INVESTIGATIONS ON SETS AND TYPES

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Wojciech Tomasz Moczydłowski

August 2007

INVESTIGATIONS ON SETS AND TYPES

Wojciech Tomasz Moczydłowski, Ph.D.

Cornell University 2007

There are two major foundational frameworks used in mathematics and computer science — set theory and type theory. The former is widely accepted as the foundation of classical mathematics, the latter is being successfully applied in computer science, for the purpose of program verification, programming languages semantics, software engineering and modeling physical systems. We investigate connections between these worlds. More specifically, we prove a normalization theorem for a constructive impredicative set theory IZF. This result makes it possible to exhibit computational content hidden in set theories. We show how to use normalization to extract programs from IZF proofs. We also demonstrate that a small change to IZF can destroy the normalization property.

Furthermore, we investigate two extensions of our framework. We first extend IZF to incorporate inaccessible sets, providing a framework powerful enough to provide constructive semantics for popular type theories. We demonstrate that the normalization property holds for the extension, thus enabling program extraction from its proofs. Second, we extend the logic of IZF to incorporate features typical of dependent type theories. We show that unless such extension is done very carefully, the theory will become inconsistent. However, we present a consistent, normalizing extension — a "dependent" set theory IZF_D. We show that the proof-theoretic power of IZF_D equals that of Zermelo-Fraenkel set theory with Choice, ZFC, the standard foundation of mathematics.

Finally, we apply our results to a constructive version of Higher-Order Logic

(HOL). Namely, we show how to refine the standard set-theoretic semantics for HOL so that it maps a constructive core of HOL to IZF. Using our normalization result for IZF, we utilize this semantics to provide the program extraction capability from constructive HOL proofs.

BIOGRAPHICAL SKETCH

The author of the thesis was born in Warsaw, Poland, where he also graduated from Warsaw University, majoring in Computer Science and Mathematics. After graduation, he moved to Cornell and spent four wonderful years working on his Ph. D. degree.

ACKNOWLEDGEMENTS

I was very lucky to have a Ph. D. committee whose members were experts in both facets of my research — computer science and mathematics. My advisor, Bob Constable, has taught me a lot about type theory and computer science. Without his will to work with me on topics that I found exciting, without his constant support and many fruitful discussions, this thesis could not have been completed.

I learned an amazing amount of mathematical logic from Richard A. Shore, my minor advisor. His classes and unwavering willingness to talk about my research were a constant source of inspiration. It was always very helpful to talk with Andrew Myers about the programming languages side of my research.

The discussions and comments of the members of the Nuprl group, in particular Stuart Allen, Mark Bickford, Rich Eaton, Christoph Kreitz and Evan Moran, played an important role in my research. My understanding of type theory would be much worse without their presence at Cornell.

I also want to extend my thanks to people whose presence made my life better during my Ph. D. studies: Siggi Cherem, Yejin Choi, Filip Dreger, Dorota Głażewska, Maya Haridasan, Katarzyna Kobyłecka-Żmijewska, Tomasz Michalski, Krzysztof Ostrowski, Andy Scukanec, Jonathan Winter, Anna Włudarska, Michał Wiktor Żmijewski and all others whom I forgot to mention.

Last, but definitely not least, I want to thank my family, for their constant support and for helping me becoming who I am. My wife Zofia Stankiewicz has been incredibly supportive and encouraging; her helpful editorial comments on various parts of this thesis made it much better. I can hardly express my thanks to her.

TABLE OF CONTENTS

Biographical Sketch . . . iii
Acknowledgements . . . iv
Table of Contents . . . v
List of Figures . . . vii

1 Introduction **1**
1.1 Sets . . . 1
1.2 Types . . . 2
1.3 Thesis map . . . 4

2 Towards computational understanding of set theory I : Propositions and Numbers **7**
2.1 Propositional calculus . . . 9
2.1.1 $\lambda^{\rightarrow}$ calculus . . . 12
2.1.2 Properties of $\lambda^{\rightarrow}$ . . . 17
2.1.3 Realizability for IPC . . . 24
2.1.4 Normalization of $\lambda^{\rightarrow}$ . . . 28
2.2 First-order arithmetic . . . 31
2.2.1 λH calculus . . . 36
2.2.2 Properties of λH . . . 41
2.2.3 Realizability for HA . . . 48
2.2.4 Normalization of λH . . . 50
2.2.5 Computation in HA . . . 54

3 Towards computational understanding of set theory II : Sets **58**
3.1 Second-order arithmetic . . . 58
3.1.1 λS calculus . . . 62
3.1.2 Properties of λS . . . 64
3.1.3 Realizability for HAS . . . 66
3.1.4 Normalization of λS . . . 71
3.2 Set theory . . . 74
3.2.1 The axioms of $\mathrm{IZF_R}$ . . . 75
3.2.2 λZ calculus . . . 82
3.2.3 Properties of λZ . . . 86
3.2.4 Realizability for $\mathrm{IZF_R}$ . . . 88
3.2.5 Normalization of λZ . . . 100
3.3 Program extraction . . . 109
3.4 Historical context . . . 115

4 Beyond IZF **117**
4.1 Inaccessible sets . . . 117
4.2 Dependent set theory . . . 128
4.2.1 $\mathrm{IZF_D}$ . . . 131
4.2.2 The terms of λD . . . 132
4.2.3 Realizability . . . 140
4.2.4 Normalization . . . 145
4.2.5 The properties of λD . . . 147
4.2.6 Program extraction . . . 150
4.2.7 Conclusion . . . 152
4.3 Historical context . . . 152

5 Program-extracting semantics **154**
5.1 Higher-order logic . . . 156
5.2 Semantics . . . 158
5.2.1 Set theory . . . 158
5.2.2 The definition of the semantics . . . 160
5.2.3 Properties . . . 162
5.2.4 Soundness . . . 163
5.3 Extraction . . . 168
5.4 Conclusion . . . 170

6 Conclusion **172**

Bibliography **174**

LIST OF FIGURES

2.1 Heyting Arithmetic 35

3.1 Second-order Heyting Arithmetic (HAS) 60

4.1 The axioms of $\mathrm{IZF_D}$ 132

CHAPTER 1

INTRODUCTION

1.1 Sets

The concept of a set is likely the most successful mathematical abstraction in the history of humankind. It is very simple: high school students can easily grasp the idea and experiment on finite sets. It can be described concisely: the number of axioms and axiom schemas in modern set theories is usually smaller than 10. Yet, at the same time, the concept of a set is powerful enough to encompass almost all of modern mathematics. And indeed, the Zermelo-Fraenkel set theory with Choice, ZFC, is widely accepted as *the* foundation of mathematics.

The origins of ZFC can be traced to Georg Cantor, a German mathematician. His work on convergence of Fourier series led him to investigate, starting in 1879, the nature of numbers and infinity [Can55]. At the time, mathematicians thought that the infinity was uniform and that there was no difference between the quantity of natural and real numbers. Cantor's results destroyed this view forever, when he invented the "diagonal method" argument to show that in fact there are more reals than natural numbers. As sets were an important tool in his studies, he started investigating them for their own sake. Soon, formal axiomatizations of set theory by Zermelo [Zer08] and Fraenkel [Fra22] followed. Later, Kuratowski invented a simple encoding of ordered pairs as unordered pairs, von Neumann showed how to uniformly encode natural numbers and ordinals as sets and Bourbaki gave a unified treatment of a large part of contemporary mathematics based on sets [Bou49, Bou68b, Bou68a]. These developments convinced most mathematicians that set theory was a valid foundation of mathematics.

This view has changed litle since then. In 2007, set theory is as relevant for mathematics as it was 60 years ago. As no widely accepted substitutes to ZFC have arised, it is still considered as *the* foundation of pure classical mathematics.

1.2 Types

In the meantime, quietly, a new world of computation was coming into existence. In 1936, Alan Turing described [Tur36] a seminal theoretical model of computation. The first digital computers were built during the Second World War. As computers evolved, so did programming languages, indispensible tools to control the complexity of tasks assigned to machines. The ARPANET was born in 1969 and evolved into the Internet through 1970s and 1980s. Search engines arrived in mid 1990s. Since then, there was no way back; nowadays, computers are an essential and indispensible part of human society and programming languages are the medium we use to communicate with machines.

Throughout the evolution of programming languages, the concept of a *type* proved to be increasingly useful. Its advantages were noticed very early; the first programming language with very rich user-defined types was ALGOL68, designed in the 1960s. Types made the task of writing correct programs much easier. They provided means to classify data and helped the compiler find common mistakes and notify the programmer.

The full importance of types came to be realized a few years later. The essential role in this discovery was *intuitionism*, also called *constructivism*, a philosophical approach to mathematics advanced by the Dutch mathematician L. E. J. Brouwer [Bro07, vS90, van99]. Briefly, constructivism is occupied with *effective* methods in mathematics. The law of excluded middle is rejected, as inherently non-constructive. A detailed description of intuitionism can be found for example in [Hey66]. Brouwer's ideas were essential to the following three mostly independent lines of research, which in the end made the *type theory* an integral part of computer science.

On the computer science side, researchers were trying to develop new tools for software correctness. Most approaches were based on axiomatic logics, such as Hoare's logic [Hoa69] or Floyd's method [Flo67]. An important exception was Robert Constable's group at Cornell, trying to apply constructive mathematics for the purpose of program synthesis [Con71]. A synthesized program could be *correct-by-construction*, with no need for further testing and verification.

On the mathematical side, logicians such as Haskell B. Curry [CFC58], Dag Prawitz [Pra65], William Howard [How80], Hans Lauchli [Lau70] and Jean-Yves Girard [Gir72] were investigating the properties of proofs in constructive logics. The Dutch mathematician N. G. de Bruijn built a system Automath [dB70] to formalize and verify mathematical proofs. In retrospect, Automath introduced and utilized important features of type theory. Dana Scott [Sco70] described a setting very close to modern type theories; had it not been for his postscript to the paper, effectively denouncing the framework, he might very well be termed as the originator of modern type theory.

The final piece of the puzzle came from the philosophical side. In 1973, a Swedish mathematician Per Martin-Löf noticed an amazing thing. He showed that the concept of a type, extended to the extreme, is a valid foundation for constructive mathematics [ML73]. His *type theory* was a new foundational basis, an important discovery for constructive mathematicians and philosophers. Soon after, Constable's group realized that as computation is an integral part of type theory, the theory can and should be applied for the purpose of assuring software correctness. They plunged forward and built a proof assistant PRL based on Martin-Löf's predicative type theory [CAB^{+}86, Con98]. An important feature of PRL and type theory was that it enabled *program extraction* from formalized proofs; the process of software development was reduced to the activity of theorem proving. The research program of Constable's group, now almost 30 years old, bore numerous fruits. See [ABC+06] for a partial account. Other computer science researchers noticed type theory as well; it is now being successfully applied on a large scale for the purpose of program verification, programming languages semantics and software engineering.

Type theory is not without its limitations, however. It lacks the power of set theory: the strongest type theories are much weaker than ZFC. It is more difficult to understand — the concept of a type, although familiar to programmers, is not nearly as intuitive as the concept of a set. Moreover, its mechanisms for abstraction are different than those that have been developed for years by mathematicians; it is therefore a remarkable challenge to code all mathematical knowledge in type theories.

On the other hand, set theory turned out to be a very inconvenient setting to reason formally about computation. As it is inherently static (sets are thought of as static objects, existing Platonically), computation needs to be somehow modelled inside. Although various embeddings of computation in the world of sets exist, from Turing machines, via Post systems and lambda calculi, to denotational and operational semantics, most turned out to be impractical. Indeed, we are aware of only one tool based on set theory [Abr96], compared with more than 20 based on different foundational bases.

1.3 Thesis map

In this thesis, we investigate whether it is possible to bring these two worlds closer. We first show in Chapter 2 that the world of sets is not as static as it seems. We investigate a constructive set theory IZF and exhibit computation hidden inside the layers of sets and logic. For this purpose, we develop a typed lambda calculus λZ, corresponding to proofs in IZF via the so-called *Curry-Howard isomorphism.* Using realizability inspired by McCarty's thesis [McC84], we show the normalization property of the calculus and explain how to use it for the purpose of providing software correct-by-construction. As the normalization proof is quite intricate, we approach IZF and λZ in stages, applying our techniques first for simpler, well-known constructive systems: propositional logic, Heyting Arithmetic and second-order Heyting Arithmetic. We then show how to apply normalization to extract programs from IZF proofs.

In Chapter 4, we discuss some possible extensions of these ideas. We first show in Section 4.1 how to enhance IZF to incorporate inaccessible sets while still supporting program extraction capability. This extension makes the set theory more powerful than all type theories used in practice. Furthermore, in Section 4.2 we show that features characteristic of the world of types can play an important role in the world of sets and we expose the danger resulting from its mixture. More specifically, we investigate a *dependent set theory* IZF_D, resulting from extending the first-order logic underlying IZF with typical type-theoretic features such as Σ-types. We show that unless Σ-types are restricted, the resulting theory is inconsistent; however, with restricted Σ-types, it has the proof-theoretic power of ZFC.

In Chapter 5 we show that our constructive set theories can be used to provide program extraction capability for existing theories. We consider the popular Higher-Order Logic (HOL), a basis for popular proof assistants such as Isabelle/HOL and PVS. We provide a constructive semantics in IZF for the constructive core of HOL and show that the semantics itself can serve as a tool for program extraction.

We believe our results bring better understanding of the nature of sets and types. Our hope is that the discovery of underlying computation can make sets more relevant in the 21st century, which is very likely to become more dominated by computation. At the same time, further integration of sets and types, such as the one we present in Section 4.2, can bring the unfamiliar world of types closer to mathematicians, in turn making it easier for computer science to utilize mathematics in formal as well as informal ways. We are hopeful that such a unified setting can be constructed.

CHAPTER 2

TOWARDS COMPUTATIONAL UNDERSTANDING OF SET THEORY I : PROPOSITIONS AND NUMBERS

In the previous chapter, we briefly presented the history of set theory, widely accepted as the foundation of mathematics. It is known how to model every mathematical object of interest as a set with little difficulty. The abstraction of a set is easy to understand and the picture of the universe built of sets is simple and compelling.

An important characteristic of the set-theoretic universe is that it is an inherently *static* entity. Sets are Platonic objects and the axioms of set theory capture some truths about them. Since the proof rules of the underlying first-order logic are intuitively true, mathematics is viewed as discovering the truth about the universe. Logic and mathematical proofs therefore serve as witnesses to externally true facts. However, in some sense they are superflous; if mathematicians could somehow tap into the fabric of the universe and "see" the sets directly, no proofs would be necessary, as the truth they are trying to discover would become self-evident. One prominent example of this view is the still unfinished search for an *intuitively appealing* axiom which could *decide* the Continuum Hypothesis.

