \\ & = & [\theta_{\mathcal{I}}u'] & \text{Prop. 4.3.6} \\ & = & [\theta_{\mathcal{I}}[u := \rho(\tau v)]u'] & \end{array}$$

□

4.4 Soundness of the Construction

After discussing generic term structures, we will now focus our attention on a specific member of this family.

Given some term structure $\mathcal{D}$, we define $\mathcal{M}$ as the unique structure over Σ such that:

- $\mathcal{M}C = \mathcal{D}C$,
- $\mathcal{M}c = \delta c$.

In contrast to generic term structures, $\mathcal{M}$ is required to interpret parameters from Σ in the same way as it is done by δ.

We claim that $\mathcal{M}$ is a model of A, provided A is plain. As a technical device to prove this claim, we introduce a restricted form of interpretations.

An interpretation $\mathcal{I}$ is called an **assignment** if the set $\{u \,|\, \mathcal{I}u \neq \delta u\}$ is finite and contains no parameters.

The notion of an assignment is defined in such a way that, independently of Σ, there always exist assignments extending $\mathcal{M}$. In particular, $\delta|_{Nam}$ is an assignment which extends Σ. Moreover, for every term t licensed by Σ and every interpretation $\mathcal{I}$, denotational coherence ensures the existence of an assignment $\mathcal{I}'$ such that $\mathcal{I}t = \mathcal{I}'t$.

Crucial for the following is the fact that every substitution corresponding to an assignment has finite domain. This allows us to reduce evaluation of terms with respect to arbitrary assignments to evaluation of their substitution instances by δ.

Lemma 4.4.1 *If t is plain and $\mathcal{I} \supseteq \mathcal{M}$ is an assignment, then $\hat{\mathcal{I}}t = \delta(\theta_{\mathcal{I}}t)$.*

Proof Let $\mathcal{I} \supseteq \mathcal{M}$ be plain. We show that, for all plain terms t: $\mathcal{I}t = \delta(\theta_{\mathcal{I}}t)$, by induction on $|t|$.

Case $t = x$. Since t is plain, x is basic and hence:

$$\begin{array}{rcll} \hat{\mathcal{I}}x & = & \mathcal{I}x & \\ & = & [\rho(\tau(\mathcal{I}x))] & \text{Prop. 4.3.1} \\ & = & [\theta_{\mathcal{I}}x] & \text{Prop. 4.3.6} \\ & = & \delta(\theta_{\mathcal{I}}x) & \text{Prop. 4.3.1} \end{array}$$

Case $t = c$. Then, since $\mathcal{I}$ is an assignment, $\hat{\mathcal{I}}c = \mathcal{I}c = \delta c = \delta(\theta_{\mathcal{I}}c)$.

Case $t = t_1 t_2$:

$$\begin{array}{rcll} \hat{\mathcal{I}}(t_1 t_2) & = & \hat{\mathcal{I}}t_1(\hat{\mathcal{I}}t_2) & \\ & = & \delta(\theta_{\mathcal{I}}t_1)(\delta(\theta_{\mathcal{I}}t_2)) & \text{IH} \\ & = & \delta((\theta_{\mathcal{I}}t_1)(\theta_{\mathcal{I}}t_2)) & \text{Prop. 4.3.5} \\ & = & \delta(\theta_{\mathcal{I}}(t_1 t_2)) & \end{array}$$

Case $t = \lambda x.s$. Let $x : T$, $s : T'$, w.l.o.g. $\mathcal{I}x = \delta x$, $x \notin \bigcup_{u \in \mathcal{N}t} \mathcal{V}(\theta_{\mathcal{I}}u)$. Hence

$$\begin{array}{rcl} \delta_{TT'}(\theta_{\mathcal{I}}(\lambda x.s)) & = & \delta_{TT'}(\lambda x.\theta_{\mathcal{I}}s) \\ & = & \lambda v \in \mathcal{D}T.\delta_{T'}((\lambda x.\theta_{\mathcal{I}}s)(\rho(\tau v))) \\ & = & \lambda v \in \mathcal{D}T.\delta_{T'}((\theta_{\mathcal{I}}s)[x := \rho(\tau v)]) \\ & = & \lambda v \in \mathcal{M}T.\delta_{T'}((\theta_{\mathcal{I}}s)[x := \rho(\tau v)]) \end{array}$$

and $\hat{\mathcal{I}}(\lambda x.s) = \lambda v \in \mathcal{M}T.\hat{\mathcal{I}}^x_v s$. Let $v \in \mathcal{M}T$. It remains to show:

$$\hat{\mathcal{I}}^x_v s = \delta((\theta_{\mathcal{I}}s)[x := \rho(\tau v)])$$

This can be done as follows:

$$\begin{array}{rcll} \hat{\mathcal{I}}^x_v s & = & \delta(\theta_{\mathcal{I}^x_v} s) & \text{IH} \\ & = & \delta(\theta_{\mathcal{I}}[x := \rho(\tau v)]s) & \text{Prop. 4.3.7} \\ & = & \delta((\theta_{\mathcal{I}}s)[x := \rho(\tau v)]) & \theta_{\mathcal{I}}x = x \end{array}$$

□

Now we are ready to prove our claim.

Theorem 4 (Soundness) *If A is plain, then $\mathcal{M} \models A$.*

Proof Let A be plain, $s = t \in A$. Since A is licensed by Σ, it suffices to show that $\mathcal{I}s = \mathcal{I}t$ for every assignment $\mathcal{I} \supseteq \mathcal{M}$. So, let $\mathcal{I} \supseteq \mathcal{M}$ be an assignment.

$$\begin{array}{rcll} \hat{\mathcal{I}}s & = & \delta(\theta_{\mathcal{I}}s) & \text{Lemma 4.4.1} \\ & = & \delta(\theta_{\mathcal{I}}t) & \text{Prop. 4.3.2} \\ & = & \hat{\mathcal{I}}t & \text{Lemma 4.4.1} \end{array}$$

□

4.5 Faithfulness and Completeness

Generally speaking, deduction is complete for a given set of axioms if for every pair of distinct convertibility classes there exists an evaluation satisfying the axioms which maps the two classes to distinct values. Similarly to our approach in Chapter 3, we are going to prove a stronger statement, namely the existence of a single evaluation that maps every pair of distinct convertibility classes to distinct values. This evaluation will extend our previously constructed model $\mathcal{M}$. So, our main goal now is to show that such an extension is possible. In the case of term structures, the desired property will be called "faithfulness".

We say that a structure $\mathcal{D}$ is **faithful** if there exists an interpretation $\mathcal{I} \supseteq \mathcal{D}$ such that $\forall t : \tau(\hat{\mathcal{I}}t) = [t]$. In such a case the interpretation $\mathcal{I}$ is said to **witness** the faithfulness of $\mathcal{D}$.

Let us first convince ourselves that faithfulness is indeed a sufficient criterion for completeness:

Proposition 4.5.1 *If A is plain and $\mathcal{M}$ is faithful, then $\mathcal{M}$ is a minimal model of A.*

Proof Let A be plain. By Theorem 4 this implies $\mathcal{M} \models A$. Now assume $\mathcal{M}$ faithful, $\mathcal{I}$ being the witness, and $\mathcal{M} \models s = t$. To show: $A \vdash s = t$.

$$\begin{array}{rcl} & \mathcal{M} \models s = t & \\ \Longrightarrow & \hat{\mathcal{I}}s = \hat{\mathcal{I}}t & \\ \Longrightarrow & \tau(\hat{\mathcal{I}}s) = \tau(\hat{\mathcal{I}}t) & \\ \Longleftrightarrow & [s] = [t] & \mathcal{I} \text{ witnesses faithfulness of } \mathcal{M} \\ \Longleftrightarrow & A \vdash s = t & \end{array}$$

□

Corollary 4.5.2 *If A is plain and $\mathcal{M}$ is faithful, then $A \models e \Longrightarrow A \vdash e$, for all equations e, i.e. A is deductively complete.*

There are several ways to prove $\mathcal{M}$ faithful. In the following, we will present two of the possible approaches. One of the approaches extends the corresponding proof by Plotkin [39] from the pure λ-theory of $\varnothing$ to larger theories, in the same way as $\mathcal{M}$

extends his model construction. While being short and intuitive, Plotkin's approach relies on the existence of η-long, β-normal forms (also $\beta\overline{\eta}$-normal forms; see [48] for details). Termination of $\beta\overline{\eta}$-normalization is, of course, a fairly non-trivial property in its own right. It is known to be reducible to that of β-normalization (see [48, 30]), which is provable using the methodology due to Tait [47] and Girard [18]. For this reason we also give a different, independent proof. This approach is slightly longer but, in return, does not rely on $\beta\overline{\eta}$-normalization. Let us begin with the latter.

4.5.1 Faithfulness Lemma

For every type T, the set of **denotable values** D_T is defined by recursion on T:

$$D_{\vec{T}C} \stackrel{\text{def}}{=} \{v \in \mathcal{D}(\vec{T}C) \mid \forall \vec{w} \in D_{\vec{T}} : \ \delta(\rho(\tau v))\vec{w} = v\vec{w}\}$$

Proposition 4.5.3 *δu is denotable, for every name u.*

Proof To show: For every type $T = \vec{T}C$ and every $u \in Nam_T$ it holds $\delta u \in D_T$, i.e. $\forall \vec{v} \in D_{\vec{T}} : \delta(\rho(\tau\delta u))\vec{v} = \delta u\vec{v}$. We show a stronger claim:

$$\begin{array}{rcll} \delta(\rho(\tau(\delta u))) & = & \delta(\rho[u]) & \text{Prop. 4.3.4} \\ & = & \delta u & \text{Prop. 4.3.2} \end{array}$$

□

The following two propositions state useful intuitions about denotable values but are not needed to prove faithfulness of $\mathcal{M}$. First, we observe that every basic value is denotable. Second, every denotable value has a non-empty image under τ.

Proposition 4.5.4 $D_C = \mathcal{D}C$

Proof It suffices to show $\mathcal{D}C \subseteq D_C$. So let $v \in \mathcal{D}C$. We show $v \in D_C$, i.e. $\delta(\rho(\tau v)) = v$:

$$\begin{array}{rcll} \delta(\rho(\tau v)) & = & [\rho(\tau v)] & \text{Prop. 4.3.1} \\ & = & [\rho v] & \text{Prop. 4.3.1} \\ & = & v & \text{Prop. 4.3.1} \end{array}$$

□

Proposition 4.5.5 $v \in D_T \Longrightarrow \rho(\tau v) \in \tau v$

Proof Let $v \in D_{\vec{T}C}$. Sufficient to show: $\forall \vec{x} \in Var_{\vec{T}} : \ [\rho(\tau v)\vec{x}] = v(\delta\vec{x})$

$$\begin{array}{rcll} [\rho(\tau v)\vec{x}] & = & [\rho(\tau v)(\rho[\vec{x}])] & \\ & = & [\rho(\tau v)(\rho(\tau(\delta\vec{x})))] & \text{Prop. 4.3.4} \\ & = & \delta(\rho(\tau v))(\delta\vec{x}) & \\ & = & v(\delta\vec{x}) & v \in D_{\vec{T}C}, \text{Prop. 4.5.3} \end{array}$$

□

An interpretation $\mathcal{I}$ is called **denotable** if it maps every name to a denotable value and $\{u \mid \mathcal{I}u \neq \delta u\}$ is finite.