In 1958, an innocous remark by an American logician Haskell B. Curry started a revolution against this view of the world: "Note the similarity of the postulates for F and those for P. If in any of the former postulates we change F to P and drop the combinator we have the corresponding postulate for P" [CFC58]. With this remark, a new, ground-breaking idea was born. Although it took a significant number of years for this idea to ripen, in our opinion the view of the world of mathematics has been changed forever. This idea is usually called *the Curry-Howard isomorphism*, although see page viii of [SU06] for a name more faithful to the contributing researchers.

The ground-breaking change brought about by this isomorphism can be summarized briefly as follows. *There is more to proofs than meets the eye.* More specifically, proofs are a link between the static, 2000 year-old world of mathematics and the young, dynamically expanding world of computation. They provide access to the computation hidden deeply among formulas, numbers, species and sets. In fact, proofs *are* the computation. A statement "p is a proof of formula ϕ" can be viewed at the same time as "p is a program satisfying the specification ϕ". The former is of fundamental importance to mathematics; the latter to computer science. The isomorphism brings these two worlds together.

It is surprisingly easy to overlook the consequences of the isomorphism on the view of the world of mathematics. The language of mathematical proofs is no longer simply a mere, imperfect tool used to discover distant, static Platonic truths. It is an essential and lively part of mathematics. It provides a connection to the world of computation and in fact *embodies* computation within. The isomorphism has therefore significantly expanded our knowledge in the fields of linguistics (as mathematical developments are usually written using informal language, close to the natural one), computer science (which is a field concerned in large part with computation), mathematics (as it greatly deepened our understanding of the foundations of mathematics) and philosophy.

We now embark with the reader on a quest of defining, understanding and applying the isomorphism to set theory. We will start from one of the simplest instances of the isomorphism, propositional logic, and culminate in a full-blown set theory. We hope that our presentation will shed light on the technical issues involved and make the relevant proofs easier to understand. Most importantly, we wish to make the reader grasp the source of computation in the world of mathematics.

The isomorphism is usually used for constructive theories, without the excluded middle rule. This is not surprising, as the excluded middle is flagrantly anti-computational. For example, using the rule one may prove that "every Turing machine either terminates or not", while it is well-known that there is no computation which could decide which is the case. From our point of view, constructivism is a tool which serves to understand the computational nature of formal systems. We view the standard classical systems as consisting of a constructive core, essential for understanding the computational content and the rest of the system, resulting from the excluded middle rule. In other words, we *factor* classical systems into a constructive core and the excluded middle rule. Whether the constructive core is in some sense a better theory than the whole system, is a question we prefer to leave to philosophers. Instead, we now plunge with the reader into the constructive cores of well-known and established formal systems, often taught in undergraduate logic courses: propositional logic and arithmetic.

2.1 Propositional calculus

We will start with one of the weakest logics in existence, which at the same time is an indispensible core of almost all formal systems designed to capture human reasoning. Before we start the formal presentation, we encourage the reader to read the previous sentence again. That nowadays we can define and discuss "logics" is a result of an important paradigm shift, which was an essential step on the road to the Curry-Howard isomorphism.

Indeed, logic became a valid subject of discourse remarkably late. Although already Aristotle discovered some laws of logic, called syllogisms, it was not until George Boole in the 19th century that logic in the modern sense started to be formalized. Boole, however, was occupied mainly with propositional equivalences between formulas, such as "$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$".

The three essential steps were undertaken by Gottlob Frege, David Hilbert and Gerhard Gentzen. Frege [Fre67] gave the first formal definition of formulas of first-order logic, in particular inventing the modern notion of a quantifier. He thus showed that the language of mathematics is amenable to formal treatment. Hilbert [HA28] provided the first *proof system* for first-order logic. He wanted to use the system for the purpose of his formalist program, to show consistency of mathematics. Although this task was doomed to failure [Göd31], his system is still used for the purpose of studying logic.

The main fault of Hilbert's system is that it is extremely inconvenient to use to formalize real mathematical arguments. A German mathematician Gerhard Gentzen thus came up with different, yet equivalent systems, called *natural deduction* and the *sequent calculus* [Gen69]. His systems allowed reasoning under assumptions, formalized the idea of *logical consequence* and were much closer to mathematical practice. He also found a way to simplify formal proofs in his systems and investigated the properties of such simplifications. This was the last necessary step on the side of proof systems to find computation in mathematics. Natural deduction is very close to the type systems of real programming languages and proof simplification corresponds to program computation. Indeed, all the systems in this section will be based on Gentzen's natural deduction, presented in a sequent calculus style.

Our starting point is one of the simplest logics in existence: Intuitionistic Propositional Calculus (IPC). It is a core of almost every reasonable logic in existence and indeed, it will be a part of all logics we investigate. IPC consists of *syntax* and *proof rules.* The syntax specifies the language of the logic. The proof rules specify the derivable statements. These are deemed to be true and the logic is supposed to capture some truths about the world.

The propositional calculus is parameterized by a countable set of *propositional variables* which we denote by $PVar$. We will use the letters p, q, r for propositional variables. The *formulas* of IPC are generated by the following abstract grammar:

$$\phi \quad ::= \quad p \mid \bot \mid \phi \wedge \phi \mid \phi \vee \phi \mid \phi \to \phi$$

Thus, IPC can be used to reason about only relatively simple formulas. The propositional variables intuitively denote statements whose precise formulation does not concern us.

The proof system for IPC allows to derive judgments of the form $\Gamma \vdash \phi$, read as "in the context Γ, the formula ϕ is derivable". A *context* is a finite set of formulas, which intuitively are assumptions, under which ϕ is true. The notation Γ, ϕ stands for the context $\Gamma \cup \phi$. The proof system is generated inductively by the following proof rules:

$$\frac{}{\Gamma, \phi \vdash \phi} \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash \phi} \qquad \frac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \phi \to \psi} \qquad \frac{\Gamma \vdash \phi \to \psi \quad \Gamma \vdash \phi}{\Gamma \vdash \psi}$$

$$\frac{\Gamma \vdash \phi}{\Gamma \vdash \phi \vee \psi} \qquad \frac{\Gamma \vdash \psi}{\Gamma \vdash \phi \vee \psi} \qquad \frac{\Gamma \vdash \phi \vee \psi \quad \Gamma, \phi \vdash \vartheta \quad \Gamma, \psi \vdash \vartheta}{\Gamma \vdash \vartheta}$$

$$\frac{\Gamma \vdash \phi \quad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi} \qquad \frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \phi} \qquad \frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \psi}$$

We shall now present the computational content of IPC proofs. To show the reader just a glimpse of its source, consider the proof: "If $\phi \wedge \psi$, then ϕ." Given a proof of $\phi \wedge \psi$, we can produce a proof of ϕ, *just as* given a pair of natural numbers, we can produce the first component of a pair. For the general, formal account of computation in IPC, we are going to introduce a *lambda calculus*, which we call $\lambda^{\to}$.

2.1.1 $\lambda^{\rightarrow}$ calculus

The last missing ingredient to define the Curry-Howard isomorphism and discover computation in proofs was invented by Alonzo Church in 1930s. This ingredient was the famous *lambda calculus*, a system of notation for functions. Church thought it could be used to serve as a foundation for mathematics. Although this application did not work in the end, a couple of years later the logicians Haskell B. Curry and William Howard noticed an amazing similarity between Church's lambda calculus and natural deduction proofs. We are now ready to present the lambda calculus and the similarity, nowadays called *isomorphism*, which we will express formally in Lemma 2.1.5.

A lambda calculus, similarly to a logic, is a purely syntactic system. We are interested in *typed* lambda calculi, which are essentially programming languages with types. The reason for our interest is that some lambda calculi correspond exactly to logical systems and capture their computational nature. The calculi we consider are also called *monomorphic intensional type theories*.

Simpler calculi can be introduced in several well-separated stages. First, the *types* and *terms* of the system are defined. Along with the terms come the reduction rules, which make the calculus a computational system. Finally, the typing system is described. We will present now present our first lambda calculus $\lambda^{\rightarrow}$. This calculus corresponds to IPC; the correspondence is captured formally by Lemma 2.1.5.

We first fix a countable set of variables Var. These are completely unrelated to $PVar$. Variables from Var will be denoted by letters x, y, z. We will usually call them *proof variables*.

The *types* of $\lambda^{\rightarrow}$ are IPC formulas. We will therefore use Greek letters ϕ, ψ, ϑ to stand for $\lambda^{\rightarrow}$ types.

The *terms* of $\lambda^{\rightarrow}$, denoted by M, N, O, are generated by the following abstract grammar:

$$
\begin{aligned}
M \quad ::= \quad & x \mid M\ N \mid \lambda x : \phi.\ M \mid \mid \mathrm{inl}(M) \mid \mathrm{inr}(M) \mid \\
& \mathrm{case}(M, x : \phi.\ N, x : \psi.\ O) \mid \langle M, N \rangle \mid \mathrm{fst}(M) \mid \mathrm{snd}(M) \mid \mathrm{magic}(M)
\end{aligned}
$$

Intuitively, these terms are notations for proofs and, at the same time, they are programs. For example, $M\ N$ is an application term; if M is a notation for a proof of $\phi \rightarrow \psi$ and N is a notation for a proof of ϕ, then $M\ N$ is a notation for a proof of ψ. This corresponds to the situation when we have a Lemma L showing $\phi \rightarrow \psi$ and a proof p of ϕ; then we can obviously apply Lemma L to p to show ψ. The terms $\mathrm{inl}(M), \mathrm{inr}(M)$ and $\mathrm{case}(M, x : \phi.\ N, x : \psi.\ O)$ correspond to the proof rules for disjunction. Similarly, the terms $\langle M, N \rangle, \mathrm{fst}(M), \mathrm{snd}(M)$ correspond to the proof rules handling conjuction. The $\mathrm{magic}(M)$ term corresponds to the rule handling absurdity ($\bot$). Finally, lambda abstractions and applications are used to handle implication. Formally, the correspondence will be captured by the typing system for $\lambda^{\rightarrow}$ and Lemma 2.1.5 below.

The variable x in $\lambda x : \phi.\ M$ and $\mathrm{case}(M, x : \phi.\ N, x : \psi.\ O)$ terms binds its occurrences in M, N, O, respectively. We consider terms differing only in their bound variables (also called α-equivalent) the same. The free variables of a term M, denoted by $FV(M)$ and capture-avoiding substitution are defined as usual. A term M such that $FV(M) = \emptyset$ is called *closed*. The notation $M[x := N]$ stands for the term M with N substituted for x. To the reader unfamiliar with these notions we recommend the first chapter of [SU06].

The computational nature of $\lambda^{\rightarrow}$ is exhibited by the deterministic reduction relation $\rightarrow$. We first define the *base reductions*:

$$\frac{}{(\lambda x : \phi.\ M)\ N \rightarrow M[x := N]}$$

$$\overline{\text{case}(\text{inl}(M), x : \phi.\ N, x : \psi.\ O) \to N[x := M]}$$

$$\overline{\text{case}(\text{inr}(M), x : \phi.\ N, x : \psi.\ O) \to O[x := M]}$$

$$\overline{\text{fst}(\langle M, N\rangle) \to M} \qquad \overline{\text{snd}(\langle M, N\rangle) \to N}$$

In an arbitrary term M there can be many subterms amenable to the base reduction rules. For example, if $I_p \equiv \lambda x : p.\ x$, then there are three ways in which we could a priori reduce $\text{fst}(\langle I_p\ I_p, I_p\ I_p\rangle)$:

$$\begin{array}{rcl}
\text{fst}(\langle I_p\ I_p, I_p\ I_p\rangle) & \to & I_p\ I_p \\
\text{fst}(\langle I_p\ I_p, I_p\ I_p\rangle) & \to & \text{fst}(\langle I_p, I_p\ I_p\rangle) \\
\text{fst}(\langle I_p\ I_p, I_p\ I_p\rangle) & \to & \text{fst}(\langle I_p\ I_p, I_p\rangle)
\end{array}$$

As we want our reduction system to be deterministic, we need to fix a strategy for applying the base reductions in an arbitrary term. A convenient method is to split the set of terms into *values* (also called *canonical forms*) and non-values. Intuitively, a value is the result of the computation process and does not allow any further reductions. Formally, we define the values of $\lambda^{\to}$ as the terms generated by the following abstract grammar, where M is an arbitrary lambda term:

$$V \quad ::= \lambda x : \phi.\ M \mid \text{inl}(M) \mid \text{inr}(M) \mid \langle M, N\rangle$$

In terms which are not values, we shall designate a *principal argument*, denoted by $[\circ]$. Informally, the reduction of a term M proceeds as follows:

- If M is a value, stop.
- If M is not a value, inspect its principal argument:
  - If it is a value, apply one of the base reduction rules.
  - If it is not a value, reduce this principal argument.

We define the principal arguments in $\lambda^{\rightarrow}$, also called *evaluation contexts*, by the following abstract grammar:

$$[\circ] \ ::= \ [\circ]\ M \mid \mathrm{case}([\circ], x:\phi.N, x:\psi.O) \mid \mathrm{fst}([\circ]) \mid \mathrm{snd}([\circ]) \mid \mathrm{magic}([\circ])$$

Formally, we extend our base reduction relation $\rightarrow$ to all lambda terms by the following inductive definition:

$$\frac{M \rightarrow M'}{M\ N \rightarrow M'\ N} \qquad \frac{M \rightarrow M'}{\mathrm{case}(M, x:\phi.\ N, x:\psi.\ O) \rightarrow \mathrm{case}(M', x:\phi.\ N, x:\psi.\ O)}$$

$$\frac{M \rightarrow M'}{\mathrm{fst}(M) \rightarrow \mathrm{fst}(M')} \qquad \frac{M \rightarrow M'}{\mathrm{snd}(M) \rightarrow \mathrm{snd}(M')} \qquad \frac{M \rightarrow M'}{\mathrm{magic}(M) \rightarrow \mathrm{magic}(M')}$$

This evaluation order is usually called *lazy evaluation* or *call-by-name* and our method of introducing the reduction relation is called *small-step* operational semantics. See [Pie02] for more information on programming languages and operational semantics.

To show one example, the reduction sequence starting from $\mathrm{fst}(\langle I_p\ I_p, I_p\ I_p\rangle)$ is:

$$\mathrm{fst}(\langle I_p\ I_p, I_p\ I_p\rangle) \rightarrow I_p\ I_p \rightarrow I_p$$

It is straightforward to translate a definition using principal arguments to the inductive rules. From now on, we will be extensively using principal arguments to define our reduction relations.

Note that there are no reductions possible from values. This confirms the intuition of a value being the result of a computation, as no further computation starting from a value is possible.

Definition 2.1.1 *We write $M\downarrow$ if the reduction sequence starting from M terminates. In this situation we also say that M* normalizes. *We write $M \downarrow v$ if we want to state that v is the term at which this reduction sequence terminates. We write $M \rightarrow^* M'$ if M reduces to M' in some number of steps.*

As $\lambda^{\rightarrow}$ is a system corresponding to a very simple logic, its computational capabilities are not very impressive. Nevertheless, we exhibit one example:

Example 2.1.2 *A term* $I_{p \to p} \equiv \lambda x : p \to p.\ x$ *applied to any term* M *returns* M*. For example:* $I_{p \to p}\ (\lambda x : p.\ x) \to \lambda x : p.\ x$*. Thus* $I_{p \to p}$ *is computationally an identity function.*

The example shows how lambda terms are interpreted as computational functions, or programs — the result of a program M on an argument N is the value to which $M\ N$ reduces. Once our lambda calculi become more complicated, we shall see more interesting examples. For now, we proceed to the type system of $\lambda^{\rightarrow}$.

The *typing judgments* of $\lambda^{\rightarrow}$ are of the form $\Gamma \vdash M : \phi$, read as "in the context Γ, the term M is of type ϕ". In $\lambda^{\rightarrow}$, contexts are sets of pairs (x, ϕ), where $x \in Var$ and ϕ is a type. The *domain* of a context Γ, denoted by $\mathrm{dom}(\Gamma)$, is the set $\{x \mid (x, \phi) \in \Gamma\}$. The *range* of a context Γ, denoted by $\mathrm{rg}(\Gamma)$, is the corresponding IPC context $\{\phi \mid (x, \phi) \in \Gamma\}$. The notation $\Gamma, x : \phi$ stands for $\Gamma \cup \{(x, \phi)\}$, where $x \notin \mathrm{dom}(\Gamma)$. The typing system follows.

$$\frac{}{\Gamma, x : \phi \vdash x : \phi} \qquad \frac{\Gamma \vdash M : \bot}{\Gamma \vdash \mathrm{magic}(M) : \phi}$$

$$\frac{\Gamma, x : \phi \vdash M : \psi}{\Gamma \vdash \lambda x : \phi.\ M : \phi \to \psi}\ x \notin \mathrm{dom}(\Gamma) \qquad \frac{\Gamma \vdash M : \phi \to \psi \quad \Gamma \vdash N : \phi}{\Gamma \vdash M\ N : \psi}$$

$$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash \mathrm{inl}(M) : \phi \vee \psi} \qquad \frac{\Gamma \vdash M : \psi}{\Gamma \vdash \mathrm{inr}(M) : \phi \vee \psi}$$

$$\frac{\Gamma \vdash M : \phi \vee \psi \quad \Gamma, x : \phi \vdash N : \vartheta \quad \Gamma, x : \psi \vdash O : \vartheta}{\Gamma \vdash \mathrm{case}(M, x : \phi.\ N, x : \psi.\ O) : \vartheta}$$

$$\frac{\Gamma \vdash M : \phi \quad \Gamma \vdash N : \psi}{\Gamma \vdash \langle M, N \rangle : \phi \wedge \psi} \qquad \frac{\Gamma \vdash M : \phi \wedge \psi}{\Gamma \vdash \mathrm{fst}(M) : \phi} \qquad \frac{\Gamma \vdash M : \phi \wedge \psi}{\Gamma \vdash \mathrm{snd}(M) : \psi}$$

Lemma 2.1.3 *If* $\Gamma \vdash M : \phi$*, then* $FV(M) \subseteq \mathrm{dom}(\Gamma)$*.*

Proof Straightforward induction on the proof tree $\Gamma \vdash M : \phi$. ∎

Definition 2.1.4 *A term M is* typable *if there is a formula ϕ such that $\vdash M : \phi$. We say that a typed lambda calculus* normalizes*, if every typable term M normalizes.*

With the typing system in hand, we can state and prove the correspondence between $\lambda^{\rightarrow}$ and IPC:

Lemma 2.1.5 *If $\Gamma \vdash O : \phi$ then $rg(\Gamma) \vdash \phi$. If $\Gamma \vdash_{IPC} \phi$, then there exists a term M such that $\overline{\Gamma} \vdash M : \phi$, where $\overline{\Gamma} = \{(x_\phi, \phi) \mid \phi \in \Gamma\}$.*

Proof Straightforward induction on the proof trees $\Gamma \vdash O : \phi$ and $\Gamma \vdash_{IPC} \phi$. Note that each typing rule corresponds to a proof rule in IPC. For the first part of the claim, simply erase lambda terms from the proof. The second part follows easily. ■

Thus indeed the lambda terms of $\lambda^{\rightarrow}$ are exactly the proofs of IPC. A different, equally valid point of view, is that any formula ϕ of IPC is a *specification* and lambda term of type ϕ realizes the specification. This is often called the *propositions-as-types* principle. The reader will need to wait until the next section to see a realistic example of the principle in action.

In order to exhibit the connection of the computational nature of $\lambda^{\rightarrow}$ with IPC and to provide a mechanism to access the computational content in IPC, we first need to show several standard technical properties of $\lambda^{\rightarrow}$.

2.1.2 Properties of $\lambda^{\rightarrow}$

We start with two technical properties. First, unnecessary extra assumptions can be added to the proof with no harm:

Lemma 2.1.6 (Weakening) *If $\Gamma \vdash Q : \Psi$ and $y \notin \mathrm{dom}(\Gamma)$, then $\Gamma, y : \psi \vdash Q : \Psi$.*

Proof Induction on $\Gamma \vdash Q : \Psi$. Most of the cases follow by a straightforward application of the induction hypothesis. Consider the following interesting cases of the last rule applied in the proof:

- $$\frac{}{\Gamma, x : \phi \vdash x : \phi}$$

  By the assumption, $y \neq x$. Thus also trivially $\Gamma, x : \phi, y : \psi \vdash y : \phi$.

- $$\frac{\Gamma, x : \phi_1 \vdash M : \phi_2}{\Gamma \vdash \lambda x : \phi_1.\ M : \phi_1 \to \phi_2}\ x \notin \mathrm{dom}(\Gamma)$$

  Take any $y \notin \mathrm{dom}(\Gamma)$. Without loss of generality we may assume that $y \neq x$. Therefore, by the induction hypothesis, $\Gamma, x : \phi_1, y : \psi \vdash M : \phi_2$, so also $\Gamma, y : \psi \vdash \lambda x : \phi_1.\ M : \phi_1 \to \phi_2$. ■

Second, the Substitution Lemma roughly says that we can replace a free variable in a term by a term of the same type. This Lemma is strictly technical and used only as a tool in the proof of Lemma 2.1.10.

Lemma 2.1.7 (Substitution Lemma) *If $\Gamma, x : \phi \vdash M : \psi$ and $\Gamma \vdash N : \phi$, then $\Gamma \vdash M[x := N] : \psi$.*

Proof By induction on $\Gamma, x : \phi \vdash M : \psi$. We show several interesting cases. Case $\Gamma, x : \phi \vdash M : \psi$ of:

- $$\frac{}{\Gamma, y : \phi_1 \vdash y : \phi_1}$$

  If $y = x$, then $M[x := N] = N$ and $\phi_1 = \phi$, so we get the claim easily. If $y \neq x$, then $M[x := N] = M = y$. We need to show that $\Gamma \setminus \{(x, \phi)\} \vdash y : \psi$. But $(y, \psi) \in (\Gamma \setminus \{(x, \phi)\})$, so we get the claim.

- $$\frac{\Gamma, x : \phi, y : \psi_1 \vdash M_1 : \psi_2}{\Gamma, x : \psi \vdash \lambda y : \psi_1.\ M_1 : \psi_1 \to \psi_2}\ y \notin \mathrm{dom}(\Gamma, x : \phi)$$

  Without loss of generality we can assume that y is fresh, so in particular $y \notin FV(N)$. By the induction hypothesis, $\Gamma, y : \psi_1 \vdash M_1[x := N] : \psi_2$, so also $\Gamma \vdash (\lambda y : \psi_1.\ M_1[x := N]) : \psi_2$. Since $y \notin FV(N)$, $\Gamma \vdash (\lambda y : \psi_1.\ M_1)[x := N] : \psi_2$.

■

The next lemma analyzes the last typing rule applied, depending on the form of the lambda term in the conclusion.

Lemma 2.1.8 (Inversion) *Suppose* $\Gamma \vdash Q : \Psi$. *Suppose* Q *is of the form:*

- $M\ N$. *Then there is* ϕ *such that* $\Gamma \vdash M : \phi \to \Psi$, $\Gamma \vdash N : \phi$ *and the proof ends with:*
  $$\frac{\Gamma \vdash M : \phi \to \Psi \quad \Gamma \vdash N : \phi}{\Gamma \vdash M\ N : \Psi}$$

- $\lambda x : \phi.\ M$. *Then for some* ψ, $\Psi = \phi \to \psi$, $\Gamma, x : \phi \vdash M : \psi$ *and the proof ends with:*
  $$\frac{\Gamma, x : \phi \vdash M : \psi}{\Gamma \vdash \lambda x : \phi.\ M : \phi \to \psi}$$

- $\mathrm{inl}(M)$. *Then for some* ϕ, ψ, $\Psi = \phi \vee \psi$, $\Gamma \vdash M : \phi$ *and the proof ends with:*
  $$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash \mathrm{inl}(M) : \phi \vee \psi}$$

- $\mathrm{inr}(M)$. *Then for some* ϕ, ψ, $\Psi = \phi \vee \psi$, $\Gamma \vdash M : \psi$ *and the proof ends with:*
  $$\frac{\Gamma \vdash M : \psi}{\Gamma \vdash \mathrm{inr}(M) : \phi \vee \psi}$$

- $\mathrm{case}(M, x : \phi.\ N, x : \psi.\ O)$. *Then* $\Gamma \vdash M : \phi \vee \psi$, $\Gamma, x : \phi \vdash N : \Psi, \Gamma, x : \psi \vdash O : \Psi$ *and the proof ends with:*
  $$\frac{\Gamma \vdash M : \phi \vee \psi \quad \Gamma, x : \phi \vdash N : \vartheta \quad \Gamma, x : \psi \vdash O : \Psi}{\Gamma \vdash \mathrm{case}(M, x : \phi.\ N, x : \psi.\ O) : \Psi}$$

- fst(M). *Then for some ψ, $\Gamma \vdash M : \Psi \wedge \psi$ and the proof ends with:*

$$\frac{\Gamma \vdash M : \Psi \wedge \psi}{\Gamma \vdash \mathrm{fst}(M) : \Psi}$$

- snd(M). *Then for some ϕ, $\Gamma \vdash M : \phi \wedge \Psi$ and the proof ends with:*

$$\frac{\Gamma \vdash M : \phi \wedge \Psi}{\Gamma \vdash \mathrm{snd}(M) : \Psi}$$

- $\langle M, N\rangle$. *Then for some ϕ, ψ, $\Psi = \phi \wedge \psi$, $\Gamma \vdash M : \phi$, $\Gamma \vdash N : \psi$ and the proof ends with:*

$$\frac{\Gamma \vdash M : \phi \quad \Gamma \vdash N : \psi}{\Gamma \vdash \langle M, N\rangle : \phi \wedge \psi}$$

- magic(M). *Then $\Gamma \vdash M : \bot$ and the proof ends with:*

$$\frac{\Gamma \vdash M : \bot}{\Gamma \vdash \mathrm{magic}(M) : \Psi}$$

Proof Straightforward inspection of the typing rules of $\lambda^{\rightarrow}$. ■

We will often invoke Inversion implicitly to determine the last rule applied in the proof.

The next lemma shows that the form of a value is determined by its type:

Lemma 2.1.9 (Canonical Forms) *If M is a value and $\vdash M : \Psi$, then if Ψ is:*

- *$\phi \rightarrow \psi$ — then $M = \lambda x : \phi.\ N$ and $x : \phi \vdash N : \psi$.*
- *$\phi \vee \psi$ — then either $M = \mathrm{inl}(N)$ and $\vdash N : \phi$ or $M = \mathrm{inr}(N)$ and $\vdash N : \psi$.*
- *$\phi \wedge \psi$ — then $M = \langle N, O\rangle$, $\vdash N : \phi$ and $\vdash O : \psi$.*
- *$\bot$ — never happens.*

Proof Straightforward inspection of the typing rules and possible values. ■i

We can now show the main properties of the computational behavior of $\lambda^{\rightarrow}$. First, we show that computation preserves meaning. In other words, program execution cannot change the program's properties, captured by its type. Formally:

Lemma 2.1.10 (Subject Reduction, Preservation) *If* $\Gamma \vdash P : \Psi$ *and* $P \to Q$*, then* $\Gamma \vdash Q : \Psi$.

Proof By induction on the definition of $P \to Q$. Case $P \to Q$ of:

- $(\lambda x : \phi.\ M)\ N \to M[x := N]$. By Inversion, $\Gamma \vdash \lambda x : \phi.\ M : \phi \to \Psi$ and $\Gamma \vdash N : \phi$. By Inversion again, $\Gamma, x : \phi \vdash M : \Psi$. By Substitution Lemma, $\Gamma \vdash M[x := N] : \Psi$.

- $\mathrm{case}(\mathrm{inl}(M), x : \phi.\ N, x : \psi.\ O) \to N[x := M]$. By Inversion, $\Gamma \vdash \mathrm{inl}(M) : \phi \vee \psi$ and $\Gamma, x : \phi \vdash N : \Psi$. By Inversion again, $\Gamma \vdash M : \phi$. By Substitution Lemma, $\Gamma \vdash M[x := N] : \Psi$, which shows the claim.

- $\mathrm{case}(\mathrm{inr}(M), x : \phi.\ N, x : \psi.\ O) \to O[x := M]$. Symmetric to the previous case.

- $\mathrm{fst}(\langle M, N\rangle) \to M$. By Inversion, $\Gamma \vdash \langle M, N\rangle : \Psi \wedge \psi$ for some ψ. By Inversion again, $\Gamma \vdash M : \Psi$ which shows the claim.

- $\mathrm{snd}(\langle M, N\rangle) \to N$. Symmetric to the previous case.