Lemma 4.5.6 (Faithfulness) *For every* $t : \vec{T}C$ *and for every denotable* $\mathcal{I} \supseteq \mathcal{M}$*:*

1. $\vec{v} \in D_{\vec{T}} \Longrightarrow \hat{\mathcal{I}}t\vec{v} = \delta(\theta_{\mathcal{I}}t)\vec{v}$
2. $\tau(\hat{\mathcal{I}}t) = [\theta_{\mathcal{I}}t]$
3. $\hat{\mathcal{I}}t \in D_{\vec{T}C}$

Proof We proceed in three steps:

1. We show (1) $\Longrightarrow$ (2). Let (1). By Proposition 4.3.3 it suffices to show $\theta_{\mathcal{I}}t \in \tau(\hat{\mathcal{I}}t)$, i.e. $\forall \vec{x} \in \mathit{Var}_{\vec{T}} : \hat{\mathcal{I}}t(\delta\vec{x}) = [(\theta_{\mathcal{I}}t)\vec{x}]$. Let $\vec{x} \in \mathit{Var}_{\vec{T}}$. By Proposition 4.5.3 and (1), Proposition 4.3.5 and Proposition 4.3.1 we have
$$\hat{\mathcal{I}}t(\delta\vec{x}) = \delta(\theta_{\mathcal{I}}t)(\delta\vec{x}) = \delta((\theta_{\mathcal{I}}t)\vec{x}) = [(\theta_{\mathcal{I}}t)\vec{x}]$$
2. We show (1) $\wedge$ (2) $\Longrightarrow$ (3). Let (1), (2) and $\vec{v} \in D_{\vec{T}}$. It suffices to show that $\delta(\rho(\tau(\hat{\mathcal{I}}t)))\vec{v} = \hat{\mathcal{I}}t\vec{v}$. We have:
$$\begin{array}{rcll} \delta(\rho(\tau(\hat{\mathcal{I}}t)))\vec{v} & = & \delta(\rho[\theta_{\mathcal{I}}t])\vec{v} & (2) \\ & = & \delta(\theta_{\mathcal{I}}t)\vec{v} & \text{Prop. 4.3.2} \\ & = & \hat{\mathcal{I}}t\vec{v} & (1) \end{array}$$
3. We show (1) by induction on $|t|$. The previous two steps imply that the inductive hypothesis can always be weakened to a corresponding instance of (2) or (3). Let $\vec{v} \in D_{\vec{T}}$.

   Case $t = x$.
$$\begin{array}{rcll} \hat{\mathcal{I}}x\vec{v} & = & \mathcal{I}x\vec{v} & \\ & = & \delta(\rho(\tau(\mathcal{I}x)))\vec{v} & \mathcal{I}x \text{ denotable} \\ & = & \delta(\theta_{\mathcal{I}}x)\vec{v} & \text{Prop. 4.3.6, 4.3.2} \end{array}$$

   Case $t = c$. We show a stronger claim: $\hat{\mathcal{I}}c = \delta c = \delta(\theta_{\mathcal{I}}c)$.

   Case $t = t_1t_2$.
$$\begin{array}{rcll} \hat{\mathcal{I}}(t_1t_2)\vec{v} & = & \hat{\mathcal{I}}t_1(\hat{\mathcal{I}}t_2)\vec{v} & \\ & = & \delta(\theta_{\mathcal{I}}t_1)(\hat{\mathcal{I}}t_2)\vec{v} & \text{IH(3)}, \text{IH(1)} \\ & = & \delta((\theta_{\mathcal{I}}t_1)(\rho(\tau(\hat{\mathcal{I}}t_2))))\vec{v} & \text{Prop. 4.3.5} \\ & = & \delta((\theta_{\mathcal{I}}t_1)(\rho[\theta_{\mathcal{I}}t_2]))\vec{v} & \text{IH(2)} \\ & = & \delta((\theta_{\mathcal{I}}t_1)(\theta_{\mathcal{I}}t_2))\vec{v} & \text{Prop. 4.3.2} \\ & = & \delta(\theta_{\mathcal{I}}(t_1t_2))\vec{v} & \end{array}$$

   Case $t = \lambda x.s$ where $\mathcal{I}x = \delta x$ and $x \notin \bigcup_{u \in \mathcal{N}t} \mathcal{V}(\theta_{\mathcal{I}}u)$. Let $x : T_1$ and $\vec{v} = w\vec{w}$. Since, by assumption, $\vec{v} \in D_{\vec{T}}$, $\mathcal{I}^x_w$ is denotable.
$$\begin{array}{rcll} \hat{\mathcal{I}}(\lambda x.s)\vec{v} & = & (\lambda v \in \mathcal{M}T_1.\hat{\mathcal{I}}^x_v s)\vec{v} & \\ & = & \hat{\mathcal{I}}^x_w s\vec{w} & \\ & = & \delta(\theta_{\mathcal{I}^x_w} s)\vec{w} & \text{IH(1)} \\ & = & \delta(\theta_{\mathcal{I}}[x := \rho(\tau w)]s)\vec{w} & \text{Prop. 4.3.7} \\ & = & \delta((\lambda x.\theta_{\mathcal{I}}s)(\rho(\tau w)))\vec{w} & \beta \\ & = & \delta(\lambda x.\theta_{\mathcal{I}}s)\vec{v} & \text{Prop. 4.3.5} \\ & = & \delta(\theta_{\mathcal{I}}(\lambda x.s))\vec{v} & \end{array}$$

□

To show faithfulness of $\mathcal{M}$ it remains to construct a witnessing interpretation. So, let $\mathcal{I}_\delta$ an interpretation extending $\mathcal{M}$ such that $\forall u \in \mathit{Nam} : \mathcal{I}_\delta u = \delta u$. Clearly, the substitution corresponding to $\mathcal{I}_\delta$, $\theta_{\mathcal{I}_\delta}$, is an identity function. Therefore, faithfulness of $\mathcal{M}$ is an immediate consequence of the second claim of Lemma 4.5.6:

Corollary 4.5.7 (Faithfulness) *$\mathcal{M}$ is faithful, with $\mathcal{I}_\delta$ being the witness.*

4.5.2 Plotkin-style Faithfulness Proof

As it was already mentioned before, the following proof idea assumes that every term can be $\beta\eta$-reduced to a $\beta\overline{\eta}$-normal form.

Plotkin's construction corresponding to τ has a noteworthy difference to our definition. While we say that a term t is in $\tau_{\vec{T}C} v$ if $v(\delta\vec{x}) = [t\vec{x}]$ for all $\vec{x} \in \mathit{Var}_{\vec{T}}$, according to Plotkin's definition it suffices if the equation holds for non-repeating $\vec{x}$. If a vector $\vec{x}$ is non-repeating, i.e. consists of pairwise distinct variables, it is possible to write an arbitrary application $t\vec{x}$ of a $\beta\overline{\eta}$-normal term t to $\vec{x}$ as $(\lambda\vec{x}.s)\vec{x}$, which can then be β-reduced to s. While Plotkin's original formulation of the proof depends on this property, in general it turns out to be non-essential. The following proposition states a generalization of the above β-reduction. This generalization will then enable us to use our initial, simpler definition of τ.

Proposition 4.5.8 *Let $\vec{x}$ be pairwise distinct and θ a substitution such that $\theta x = x$ if x does not occur in $\vec{x}$. Then $\vdash (\lambda\vec{x}.t)(\theta\vec{x}) = \theta t$*

Proof By induction on $|\vec{x}|$.

Case $\vec{x} = \langle\rangle$. Then $\theta x = x$ holds for all $x \in \mathit{Var}$, so $t = \theta t$.

Case $\vec{x} = \vec{y}x$.

$$
\begin{array}{lll}
(\lambda\vec{x}.t)(\theta\vec{x}) & & \\
\quad = & (\lambda\vec{y}x.t)(\theta\vec{y})(\theta x) & \\
\quad = & (\lambda\vec{y}x.t)(\theta[x := x]\vec{y})(\theta x) & x \notin \vec{y} \\
\quad = & \theta[x := x](\lambda x.t)(\theta x) & \text{IH} \\
\quad = & \theta(\lambda x.t)(\theta x) & x \notin \mathcal{V}(\lambda x.t) \\
\quad = & \theta((\lambda x.t)x) & \\
\quad = & \theta t & \beta
\end{array}
$$

□

For the final step we would like to re-use the interpretation $\mathcal{I}_\delta$, which we have introduced for Corollary 4.5.7 to witness faithfulness of $\mathcal{M}$. It is not surprising that $\mathcal{I}_\delta$ does the same job here.

Lemma 4.5.9 (Plotkin) *If t is $\beta\overline{\eta}$-normal, then $\tau(\hat{\mathcal{I}}_\delta t) = [t]$.*

Proof By induction on $|t|$. Let $t : \vec{T}C$ be $\beta\overline{\eta}$-normal. By Proposition 4.3.3 it suffices to show that $t \in \tau(\hat{\mathcal{I}}_\delta t)$, i.e. $\forall\vec{x} \in \mathit{Var}_{\vec{T}} : \hat{\mathcal{I}}_\delta t(\delta\vec{x}) = [t\vec{x}]$.

So let $\vec{x} \in Var_{\vec{T}}$, $t = \lambda\vec{y}.t_0\vec{t}$ where t_0 is atomic and, w.l.o.g., $\vec{y}$ pairwise distinct. Let $\theta = [\vec{y} := \vec{x}]$. Then, by Proposition 4.5.8:

$$\vdash t\vec{x} = t(\theta\vec{y}) = \theta(t_0\vec{t}) = (\theta t_0)(\theta\vec{t}) \qquad (*)$$

Hence

$$\begin{array}{rcll} \hat{\mathcal{I}}_\delta t(\delta\vec{x}) & = & \hat{\mathcal{I}}_\delta t(\hat{\mathcal{I}}_\delta\vec{x}) & \\ & = & \hat{\mathcal{I}}_\delta(t\vec{x}) & \\ & = & \hat{\mathcal{I}}_\delta((\theta t_0)(\theta\vec{t})) & (*) \\ & = & \hat{\mathcal{I}}_\delta(\theta t_0)(\hat{\mathcal{I}}_\delta(\theta\vec{t})) & \\ & = & \delta(\theta t_0)(\hat{\mathcal{I}}_\delta(\theta\vec{t})) & t_0 \text{ atomic} \\ & = & \delta((\theta t_0)(\rho(\tau(\hat{\mathcal{I}}_\delta(\theta\vec{t}))))) & \text{Prop. 4.3.5} \\ & = & [(\theta t_0)(\rho(\tau(\hat{\mathcal{I}}_\delta(\theta\vec{t}))))] & \text{Prop. 4.3.1} \\ & = & [(\theta t_0)(\rho[\theta\vec{t}])] & \text{IH} \\ & = & [(\theta t_0)(\theta\vec{t})] & \\ & = & [t\vec{x}] & (*) \end{array}$$

□

Since $\mathcal{M}$ is standard and hence compatible with $\beta\eta$-conversion, we immediately obtain the desired stronger claim:

Corollary 4.5.10 $\tau(\mathcal{M}\delta t) = [t]$

4.6 Consequences

Plain axioms allow us to describe a wide class of interesting equational λ-theories. Since the class of plain terms is a proper superset of both one-sorted and many-sorted first-order formulas, it is not very surprising that, as far as deduction is concerned, first-order logic can easily be simulated by an adequate plain axiomatization. In Figure 4.1 we saw an example of such an axiomatization. It can be extended in an intuitive way to describe arbitrary constructions formulated in many-sorted predicate logic. Obviously, plain axioms can also be used to deductively simulate fragments of first-order logic, like propositional logic or modal logics.

What about semantics? Traditionally formal semantics of both first-order and higher-order logic always includes the definition of a distinguished domain, namely the domain of truth values. It is assumed to consist of exactly two values which are commonly referred to as "truth" and "falsehood". Models which satisfy this requirement are said to have **Boolean extensionality**. Truth values play a crucial role in traditional semantic constructions in so far as they are used to define the notion of semantic validity of formulas. Their syntactic counterparts, propositional constants and connectives, are usually integrated into rules of inference, which makes them an essential part of the overall logical formalism.

In equational logic on the other hand, semantic validity does not depend in the least on any particular definition of truth values, just as equational deduction does not depend on propositional constants. Indeed, equational logic is often used in a context where an explicit notion of truth is unnecessary, and hence, the domain of truth values is not even defined. The prototypical example of such an application is formal group theory. As a consequence of this practice, equational semantics typically does not make any statements about the interpretation of Boolean constants or the structure of the propositional domain. Instead, all the essential semantic properties of a formal system are assumed to be specified by corresponding axioms. This is also the approach that we pursue in the definition of S.

To simulate traditional first-order or higher-order logic in S, it therefore seems necessary to provide a sufficiently precise equational specification of the propositional domain and a reasonable selection of Boolean constants. We want to know in how far this task can be accomplished by using only plain axioms.

Given a structure $\mathcal{A}$ and a type T licensed by $\Sigma_{\mathcal{A}}$, the domain $\mathcal{A}T$ is called **trivial** if $|\mathcal{A}T| = 1$. A structure $\mathcal{A}$ is called trivial if $\mathcal{A}T$ is trivial for all types T licensed by $\Sigma_{\mathcal{A}}$.