- $$\frac{M \to M_1}{M\ N \to M_1\ N}$$

  By Inversion, for some ϕ, $\Gamma \vdash M : \phi \to \Psi$ and $\Gamma \vdash N : \phi$. By the induction hypothesis, $\Gamma \vdash M_1 : \phi \to \Psi$, thus also $\Gamma \vdash M_1\ N : \Psi$.

- $$\frac{M \to M_1}{\mathrm{case}(M, x : \phi.\ N, x : \psi.\ O) \to \mathrm{case}(M_1, x : \phi.\ N, x : \psi.\ O)}$$

  By Inversion, $\Gamma \vdash M : \phi \vee \psi$, $\Gamma, x : \phi \vdash N : \Psi$, $\Gamma, x : \psi \vdash O : \Psi$. By the induction hypothesis $\Gamma \vdash M_1 : \phi \vee \psi$, so also $\Gamma \vdash \mathrm{case}(M_1, x : \phi.\ N, x : \psi.\ O) : \Psi$.

- $$\frac{M \to M_1}{\mathrm{fst}(M) \to \mathrm{fst}(M_1)}$$

  By Inversion, $\Gamma \vdash M : \Psi \wedge \psi$ for some ψ, by the induction hypothesis $\Gamma \vdash M_1 : \Psi \wedge \psi$, so also $\Gamma \vdash \mathrm{fst}(M_1) : \Psi$.

- $$\frac{M \to M_1}{\mathrm{snd}(M) \to \mathrm{snd}(M_1)}$$

  Symmetric to the previous case.

- $$\frac{M \to M_1}{\mathrm{magic}(M) \to \mathrm{magic}(M_1)}$$

  Straightforward. ■

Furthermore, the computation cannot get "stuck" before it reaches a value, considered to be the result of the computation.

Lemma 2.1.11 (Progress) *If $\vdash P : \Psi$ then either P is a value or there is Q such that $P \to Q$.*

Proof By induction on the length of P. Case P of:

- x. By Lemma 2.1.3, this situation cannot happen.
- $\lambda x : \phi.\ M$. Then P is a value.
- $M\ N$. By Inversion, for some ϕ we have $\vdash M : \phi \to \Psi$. By the induction hypothesis, either M is a value or for some Q, $M \to Q$. In the latter case, $M\ N \to Q\ N$. In the former, by Canonical Forms $M = \lambda x : \phi.\ O$ for some O. Therefore $M\ N = (\lambda x : \phi.\ O)\ N \to O[x := N]$.
- $\mathrm{inl}(M), \mathrm{inr}(M)$. These are values.
- $\mathrm{case}(M, x : \phi.\ N, x : \psi.\ O)$. Similar to the case where $P = M\ N$.

- $\langle M, N\rangle$ is a value.
- $\mathrm{fst}(M), \mathrm{snd}(M)$. Similar to the case where $P = M\ N$.
- $\mathrm{magic}(M)$. By Inversion, $\vdash M : \bot$. By the induction hypothesis, either M is a value or there is M' such that $M \to M'$. By Canonical Forms the former case is impossible. In the latter case, $\mathrm{magic}(M) \to \mathrm{magic}(M')$. ■

Subject Reduction and Progress together provide useful corollaries.

Corollary 2.1.12 *If $\vdash M : \phi$ and $M \downarrow v$, then $\vdash v : \phi$ and v is a value.*

Proof By Subject Reduction, $\vdash v : \phi$. By Progress, v is a value. ■

Corollary 2.1.13 *If $\vdash M : \bot$, then M does not normalize.*

Proof If M normalized, then by Corollary 2.1.12 we would have a value of type $\bot$, which by Canonical Forms is impossible. ■

The importance of normalization from the logical point of view is shown in the following Corollary:

Corollary 2.1.14 *Normalization of $\lambda^{\to}$ implies consistency of IPC.*

Proof Suppose $\lambda^{\to}$ normalizes and there is an IPC proof of $\bot$. By Lemma 2.1.5, we can find a term M such that $\vdash M : \bot$. By Corollary 2.1.13, M does not normalize. This contradiction shows the claim. ■

From our point view, however, an equally important property of normalization is that it provides a means to access computational information hidden in proofs. The reader will need to wait until the section 2.2 to see this mechanism in action, as IPC is a bit too simple for realistic examples.

Having established the importance of normalization, we proceed to show that $\lambda^{\to}$ indeed normalizes. There are many known techniques used to prove normalization. Our choice of a technique called *realizability* stems from the fact that it is the only known technique which generalizes smoothly from $\lambda^{\to}$ to set theory.

2.1.3 Realizability for IPC

Realizability is a technique introduced by Kleene in 1945 [Kle45]; see [vO02] for a historical account. It provides a formal account of the so-called Brouwer-Heyting-Kolmogorov (BHK) interpretation of constructive logic. The BHK interpretation explains what the *construction*, or a constructive proof of a formula is. We present the clauses following [SU06].

- The construction of a propositional variable p is unspecified. This is because intuitively propositional logic only captures generic statements which hold no matter what the actual content of p is.
- There is no construction of $\bot$.
- The construction of a conjunction $\phi \wedge \psi$ is a pair consisting of a construction of ϕ and a construction of ψ.
- The construction of a disjunction $\phi \vee \psi$ is either a construction of ϕ or a construction of ψ.
- The construction of an implication $\phi \to \psi$ is a method, which transforms every construction of ϕ to a construction of ψ.

An important thing to note about the interpretation, often confusing to newcomers, is that it is *not* a mathematical definition. In particular, the notion of a method in the clause for implication is left unspecified. The BHK interpretation provides a set of intuitions which can be formalized in various ways. One such interpretation, which we pursue in this section, is realizability.

Traditionally, a realizability relation, written as $n \Vdash \phi$, relates natural numbers with formulas. The natural numbers are constructions from the BHK interpretation; some are interpreted as pairs, some as methods, implemented as Turing machine indices.

We are interested in using realizability to show normalization of lambda calculi. For this purpose, it is much more convenient to use lambda terms as realizers. However, lambda terms of $\lambda^{\rightarrow}$ are slightly inconvenient for this purpose. They contain more information than necessary; while in $\lambda^{\rightarrow}$ the resulting nuisance is not significant, with more complicated calculi it would significantly obscure the presentation. We therefore use a simplified lambda calculus, which we call $\overline{\lambda^{\rightarrow}}$, for realizability.

Realizability terms

The terms of $\overline{\lambda^{\rightarrow}}$ arise by erasing formulas from the terms of $\lambda^{\rightarrow}$. Formally, $\overline{\lambda^{\rightarrow}}$ terms are an image of the following erasure map $M \rightarrow \overline{M}$ on terms of $\lambda^{\rightarrow}$:

$$\overline{x} \equiv x \qquad \overline{M\ N} \equiv \overline{M}\ \overline{N} \qquad \overline{\lambda x : \phi.\ M} \equiv \lambda x.\ \overline{M}$$

$$\overline{\mathrm{inl}(M)} \equiv \mathrm{inl}(\overline{M}) \qquad \overline{\mathrm{inr}(M)} \equiv \mathrm{inr}(\overline{M})$$

$$\overline{\mathrm{case}(M, x : \phi.\ N, x : \psi.\ O)} \equiv \mathrm{case}(\overline{M}, x.\overline{N}, x.\overline{O})$$

$$\overline{\langle M, N\rangle} \equiv \langle \overline{M}, \overline{N}\rangle \qquad \overline{\mathrm{fst}(M)} \equiv \mathrm{fst}(\overline{M}) \qquad \overline{\mathrm{snd}(M)} = \mathrm{snd}(\overline{M})$$

$$\overline{\mathrm{magic}(M)} = \mathrm{magic}(\overline{M})$$

We will use letters M, N to denote the terms of $\overline{\lambda^{\rightarrow}}$. This will not lead to any confusion, as the context will make it clear whether we are discussing the terms of $\lambda^{\rightarrow}$ or of $\overline{\lambda^{\rightarrow}}$.

The notions of reductions and values are induced from $\lambda^{\rightarrow}$ in an obvious way. Formally, the reduction relation is generated by the following rules:

$$(\lambda x.\ M)\ N \rightarrow M[x := N]$$

$$\mathrm{case}(\mathrm{inl}(M), x.N, x.O) \rightarrow N[x := M] \qquad \mathrm{case}(\mathrm{inr}(M), x.N, x.O) \rightarrow O[x := M]$$

$$\mathrm{fst}(\langle M, N\rangle) \rightarrow M \qquad \mathrm{snd}(\langle M, N\rangle) \rightarrow N$$

and evaluation contexts:

$$[\circ] \quad ::= \quad [\circ]\ M \mid \mathrm{case}([\circ], x.N, x.O) \mid \mathrm{fst}([\circ]) \mid \mathrm{snd}([\circ]) \mid \mathrm{magic}([\circ])$$

The values are generated by the following abstract grammar:

$$V \quad ::= \lambda x.\ M \mid \mathrm{inl}(M) \mid \mathrm{inr}(M) \mid \langle M, N\rangle$$

The following intuitively obvious properties of the erasure hold:

Lemma 2.1.15 $\overline{M[x := N]} = \overline{M}[x := \overline{N}]$

Proof Straightforward induction on M. ■

Lemma 2.1.16 $M \to N$ *implies* $\overline{M} \to \overline{N}$.

Proof By induction on $M \to N$. We show two representative cases of the proof:

- $(\lambda x : \phi.\ M)\ N \to M[x := N]$. Then $(\lambda x.\ \overline{M})\ \overline{N} \to \overline{M}[x := \overline{N}]$. By Lemma 2.1.15 we get the claim.
- $M\ N \to M'\ N$, provided that $M \to M'$. By the induction hypothesis, $\overline{M} \to \overline{M'}$, so also $\overline{M\ N} = \overline{M}\ \overline{N} \to \overline{M'}\ \overline{N} = \overline{M'\ N}$. ■

Lemma 2.1.17 *If* $\overline{P} \to Q'$, *then* $P \to Q$ *and* $\overline{Q} = Q'$.

Proof By induction on $\overline{P} \to Q'$. We show two representative cases of the proof:

- $(\lambda x.\ M')\ N' \to M'[x := N']$. By the definition of the erasure map, $P = (\lambda x : \phi.\ M)\ N$ for some ϕ and $\overline{M} = M'$, $\overline{N} = N'$. Thus $P \to M[x := N]$. Lemma 2.1.15 shows the claim.
- $\overline{M}\ \overline{N} \to O\ \overline{N}$, provided that $\overline{M} \to O$. By the induction hypothesis, $M \to M'$ and $\overline{M'} = O$. Thus $M\ N \to M'\ N$ and we get the claim. ■

These properties make it possible to prove the following, intuitively obvious lemma:

Lemma 2.1.18 *If* $\overline{M} \downarrow$ *then* $M \downarrow$. *In other words, the erasure map preserves normalization.*

Proof Lemma 2.1.17 shows that the reduction sequence $\overline{M} \to \ldots \to v'$ entails the existence of the sequence $M \to \ldots \to v$, where $\overline{v} = v'$. As it is easy to see that v' being a value entails v being a value, the claim follows. ■

The crucial feature of $\lambda^{\to}$ which makes the proof of Lemma 2.1.18 possible, is that types in lambda terms do not play any role in reductions. In other words, evaluation is *type-oblivious*. While this might not seem to be that interesting in the case of $\lambda^{\to}$, we shall see more significant examples of obliviousness of calculi corresponding to stronger systems later.

Realizability relation

Having defined the terms of $\overline{\lambda^{\to}}$, we proceed to define the realizability relation. We remark that our presentation is very close to *logical relations*, used in the world of programming languages. See [Mit96] for more information on this subject.

Definition 2.1.19 *A* realizer *is a closed term of* $\overline{\lambda^{\to}}$.

Definition 2.1.20 *The binary realizability relation* $\Vdash$ *relates realizers to formulas of IPC. We write the relation as an infix operator:* $M \Vdash \phi$ *should be read as "*M *realizes* ϕ*". The relation is defined by structural induction on* ϕ*:*

- $M \Vdash p \equiv M \downarrow$
- $M \Vdash \bot \equiv \bot$
- $M \Vdash \phi \wedge \psi \equiv M \downarrow \langle M_1, M_2 \rangle \wedge (M_1 \Vdash \phi) \wedge (M_2 \Vdash \psi)$

- $M \Vdash \phi \vee \psi \equiv (M \downarrow \mathrm{inl}(M_1) \wedge M_1 \Vdash \phi) \vee (M \downarrow \mathrm{inr}(M_1) \wedge M_1 \Vdash \psi)$
- $M \Vdash \phi \to \psi \equiv (M \downarrow \lambda x.\ M_1) \wedge \forall N.\ (N \Vdash \phi) \to (M_1[x := N] \Vdash \psi)$

The reader should now go back to the BHK interpretation and convince herself that realizability does provide a reasonable implementation of the interpretation.

We proceed to show several easy properties of realizability, which are crucial to the normalization proof and which hold for all realizability relations we consider in this thesis.

Lemma 2.1.21 *If $M \Vdash \phi$, then $M \downarrow$.*

Proof Straightforward — if $\phi = \bot$, the claim is trivial, all other cases of the definition start with a clause assuring normalization of M. ■

Lemma 2.1.22 *If $M \to^* N$, then $M \Vdash \phi$ iff $N \Vdash \phi$.*

Proof For $\phi = \bot$, the proof is straightforward. For the rest of the cases, M realizing ϕ depends only on its normalization and value which do not change with reductions. ■

Lemma 2.1.23 *If $M \Vdash \phi \to \psi$ and $N \Vdash \psi$, then $M\ N \Vdash \psi$.*

Proof By $M \Vdash \phi \to \psi$, then $M \downarrow \lambda x.\ O$ and $O[x := N] \Vdash \psi$. Since $M\ N \to^* (\lambda x.\ O)\ N \to O[x := N]$, Lemma 2.1.22 shows the claim. ■

2.1.4 Normalization of $\lambda^{\to}$

With realizability, we have at our disposal sufficient means to prove normalization of $\lambda^{\to}$. We need the following technical tool to handle possible free variables in lambda terms:

Definition 2.1.24 *An* environment*, denoted by ρ, is a finite partial function from Var to realizers. For a term M, $M[\rho]$ denotes $M[x_1 := \rho(x_1), \ldots, x_n := \rho(x_n)]$. We write $\rho \models \Gamma$ if for all $(x, \phi) \in \Gamma$, $\rho(x) \Vdash \phi$.*

Theorem 2.1.25 *If $\Gamma \vdash M : \Psi$, then for all $\rho \models \Gamma$, $\overline{M}[\rho] \Vdash \Psi$.*

Proof The proof proceeds by induction on the proof $\Gamma \vdash M : \Psi$. To increase readability, we will write M' in the proof to denote $\overline{M}[\rho]$, where M and ρ are clear from the context. Note that by Lemma 2.1.3 and the definition of the erasure map, $\overline{M}[\rho]$ is closed and so $\overline{M}[\rho] \Vdash \Psi$ is defined. Case $\Gamma \vdash M : \Psi$ of:

- $$\frac{}{\Gamma, x : \phi \vdash x : \phi}$$

  Then $M' = \rho(x)$ and the claim follows.

- $$\frac{\Gamma \vdash M : \phi \to \psi \quad \Gamma \vdash N : \phi}{\Gamma \vdash M\ N : \psi}$$

  By the induction hypothesis, $M' \Vdash \phi \to \psi$ and $N' \Vdash \phi$. Lemma 2.1.23 gives the claim.

- $$\frac{\Gamma, x : \phi \vdash M : \psi}{\Gamma \vdash \lambda x : \phi.\ M : \phi \to \psi}$$

  We need to show that for any $N \Vdash \phi$, $M'[x := N] \Vdash \psi$. Take any such N. Let $\rho' = \rho[x := N]$. Then $\rho' \models \Gamma, x : \phi$, so by the induction hypothesis $\overline{M}[\rho'] \Vdash \psi$. As it is easy to see that $\overline{M}[\rho'] = \overline{M}[\rho][x := N] = M'[x := N]$, we get $M'[x := N] \Vdash \psi$.

- $$\frac{\Gamma \vdash M : \bot}{\Gamma \vdash \text{magic}(M) : \phi}$$

  By the induction hypothesis, $M' \Vdash \bot$, which is not the case, so anything holds, in particular $\text{magic}(M') \Vdash \phi$.

- $$\frac{\Gamma \vdash M : \phi \wedge \psi}{\Gamma \vdash \mathrm{fst}(M) : \phi}$$

  By the induction hypothesis, $M' \Vdash \phi \wedge \psi$, so $M' \downarrow \langle M_1, M_2 \rangle$ and $M_1 \Vdash \phi$. Therefore $\mathrm{fst}(M) \to^* \mathrm{fst}(\langle M_1, M_2 \rangle) \to M_1$. Lemma 2.1.22 gives the claim.

- $$\frac{\Gamma \vdash M : \phi \wedge \psi}{\Gamma \vdash \mathrm{snd}(M) : \psi}$$

  Symmetric to the previous case.

- $$\frac{\Gamma \vdash M : \phi \quad \Gamma \vdash N : \psi}{\Gamma \vdash \langle M, N \rangle : \phi \wedge \psi}$$

  All we need to show is $M' \Vdash \phi$ and $N' \Vdash \psi$, which we get from the induction hypothesis.

- $$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash \mathrm{inl}(M) : \phi \vee \psi}$$

  We need to show that $M' \Vdash \phi$, which we get from the induction hypothesis.

- $$\frac{\Gamma \vdash M : \psi}{\Gamma \vdash \mathrm{inr}(M) : \phi \vee \psi}$$

  Symmetric to the previous case.

- $$\frac{\Gamma \vdash M : \phi \vee \psi \quad \Gamma, x : \phi \vdash N : \vartheta \quad \Gamma, x : \psi \vdash O : \vartheta}{\Gamma \vdash \mathrm{case}(M, x : \phi.\ N, x : \psi.\ O) : \vartheta}$$

  By the induction hypothesis, $M' \Vdash \phi \vee \psi$. Therefore either $M' \downarrow \mathrm{inl}(M_1)$ and $M_1 \Vdash \phi$ or $M' \downarrow \mathrm{inr}(M_1)$ and $M_1 \Vdash \psi$. We only treat the former case, the latter is symmetric. Since $\rho[x := M_1] \Vdash \Gamma, x : \phi$, by the induction hypothesis we get $\overline{N}[\rho[x := M_1]] \Vdash \vartheta$. We also have $\mathrm{case}(M', x.N', x.O') \to^* \mathrm{case}(\mathrm{inl}(M_1), x.N', x.O') \to N'[x := M_1]$. It is easy to see that $N'[x := M_1] = \overline{N}[\rho[x := M_1]]$, so Lemma 2.1.22 gives us the claim. ■

Corollary 2.1.26 (Normalization) *If* $\vdash M : \phi$*, then* $M \downarrow$*.*

Proof Take the empty ρ. Then $\rho \models \emptyset$. By Theorem 2.1.25, $\overline{M}[\rho]$ normalizes. By the definition of ρ, $\overline{M}[\rho] = \overline{M}$. By Lemma 2.1.18, M normalizes. ■

Corollary 2.1.27 *IPC is consistent.*

As stated before, we will present some realistic applications of applying normalization to extract computational content from proofs in the next section. Here we present, however, an important theoretical application.

Corollary 2.1.28 (Disjunction Property) *If* $\vdash \phi \vee \psi$*, then either* $\vdash \phi$ *or* $\vdash \psi$*.*

Proof Suppose $\vdash \phi \vee \psi$. Then there is a term M such that $\vdash M : \phi \vee \psi$. Since $\lambda^{\rightarrow}$ normalizes, $M \downarrow v$ and by Lemma 2.1.12 $\vdash v : \phi \vee \psi$. By Canonical Forms, either $v = \mathrm{inl}(N)$ and $\vdash N : \phi$ or $v = \mathrm{inr}(M)$ and $\vdash N : \psi$. By Lemma 2.1.5, in the first case $\vdash \phi$, in the second $\vdash \psi$. ■

This concludes our account of propositional logic. We now move to more sophisticated systems. However, the list of lemmas in all systems we consider in this chapter will always include the lemmas we showed in this section. Moreover, the proofs and systems we consider are modular — proofs of cases in lemmas for $\lambda^{\rightarrow}$ and IPC still work, possibly with minor modifications, in more complicated systems. This will vastly simplify our account, as we will need to show only the new cases in proofs.

2.2 First-order arithmetic

We now extend the system presented in the previous section to the constructive version of the first-order arithmetic, called Heyting Arithmetic (HA).

Heyting Arithmetic is based on the constructive version of the first-order logic, one of the most successful logics in existence. First-order logic is widely used as a basis for theorem provers. Moreover, it is an underlying logic of set theory, the foundation of mathematics.

The constructive version of first-order logic is called intuitionistic first-order logic (IFOL). The propositional variables from IPC are replaced by much more concrete entities in IFOL. The logic allows its user to make statements about arbitrary domains. These domains can have distinguished elements, operations on such elements and their properties. The syntactic counterparts in IFOL are called *constants*, *function symbols* and *relational symbols*, respectively. Their list is called a *signature*; any IFOL theory is parameterized by a signature.

We are interested in the domain of natural numbers, formalized as HA. HA has one constant 0, one unary function symbol S, which intuitively corresponds to the successor function, two binary function symbols $+, *$ and one relational symbol $=$.

For the formal presentation, we first fix a countable set $FVar$ of the *first-order variables*. We will use the letters a, b, c, n, m for the first-order variables. The syntactic counterpart of an element of a domain is called a *term*:

Definition 2.2.1 *The* terms *of HA are generated by the following abstract grammar:*

$$t \quad ::= \quad a \mid 0 \mid S(t) \mid t + t \mid t * t$$

In this section, letters t, s, u will denote exclusively the terms of HA. A term is *closed* if it has no variables. The set of all HA terms will be denoted by Tms and the set of all closed terms Tms_c.

Definition 2.2.2 *We call any term of the form $S(S(S(\ldots(0))))$ a* numeral.

Definition 2.2.3 *The* formulas *of IFOL are generated by the following abstract grammar:*

$$\phi \quad ::= \quad t = t \mid \bot \mid \phi \to \psi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \forall n.\ \phi \mid \exists n.\ \phi$$

The variable n in quantifiers binds its occurrences in respective formulas. The free first-order variables of a formula ϕ are denoted by $FV_F(\phi)$ and defined in a standard way along with substitution. The notation $\vec{a}$ is used for sequences, treated as sets when convenient. The notation $\phi(\vec{a})$ is used in the situation where all free variables of ϕ are among $\vec{a}$. The following standard substitution lemma is proven easily:

Lemma 2.2.4 *For any formula ϕ, $\phi[a := t][b := u[a := t]] = \phi[b := u][a := t]$, for $b \notin FV(t)$.*

Proof Straightforward structural induction on ϕ. ∎

The proof system, similarly to IPC, is used to derive judgments of the form "$\Gamma \vdash \phi$". The contexts are still finite sets of variables. The logic arises by extending the rules of IPC by the following clauses:

$$\frac{\Gamma \vdash \phi}{\Gamma \vdash \forall a.\ \phi}\ a \notin FV_F(\Gamma) \qquad \frac{\Gamma \vdash \forall a.\ \phi}{\Gamma \vdash \phi[a := t]}$$

$$\frac{\Gamma \vdash \phi[a := t]}{\Gamma \vdash \exists a.\ \phi} \qquad \frac{\Gamma \vdash \exists a.\ \phi \quad \Gamma, \phi \vdash \psi}{\Gamma \vdash \psi}\ a \notin FV_F(\Gamma) \cup FV_F(\psi)$$

Finally, as HA is an axiomatic theory, we list its axioms:

- (eqRefl) $\forall n.\ n = n$
- (eqSymm) $\forall n, m.\ n = m \to m = n$
- (eqTrans) $\forall n, m, o.\ n = m \wedge m = o \to n = o$
- (eqS) $\forall n, m.\ n = m \to S(n) = S(m)$

- (P3) $\forall n.\ S(n) = 0 \to \bot$
- (P4) $\forall n, m.\ S(n) = S(m) \to n = m$
- (plusZ) $\forall n.\ n + 0 = 0$
- (plusS) $\forall n, m.\ n + S(m) = S(n + m)$
- (mulZ) $\forall n.\ n * 0 = 0$
- (mulS) $\forall n, m.\ n * S(m) = n * m + m$
- ($\mathrm{ind}_{\phi(n,\vec{a})}$) $\forall \vec{a}.\ \phi(0, \vec{a}) \to (\forall n.\ \phi(n, \vec{a}) \to \phi(S(n), \vec{a})) \to \forall n.\ \phi(n, \vec{a})$.

The last axiom is the induction axiom *schema*. This means that this is actually an abbreviation for an infinite family of axioms — there is one instance of the schema for every possible formula $\phi(n, \vec{a})$. We use the notation $\Gamma \vdash_{HA} \phi$ if ϕ can be derived from Γ and the axioms of HA.

For any closed term t, there is a unique natural number denoted by t, which we denote by $[\![t]\!]$ and define as follows:

- $[\![0]\!] = 0$.
- $[\![S(t)]\!] = [\![t]\!] + 1$.
- $[\![t + u]\!] = [\![t]\!] + [\![u]\!]$.
- $[\![t * u]\!] = [\![t]\!] * [\![u]\!]$.

Definition 2.2.5 *We denote by t_n the unique numeral such that $[\![t]\!] = [\![t_n]\!]$. Similarly, for any natural number m we denote by m_n the unique numeral such that $[\![m_n]\!] = m$.*

Lemma 2.2.6 *For any formula ϕ, closed term t and variable a, $\Gamma \vdash_{HA} \phi[a := t]$ iff $\Gamma \vdash_{HA} \phi[a := t_n]$.*

Proof Easy induction on the proof tree. ∎

$$
\begin{array}{rcl}
t & ::= & a \mid 0 \mid S(t) \mid t + t \mid t * t \\
\phi & ::= & t = t \mid \bot \mid \phi \to \psi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \forall n.\ \phi \mid \exists n.\ \phi
\end{array}
$$

$$
\frac{}{\Gamma, \phi \vdash \phi} \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash \phi} \qquad \frac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \phi \to \psi} \qquad \frac{\Gamma \vdash \phi \to \psi \quad \Gamma \vdash \phi}{\Gamma \vdash \psi}
$$

$$
\frac{\Gamma \vdash \phi}{\Gamma \vdash \phi \vee \psi} \qquad \frac{\Gamma \vdash \psi}{\Gamma \vdash \phi \vee \psi} \qquad \frac{\Gamma \vdash \phi \vee \psi \quad \Gamma, \phi \vdash \vartheta \quad \Gamma, \psi \vdash \vartheta}{\Gamma \vdash \vartheta}
$$

$$
\frac{\Gamma \vdash \phi \quad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi} \qquad \frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \phi} \qquad \frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \psi}
$$

$$
\frac{\Gamma \vdash \phi}{\Gamma \vdash \forall a.\ \phi}\ a \notin FV_F(\Gamma) \qquad \frac{\Gamma \vdash \forall a.\ \phi}{\Gamma \vdash \phi[a := t]}
$$

$$
\frac{\Gamma \vdash \phi[a := t]}{\Gamma \vdash \exists a.\ \phi} \qquad \frac{\Gamma \vdash \exists a.\ \phi \quad \Gamma, \phi \vdash \psi}{\Gamma \vdash \psi}\ a \notin FV_F(\Gamma) \cup FV_F(\psi)
$$

(eqRefl)	$\forall n.\ n = n$	(eqSymm)	$\forall n, m.\ n = m \to m = n$
(eqTrans)	$\forall n, m, o.\ n = m \wedge m = o \to n = o$	(eqS)	$\forall n, m.\ n = m \to S(n) = S(m)$
(P3)	$\forall n.\ S(n) = 0 \to \bot$	(P4)	$\forall n, m.\ S(n) = S(m) \to n = m$
(plusZ)	$\forall n.\ n + 0 = 0$	(plusS)	$\forall n, m.\ n + S(m) = S(n + m)$
(mulZ)	$\forall n.\ n * 0 = 0$	(mulS)	$\forall n, m.\ n * S(m) = n * m + m$

$$
(\mathrm{ind}_{\phi(n,\vec{a})})\ \forall \vec{a}.\ \phi(0, \vec{a}) \to (\forall n.\ \phi(n, \vec{a}) \to \phi(S(n), \vec{a})) \to \forall n.\ \phi(n, \vec{a})
$$

Figure 2.1: Heyting Arithmetic

2.2.1 λH calculus

We now extend $\lambda^{\rightarrow}$ from the previous section to encompass HA. The new calculus will be called λH. Its terms arise by extending the grammar generating terms of $\lambda^{\rightarrow}$ by the following clauses. The first group of new clauses corresponds to the proof rules of the first-order logic. The correspondence will be captured formally by the type system.

$$M \quad ::= \quad \ldots \mid \lambda a.\ M \mid M\ t \mid [t, M] \mid \text{let } [a, x : \phi] := M \text{ in } N$$

The second group corresponds to HA axioms:

$$\begin{aligned} M \quad ::= \quad & \ldots \mid \mathrm{eqReflRep}(t) \mid \mathrm{eqSymmRep}(t, s, M) \mid \mathrm{eqTransRep}(t, s, u, M) \mid \\ & \mathrm{eqSRep}(t, s, M) \mid \mathrm{p3Rep}(t, M) \mid \mathrm{p3Rep}(t, s, M) \mid \\ & \mathrm{plusZRep}(t) \mid \mathrm{plusSRep}(t, s) \mid \mathrm{mulZRep}(t) \mid \\ & \mathrm{mulSRep}(t, s) \mid \mathrm{ind}_{n, \vec{a}.\ \phi(n, \vec{a})}(t, \vec{t}, M) \end{aligned}$$

The Rep suffix in these terms refers to the fact that these terms are *representatives* of the corresponding axioms — given a proof M of $t = s$, $\mathrm{eqSRep}(t, s, M)$ represents a proof of $S(t) = S(s)$. The notation $a, \vec{n}.\ \phi(n, \vec{a})$ denotes a formula ϕ with its variables $a, \vec{n}$ bound. The $\mathrm{ind}_{a, \vec{n}.\ \phi(n, \vec{a})}(t, \vec{t}, M)$ term in the grammar is a *term schema*, describing a family of terms and parameterized by the formula ϕ. There is thus one term for each formula $n, \vec{a}.\ \phi(n, \vec{a})$. In any such term, the number of terms in the sequence $\vec{t}$ is the same as the number of variables in the sequence $\vec{a}$.