Note that triviality on base types implies triviality on certain functional types:

Proposition 4.6.1 *A structure $\mathcal{A}$ is trivial on a sort C if and only if it is trivial on all types of the form $\vec{T}C$.*

Proof One direction is immediate. The other one is proved by induction on $|\vec{T}|$. □

In analogy to satisfiability with respect to general semantics, we say that a set A of equations is **satisfiable** at type T (with respect to standard semantics) if it has a model $\mathcal{A}$ such that $\mathcal{A}T$ is non-trivial. A is called satisfiable if it has a non-trivial model. Note that in every term structure $\mathcal{D}$, a basic domain $\mathcal{D}C$ is non-trivial if and only if $|\mathcal{D}C| = \aleph_0$, which follows by essentially the same chain of reasoning as the one used for proving Proposition 3.4.1, now limited to base types:

Proposition 4.6.2 *For every set A of plain equations there exists a model $\mathcal{A} \models A$ such that for every $C \in Sor$: $|\mathcal{A}C| = 1$ or $|\mathcal{A}C| = \aleph_0$.*

Proof Non-triviality of $\mathcal{D}C$ requires that there exist at least two terms of type C which are not deductively equivalent. This implies that no two distinct variables of type C may be deductively equivalent, i.e. contained in the same convertibility class. Therefore, there must exist at least as many convertibility classes in $\mathcal{D}C$ as there are distinct variables of type C. So, the definitions of variables and typings give us $\aleph_0$ as a lower bound on the cardinality of $\mathcal{D}C$. The upper bound is provided by the fact that we have only countably many terms. □

Corollary 4.6.3 (Löwenheim-Skolem) *A set A of plain equations is satisfiable at a base type C if and only if it has a model $\mathcal{A}$ such that $|\mathcal{A}C| = \aleph_0$.*

Hence, every plain axiomatization of FOL in S will allow models which interpret the Boolean type by an infinite set. Such models of S clearly cannot be considered models of FOL or HOL in the traditional sense. On the other hand, a traditional model of FOL or a traditional standard model of HOL satisfies all the requirements to a model of S. In this sense, the notion of a model in S is weaker than the traditional notion.

This weakness becomes deductively apparent when we try to use a plain axiomatization of FOL to derive higher-order theorems. Consider, for instance, the equation $yx = y(y(yx))$, where $x : \mathsf{B}$ and $y : \mathsf{BB}$. It is not hard to check that this equation is valid in a model if and only if it interprets the Boolean type by a two-valued set (see [45]). Therefore the equation is certainly valid in every traditional model of FOL or HOL. Indeed, it turns out to be a theorem of traditional higher-order logic with equality. In FOL, the equation, while expressing a valid statement when considered at the meta-level, cannot be expressed at the object level because it involves Boolean and higher-order variables. Although our deduction formalism does allow us to express the above equation, because of Proposition 4.6.2 the equation does not follow semantically and hence is not derivable from any set of plain axioms which is satisfiable at B.

On first-order terms however, validity with respect to equational semantics of S can coincide with validity with respect to traditional semantics. To convince ourselves of this fact, we observe that, while being able to deductively simulate traditional formulations of FOL, our axiomatization of FOL is sound with respect to traditional semantics. The claim follows by deductive completeness of FOL with respect to traditional semantics, first shown by Gödel [19].

Since plain terms are a superset of first-order formulas, this implies that plain axioms in S are semantically at least as expressive as traditional FOL. The converse does not hold. Although plain axioms are semantically less expressive than unrestricted HOL, they allow to specify certain properties of higher-order objects which cannot be described by first-order terms. Consider, for instance, the equation $\forall_{\mathsf{IB}} x.1 = 1$. It partially describes second-order universal quantification. More specifically, every model $\mathcal{A}$ satisfying this equation has to interpret the parameter $\forall_{\mathsf{IB}}$ by a function that, whenever applied to $\lambda v \in \mathcal{A}(\mathsf{IB}).\mathcal{A}1$, returns $\mathcal{A}1$. Since $\forall_{\mathsf{IB}}$ is not a first-order constant, it may not appear in a first-order formula. Hence, every satisfiable set of first-order formulas will allow models $\mathcal{A}$ which interpret $\forall_{\mathsf{IB}}$ by $\lambda v \in \mathcal{A}((\mathsf{IB})\mathsf{B}).\mathcal{A}0$, contradicting the above equation whenever $\mathcal{A}0 \neq \mathcal{A}1$.

We have already seen that semantic expressiveness of equational theories induced by plain axioms is greater than that of traditional FOL. On the other hand, we know that both traditional HOL and unrestricted, recursively generated λ-theories provide semantic expressiveness which is sufficient to characterize the natural numbers or non-trivial finite domains (see Section 2.2 and [5, 30]). So, independently of each other, Proposition 4.6.2 and the main completeness result for plain axioms imply that λ-theories induced by plain axioms are semantically strictly less expressive than unrestricted higher-order calculi.

Let us conclude our discussion of plain axiom systems by proving their compactness:

Proposition 4.6.4 (Compactness) *A set of plain equations A is satisfiable at a type T if and only if every finite subset of A is satisfiable at T.*

Proof Let A be a set of plain equations and $T = \vec{T}C$ a type such that every finite subset of A is satisfiable at T. By Proposition 4.6.1, every subset of A is also satisfiable at C. Let $x, y : C$ be two distinct variables. By soundness, the equation $x = y$ is not provable from any of the finite subsets of A. Consequently, since all proofs from A are finite,

$A \not\vdash x = y$, i.e. $[x]_A \neq [y]_A$. Hence $|\mathcal{MC}| > 1$. Then A is satisfiable by Theorem 4. The converse is immediate. □

4.7 Excursion: Fixed Points and Incompleteness of S

In Section 2.2 we saw an axiomatic system of S that is not complete with respect to standard semantics. We want to present another, conceptually much simpler example of an axiomatic system with incomplete deduction.

Let us look at a typical definition of a fixed point combinator $\boldsymbol{F}$: $\boldsymbol{F}f = f(\boldsymbol{F}f)$. We know that there exists no standard model satisfying the equation that is non-trivial at $\boldsymbol{\tau}f$, simply because non-trivial standard domains always contain functions which do not have a fixed point. From the study of PCF [41], probably the most important system containing fixed point combinators, we know that the above equation does not suffice to make every two terms of type $\boldsymbol{\tau}f$ convertible (see also [33]). So, given two sequences $\langle \boldsymbol{F}_T \,|\, T \in \mathit{Ty} \rangle$ and $\langle f_T \,|\, T \in \mathit{Ty} \rangle$ such that, for every type T, $\boldsymbol{F}_T \in \mathit{Par}_{(TT)T}$ and $f_T \in \mathit{Var}_{TT}$, the set $\{\boldsymbol{F}_T f_T = f_T(\boldsymbol{F}_T f_T) \,|\, T \in \mathit{Ty}\}$ is an example of an axiomatic system that is incomplete with respect to standard semantics.

4.8 Preceding and Further Work

Unlike general semantic constructions by Friedman [16], Mitchell [33] or our own construction in Chapter 3, the functional domains in $\mathcal{D}$ do not consist of convertibility classes, but have to contain actual functions. Since for all satisfiable axiom systems the standard term model contains infinite domains, it necessarily has functional types with uncountable domains. Therefore in general, there exists no bijection between terms or classes of terms of a functional type and elements of the corresponding domain. This is a major complication which arises when one tries to generalize Birkhoff's completeness proof [10] to higher-order calculi. The solution to this problem is to establish a homomorphic connection between functional convertibility classes and selected subsets of the corresponding domains. This idea was first exploited by Friedman in his completeness proof for the pure simply typed λ-calculus [16]. Using such a partial homomorphism from standard domains to convertibility classes, he reduced the completeness proof for standard models to a proof of completeness with respect to general semantics, similar to the one in Chapter 3. Later Plotkin presented a modified version of the completeness proof [39], which no longer relied on general semantics and made the connection between standard domains and convertibility classes less explicit (see also [37, 46]). While our proof relies on an extension of Plotkin's model construction, we also make explicit the connections between domains of $\mathcal{D}$ and convertibility classes modulo A. The detailed analysis of these connections gives us the insights which are necessary to prove soundness of our construction. It was first applied, in a simplified setting, in [30] to prove standard completeness of the simply typed λ-calculus with algebraic axioms. Unfortunately, the proof of faithfulness for the construction presented in [30] contained a flaw. This gap can now be closed by adapting the corresponding proof by Plotkin [39], analogously to how it was done in Section 4.5.

As we saw in Section 2.2 and 4.7, in general, standard completeness cannot be achieved in the presence of axioms with higher-order variables. Nevertheless, as we will see in the following chapter, there also exist non-trivial deductively complete axiomatic systems which contain higher-order variables. It remains a challenge to find a more precise generic sufficient criterion for deductive completeness with respect to standard semantics which would extend our result for plain axioms to systems with higher-order variables.

5 Propositional Type Theory

Propositional Type Theory was introduced in 1963 by Leon Henkin [25]. Its main difference from Church's type theory [13] is the absence of any base types apart from the type of truth values. Therefore, one can say PTT is a higher-order extension of propositional logic in the same sense as Church's formulation of HOL is a higher-order extension of the first-order predicate calculus. Henkin's formulation of PTT, while based on Church's formulation of type theory, pioneers a different way of introducing logical constants. For every type T, the formulation contains a single primitive constant $Q_T : TT\mathsf{B}$ which denotes the identity relation over the domain of T. Boolean connectives, quantifiers and the description operator are then defined in terms of these primitive constants. Later, this approach was carried over to full HOL by Andrews [2, 4] (see also [26, 5]).

In contrast to the formerly introduced notion of external equality, we call an equivalence relation **internal** if it is the denotation of an object-level constant, like the one introduced by Henkin.

The semantics of PTT is based on the interpretation of the type B of truth values by a two-element set. Functional domains are standard. Henkin shows that logical constants defined in terms of the primitive equality constants Q_T are interpreted in the traditional way. One of the key features of PTT shown by Henkin is the possibility to express every value of an arbitrary domain by a closed term. By exploiting this property Henkin proves deductive completeness of PTT with respect to the above standard semantics.

In the following, we want to give an equational formulation of PTT in S and show its completeness and decidability with respect to the original semantics. We will see that a formulation of PTT in S can be made significantly simpler than Henkin's original approach and moreover, allows us to considerably simplify the corresponding completeness proof. Let us first discuss why this is the case.

Henkin's approach to defining propositional constants in terms of equality has several important implications on the structure of his axioms. His original axiomatization consists of infinitely many axioms corresponding to seven axiom schemes, three of which were immediately shown by Andrews [1] to be derivable from the remaining four. These still contain infinitely many axioms and state, among other things, that, for every type T, the equality relation denoted by Q_T is a congruence which is extensional and compatible with β-conversion.

As an equational formalism, S has a built-in notion of equality, given by the external equality relation. Because of the rules **Cong** and β we know that this relation is a congruence compatible with β-conversion. As it was shown in Section 4.2, external equality in S is also known to be extensional. As we can see, the essential properties of internal equality which have to be axiomatized in Henkin's formulation of PTT come for free with external equality in S. Therefore, it should be possible for us to achieve the

Name	**MT**			
Base Type	B	truth values		
Constants	$0, 1 : \mathsf{B}$	truth and falsehood		
	$(\to) : \mathsf{BBB}$	implication (right associative)		
Variables	$x, y : \mathsf{B}$			
	$f : \mathsf{BB}$			
Notation	$x \vee y$	$\stackrel{\text{def}}{=}$	$(x \to y) \to y$	disjunction
Axioms	$0 \to x$	$=$	1	I0
	$1 \to x$	$=$	x	I1
	$f0 \to f1 \to fx$	$=$	1	BCA
	$x \vee y$	$=$	$y \vee x$	Comm

Figure 5.1: MT: Axiomatization of PTT

same level of deductive expressiveness by relying entirely on external equality, without having to provide an infinite axiomatization of an object-level constant with the same semantics.

5.1 Semantics and Axiomatization of PTT

For the following discussion it is useful to have a notion of isomorphism on structures. So, let $\mathcal{A}$ and $\mathcal{B}$ be two structures over the same signature Σ. An **isomorphism** $h : \mathcal{A} \to \mathcal{B}$ is a family of functions indexed by types $\langle h_T \,|\, T \in Ty \rangle$ such that, for every two types T, T' licensed by Σ:

1. $h_T \in \mathcal{A}T \to \mathcal{B}T$ is bijective,
2. for every $c : T$ in Σ: $h_T(\mathcal{A}c) = \mathcal{B}c$,
3. $h_{TT'} = \lambda v \in \mathcal{A}(TT')\lambda w \in \mathcal{B}T.h_{T'}(v(h_T^{-1}w))$.