A subtle point in the definition of first-order substitution on lambda terms is that it is extended to the formulas parameterizing ind terms:

$$(\mathrm{ind}_{a, \vec{n}.\ \phi(n, \vec{a})}(t, \vec{t}, M))[b := u] \equiv \mathrm{ind}_{a, \vec{n}.\ \phi[b := u](n, \vec{a})}(t[b := u], \overrightarrow{t[b := u]}, M[b := u])$$

This is the reason for the adoption of binders in the parameterizing formula; had it not been for them, the variables $n, \vec{a}$ in the term $\text{ind}_{\phi(n,\vec{a})}$ might be considered free and a subject to substitution, which is not a desired behavior.

The reduction system is expanded by adding the following clauses to the reduction relation.

$$(\lambda a.\ M)\ t \to M[a := t] \qquad \text{let } [a, x : \phi] := [t, M] \text{ in } N \to N[a := t][x := M]$$

$$\text{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}(0, \vec{t}, M) \to \text{fst}(M)$$

$$\text{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}(S(t), \vec{t}, M) \to \text{snd}(M)\ t\ \text{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}(t, \vec{t}, M), \text{where } t \text{ is a numeral.}$$

$$t \to t_n, \text{for any closed } t \text{ that is not a numeral.}$$

The new evaluation contexts are:

$$[\circ] \quad ::= \quad \ldots \mid [\circ]\ t \mid \text{let } [a, x : \phi] := [\circ] \text{ in } N \mid \text{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}([\circ], \vec{t}, M)$$

The new values are $\lambda a.\ M$, $[t, M]$ and all Rep terms.

A new feature in λH is a *second* reduction relation, defined on formulas and denoted by $\to_\omega$. Intuitively, we will consider formulas with closed terms denoting the same natural numbers as being the same. This is because the $\text{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}(t, \vec{t}, M)$ term can only reduce if t is a numeral. For this reason we have a rule $t \to t'$ along with the evaluation context $\text{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}([\circ], \vec{t}, M)$, in order to force the first argument of the ind term to reduce to the corresponding numeral. If we want to be able to prove Subject Reduction, these reductions must have their counterpart in the logic. This is the reason for the new relation $\to_\omega$, which is formally defined as follows:

$$\phi[a := t] \to_\omega \phi[a := t_n], \text{for any } \phi, a \text{ and closed term } t.$$

Definition 2.2.7 *We write $\phi \leftrightarrow_\omega \psi$ if either $\phi \to_\omega \psi$ or $\psi \to_\omega \phi$. We will denote by $=_\omega$ the smallest equivalence relation extending $\to_\omega$.*

We now move on to define the type system for λH. It arises by extending the system for $\lambda^{\to}$ presented in the previous system. The contexts are still finite sets of pairs (x, ϕ). The new rules corresponding to the first-order logic are:

$$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash \lambda a.\ M : \forall a.\ \phi}\ a \notin FV_F(\Gamma) \qquad \frac{\Gamma \vdash M : \forall a.\ \phi}{\Gamma \vdash M\ t : \phi[a := t]}$$

$$\frac{\Gamma \vdash M : \phi[a := t]}{\Gamma \vdash [t, M] : \exists a.\ \phi} \qquad \frac{\Gamma \vdash M : \exists a.\ \phi \quad \Gamma, x : \phi \vdash N : \psi}{\Gamma \vdash \text{let } [a, x : \phi] := M \text{ in } N : \psi}\ a \notin FV_F(\Gamma, \psi)$$

The rules corresponding to HA axioms are:

$$\frac{}{\Gamma \vdash \text{eqReflRep}(t) : t = t} \qquad \frac{\Gamma \vdash M : s = t}{\Gamma \vdash \text{eqSymmRep}(t, s, M) : t = s}$$

$$\frac{\Gamma \vdash M : t = s \wedge s = u}{\Gamma \vdash \text{eqTransRep}(t, s, u, M) : t = u}$$

$$\frac{\Gamma \vdash M : t = s}{\Gamma \vdash \text{eqSRep}(t, s, M) : S(t) = S(s)}$$

$$\frac{\Gamma \vdash M : S(t) = 0}{\Gamma \vdash \text{p3Rep}(t, M) : \bot} \qquad \frac{\Gamma \vdash M : S(t) = S(s)}{\Gamma \vdash \text{p4Rep}(t, s, M) : t = s}$$

$$\frac{}{\Gamma \vdash \text{plusZRep}(t) : t + 0 = 0} \qquad \frac{}{\Gamma \vdash \text{plusSRep}(t, s) : t + S(s) = S(t + s)}$$

$$\frac{}{\Gamma \vdash \text{mulZRep}(t) : t * 0 = 0} \qquad \frac{}{\Gamma \vdash \text{mulSRep}(t, s) : t * (S(s)) = t * s + t}$$

$$\frac{\Gamma \vdash M : \phi(0, \vec{t}) \wedge \forall n.\ \phi(n, \vec{t}) \to \phi(S(n), \vec{t})}{\Gamma \vdash \text{ind}_{n, \vec{a}.\ \phi(n, \vec{a})}(t, \vec{t}, M) : \phi(t, \vec{t})}$$

In addition, there is a rule corresponding to Lemma 2.2.6:

$$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash M : \psi}\ \phi \leftrightarrow_\omega \psi$$

Definition 2.2.8 *We call the last proof rule* inessential *and all other proof rules* essential*.*

Lemma 2.2.9 *If $\phi =_\omega \psi$, then $\Gamma \vdash M : \phi$ iff $\Gamma \vdash M : \psi$.*

Proof Straightforward induction on the definition of $=_\omega$. ■

In order to smoothen further presentation, we remark that all rules involving Rep terms are of the same form:

$$\frac{\Gamma \vdash M : \phi_A(\vec{t})}{\Gamma \vdash \mathrm{axRep}(\vec{t}, M) : \psi_A(\vec{t})} \ ,$$

for appropriate number of terms t and formulas ϕ_A and ψ_A. The term M and the assumptions might be not present. For example, for ax $\equiv$ eqReflRep we have $\vec{t} \equiv t$, M is not present and $\psi_A \equiv t = t$. For ax $\equiv$ eqSRep, $\vec{t} \equiv t, s$, $\phi_A \equiv t = s$ and $\psi_A \equiv S(t) = S(s)$. We will therefore use meta-level schemas using axRep, ϕ_A and ψ_A to talk about all these terms and rules at once. For example, using this convention, we can specify the typing rules corresponding to HA axioms apart from the induction axiom concisely as:

$$\frac{\Gamma \vdash M : \phi_A(\vec{t})}{\Gamma \vdash \mathrm{axRep}(\vec{t}, M) : \psi_A(\vec{t})}$$

The following Lemma shows that λH calculus is a faithful representation of HA.

Lemma 2.2.10 (Curry-Howard isomorphism) *If* $\Gamma \vdash O : \Psi$ *then* $rg(\Gamma) \vdash_{HA} \Psi$. *If* $\Gamma \vdash_{HA} \Psi$, *then there exists a term* M *such that* $\overline{\Gamma} \vdash M : \Psi$, *where* $\overline{\Gamma} = \{(x_\phi, \phi) \mid \phi \in \Gamma\}$.

Proof The first part is obvious — for the new typing rules use the corresponding HA axioms and Lemma 2.2.6 to derive the formulas. For the second part, we proceed by induction on the proof tree just as in case of $\lambda^\to$. The proofs of the cases corresponding to the rules of $\lambda^\to$ are the same as before. We show the new cases in the proof. Case the last rule in the proof $\Gamma \vdash_{HA} \Psi$ of:

- $$\frac{\Gamma \vdash \exists a.\ \phi \quad \Gamma, \phi \vdash \Psi}{\Gamma \vdash \Psi}\ a \notin FV_F(\Gamma) \cup FV_F(\psi)$$

By the induction hypothesis we get terms M, N such that $\overline{\Gamma} \vdash M : \exists a.\ \phi$ and $\overline{\Gamma}, x_\phi : \phi \vdash N : \Psi$. The following proof tree shows the claim:

$$\frac{\overline{\Gamma} \vdash M : \exists a.\ \phi \quad \overline{\Gamma}, x_\phi : \phi \vdash N : \Psi}{\overline{\Gamma} \vdash \text{let } [a, x_\phi : \phi] := M \text{ in } N : \Psi}$$

- $$\frac{\Gamma \vdash \phi}{\Gamma \vdash \forall a.\ \phi}\ a \notin FV_F(\Gamma)$$

  By the induction hypothesis we get a lambda term M such that $\overline{\Gamma} \vdash M : \phi$. Therefore $\overline{\Gamma} \vdash \lambda a.\ M : \forall a.\ \phi$.

- $$\frac{\Gamma \vdash \forall a.\ \phi}{\Gamma \vdash \phi[a := t]}$$

  By the induction hypothesis we get a lambda term M such that $\overline{\Gamma} \vdash M : \forall a.\ \phi$. Therefore $\overline{\Gamma} \vdash M\ t : \phi[a := t]$.

- $$\frac{\Gamma \vdash \phi[a := t]}{\Gamma \vdash \exists a.\ \phi}$$

  By the induction hypothesis we get a lambda term M such that $\overline{\Gamma} \vdash M : \phi[a := t]$. Therefore $\overline{\Gamma} \vdash [t, M] : \phi[a := t]$.

- One of the axioms "ax″ of HA. Then $\Psi = \forall \vec{n}.\ \phi_A(\vec{n}) \to \psi_A(\vec{n})$. The following proof tree shows the claim. To increase readability, we compress several steps introducing the universal quantifier into one:

$$\dfrac{\dfrac{\dfrac{x : \phi_A(\vec{n}) \vdash x : \phi_A(\vec{n})}{x : \phi_A(\vec{n}) \vdash \text{axRep}(\vec{n}, x) : \psi_A(\vec{n})}}{\vdash \lambda x : \phi_A(\vec{n}).\ \text{axRep}(\vec{n}, x) : \phi_A(\vec{n}) \to \psi_A(\vec{n})}}{\vdash \lambda \vec{n}.\ \lambda x : \phi_A(\vec{n}).\ \text{axRep}(\vec{n}, x) : \forall \vec{n}.\ \phi_A(\vec{n}) \to \psi_A(\vec{n})}$$

- The induction axiom. Then $\Psi \equiv \forall \vec{a}.\ \phi(0, \vec{a}) \wedge \forall n.\ \phi(n, \vec{a}) \to \phi(S(n), \vec{a}) \to$

$\forall n.\ \phi(n, \vec{a})$. The following proof tree shows the claim.

$$\dfrac{\dfrac{\dfrac{\overline{x : \phi(0, \vec{a}) \wedge \forall n.\ \phi(n, \vec{a}) \rightarrow \phi(S(n), \vec{a}) \vdash x : \phi(0, \vec{a}) \wedge \forall n.\ \phi(n, \vec{a}) \rightarrow \phi(S(n), \vec{a})}}{x : \phi(0, \vec{a}) \wedge \forall n.\ \phi(n, \vec{a}) \rightarrow \phi(S(n), \vec{a}) \vdash \mathrm{ind}_{n, \vec{a}.\ \phi(n, \vec{a})}(n, \vec{a}, x) : \phi(n, \vec{a})}}{x : \phi(0, \vec{a}) \wedge \forall n.\ \phi(n, \vec{a}) \rightarrow \phi(S(n), \vec{a}) \vdash \lambda n.\ \mathrm{ind}_{n, \vec{a}.\ \phi(n, \vec{a})}(n, \vec{a}, x) : \forall n.\ \phi(n, \vec{a})}}{\vdash \lambda \vec{a} \lambda x : \phi(0, \vec{a}) \wedge \forall n.\ \phi(n, \vec{a}) \rightarrow \phi(S(n), \vec{a}) \lambda n.\ \mathrm{ind}_{n, \vec{a}.\ \phi(n, \vec{a})}(n, \vec{a}, x) : \Psi}$$

■

2.2.2 Properties of λH

The lemmas proved for $\lambda^{\rightarrow}$ extend in a natural way to encompass λH.

Lemma 2.2.11 (Weakening) *If $\Gamma \vdash Q : \Psi$ and y and $FV_F(\psi)$ are fresh to the proof tree $\Gamma \vdash Q : \Psi$, then $\Gamma, y : \psi \vdash Q : \Psi$.*

Proof By induction on $\Gamma \vdash Q : \Psi$. We show the new cases in the proof. Case $\Gamma \vdash Q : \Psi$ of:

- $$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash \lambda a.\ M : \forall a.\ \phi}\ a \notin FV_F(\Gamma)$$

  By the induction hypothesis, $\Gamma, y : \psi \vdash M : \phi$. By y and $FV_F(\psi)$ being fresh to the proof tree, $a \notin FV_F(\Gamma, y : \psi)$. Therefore, $\Gamma, y : \psi \vdash \lambda a.\ M : \forall a.\ \phi$.

- $$\frac{\Gamma \vdash M : \forall a.\ \phi}{\Gamma \vdash M\ t : \phi[a := t]}$$

  Straightforward.

- $$\frac{\Gamma \vdash M : \phi[a := t]}{\Gamma \vdash [t, M] : \exists a.\ \phi}$$

  Straightforward.

- $$\frac{\Gamma \vdash M : \exists a.\, \phi \quad \Gamma, x : \phi \vdash N : \psi}{\Gamma \vdash \text{let } [a, x : \phi] := M \text{ in } N : \psi}\; a \notin FV(\Gamma, \psi)$$

  Similar to the first case in the proof.

- $$\frac{\Gamma \vdash M : \phi_A(\vec{t})}{\Gamma \vdash \text{axRep}(\vec{t}, M) : \psi_A(\vec{t})}$$

  Straightforward.

- $$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash M : \psi}\; \phi \leftrightarrow_\omega \psi$$

  Straightforward. ■

The "propositional" substitution lemma is proved in exactly the same way as its counterpart in IPC. As the new typing rules do not interact with propositional substitution, the proof for the corresponding cases follows by a straightforward application of the induction hypothesis.

Lemma 2.2.12 *If $\Gamma, x : \phi \vdash M : \psi$ and $\Gamma \vdash N : \phi$, then $\Gamma \vdash M[x := N] : \psi$.*

Since λH introduces first-order binders, a new, first-order substitution lemma is necessary. Similarly to Lemma 2.2.12, it is used only in the proof of Subject Reduction.

Lemma 2.2.13 *If $\Gamma \vdash Q : \Psi$, then for any first-order variable b and term u, $\Gamma[b := u] \vdash Q[b := u] : \Psi[b := u]$.*

Proof By induction on the proof tree $\Gamma \vdash Q : \Psi$. Most of the rules do not interact with first-order substitution, so we show the proof only for the four of them that do. Case $\Gamma \vdash Q : \Psi$ of:

- $$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash \lambda a.\ M : \forall a.\ \phi}\ a \notin FV_F(\Gamma)$$

  Without loss of generality we can assume that $a \notin FV_F(u) \cup \{b\}$. By the induction hypothesis, $\Gamma[b := u] \vdash M[b := u] : \phi[b := u]$. Therefore $\Gamma[b := u] \vdash \lambda a.\ M_1[b := u] : \forall a.\ \phi[b := u]$ and by the choice of a, $\Gamma[b := u] \vdash (\lambda a.\ M)[b := u] \vdash (\forall a.\ \phi)[b := u]$.

- $$\frac{\Gamma \vdash M : \forall a.\ \phi}{\Gamma \vdash M\ t : \phi[a := t]}$$

  Choosing a to be fresh, by the induction hypothesis we get $\Gamma[b := u] \vdash M[b := u] : \forall a.\ (\phi[b := u])$, so $\Gamma[b := u] \vdash M[b := u]\ t[b := u] : \phi[b := u][a := t[b := u]]$. By Lemma 2.2.4 and $a \notin FV(t)$, we get $\Gamma[b := u] \vdash (M\ t)[b := u] : \phi[a := t][b := u]$.

- $$\frac{\Gamma \vdash M : \phi(0, \vec{t}) \land \forall n.\ \phi(n, \vec{t}) \to \phi(S(n), \vec{t})}{\Gamma \vdash \mathrm{ind}_{n, \vec{a}.\ \phi(n, \vec{a})}(t, \vec{t}, M) : \phi(t, \vec{t})}$$

  By the induction hypothesis, $\Gamma[b := u] \vdash M[b := u] : \phi[b := u](0, \overrightarrow{t[b := u]}) \land \forall n.\ \phi[b := u](n, \overrightarrow{t[b := u]}) \to \phi[b := u](S(n), \overrightarrow{t[b := u]})$. Therefore $\Gamma[b := u] \vdash \mathrm{ind}_{n, \vec{a}.\ \phi[b:=u](n, \vec{a})}(t[b := u], \overrightarrow{t[b := u]}, M[b := u]) : \phi[b := u](t[b := u], \overrightarrow{t[b := u]})$. The claim follows.

- $$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash M : \psi}\ \phi \leftrightarrow_\omega \psi$$

  To fix our attention, assume $\phi \to_\omega \psi$. Therefore for some ϕ_1 and a fresh variable a, we have:
  $$\frac{\Gamma \vdash M : \phi_1[a := t]}{\Gamma \vdash M : \phi_1[a := t_n]}$$

  By the induction hypothesis, $\Gamma[b := u] \vdash M[b := u] : \phi_1[a := t][b := u]$. As t is closed and a is fresh, $\Gamma[b := u] \vdash M[b := u] : \phi_1[b := u][a := t]$, so $\Gamma[b := u] \vdash M[b := u] : \phi_1[b := u][a := t_n]$. As t_n is closed as well, $\Gamma[b := u] \vdash M[b := u] : \phi_1[a := t_n][b := u]$, which shows the claim. ■

Taking into account the relation $\leftrightarrow_\omega$, the lambda terms still determine formulas:

Lemma 2.2.14 (Inversion) *Suppose $\Gamma \vdash Q : \Psi$. Suppose Q is of the form:*

- $M\ N$. *Then there is ϕ such that $\Gamma \vdash M : \phi \to \Psi_1$, $\Gamma \vdash N : \phi$ and $\Psi_1 =_\omega \Psi$.*
- $\lambda x : \phi.\ M$. *Then for some ϕ, ψ, $\Psi =_\omega \phi \to \psi$ and $\Gamma, x : \phi \vdash M : \psi$.*
- $\mathrm{inl}(M)$. *Then for some ϕ, ψ, $\Psi =_\omega \phi \vee \psi$ and $\Gamma \vdash M : \phi$.*
- $\mathrm{inr}(M)$. *Then for some ϕ, ψ, $\Psi =_\omega \phi \vee \psi$ and $\Gamma \vdash M : \psi$.*
- $\mathrm{case}(M, x : \phi.N, x : \psi.O)$. *Then $\Gamma \vdash M : \phi \vee \psi$, $\Gamma, x : \phi \vdash N : \Psi_1, \Gamma, x : \psi \vdash O : \Psi_1$ and $\Psi_1 =_\omega \Psi$.*
- $\mathrm{fst}(M)$. *Then for some ψ, $\Gamma \vdash M : \Psi_1 \wedge \psi$ and $\Psi_1 =_\omega \Psi$.*
- $\mathrm{snd}(M)$. *Then for some ϕ, $\Gamma \vdash M : \phi \wedge \Psi_1$ and $\Psi_1 =_\omega \Psi$.*
- $\langle M, N \rangle$. *Then for some ϕ, ψ, $\Psi =_\omega \phi \wedge \psi$, $\Gamma \vdash M : \phi$ and $\Gamma \vdash N : \psi$.*
- $\mathrm{magic}(M)$. *Then $\Gamma \vdash M : \bot$.*
- $\lambda a.\ M$. *Then $\Psi =_\omega \forall a.\ \phi$, $a \notin FV(\Gamma)$ and $\Gamma \vdash M : \phi$.*
- $M\ t$. *Then for some term t and a formula ϕ, $\Psi =_\omega \phi[a := t]$ and $\Gamma \vdash M : \forall a.\ \phi$.*
- $[t, M]$. *Then $\Psi =_\omega \exists a.\ \phi$ and $\Gamma \vdash M : \phi[a := t]$.*
- $\mathrm{let}\ [a, x : \phi] := M\ \mathrm{in}\ N$. *Then $a \notin FV_F(\Gamma, \psi), \Gamma \vdash M : \exists a.\ \phi$, $\Gamma, x : \phi \vdash N : \Psi_1$ and $\Psi_1 =_\omega \Psi$.*
- $\mathrm{axRep}(\vec{t}, M)$. *Then $\Psi =_\omega \psi_A(\vec{t})$ and $\Gamma \vdash M : \phi_A(\vec{t})$.*
- $\mathrm{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}(t, \vec{t}, M)$. *Then $\Psi =_\omega \phi(t, \vec{t})$, $\Gamma \vdash M : \phi(0, \vec{t}) \wedge \forall n.\ \phi(n, \vec{t}) \to \phi(S(n), \vec{t})$.*

Moreover, in all the cases, the proof trees in the conclusion are subtrees of $\Gamma \vdash Q : \Psi$.

Proof The proof tree ends with (possibly zero) applications of the inessential proof rule, preceded by an essential rule. An inspection of the latter shows the claim. ■

The new formulas still determine values:

Lemma 2.2.15 (Canonical Forms) *Suppose M is a value, $\vdash M : \Psi$ and Ψ is of the form:*

- $\phi \to \psi$. *Then $M = \lambda x : \phi_1.\ N$ and $x : \phi_1 \vdash N : \psi_1$, where $\phi \to \psi =_\omega \phi_1 \to \psi_1$.*
- $\phi \vee \psi$. *Then either $M = \mathrm{inl}(N)$ and $\vdash N : \phi_1$ or $M = \mathrm{inr}(N)$ and $\vdash N : \psi_1$, where $\phi_1 =_\omega \phi$ and $\psi_1 =_\omega \psi$.*
- $\phi \wedge \psi$. *Then $M = \langle N, O \rangle$, $\vdash N : \phi_1$ and $\vdash O : \psi_1$, where $\phi =_\omega \phi_1$ and $\psi =_\omega \psi_1$.*
- $\bot$. *This never happens.*
- $\forall a.\ \phi$. *Then $M = \lambda a.\ N$ and $\vdash N : \phi_1$, where $\phi_1 =_\omega \phi$.*
- $\exists a.\ \phi$. *Then $M = [t, N]$ and $\vdash N : \phi_1[a := t]$, where $\phi_1 =_\omega \phi$.*
- $t = u$. *Then M is one of* axRep *terms.*

Proof Straightforward. ■

Lemma 2.2.16 (Subject Reduction, Preservation) *If $\Gamma \vdash P : \Psi$ and $P \to Q$, then $\Gamma \vdash Q : \Psi$.*

Proof By induction on the definition of $P \to Q$. Case $P \to Q$ of:

- $(\lambda x : \phi.\ M)\ N \to M[x := N]$. By Inversion, there is a ϕ_1 such that $\Gamma \vdash \lambda x : \phi.\ M : \phi_1 \to \Psi_1$, $\Gamma \vdash N : \phi_1$ and $\Psi_1 =_\omega \Psi$. By Inversion again, $\Gamma, x : \phi \vdash M : \Psi_2$, $\phi_1 =_\omega \phi$ and $\Psi_2 =_\omega \Psi_1$. Therefore $\Gamma \vdash N : \phi$, so by Lemma 2.2.12, $\Gamma \vdash M[x := N] : \Psi_2$. Since $\Psi_2 =_\omega \Psi$, by Lemma 2.2.9 $\Gamma \vdash M[x := N] : \Psi$.

- $\mathrm{case}(\mathrm{inl}(M), x : \phi.\ N, x : \psi.\ O) \to N[x := M]$. By Inversion, $\Gamma \vdash \mathrm{inl}(M) : \phi \vee \psi$, $\Gamma, x : \phi \vdash N : \Psi_1$ and $\Psi_1 =_\omega \Psi$. By Inversion again, $\Gamma \vdash M : \phi_1$ and $\phi_1 =_\omega \phi$, so also $\Gamma \vdash M : \phi$. By Lemma 2.2.12, $\Gamma \vdash M[x := N] : \Psi_1$, so by Lemma 2.2.9 also $\Gamma \vdash M[x := N] : \Psi$.

- $\mathrm{case}(\mathrm{inr}(M), x : \phi.\ N, x : \psi.\ O) \to O[x := M]$. Symmetric to the previous case.

- $\mathrm{fst}(\langle M, N \rangle) \to M$. By Inversion, $\Gamma \vdash \langle M, N \rangle : \Psi_1 \wedge \psi$ for some ψ and $\Psi_1 =_\omega \Psi$. By Inversion again, $\Gamma \vdash M : \Psi_2$ and $\Psi_2 =_\omega \Psi_1$, so also $\Gamma \vdash M : \Psi$.

- $\mathrm{snd}(\langle M, N \rangle) \to N$. Symmetric to the previous case.

- $(\lambda a.\ M)\ t \to M[a := t]$. By Inversion, $\Gamma \vdash \lambda a.\ M : \forall a.\ \phi$ and $\Psi =_\omega \phi[a := t]$. By Inversion again, $\Gamma \vdash M : \phi_1$, $\phi_1 =_\omega \phi$ and $a \notin FV_F(\Gamma)$, so $\Gamma[a := t] = \Gamma$. By Lemma 2.2.13, $\Gamma \vdash M[a := t] : \phi_1[a := t]$. As it is easy to see that $\phi[a := t] =_\omega \phi_1[a := t]$, the claim follows.

- $\mathrm{let}\ [a, x : \phi] := [t, M]\ \mathrm{in}\ N \to N[a := t][x := M]$. Choose a to be fresh, so in particular $M[a := t] = M$. By Inversion, $a \notin FV(\Gamma, \Psi)$, $\Gamma \vdash [t, M] : \exists a.\ \phi$, $\Gamma, x : \phi \vdash N : \Psi_1$, $\Gamma[a := t] = \Gamma$ and $\Psi_1 =_\omega \Psi$. By Inversion again, $\Gamma \vdash M : \phi_1[a := t]$ and $\phi_1 =_\omega \phi$. Therefore, $\Gamma \vdash M : \phi[a := t]$. As $a \notin FV(\Psi)$, also $a \notin FV(\Psi_1)$, so $\Psi_1[a := t] = \Psi_1$. By Lemma 2.2.13 we get $\Gamma[a := t], x : \phi[a := t] \vdash N[a := t] : \Psi_1[a := t]$, so also $\Gamma, x : \phi[a := t] \vdash N[a := t] : \Psi_1$. By Lemma 2.2.12, we get $\Gamma \vdash N[a := t][x := M] : \Psi_1$, so also $\Gamma \vdash N[a := t][x := M] : \Psi$.

- $\mathrm{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}(t,\vec{t},M) \to \mathrm{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}(t_n,\vec{t},M)$. This reduction rule is the reason for the inessential proof rule in the typing system for λH. By Inversion, $\Gamma \vdash M : \phi(0,\vec{t}) \wedge \forall n.\ \phi(n,\vec{t}) \to \phi(S(n),\vec{t})$ and $\Psi =_\omega \phi(t,\overline{t})$. Thus also $\Gamma \vdash \mathrm{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}(t_n,\vec{t},M) : \phi(t_n,\overline{t})$.