$\mathcal{A}$ and $\mathcal{B}$ are called **isomorphic** (notation $\mathcal{A} \cong \mathcal{B}$) if there exists an isomorphism $\mathcal{A} \to \mathcal{B}$.

Note that by the third property of its definition, the behaviour of a structure isomorphism on functional domains is uniquely determined by its behaviour on basic domains, i.e. for every family of bijections $\{h_C : \mathcal{A}C \to \mathcal{B}C \,|\, C \in \Sigma \cap Sor\}$ there exists a unique minimal extension h satisfying the first and the third property of the definition. To show that h is a structure isomorphism it then suffices to prove the second property.

Given two isomorphic models $\mathcal{A}$ and $\mathcal{B}$, one can show that for every set A of equations licensed by Σ: $\mathcal{A} \vDash A \iff \mathcal{B} \vDash A$.

Now the semantics of PTT can be completely represented as properties of a single structure:

Let $\mathcal{T}_2$ be a structure over the signature consisting of a single sort B and the parameters $0 : \mathsf{B}$ (falsehood), $1 : \mathsf{B}$ (truth) and $(\rightarrow) : \mathsf{BB}$ (implication), such that:

1. $\mathcal{T}_2 0 = \mathsf{F}$ and $\mathcal{T}_2 1 = \mathsf{T}$ where $\mathsf{T} \neq \mathsf{F}$,
2. $\mathcal{T}_2 \mathsf{B} = \{\mathsf{F}, \mathsf{T}\}$,
3. $\mathcal{T}_2(\rightarrow)$ has the usual truth table semantics of implication.

We assume $\mathcal{T}_2$ to be the unique non-trivial model of PTT up to isomorphism.

To simplify matters, for the rest of the chapter we consider only types, terms and equations which are licensed by $\Sigma_{\mathcal{T}_2}$, unless stated otherwise.

In the following, we consider several equational axiomatizations of PTT. The first and conceptually simplest one, MT, is presented in Figure 5.1. The initial formulation of MT as well as some discussion of its deductive and semantic properties can be found in [45]. It is fairly easy to check soundness of MT:

Proposition 5.1.1 $\mathcal{T}_2 \vDash \text{MT}$

MT is interesting because it satisfies an important prerequisite for deductive completeness. It entails semantically precisely the valid equations of $\mathcal{T}_2$:

Proposition 5.1.2 $\mathcal{T}_2 \vDash e \iff \text{MT} \vDash e$

Proof

- "$\Leftarrow$": Follows by Proposition 5.1.1.
- "$\Rightarrow$": It suffices to show that every non-trivial model of MT (over $\Sigma_{\mathcal{T}_2}$) is isomorphic to $\mathcal{T}_2$. So, let $\mathcal{A}$ be a non-trivial structure over $\Sigma_{\mathcal{T}_2}$ that satisfies MT. First we have to show that $\mathcal{A}\mathsf{B} \cong \mathcal{T}_2\mathsf{B}$. This can be done in two steps:
  1. We show: $\mathcal{A}0 \neq \mathcal{A}1$. Assume, for contradiction, $\mathcal{A}0 = \mathcal{A}1$. Consequently, we have: $\mathcal{A} \vDash 1 \rightarrow x = 0 \rightarrow x$. By I1 and I0 we obtain $\mathcal{A} \vDash x = 1$, contradicting non-triviality of $\mathcal{A}$.
  2. We show: $\mathcal{A}\mathsf{B} = \{\mathcal{A}0, \mathcal{A}1\}$. Assume, for contradiction, there exists an element $v \in \mathcal{A}\mathsf{B}$ which is distinct from $\mathcal{A}0$ and $\mathcal{A}1$. Let $\mathcal{I} \supseteq \mathcal{A}$ be an interpretation such that $\mathcal{I}x = v$ and $\mathcal{I}f = \lambda w \in \mathcal{A}\mathsf{B}.\text{if } w \in \{\mathcal{A}0, \mathcal{A}1\} \text{ then } \mathcal{A}1 \text{ else } w$. But then $\hat{\mathcal{I}}(f0 \rightarrow f1 \rightarrow fx) = w \neq \mathcal{A}1$, contradicting $\mathcal{A} \vDash \text{BCA}$.

  Now we can construct an isomorphism $h : \mathcal{A} \rightarrow \mathcal{T}_2$. Let, for $c \in \{0, 1\}$, $h_{\mathsf{B}}(\mathcal{A}c) = \mathcal{T}_2 c$. Thus we have specified the behaviour of h on all domains. It remains to show that $h_{\mathsf{BB}}(\mathcal{A}(\rightarrow)) = \mathcal{T}_2(\rightarrow)$. For this purpose it suffices to prove that, for all $v, w \in \mathcal{A}\mathsf{B}$: $h_{\mathsf{B}}(\mathcal{A}(\rightarrow)vw) = \mathcal{T}_2(\rightarrow)(h_{\mathsf{B}}v)(h_{\mathsf{B}}w)$. Since $\mathcal{A}\mathsf{B} = \{\mathcal{A}0, \mathcal{A}1\}$, this fact can be verified by a simple case analysis, using the axioms I0 and I1. □

Next we look at the deductive power of MT. The first thing we observe is that MT is deductively complete for closed basic equations:

Proposition 5.1.3 *Let t be closed and basic. Then either $\mathcal{T}_2 \vdash t = 0$ or $\mathcal{T}_2 \vdash t = 1$.*

Proof By induction on the size of the $\beta\eta$-normal form of t. $\square$

Proposition 5.1.4 *If e is closed and basic, then $\mathcal{T}_2 \vDash e \implies \mathrm{MT} \vdash e$.*

Proof Follows by Proposition 5.1.3 and 5.1.1. $\square$

The next question to ask is whether MT is deductively complete in general, i.e. whether for every equation e:

$$\mathcal{T}_2 \vDash e \implies \mathrm{MT} \vdash e$$

Surprisingly enough, this is the case. The four simple axioms of MT already suffice to completely specify the properties of $\mathcal{T}_2$.

Before we proceed to prove this claim, let us note two useful deductive properties of MT. First, the rule of **modus ponens** (MP) is derivable from MT:

Proposition 5.1.5 (MP) *For all terms s and t, not necessarily licensed by $\Sigma_{\mathcal{T}_2}$:* $\mathrm{MT}, s \to t = 1, s = 1 \vdash t = 1$.

Proof Proceeds analogously to the corresponding proof for FOL, see Section 4.1. $\square$

Second, MT satisfies the following **deduction theorem**:

Proposition 5.1.6 (Deductivity) *For all sets $A \supseteq \mathrm{MT}$, terms t and closed terms s, not necessarily licensed by $\Sigma_{\mathcal{T}_2}$: $A, s = 1 \vdash t = 1 \implies A \vdash s \to t = 1$.*

Proof Let s be closed, $A \supseteq \mathrm{MT}$ and $A, s = 1 \vdash t = 1$. By Proposition 2.1.3, the derivation of $t = 1$ from $A \cup \{s = 1\}$ can be written as a conversion proof $t_0 = \ldots = t_n$, where $n \in \mathbb{N}$, $t_0 = t$, $t_n = 1$. By **Rep**, this proof can be transformed into $s \to t = \ldots = s \to 1$.

Since s is closed and therefore has no substitution instances, in order to show that the new proof no longer depends on $s = 1$, it suffices to prove that for all contexts k: $A \vdash s \to k[s] = s \to k[1]$. Again since s is closed, the target equation can be β-converted to $s \to (\lambda x.k[x])s = s \to (\lambda x.k[x])1$, for some x, which is a substitution instance of $x \to fx = x \to f1$. The subclaim follows by appropriate application of I0, reflexivity and BCA.

By the above, we obtain: $A \vdash s \to t = s \to 1$. It remains to show that $A \vdash s \to 1 = 1$, which is easily doable by I0, I1 and BCA. $\square$

5.2 Completeness of MT

5.2.1 Boolean Axioms and Algebraic Completeness

Before we proceed to the higher-order case, let us have a brief look at the behaviour of MT on algebraic terms. Since $\mathcal{T}_2$ is a higher-order extension of a two-element Boolean algebra, we know that all of its algebraic properties can also be specified by means of Boolean axioms. A reference axiomatization BA is given in Figure 5.2. Such a specification by Boolean axioms is known to be deductively complete for algebraic equations:

Name	**BA**			
Variables	$x, y, z : \mathsf{B}$			
Notation	$\neg x$	$\stackrel{\text{def}}{=}$	$x \to 0$	negation
	$x \vee y$	$\stackrel{\text{def}}{=}$	$(x \to y) \to y$	disjunction
	$x \wedge y$	$\stackrel{\text{def}}{=}$	$\neg(\neg x \vee \neg y)$	conjunction
	$x \equiv y$	$\stackrel{\text{def}}{=}$	$(x \to y) \wedge (y \to x)$	equivalence

Axioms

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

Figure 5.2: BA: Boolean Axioms (for $\Sigma_{\mathcal{T}_2}$)

Proposition 5.2.1 *If e is algebraic, then* $\mathcal{T}_2 \models e \implies \mathrm{BA} \vdash e$.

By proving all of the Boolean axioms derivable from MT we can show that:

Proposition 5.2.2 $\mathrm{BA} \vdash e \implies \mathrm{MT} \vdash e$

Hence, just like BA, MT is deductively complete for algebraic equations:

Corollary 5.2.3 *If e is algebraic, then* $\mathcal{T}_2 \models e \implies \mathrm{MT} \vdash e$.

Boolean axioms have the advantage of being a well-understood system with a lot of useful theorems which facilitate the task of finding formal proofs. Therefore, we want to use BA to establish the following two propositions:

Proposition 5.2.4 *For all terms s and t, not necessarily licensed by $\Sigma_{\mathcal{T}_2}$:*
$\mathrm{BA}, s \equiv t = 1 \dashv\vdash \mathrm{BA}, s = t$

Proof

- "$\dashv$": It suffices to show that $\mathrm{BA} \vdash t \equiv t = 1$, which is easy.
- "$\vdash$": Let BA and $s \equiv t = 1$ be derivable. Then:

$$s = s \wedge s \equiv t = s \wedge (s \to t) \wedge (t \to s) = s \wedge t$$

Analogously, we derive $t = s \wedge t$. The claim follows. □

$$
\begin{array}{rcll}
\uparrow_{\vec{T}\mathsf{B}} v & = & \lambda \vec{x}. \bigvee\limits_{\substack{\vec{w} \in \mathcal{T}_2 \vec{T} \\ v\vec{w} = \mathcal{T}_2 1}} \vec{x} \doteq_{\vec{T}} (\uparrow_{\vec{T}} \vec{w}) & \text{Rei}_{\vec{T}\mathsf{B}} \\
\forall_T & = & \lambda f. \bigwedge\limits_{v \in \mathcal{T}_2 T} f(\uparrow_T v) & \forall\text{Enum}_T \\
(\doteq_{\mathsf{B}}) & = & (\equiv) & \text{BExt}_{\mathsf{B}} \\
(\doteq_{T_1 T_2}) & = & \lambda f g. \forall_{T_1} x. fx \doteq_{T_2} gx & \text{BExt}_{T_1 T_2}
\end{array}
$$

Figure 5.3: MT: Derived Notation

Proposition 5.2.5 *Let X, Y be finite sets of variables. Then*

$$
\text{BA} \vdash \bigwedge_{x \in X} \bigvee_{y \in Y} fxy = \bigvee_{g \in X \to Y} \bigwedge_{x \in X} fx(gx)
$$

Proof By induction on $|X|$.

Case $X = \varnothing$. The claim holds since $\bigwedge\limits_{x \in \varnothing} fx = 1$ and $\bigvee\limits_{g \in \varnothing \to Y} fg = f\varnothing$.

Case $X = \{x_1\} \cup X'$ where $x_1 \notin X'$. Starting from the right-hand side we have:

$$
\begin{array}{rcll}
\bigvee\limits_{g \in X \to Y} \bigwedge\limits_{x \in X} fx(gx) & = & \bigvee\limits_{g \in X \to Y} (\bigwedge\limits_{x \in X'} fx(gx)) \wedge fx_1(gx_1) & \\
& = & \bigvee\limits_{g \in X' \to Y} \bigvee\limits_{y \in Y} (\bigwedge\limits_{x \in X'} fx(gx)) \wedge fx_1 y & \\
& = & (\bigvee\limits_{g \in X' \to Y} \bigwedge\limits_{x \in X'} fx(gx)) \wedge \bigvee\limits_{y \in Y} fx_1 y & \\
& = & (\bigwedge\limits_{x \in X'} \bigvee\limits_{y \in Y} fxy) \wedge \bigvee\limits_{y \in Y} fx_1 y & \text{IH} \\
& = & \bigwedge\limits_{x \in X} \bigvee\limits_{y \in Y} fxy &
\end{array}
$$

□

By Proposition 5.2.2, both of the above propositions remain valid if we replace BA by MT. Hence, we can freely use them in our completeness proof.

5.2.2 Derived Notation

For convenience, we introduce some notational abbreviations, shown in Figure 5.3.

For every type T and every value $v \in \mathcal{T}_2 T$ we define $\uparrow_T v$ to be the term denoting v, also called **reification** of v. Moreover, for every type T we define a term $\forall_T$, denoting universal quantification over $\mathcal{T}_2 T$, and $\doteq_T$, the identity test for elements of $\mathcal{T}_2 T$. $\uparrow_T v$, $\forall_T$ and $\doteq_T$ are defined by mutual recursion on T.

To convince ourselves that the notational extension is well-defined we observe that every instance of the mutually recursive definitions can always be expanded in such a way that at the end the right-hand side contains no derived notation, i.e. the recursive formulation of the defining equations is just a convenient abbreviation for an equivalent non-recursive definition.

It is not hard to check that the abbreviations indeed have the intended semantics:

Proposition 5.2.6 *For all interpretations $\mathcal{I} \supseteq \mathcal{T}_2$:*
1. $\hat{\mathcal{I}}(\uparrow_T v) = v$,
2. $\hat{\mathcal{I}}\forall_T v = (\textit{if } v = (\lambda v \in \mathcal{T}_2 T.\mathcal{T}_2 1) \textit{ then } \mathcal{T}_2 1 \textit{ else } \mathcal{T}_2 0)$,
3. $\hat{\mathcal{I}}(\doteq_T) v w = (\textit{if } v = w \textit{ then } \mathcal{T}_2 1 \textit{ else } \mathcal{T}_2 0)$.

Proof By mutual induction on T. □

Even easier to check is that the derived notation introduces only closed terms:

Proposition 5.2.7 $\uparrow_T v$, $\forall_T$ *and* $\doteq_T$ *are closed.*

Proof By mutual induction on T. □

5.2.3 ∀I: A Sufficient Criterion for Completeness of MT

Now we can formulate a necessary and sufficient criterion for the deductive completeness of MT:

Proposition 5.2.8 MT *is complete if and only if for all types T there exist variables f and x such that:* $\text{MT} \vdash \forall_T f \to f x = 1$.

Proof

- "⇒": Assume MT is complete. It suffices to check that $\mathcal{T}_2 \models \forall_T f \to f x = 1$, which can easily be done by using Proposition 5.1.1 and 5.2.6.
- "⇐": Assume that $\forall_T f \to f x = 1$ is derivable for every T, and that, for some s and t, $\mathcal{T}_2 \models s = t$. To show: $\text{MT} \vdash s = t$. By Corollary 4.2.3 we can assume w.l.o.g. that s and t are basic. By Proposition 5.2.4 it then suffices to show that $\text{MT} \vdash s \equiv t = 1$. Let $\vec{x}$ be the set $\mathcal{V}s \cup \mathcal{V}t$. Since $\mathcal{T}_2 \models s \equiv t = 1$, by Proposition 5.2.6 it also holds: $\mathcal{T}_2 \models \forall \vec{x}.s \equiv t = 1$. By Proposition 5.2.7, $\forall \vec{x}.s \equiv t = 1$ is closed. Hence, by Proposition 5.1.4, $\text{MT} \vdash \forall \vec{x}.s \equiv t = 1$. The claim follows by our assumption and MP. □

The criterion expresses the principle of universal instantiation, which is well known from predicate logic. If a predicate is true universally, then also every particular application of this predicate will evaluate to truth. Therefore in the following, the criterion will often be referred to by the abbreviation ∀I. More precisely, the definition of ∀I together with a few additional theorems is given in Figure 5.4.

$$\begin{array}{ll} \text{MT}, s \doteq_T t = 1 \vdash s = t & \text{Eq}_T \\ x \doteq_T y \to fx \to fy = 1 & \text{Rep}_T \\ \bigvee_{v \in \mathcal{T}_2 T} x \doteq_T (\uparrow v) = 1 & \text{Enum}_T \\ \forall_T f \to fx = 1 & \forall\text{I}_T \end{array}$$

Figure 5.4: MT: Derived Rules and Theorems

The derived rule of inference Eq states that equality of two terms s and t according to the internal relation defined by BExt can be deductively turned into external equality of s and t. In the formulation of Eq, s and t are assumed to be type-licensed by $\Sigma_{\mathcal{T}_2}$; they may contain parameters which are not in $\Sigma_{\mathcal{T}_2}$. Rep states that if two terms are related by internal equality, you can substitute one by the other in every context that does not change the bindings of the occurring variables. Finally, Enum states that in $\mathcal{T}_2$ every value can be expressed by some closed term.

To prove completeness of MT it now suffices to show that ∀I is indeed a theorem of MT. Since the other theorems from Figure 5.4 turn out to be useful for the proof of ∀I, they will be proven simultaneously.

5.2.4 Proof of ∀I

By a simple inspection of the definition of reification at type B one can see that reification agrees with the interpretation of 0 and 1 by $\mathcal{T}_2$. That is, the values $\mathcal{T}_2 0$ and $\mathcal{T}_2 1$ are reified to the initial constants 0 and 1, respectively:

Proposition 5.2.9 (Rei$_\mathsf{B}$) *If* $c \in \{0, 1\}$, *then* $\uparrow_\mathsf{B}(\mathcal{T}_2 c) = c$.

Next we show that reflexivity of internal equality is deductively entailed by MT:

Proposition 5.2.10 $\text{MT} \vdash t \doteq_T t = 1$

Proof By induction on T. □

The following proposition states two important properties of reification. First, reification is a congruence, in the sense that the syntactic application of two reified values v and w of appropriate types is deductively equivalent to the reification of the value vw. Second, every pair of distinct values is reified to terms which are provably non-convertible (their convertibility would contradict Proposition 5.2.10).

Proposition 5.2.11

1. If $T = T_1T_2$*, then* $\mathrm{MT} \vdash (\uparrow_T v)(\uparrow_{T_1} w) = \uparrow_{T_2}.vw$*.*

2. If $v, w \in \mathcal{T}_2 T$ *are distinct, then* $\mathrm{MT} \vdash (\uparrow_T v) \doteq_T (\uparrow_T w) = 0$*.*

Proof By mutual induction on T.

Case $T = \mathsf{B}$. The first claim holds vacuously. For the second claim it suffices to check that both $0 \equiv 1 = 0$ and $1 \equiv 0 = 0$ are tautologous.

Case $T = T_1T_2$ where $T_2 = \vec{T}\mathsf{B}$.

1. Then

$$\begin{aligned}(\uparrow_{T_1\vec{T}\mathsf{B}} v)(\uparrow_{T_1} w) &= (\lambda x\vec{x}. \bigvee_{\substack{u\vec{u}\in\mathcal{T}_2(T_1\vec{T})\\ vu\vec{u}=_{\mathcal{T}_2}1}} x \doteq_{T_1} (\uparrow_{T_1} u) \wedge \vec{x} \doteq_{\vec{T}} (\uparrow_{\vec{T}} \vec{u}))(\uparrow_{T_1} w) \\ &= \lambda\vec{x}. \bigvee_{\substack{u\vec{u}\in\mathcal{T}_2(T_1\vec{T})\\ vu\vec{u}=_{\mathcal{T}_2}1}} (\uparrow_{T_1} w) \doteq_{T_1} (\uparrow_{T_1} u) \wedge \vec{x} \doteq_{\vec{T}} (\uparrow_{\vec{T}} \vec{u}) \qquad \beta\end{aligned}$$

By Proposition 5.2.10 and the second inductive hypothesis respectively, we know:

- $\mathrm{MT} \vdash (\uparrow_{T_1} w) \doteq_{T_1} (\uparrow_{T_1} w) = 1$
- $\mathrm{MT} \vdash (\uparrow_{T_1} w) \doteq_{T_1} (\uparrow_{T_1} u) = 0$ if $u \neq w$

And therefore

$$\begin{aligned}&\lambda\vec{x}. \bigvee_{\substack{u\vec{u}\in\mathcal{T}_2(T_1\vec{T})\\ vu\vec{u}=_{\mathcal{T}_2}1}} (\uparrow_{T_1} w) \doteq_{T_1} (\uparrow_{T_1} u) \wedge \vec{x} \doteq_{\vec{T}} (\uparrow_{\vec{T}} \vec{u}) \\ &= \lambda\vec{x}. \bigvee_{\substack{\vec{w}\in\mathcal{T}_2\vec{T}\\ vw\vec{w}=_{\mathcal{T}_2}1}} \vec{x} \doteq_{\vec{T}} (\uparrow_{\vec{T}} \vec{w}) \\ &= \uparrow_{\vec{T}\mathsf{B}}.vw\end{aligned}$$

2. Let $v, w \in \mathcal{T}_2 T$ be distinct. Then there exist a value $u \in \mathcal{T}_2 T_1$ such that vu, wu are distinct. Thus

$$\begin{aligned}\mathrm{MT} &\vdash \mathrm{MT}, (\uparrow.v\vec{u}) \doteq_{T_2} (\uparrow.w\vec{u}) = 0 && \mathrm{IH}(2) \\ &\vdash \mathrm{MT}, (\uparrow v)(\uparrow\vec{u}) \doteq_{T_2} (\uparrow w)(\uparrow\vec{u}) = 0 && (1) \\ &\vdash \forall_{T_1}(\lambda x.(\uparrow v)x \doteq_{T_2} (\uparrow w)x) = 0 && \\ &\dashv\vdash (\uparrow v) \doteq_T (\uparrow w) = 0 && \mathrm{BExt}\end{aligned}$$

□

Now we are ready to prove the first of our key propositions on the way to ∀I, which is Enum:

Proposition 5.2.12 (Enum$_T$) $\mathrm{MT} \vdash \bigvee_{v\in\mathcal{T}_2 T} x \doteq (\uparrow v) = 1$

Proof By induction on T.

Case $T = \mathsf{B}$. The claim follows by BCA and Proposition 5.2.10.

Case $T = T_1T_2$.

$$\begin{array}{rcll} \bigvee\limits_{v\in\mathcal{T}_2T} x \doteq (\uparrow v) & = & \bigvee\limits_{v\in\mathcal{T}_2T} \forall_{T_1} y.xy \doteq (\uparrow v)y & \text{BExt} \\ & = & \bigvee\limits_{v\in\mathcal{T}_2T} \bigwedge\limits_{w\in\mathcal{T}_2T_1} x(\uparrow w) \doteq (\uparrow v)(\uparrow w) & \forall\text{Enum} \\ & = & \bigvee\limits_{v\in\mathcal{T}_2T_1\to\mathcal{T}_2T_2} \bigwedge\limits_{w\in\mathcal{T}_2T_1} x(\uparrow w) \doteq (\uparrow .vw) & \text{Prop. 5.2.11(1)} \\ & = & \bigwedge\limits_{w\in\mathcal{T}_2T_1} \bigvee\limits_{u\in\mathcal{T}_2T_2} x(\uparrow w) \doteq (\uparrow u) & \text{Prop. 5.2.5} \\ & = & \bigwedge\limits_{w\in\mathcal{T}_2T_1} 1 & \text{IH} \\ & = & 1 & \end{array}$$

□

For the following, it is convenient to re-state Rep in a slightly different form:

Proposition 5.2.13 $\mathrm{MT}, \mathrm{Rep}_T \vdash x \doteq_T y \wedge fx = x \doteq_T y \wedge fy$

Proof It suffices to convert both sides of the equation to $x \doteq_T y \wedge fx \wedge fy$, which can be done by using MT, or equivalently BA, and Rep_T. □

Finally we come to the proof of the remaining theorems from Figure 5.4, including ∀I. This concludes our completeness proof for PTT axiomatized by MT.

Proposition 5.2.14

1. Eq_T

2. $\mathrm{MT} \vdash \mathrm{Rep}_T$

3. $\mathrm{MT} \vdash \forall\mathrm{I}_T$

Proof We proceed in three steps:

1. We show (1) $\Longrightarrow$ (2). Assume (1). By the deduction theorem and stability of deduction under substitution it suffices to show, for fresh parameters $a, b : \mathsf{B}$ and $c : \mathsf{BB}$: $\mathrm{MT}, a \doteq b = 1, ca = 1 \vdash cb = 1$. By (1) we have $\mathrm{MT}, a \doteq b = 1, ca = 1 \vdash a = b$. Therefore $\mathrm{MT}, a \doteq b = 1, ca = 1 \vdash cb = ca = 1$.
2. We show (2) $\Longrightarrow$ (3). Assume (2) and let $v \in \mathcal{T}_2T$. Then:

$$\begin{array}{rcll} x \doteq (\uparrow v) & = & x \doteq (\uparrow v) \wedge ((\bigwedge\limits_{w\in\mathcal{T}_2T} f(\uparrow w)) \to f(\uparrow v)) & \\ & = & x \doteq (\uparrow v) \wedge (\forall f \to f(\uparrow v)) & \forall\text{Enum} \\ & = & x \doteq (\uparrow v) \wedge (\forall f \to fx) & \text{Prop. 5.2.13} \end{array}$$

By the above we then have

$$\begin{array}{rcll}
\forall f \to fx & = & (\bigvee_{v \in \mathcal{T}_2 T} x \doteq (\uparrow v)) \wedge (\forall f \to fx) & \text{Prop. 5.2.12} \\
& = & \bigvee_{v \in \mathcal{T}_2 T} x \doteq (\uparrow v) \wedge (\forall f \to fx) & \\
& = & \bigvee_{v \in \mathcal{T}_2 T} x \doteq (\uparrow v) & \\
& = & 1 & \text{Prop. 5.2.12}
\end{array}$$

3. We show (1) by induction on T. By the above, the inductive hypothesis can always be weakened to an appropriate instance of (2) or (3).

   Case $T = \mathsf{B}$. By BExt, $\mathrm{MT} \vdash s \doteq_{\mathsf{B}} t = s \equiv t$. Hence, the claim follows by Proposition 5.2.4.

   Case $T = T_1 T_2$.

$$\begin{array}{rcll}
\mathrm{MT}, s \doteq_T t = 1 & \dashv\vdash & \mathrm{MT}, \forall_{T_1}(\lambda x.sx \doteq_{T_2} tx) = 1 & \text{BExt} \\
& \vdash & \mathrm{MT}, sx \doteq_{T_2} tx = 1 & \text{IH(3), MP} \\
& \vdash & sx = tx & \text{IH(1)} \\
& \dashv\vdash & s = t & \text{Cor. 4.2.3}
\end{array}$$

□

5.3 Decidability of PTT

Since $\mathcal{T}_2$ is the only model of PTT and all domains of $\mathcal{T}_2$ are finite, it is obvious that validity of equations in PTT is decidable by means of an exhaustive case analysis on the level of semantics. What we are going to show now is that a decision procedure for PTT can also be based entirely on equational deduction from MT.

Our main idea here is convert an arbitrary equation e into a closed basic formula which is convertible to 1 if e is provable, and to 0 otherwise.

Proposition 5.3.1 *Let s and t be terms, and let $\vec{x}$ contain $\mathcal{V}s \cup \mathcal{V}t$. Then:* $\mathcal{T}_2 \models s = t \iff \mathrm{MT} \vdash \forall \vec{x}.s \doteq t = 1.$

Proof Let s, t and $\vec{x}$ be as required. Then:

$$\begin{array}{rll}
& \mathcal{T}_2 \models s = t & \\
\iff & \mathcal{T}_2 \models \forall \vec{x}.s \doteq t = 1 & \text{Prop. 5.2.6} \\
\iff & \mathrm{MT} \vdash \forall \vec{x}.s \doteq t = 1 & \text{Prop. 5.1.1, 5.2.7, 5.1.4}
\end{array}$$

□

Note that the above proof does not depend on the general completeness result, but only on completeness for closed basic equations.

Proposition 5.3.2 *Let s and t be terms, and let $\vec{x}$ contain $\mathcal{V}s \cup \mathcal{V}t$. Then:* $\mathcal{T}_2 \not\models s = t \iff \mathrm{MT} \vdash \forall \vec{x}.s \doteq t = 0.$

Proof

- "⇐": By soundness (Proposition 5.1.1 and 5.2.6).
- "⇒": Let s, t and $\vec{x}$ be as required. Assume, for contradiction, $\mathcal{T}_2 \not\models s = t$ and MT $\not\vdash \forall\vec{x}.s \doteq t = 0$. Then:

$$\begin{array}{lll} & \mathrm{MT} \vdash \forall\vec{x}.s \doteq t = 1 & \text{Prop. 5.2.7, 5.1.3} \\ \Longleftrightarrow & \mathcal{T}_2 \models s = t & \text{Prop. 5.3.1} \\ \Longrightarrow & \text{contradiction} & \end{array}$$

□

Now a decision procedure for PTT can be defined as follows. Given an equation $s = t$, it enumerates proofs from MT until it encounters either a proof of $\forall\vec{x}.s \doteq t = 0$ or a proof of $\forall\vec{x}.s \doteq t = 1$. Since by the above, exactly one of the statements is provable, the procedure terminates returning the appropriate result.

Complexity of PTT has been a subject of study for several decades and, due to results by Meyer [32] and Vorobyov [50], we know that the theory is not efficiently decidable. In fact, PTT is the most nonelementary theory currently known [50].

5.4 Definitional Extensions

When working with MT we often have to deal with huge terms, which are only manageable by using notational abbreviations. Sometimes, instead of using derived notation, it is more convenient to abbreviate large terms by defining new constants and introducing axioms capturing their intended deductive and semantic properties. For instance, let us have a look at a system MT′, which differs from MT in the fact that $\doteq_T$, $\forall_T$ and $\uparrow_T v$ (for every $v \in \mathcal{T}_2 T$) are not considered derived notation, but parameters whose semantics is defined by the equations in Figure 5.3 and MT. Of course, we would like to show that such a system is still deductively complete. To do so, we introduce an approach which allows us to prove completeness of systems like MT′ by reduction to the respective theories without the additional constants. A key role in our approach plays the concept of a definitional extension. Intuitively, a set of axioms B is called a definitional extension of another set A if B differs from A only in the fact that it defines additional constants that denote values already expressible in A.

Given a set A of equations, the **signature** of A (notation Σ_A) is the smallest signature which contains all the parameters occurring in A.

A set of equations B is a **definitional extension** of a set A if $\Sigma_A \cap \mathit{Sor} = \Sigma_B \cap \mathit{Sor}$ and $B - A$ is deductively equivalent to some set D such that all equations $e \in D$ are of the form $c = t$ where

1. $c \notin \Sigma_A$,
2. $c \notin \mathcal{P}(D - \{c = t\})$,
3. $t \in \mathit{Ter}(\Sigma_A)$ is closed.

The set D is called the **definitional part** of B.

Given A, B and D as above, we define $\downarrow_{\mathrm{def}}$ as the following substitution:

$$\downarrow_{\mathrm{def}} = \lambda u \in \mathit{Nam}.\text{if } u \in \operatorname{dom} D \text{ then } Du \text{ else } u$$

It is important to convince ourselves that applications of $\downarrow_{\mathrm{def}}$ to terms yield deductively equivalent terms: $B \vdash t = \downarrow_{\mathrm{def}} t$. This can be seen as a consequence of the following more general theorem:

Proposition 5.4.1 *If, for all $u \in \mathcal{N}t$, $A \vdash u = \theta u$, then $A \vdash t = \theta t$.*

Proof By induction on $|t|$. □

Proposition 5.4.2 *Let B be a definitional extension of A. Then:*

1. *Every model $\mathcal{A}$ of A can be extended to a model of B.*
2. *Every model $\mathcal{B}$ of B is a model of A.*

Proof The second claim is obvious, so we only show the first claim. Let $\mathcal{A}$ be a model of A, and $\mathcal{I} \supseteq \mathcal{A}$ an interpretation. We have to construct a model $\mathcal{B}$ of B which extends $\mathcal{A}$ to parameters $c \in \Sigma_B - \Sigma_A$. Consider the deductively and hence semantically equivalent re-formulation of B as $A \cup D$ where D is the definitional part of B. For every equation $(c = t) \in D$, define $\mathcal{B}c = \hat{\mathcal{I}}t$. Note that since t is closed, $\hat{\mathcal{I}}t$ does not depend on the choice of $\mathcal{I}$, so $\mathcal{B}$ is uniquely determined by $\mathcal{A}$ and D. Since every interpretation $\mathcal{I}' \supseteq \mathcal{B}$ also extends $\mathcal{A}$, by coherence $\hat{\mathcal{I}}'c = \mathcal{B}c = \hat{\mathcal{I}}t = \hat{\mathcal{I}}'t$, i.e. $\mathcal{B}$ is a model of D and hence also a model of B. □

Proposition 5.4.3 *Let B be a definitional extension of A, and let e be licensed by Σ_A. Then: $A \vDash e \iff B \vDash e$.*

Proof We show "$\Leftarrow$"; the reverse direction is proven analogously but simpler. Let B be a definitional extension of A. It suffices to show that for every model $\mathcal{A}$ and every interpretation $\mathcal{I} \supseteq \mathcal{A}$ there exists a model $\mathcal{B}$ and an interpretation $\mathcal{I}' \supseteq \mathcal{B}$ such that for every term $t \in \mathit{Ter}(\Sigma_A)$: $\hat{\mathcal{I}}t = \hat{\mathcal{I}}'t$. So, let $\mathcal{A}$ be a model of A and $\mathcal{I} \supseteq \mathcal{A}$ an interpretation. By Proposition 5.4.2, we can extend $\mathcal{A}$ to a model $\mathcal{B}$ of B. Let $\mathcal{I}'' \supseteq \mathcal{B}$ arbitrary, and let $\mathcal{I}' = \lambda u \in \mathit{Nam}.$if u licensed by Σ_A then $\mathcal{I}u$ else $\mathcal{I}''u$. Clearly, $\mathcal{I}'$ is an extension of $\mathcal{B}$. By coherence, $\hat{\mathcal{I}}t = \hat{\mathcal{I}}'t$ holds for all terms t licensed by Σ_A. □

Lemma 5.4.4 *For every context k, term t and substitution θ such that $\operatorname{dom}\theta \subseteq \mathit{Par}$ and $\bigcup_{c \in \mathit{Par}} \mathcal{V}(\theta c) = \varnothing$ (independently of Σ): $\theta(k[t]) = (\theta k)[\theta t]$.*

Proof Let k, t and θ be as required. We proceed by induction on $|k|$.

Case $k = \bullet$. Then: $\theta(\bullet[t]) = \theta t = \bullet[\theta t] = (\theta\bullet)[\theta t]$.

Case $k = k's$. By the inductive hypothesis:

$\theta((k's)[t]) = \theta(k'[t]s) = (\theta(k'[t]))(\theta s) = (\theta k')[\theta t](\theta s) = (\theta(k's))[\theta t]$.

Case $k = sk'$ proceeds analogously to the preceding case.

Case $k = \lambda x.k'$. Then:

$$\begin{array}{rcll} \theta((\lambda x.k')[t]) & = & \theta(\lambda x.k'[t]) & \\ & = & \lambda x.\theta(k'[t]) & \\ & = & \lambda x.(\theta k')[\theta t] & \text{IH} \\ & = & (\lambda x.\theta k')[\theta t] & \\ & = & (\theta(\lambda x.k'))[\theta t] & \end{array}$$

□

Theorem 5 *Let B be a definitional extension of A. Then A is deductively complete if and only if B is deductively complete.*

Proof Let B be a definitional extension of A.

- "⇒": Assume A is complete and let $B \vDash s = t$, for some $s, t \in \mathit{Ter}(\Sigma_B)$. Then:

$$\begin{array}{rll} & B \vDash \downarrow_{\text{def}} s = \downarrow_{\text{def}} t & \text{Prop. 5.4.1, soundness (Prop. 2.1.4)} \\ \Longleftrightarrow & A \vDash \downarrow_{\text{def}} s = \downarrow_{\text{def}} t & \text{Prop. 5.4.3} \\ \Longrightarrow & A \vdash \downarrow_{\text{def}} s = \downarrow_{\text{def}} t & \text{assumption (completeness of } A\text{)} \\ \Longrightarrow & B \vdash \downarrow_{\text{def}} s = \downarrow_{\text{def}} t & \\ \Longleftrightarrow & B \vdash s = t & \text{Prop. 5.4.1} \end{array}$$

- "⇐": Assume B is complete and let $A \vDash s = t$, for some $s, t \in \mathit{Ter}(\Sigma_A)$. By Proposition 5.4.3, $B \vDash s = t$. By completeness of B, $B \vdash s = t$. Consider the corresponding conversion proof $t_0 = \ldots = t_n$ ($n \in \mathbb{N}$, $t_0 = s$, $t_n = t$) from the axioms $A \cup D$ where D is the definitional part of B. Since $\downarrow_{\text{def}} s = s$ and $\downarrow_{\text{def}} t = t$, it suffices to show that $\downarrow_{\text{def}} t_0 = \downarrow_{\text{def}} t_1 = \ldots = \downarrow_{\text{def}} t_n$ is a conversion proof from A, i.e. for all $i \in \{1, \ldots, n\}$: $A \vdash \downarrow_{\text{def}} t_{i-1} = \downarrow_{\text{def}} t_i$. Let, for some k, s', t' and θ with $\operatorname{dom}\theta \subseteq \mathit{Var}$, $t_{i-1} = k[\theta s']$ and $t_i = k[\theta t']$.

  Case $(s' = t') \in D$. Then $\downarrow_{\text{def}} t_{i-1} = \downarrow_{\text{def}} t_i$, and we are done.

  Case $(s' = t') \in A$ or $(t' = s') \in A$. By Lemma 5.4.4, $\downarrow_{\text{def}} t_{i-1} = (\downarrow_{\text{def}} k)[\downarrow_{\text{def}}(\theta s')]$ and $\downarrow_{\text{def}} t_i = (\downarrow_{\text{def}} k)[\downarrow_{\text{def}}(\theta t')]$. Since substitutions are closed under composition, $A \vdash \downarrow_{\text{def}} t_{i-1} = \downarrow_{\text{def}} t_i$.

  Case $s' = t'$ is an instance of β or η and θ is the identity substitution. We proceed analogously to the preceding case. □

It easily can be verified that MT′ is a definitional extension of MT. Therefore:

Corollary 5.4.5 MT′ *is deductively complete.*

5.5 Alternative Axiomatizations

Now that we have proven MT deductively complete, we want to consider two more possibilities to axiomatize PTT and show their completeness by reduction to MT.

Name	**PT**				
Extends	**MT**				
Constant	$\stackrel{\circ}{=}_T : TT\mathsf{B}$	internal equality			
Variables	$x, y : T$ $f : T\mathsf{B}$				
Axioms	$x \stackrel{\circ}{=} x$	$=$	1	Ref	
	$x \stackrel{\circ}{=} y \to fx \to fy$	$=$	1	Rep	

Figure 5.5: PT as Extension of MT

5.5.1 PT

First we want to consider an axiomatization of PTT which is relatively close to Henkin's original formulation. Just like Henkin, we introduce an internal equality constant for every type and specify its semantics by appropriate axioms, given in Figure 5.5. Unlike Henkin, we do not consider internal equality as the only primitive constant. Instead PT is assumed to extend MT.

Soundness of PT with respect to $\mathcal{T}_2$ is obvious and easily provable. Now to the reverse direction. It is not hard to show that the internal equality relation as defined by PT is symmetric:

Proposition 5.5.1 $\text{PT} \vdash x \stackrel{\circ}{=} y = y \stackrel{\circ}{=} x$

Proof By Proposition 5.2.4, stability of deduction under substitution and the deduction theorem, it suffices to show that, for every type T and fresh parameters $a, b : T$, $\text{PT}, a \stackrel{\circ}{=} b = 1 \vdash (\lambda x. x \stackrel{\circ}{=} a)b = 1$, which follows by Rep, Ref and MP. □

The following proposition is more interesting. In conjunction with Ref, it states that internal equality is deductively equivalent to external equality:

Proposition 5.5.2 *For all terms s and t, not necessarily licensed by Σ_{PT}:*
$\text{PT}, s \stackrel{\circ}{=} t = 1 \vdash s = t$.

Proof Let $a, b : \vec{T}\mathsf{B}$ be fresh constants. By Corollary 4.2.3 and Proposition 5.2.4 it suffices to show:

1. $\text{PT}, s \stackrel{\circ}{=} t = 1 \vdash s\vec{x} \to t\vec{x} = 1$
2. $\text{PT}, s \stackrel{\circ}{=} t = 1 \vdash t\vec{x} \to s\vec{x} = 1$

The first subclaim follows by Rep and MP. The second one additionally needs Proposition 5.5.1. □

PT differs from MT only in the fact that it defines one additional constant, namely $\stackrel{\circ}{=}$. Thus, an appealing way of proving completeness of PT seems to be showing it a definitional extension of MT. In order to do so, we define the following set of axioms: $\mathrm{MT}_{\stackrel{\circ}{=}} = \mathrm{MT} \cup \{(\stackrel{\circ}{=}_T) = (\doteq_T)\}_{T \in Ty(\Sigma_{\mathrm{MT}})}$. Unlike PT, $\mathrm{MT}_{\stackrel{\circ}{=}}$ is obviously a definitional extension of MT. Hence, it now suffices to show: $\mathrm{PT} \dashv\vdash \mathrm{MT}_{\stackrel{\circ}{=}}$. By completeness of MT, we immediately get that both Ref and Rep are theorems of $\mathrm{MT}_{\stackrel{\circ}{=}}$, and therefore: $\mathrm{MT}_{\stackrel{\circ}{=}} \vdash \mathrm{PT}$. So, to prove the deductive equivalence it suffices to show that $\mathrm{PT} \vdash (\stackrel{\circ}{=}_T) = (\doteq_T)$, for all types T.

For the base type, the proof is easy:

Proposition 5.5.3 $\mathrm{PT} \vdash (\stackrel{\circ}{=}_{\mathsf{B}}) = (\doteq_{\mathsf{B}})$

Proof By Corollary 4.2.3, Proposition 5.2.4, stability of deduction under substitution and the deduction theorem, it suffices to show that, for fresh parameters $a, b : \mathsf{B}$: $\mathrm{PT}, a \stackrel{\circ}{=}_{\mathsf{B}} b \dashv\vdash \mathrm{PT}, a \doteq_{\mathsf{B}} b$. By $\mathrm{BExt}_{\mathsf{B}}$ and Proposition 5.2.4 this is equivalent to showing $\mathrm{PT}, a \stackrel{\circ}{=}_{\mathsf{B}} b \dashv\vdash \mathrm{PT}, a = b$, which follows by Ref and Proposition 5.5.2. □

Before we proceed to functional types, it is helpful to prove that $\stackrel{\circ}{=}$ is extensional:

Proposition 5.5.4 $\mathrm{PT} \vdash \forall_{T_1} x.fx \stackrel{\circ}{=}_{T_2} gx = f \stackrel{\circ}{=}_{T_1 T_2} g$

Proof By Proposition 5.2.4 it suffices to show: $\mathrm{PT} \vdash f \stackrel{\circ}{=} g \equiv \forall x.fx \stackrel{\circ}{=} gx = 1$. This can be done in two steps:

- "→": By stability of deduction under substitution and the deduction theorem, it is sufficient to show, for fresh parameters $a, b : T_1 T_2$: $\mathrm{PT}, \forall_{T_1} x.ax \stackrel{\circ}{=}_{T_2} bx \vdash a \stackrel{\circ}{=}_{T_1 T_2} b$. This claim is an easy consequence of Proposition 5.2.14(3), MP and Corollary 4.2.3.
- "←": Follows similarly to the first step by stability of deduction under substitution, the deduction theorem, Proposition 5.5.2 and Ref. □

Proposition 5.5.5 $\mathrm{PT} \vdash (\stackrel{\circ}{=}_T) = (\doteq_T)$

Proof By induction on T.

Case $T = \mathsf{B}$ follows by Proposition 5.5.3.

Case $T = T_1 T_2$. We show:

$$\begin{array}{rcll} f \stackrel{\circ}{=}_{T_1 T_2} g & = & \forall_{T_1} x.fx \stackrel{\circ}{=}_{T_2} gx & \text{Prop. 5.5.4} \\ & = & \forall_{T_1} x.fx \doteq_{T_2} gx & \text{IH} \\ & = & f \doteq_{T_1 T_2} g & \text{BExt} \end{array}$$

The claim follows by Corollary 4.2.3. □

By Theorem 5 we obtain:

Corollary 5.5.6 PT *is deductively complete.*

Name	**QT**			
Variables	$x : \mathsf{B}$ $y : T$ $f : \mathsf{BB}$			
Rules	$\dfrac{s \mathrel{\dot{=}} t = 1}{s = t}$			Eq
Axioms	$0 \to x$	$=$	1	I0
	$1 \to x$	$=$	x	I1
	$f0 \to f1 \to fx$	$=$	1	BCA
	$y \mathrel{\dot{=}} y$	$=$	1	Ref

Figure 5.6: QT: Equational Type Theory (Constants as in PT)

5.5.2 QT

In our last formulation of PTT we treat the internal equality constant $\mathrel{\dot{=}}$ in a special way by embedding it into a rule of inference. Compared to PT, this allows us to omit two axioms: Rep and Comm.

As usual, we can easily check that QT is sound with respect to $\mathcal{T}_2$. Alternatively, soundness of QT can be reduced to soundness of PT by showing: PT $\vdash$ QT. Since $\Sigma_{\mathrm{QT}} = \Sigma_{\mathrm{PT}}$, to prove completeness of QT it suffices to show the reverse direction: QT $\vdash$ PT. So, let us this time prove both soundness and completeness of QT by showing: QT $\dashv\vdash$ PT. More precisely, we are going to show that:

1. The axioms Rep and Comm of PT are derivable in QT.
2. The rule of inference Eq of QT is derivable in PT.

Proposition 5.5.7 (Replacement) QT $\vdash x \mathrel{\dot{=}} y \to fx \to fy = 1$

Proof Proceeds analogously to the first step of the proof of Proposition 5.2.14. □

Proposition 5.5.8 (Commutativity) QT $\vdash (x \to y) \to y = (y \to x) \to x$

Proof By repeated application of I0, I1 and Ref we obtain

$$\mathrm{QT} \vdash ((a \to b) \to b) \mathrel{\dot{=}} ((b \to a) \to a) = 1$$

for every combination of $a, b \in \{0, 1\}$. By BCA and MP we conclude:

$$\mathrm{QT} \vdash ((x \to y) \to y) \mathrel{\dot{=}} ((y \to x) \to x) = 1$$

The claim follows by Eq. □

Luckily, the derivability of Eq from PT was already shown when we discussed PT (see Proposition 5.5.2). Hence we conclude:

Proposition 5.5.9 QT $\dashv\vdash$ PT

Proof The claim follows by Propositions 5.5.2, 5.5.7 and 5.5.8. □

Corollary 5.5.10 QT *is deductively complete.*

5.6 Consequences and Connections with General Semantics

We have discussed several axiomatizations of PTT. Among these, MT plays a special role. In contrast to Henkin's original axiomatization of PTT, which has infinitely many axioms, indexed by pairs of types, MT is finite. It contains only four axioms which describe only three domains, namely those of type B, BB and BBB. We conclude:

Proposition 5.6.1 *PTT has a finite axiomatization in* S *which is deductively complete, i.e. there exists a set* A *of equations such that, for every equation* e*:* $\mathcal{T}_2 \models e \implies A \vdash e$.

Although so far we have only considered standard semantics of PTT, completeness of MT with respect to $\mathcal{T}_2$ has noteworthy consequences for the general semantics of the formalism. The most interesting observation is probably that MT has no non-standard models in Henkin's [24] sense:

Proposition 5.6.2 *Every model of* MT *which has Boolean extensionality is standard at* $Ty(\Sigma_{\mathcal{T}_2})$*, up to isomorphism.*

Proof We are aware of two ways to prove this claim. One possibility to show the claim is by a simple cardinality argument, exploiting functional extensionality of S, Proposition 5.2.6(1) and 5.2.11(2). The main idea is to show that functional domains corresponding to types licensed by $Ty(\Sigma_{\mathcal{T}_2})$ must be isomorphic to standard domains in every general model of MT because every two distinct elements of such a standard domain can be represented by two closed, non-convertible terms.

Another possibility is to consider general models of PTT as particular models of full HOL in which the Boolean and the individual domains coincide. Since, by our cardinality assumption for the Boolean domain, every such model is finite, we can adapt the proof by Andrews [5], who shows that every finite general model of HOL which has Boolean extensionality is standard. The fact that Andrews relies on the description operator is not a problem, since our system allows us to define the operator as a closed term (Henkin's [25] original formulation of PTT contains a suitable definition of the description operator).

When we began our discussion of PTT we saw that, up to isomorphism, $\mathcal{T}_2$ is the unique non-trivial standard model of MT (over $\Sigma_{\mathcal{T}_2}$; see Proposition 5.1.2). Now we can strengthen this statement:

Corollary 5.6.3 *Up to isomorphism, $\mathcal{T}_2$ is the unique general model of* MT *over* $\Sigma_{\mathcal{T}_2}$ *which has Boolean extensionality.*

We have not investigated whether MT has non-standard models which interpret the Boolean type by larger sets (but see Section 5.7).

If one compares PT to Henkin's axiomatization of PTT, one may notice that Henkin's axioms contain a distinguished axiom schema describing functional extensionality of the internal equality relation, while PT does not. In fact, it is not hard to derive from PT various formulations of extensionality, in particular the one used by Henkin. The semantic reason for this is that in $\mathcal{T}_2$, the axiom of replacement entails extensionality of $\mathrel{\mathring{=}}$. It states that if two values $v, v' \in \mathcal{T}_2(\vec{T}\mathsf{B})$ are in the relation denoted by $\mathrel{\mathring{=}}$, then every function $w \in \mathcal{T}_2((\vec{T}\mathsf{B})\mathsf{B})$ yields the same value when applied to v and to v'. Since $\mathcal{T}_2$ is standard, possible values for w include the functions $\lambda v \in \mathcal{T}_2(\vec{T}\mathsf{B}).v\vec{w}$ for all $\vec{w} \in \mathcal{T}_2\vec{T}$. Thus v and v' are in the internal equality relation only if they are extensionally equal. So, as a consequence of Corollary 5.6.3 and Henkin's [24] completeness theorem, extensionality of $\mathrel{\mathring{=}}$ has to be derivable from PT.

Since for fragments of HOL over richer signatures than that of PTT general and standard models no longer coincide, in these more general cases it is no longer possible to axiomatize equality without axioms of extensionality. This can be seen as a consequence of a result by Andrews [3], who observed that internal equality in general models which have both Boolean and functional extensionality nevertheless does not necessarily correspond to identity.

The argument presented in Section 5.2 proves deductive completeness of MT, $\mathcal{T}_2 \models e \implies \mathrm{MT} \vdash e$, only for equations e which are licensed by $\Sigma_{\mathcal{T}_2}$. Otherwise, Proposition 5.2.8 is no longer applicable. However using Proposition 5.6.2, the completeness result can be strengthened:

Theorem 6 *If e is type-licensed by $\Sigma_{\mathcal{T}_2}$, then: $\mathcal{T}_2 \models e \implies \mathrm{MT} \vdash e$.*

Proof Let e be type-licensed by $\Sigma_{\mathcal{T}_2}$ and $\mathcal{T}_2 \models e$. By Proposition 5.6.2 all general models of MT which have Boolean extensionality agree with $\mathcal{T}_2$ on the interpretation of terms and types licensed by $\Sigma_{\mathcal{T}_2}$. Since e only involves values from $\mathcal{T}_2$, we have: $\mathcal{T}_2 \models e \iff$ for all general models $\mathcal{H} \models \mathrm{MT}$ with Boolean extensionality: $\mathcal{H} \models e$. By Henkin's [24] completeness result, e is provable in the full HOL. Since e also holds in models $\mathcal{H}$ in which the Boolean and the individual domains coincide, e has a proof which does not contain any terms that are not type-licensed by $\Sigma_{\mathcal{T}_2}$. Now it is easily provable that every instance of Henkin's [24] axioms or rules of inference for HOL which is licensed by $\Sigma_{\mathcal{T}_2}$ is can be derived in S from MT, which completes the argument. $\square$

Corollary 5.6.4 (Standard Soundness and Completeness) *If e is type-licensed by $\Sigma_{\mathcal{T}_2}$, then:* $\mathcal{A} \models e$ for all standard models $\mathcal{A}$ of MT $\iff$ MT $\vdash e$.

Corollary 5.6.5 (General Soundness and Completeness) *If e is type-licensed by $\Sigma_{\mathcal{T}_2}$, then:* $\mathcal{H} \models e$ for all general models $\mathcal{H}$ of MT $\iff$ MT $\vdash e$.

An interesting consequence of Corollary 5.6.3 is standard (strong) compactness of PTT, which can be formulated as follows:

Proposition 5.6.6 (Compactness) *An extension A of* MT *type-licensed by $\Sigma_{\mathcal{T}_2}$ is satisfiable at* B *if and only if every finite subset of A is satisfiable at* B.

Proof Let A be an extension of MT type-licensed by $\Sigma_{\mathcal{T}_2}$, and let every finite subset of A be satisfiable at B. Then, in particular, every finite subset of A which extends MT is satisfiable at B. Let B be an arbitrary subset of A extending MT, and let $\mathcal{B}$ be a model of B. By Corollary 5.6.3 we know that $\mathcal{B}$ is, up to isomorphism, an extension of $\mathcal{T}_2$. Therefore, the equation $0 = 1$ is not valid in $\mathcal{B}$ and hence, by soundness, is not provable from B. Since every proof from A is finite, i.e. can be seen as a proof from a finite subset B of A extending MT, by the above reasoning $0 = 1$ is not provable from A. Hence, Henkin's [24] model existence theorem guarantees that A has a general model $\mathcal{H}'$ which has Boolean extensionality. By Corollary 5.6.3 we can construct a standard extension $\mathcal{A}$ of $\mathcal{T}_2$ which agrees with $\mathcal{H}'$ on all parameters that are type-licensed by $\Sigma_{\mathcal{T}_2}$. By denotational coherence $\mathcal{A}$ is a model of A. The converse is immediate. $\square$

The above compactness property is called strong because it makes a statement about satisfiability with respect to standard models. An analogous statement with respect to general models is called weak compactness. Full HOL has been shown weakly compact by Henkin [24], but at the same time is known to violate strong compactness (see Andrews [5] for a proof).

5.7 Further Work

An essential result obtained in the preceding section is that all standard models of MT have Boolean extensionality (see proof of Proposition 5.1.2). Moreover, we observed that MT has no non-standard models with Boolean extensionality (Proposition 5.6.2). It remains open whether there exist non-standard models of MT that have no Boolean extensionality. Intuitively, functions violating the axiom BCA of MT do not seem representable by λ-terms licensed by Σ_{MT}, which supports the conjecture that such non-standard models indeed exist. A construction based on appropriate logical relation frames (see [3, 11]) is likely to yield models with the required properties, but has not yet been attempted.

Dropping the axiom Comm from MT has no semantic consequences as long as we consider models of PTT with Boolean extensionality. So, constructing and analyzing models of MT and MT $-$ {Comm} that do not have Boolean extensionality seems essential for proving the independence of Comm.

Bibliography

[1] ANDREWS, P. B. A reduction of the axioms for the theory of propositional types. *Fundamenta Mathematicae 52* (1963), 345–350.

[2] ANDREWS, P. B. A transfinite type theory with type variables. *Journal of Symbolic Logic 33*, 1 (March 1965), 112–113.

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

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

[5] 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.

[6] BAADER, F., AND NIPKOW, T. *Term Rewriting and All That*. Cambridge University Press, 1998.

[7] BARENDREGT, H. P. *The Lambda Calculus: Its Syntax and Semantics*, 2nd revised ed., vol. 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1984.

[8] BARENDREGT, H. P. Lambda calculi with types. In *Handbook of Logic in Computer Science*, S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, Eds., vol. 2. Oxford University Press, 1992.

[9] BENZMÜLLER, C., BROWN, C. E., AND KOHLHASE, M. Higher-order semantics and extensionality. *Journal of Symbolic Logic 69*, 4 (December 2004), 1027–1088.

[10] BIRKHOFF, G. On the structure of abstract algebras. *Proceedings of the Cambridge Philosophical Society 31* (1935), 433–454.

[11] BROWN, C. E. *Set Comprehension in Church's Type Theory*. PhD thesis, Department of Mathematical Sciences, Carnegie Mellon University, 2004.

[12] BURRIS, S., AND SANKAPPANAVAR, H. P. *A Course in Universal Algebra*. Graduate Texts in Mathematics. Springer-Verlag, 1981.

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

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

[15] Fitting, M. *First-Order Logic and Automated Theorem Proving*, second ed. Graduate Texts in Computer Science. Springer-Verlag, 1996.

[16] 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.

[17] Gandy, R. O. An early proof of normalization by A. M. Turing. 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. 453–455.

[18] Girard, J.-Y. *Interprétation fonctionelle et élimination des coupures dans l'arithmétique d'ordre supérieur.* Thèse de doctorat d'état, Université de Paris VII, 1972.

[19] Gödel, K. Die vollständigkeit der Axiome des logischen Funktionenekalküls. *Monatshefte für Mathematik und Physik 37* (1930), 349–360. Translated in [49], pp. 582–591.

[20] Gödel, K. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme, I. *Monatshefte für Mathematik und Physik 38* (1931), 173–198. Translated in [49], pp. 596–616.

[21] Gödel, K. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica 12* (1958), 280–287. Translated in [22], pp. 240–252.

[22] Gödel, K. *Collected Works, Volume II, Publications 1938–1974*. Oxford University Press, 1990.

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

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

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

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

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

[28] Hindley, J. R., and Seldin, J. P. *Introduction to Combinators and λ-calculus.* Cambridge University Press, 1986.

[29] Hrbacek, K., and Jech, T. *Introduction to Set Theory*, third ed. Pure and Applied Mathematics. Marcel Dekker, 1999.

[30] Kaminski, M. *Studies in Higher-Order Equational Logic.* Bachelor's thesis, Saarland University, 2005.

[31] Manca, V., and Salibra, A. First-order theories as many-sorted algebras. *Notre Dame Journal of Formal Logic 25*, 1 (1984), 86–94.

[32] Meyer, A. R. The inherent computational complexity of theories of ordered sets. In *Proceedings of the International Congress of Mathematicians* (Vancouver, August 1974), pp. 477–482.

[33] Mitchell, J. C. *Foundations for Programming Languages.* Foundations of Computing. The MIT Press, 1996.

[34] Mycielski, J. Equational treatment of first-order logic. *Algebra Universalis 33*, 1 (1995), 26–39.

[35] Peano, G. Arithmetices principia, nova methodo exposita. Turin, 1889. Translated in [49], pp. 83–97.

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

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

[38] Plotkin, G. D. LCF considered as a programming language. *Theoretical Computer Science 5*, 3 (1977), 223–255.

[39] Plotkin, G. D. Lambda-definability in the full type hierarchy. 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.

[40] Sazonov, V. Expressibility of functions in D. Scott's LCF. *Algebra and Logic 15*, 3 (1976), 192–206.

[41] Scott, D. S. A type-theoretical alternative to ISWIM, CUCH, OWHY. Unpublished manuscript, 1969.

[42] Scott, D. S. A type-theoretical alternative to ISWIM, CUCH, OWHY. *Theoretical Computer Science 121*, 1-2 (1993), 411–440.

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

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

[45] SMOLKA, G. *Introduction to Computational Logic: Lecture Notes*. Universität des Saarlandes, 2006. www.ps.uni-sb.de/courses/cl-ss06/script/.

[46] 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.

[47] TAIT, W. Intensional interpretations of functionals of finite type I. *Journal of Symbolic Logic 32*, 2 (1967), 198–212.

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

[49] 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.

[50] VOROBYOV, S. The most nonelementary theory. *Information and Computation 190*, 2 (May 2004), 196–219.

Notation

Index