- $\mathrm{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}(0,\vec{t},M) \to \mathrm{fst}(M)$. Then $\Psi =_\omega \phi(0,\vec{t})$. By Inversion, $\Gamma \vdash M : \phi(0,\vec{t}) \wedge \forall n.\ \phi(n,\vec{t}) \to \phi(S(n),\vec{t})$. Therefore, $\Gamma \vdash \mathrm{fst}(M) : \phi(0,\vec{t})$ and consequently $\Gamma \vdash \mathrm{fst}(M) : \Psi$.

- $\mathrm{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}(S(t),\vec{t},M) \to \mathrm{snd}(M)\, t\, \mathrm{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}(t,\vec{t},M)$. Then $\Psi =_\omega \phi(S(t),\vec{t})$. By Inversion, $\Gamma \vdash M : \phi(0) \wedge \forall n.\ \phi(n,\vec{t}) \to \phi(S(n),\vec{t})$. The following proof tree shows the claim.

$$\dfrac{\dfrac{\Gamma \vdash \mathrm{snd}(M) : \forall n.\ \phi(n,\vec{t}) \to \phi(S(n),\vec{t})}{\Gamma \vdash \mathrm{snd}(M)\ t : \phi(t,\vec{t}) \to \phi(S(t),\vec{t})} \quad \dfrac{\Gamma \vdash M : \phi(0) \wedge \forall n.\ \phi(n,\vec{t}) \to \phi(S(n),\vec{t})}{\Gamma \vdash \mathrm{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}(t,\vec{t},M) : \phi(t,\vec{t})}}{\dfrac{\Gamma \vdash \mathrm{snd}(M)\ t\ \mathrm{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}(t,\vec{t},M) : \phi(S(t),\vec{t})}{\Gamma \vdash \mathrm{snd}(M)\ t\ \mathrm{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}(t,\vec{t},M) : \Psi}}$$

- The induction steps are easy. ∎

Lemma 2.2.17 (Progress) *If $\vdash P : \Psi$ and $FV_F(P) = \emptyset$, then either P is a value or there is Q such that $P \to Q$ and $FV_F(Q) = \emptyset$.*

Proof Induction on the length of P. We show the cases for the new terms. In all cases, the claim about $FV_F(Q)$ is trivial to verify. Case P of:

- $\lambda a.\ M, [t, M]$. These are values.

- $M\ t$. By Inversion and Canonical Forms, $\Psi =_\omega \phi[a := t]$ and $\vdash M : \forall a.\ \phi$. By the induction hypothesis, either $M = \lambda a.\ N$ in which case $P \to N[a := t]$, or $M \to M'$ and also $M\ t \to M'\ t$.

- let $[a, x : \phi] := M$ in N. By Inversion, $\vdash M : \exists a.\ \phi$. By the induction hypothesis and Canonical Forms, either $M = [t, O]$ in which case $P \to$

$N[a := t][x := O]$, or $M \to M'$ when also let $[a, x : \phi] := M$ in $N \to$ let $[a, x : \phi] := M'$ in N.

- $\mathrm{axRep}(\vec{t}, M)$. These terms are values.
- $\mathrm{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}(t, \vec{t}, M)$. Since $FV_F(P) = \emptyset$, t is closed. If t is a numeral, the term immediately reduces. If not, then $t \to t_n$ and $\mathrm{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}(t, \vec{t}, M) \to \mathrm{ind}_{n,\vec{a}.\ \phi(n,\vec{a})}(t_n, \vec{t}, M)$. ■

By composing Subject Reduction and Preservation as before, we get:

Corollary 2.2.18 *If $\vdash M : \phi$, M is closed and $M \downarrow v$, then $\vdash v : \phi$ and v is a value.*

Corollary 2.2.19 *If $\vdash M : \bot$ and M is closed, then M does not normalize.*

2.2.3 Realizability for HA

We will extend realizability for IPC to encompass HA. For this purpose, we will first present a BHK interpretation for the first-order arithmetic:

- The construction of an atomic formula $t = s$ exists only if if the numbers corresponding to the terms are equal.
- There is no construction of $\bot$.
- The construction of a conjunction $\phi \wedge \psi$ is a pair consisting of a construction of ϕ and a construction of ψ.
- The construction of a disjunction $\phi \vee \psi$ is either a construction of ϕ or a construction of ψ.
- The construction of an implication $\phi \to \psi$ is a method, which transforms every construction of ϕ to a construction of ψ.

- The construction of an existential $\exists n.\ \phi$ is a pair consisting of a term t and a construction of $\phi[n := t]$.
- The construction of $\forall n.\ \phi$ is a method transforming any term t into $\phi[a := t]$.

We proceed to implement the new BHK interpretation via realizability. We first extend the erasure map so that it maps λH to $\overline{\lambda H}$, by adding the following clauses to its definition:

$$\overline{[t, M]} \equiv [t, \overline{M}] \qquad \overline{\text{let } [a, y : \phi] := M \text{ in } N} \equiv \text{let } [a, y] := \overline{M} \text{ in } \overline{N}$$

$$\overline{M\ t} \equiv \overline{M}\ t \qquad \overline{\lambda a.\ M} \equiv \lambda a.\ \overline{M}$$

$$\overline{\text{axRep}(\vec{t}, M)} \equiv \text{axRep}(\vec{t}, \overline{M}) \qquad \overline{\text{ind}_{n,\vec{a}.\ \phi(n,\vec{a}))}(t, \vec{t}, M)} \equiv \text{ind}(t, \vec{t}, \overline{M})$$

It is easy to see that the main lemma still holds:

Lemma 2.2.20 *If $\overline{M} \downarrow$ then $M \downarrow$.*

IFOL expands IPC by quantifiers, terms and equality relational symbols. The realizability relation has to account for these new features. We define realizability only on closed formulas. The definition of realizers stays unchanged:

Definition 2.2.21 *A* realizer *is a closed term of $\lambda\overline{D}$.*

Definition 2.2.22 *The realizability relation $M \Vdash \phi$ between realizers M and closed formulas ϕ is defined by the following clauses:*

$$\begin{array}{lcl}
M \Vdash t = s & \equiv & M \downarrow \wedge [\![t]\!] = [\![s]\!] \\
M \Vdash \bot & \equiv & \bot \\
M \Vdash \phi \wedge \psi & \equiv & M \downarrow \langle M_1, M_2 \rangle \wedge (M_1 \Vdash \phi) \wedge (M_2 \Vdash \psi) \\
M \Vdash \phi \vee \psi & \equiv & (M \downarrow \text{inl}(M_1) \wedge M_1 \Vdash \phi) \vee (M \downarrow \text{inr}(M_1) \wedge M_1 \Vdash \psi) \\
M \Vdash \phi \to \psi & \equiv & (M \downarrow \lambda x.\ M_1) \wedge \forall N.\ (N \Vdash \phi) \to (M_1[x := N] \Vdash \psi) \\
M \Vdash \exists a.\ \phi & \equiv & M \downarrow [t, N] \wedge N \Vdash \phi[a := t] \\
M \Vdash \forall a.\ \phi & \equiv & M \downarrow \lambda a.\ N \wedge \forall t \in Tms_c.\ N[a := t] \Vdash \phi[a := t]
\end{array}$$

It is easy to see that all the properties of realizability proved for $\lambda^{\rightarrow}$ hold for λH as well:

Lemma 2.2.23 *If $M \Vdash \phi$, then $M \downarrow$.*

Lemma 2.2.24 *If $M \to^* N$, then $M \Vdash \phi$ iff $N \Vdash \phi$.*

Lemma 2.2.25 *If $M \Vdash \phi \to \psi$ and $N \Vdash \psi$, then $M\ N \Vdash \psi$.*

One new property is necessary in order to treat applications of the inessential proof rule:

Lemma 2.2.26 *If $\phi \leftrightarrow_\omega \phi'$ then $M \Vdash_\rho \phi$ iff $M \Vdash_\rho \phi'$.*

Proof Straightforward induction on the definition of realizability, using the fact that for any term t, $[\![t]\!] = [\![t_n]\!]$. ■

2.2.4 Normalization of λH

The normalization proof proceeds mostly unchanged. The definition of an environment must be updated to account for the free first-order variables in the sequent.

Definition 2.2.27 *An* environment*, denoted by ρ, is a finite partial function from Var to $\overline{\lambda H}$ and from $FVar$ to closed HA terms. For a term M, $M[\rho]$ denotes $M[x_1 := \rho(x_1), \ldots, x_n := \rho(x_n), a_1 := \rho(a_1), \ldots, a_m := \rho(a_m)]$. Similarly, $\phi[\rho]$ denotes $\phi[a_1 := \rho(a_1), \ldots, a_m := \rho(a_m)]$. We write $\rho \models \Gamma \vdash M : \phi$ if for all $a \in FV_F(\Gamma, M, \phi)$, $\rho(a)$ is defined and for all $(x, \phi) \in \Gamma$, $\rho(x) \Vdash \phi[\rho]$.*

Theorem 2.2.28 (Normalization) *If $\Gamma \vdash Q : \Psi$, then for all $\rho \models \Gamma \vdash Q : \Psi$, $\overline{M}[\rho] \Vdash \Psi[\rho]$.*

Proof For any λH term M, M' in the proof denotes $\overline{M}[\rho]$ and ϕ' denotes $\phi[\rho]$. We proceed by induction on the proof tree $\Gamma \vdash Q : \Psi$. As the cases corresponding to $\lambda^{\rightarrow}$ do not interact with first-order substitutions, the proofs remain unchanged from Theorem 2.1.25. We therefore show only the new cases. Case $\Gamma \vdash Q : \Psi$ of:

- $$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash \lambda a.\ M : \forall a.\ \phi}$$

  By the induction hypothesis, for all $\rho \models \Gamma \vdash M : \phi$, $\overline{M}[\rho] \Vdash \phi[\rho]$. We need to show that for all $\rho \models \Gamma \vdash \lambda a.\ M : \forall a.\ \phi$, $\overline{(\lambda a.\ M)}[\rho] \Vdash (\forall a.\ \phi)[\rho]$. Take any such ρ. It suffices to show $\lambda a.\ \overline{M}[\rho] \Vdash \forall a.\ \phi[\rho]$, which is equivalent to $\forall t \in Tms_c.\ \overline{M}[\rho][a := t] \Vdash \phi[\rho][a := t]$. Take any $t \in Tms_c$. We know that $\rho[a := t] \models \Gamma \vdash M : \phi$, $\overline{M}[\rho][a := t] = \overline{M}[\rho[a := t]]$ and $\phi[\rho][a := t] = \phi[\rho[a := t]]$. Thus, by the induction hypothesis $\overline{M}[\rho][a := t] \Vdash \phi[\rho][a := t]$ which is what we want.

- $$\frac{\Gamma \vdash M : \forall a.\ \phi}{\Gamma \vdash M\ t : \phi[a := t]}$$

  Take any $\rho \models \Gamma \vdash M\ t : \phi[a := t]$. By the induction hypothesis, $M' \Vdash \forall a.\ \phi'$, so $M' \downarrow \lambda a.\ N$ and $\forall u \in Tms_c.\ N[a := u] \Vdash \phi'[a := u]$. As $t[\rho] \in Tms_c$, we get in particular $N[a := t[\rho]] \Vdash \phi'[a := t[\rho]]$. Since $\overline{M\ t}[\rho] = M'\ (t[\rho]) \rightarrow^* (\lambda a.\ N)\ t[\rho] \rightarrow N[a := t[\rho]]$ and $(\phi[a := t])' = \phi'[a := t[\rho]]$, Lemma 2.2.24 gives us the claim.

- $$\frac{\Gamma \vdash M : \phi[a := t]}{\Gamma \vdash [t, M] : \exists a.\ \phi}$$

  By the induction hypothesis, $M' \Vdash \phi'[a := t[\rho]]$. Thus $\overline{[t, M]}[\rho] = [t[\rho], M'] \Vdash \exists a.\phi'$ which is what we want.

- $$\frac{\Gamma \vdash M : \exists a.\ \phi \quad \Gamma, x : \phi \vdash N : \psi}{\Gamma \vdash \text{let } [a, x : \phi] := M \text{ in } N : \psi}\ a \notin FV(\Gamma, \psi)$$

  Let $\rho \models \Gamma \vdash \text{let } [a, x : \phi] := M \text{ in } N : \psi$. We need to show that $\overline{\text{let } [a, x : \phi] := M \text{ in } N}[\rho] = \text{let } [a, x] := M' \text{ in } \overline{N}[\rho] \Vdash \psi'$. By the induction hypothesis, $M' \Vdash \exists a.\ \phi'$, so $M' \downarrow [t, M_1]$ and $M_1 \Vdash \phi'[a := t]$. By the induction hypothesis again, for any $\rho' \models \Gamma, x : \phi \vdash N : \psi$ we have $\overline{N}[\rho'] \Vdash \psi[\rho']$. Take $\rho' = \rho[x := M_1, a := t]$. Since $a \notin FV(\psi)$, $\psi[\rho'] = \psi'$, so $\overline{N}[\rho'] \Vdash \psi'$. Now, $\text{let } [a, x : \phi] := M' \text{ in } \overline{N}[\rho] \to^* \text{let } [a, x] := [t, M_1] \text{ in } \overline{N}[\rho] \to \overline{N}[\rho][a := t][x := M_1] = \overline{N}[\rho']$. Lemma 2.2.24 gives us the claim.

- $$\frac{\Gamma \vdash M : \phi_A(\vec{t})}{\Gamma \vdash \text{axRep}(\vec{t}, M) : \psi_A(\vec{t})}$$

  By the induction hypothesis, $M' \Vdash \phi'_A(\overrightarrow{t[\rho]})$. Since possible ϕ'_A are all atomic formulas, this means that $\phi'_A(\overrightarrow{t[\rho]})$ holds in the real world. It suffices to show that $\text{axRep}(\vec{t}[\rho], M') \Vdash \psi'_A(\overrightarrow{t[\rho]})$. Again, all possible ψ'_A are atomic, so since $\text{axRep}(\overrightarrow{t[\rho]}, M')$ is a value, we only need to verify the truth of $\psi'_A(\overrightarrow{t[\rho]})$ in the real world. As the real natural numbers satisfy axioms of HA, the claim follows.

- $$\frac{\Gamma \vdash M : \phi(0, \vec{t}) \land \forall n.\ \phi(n, \vec{t}) \to \phi(S(n), \vec{t})}{\Gamma \vdash \text{ind}_{n, \vec{a}.\ \phi(n, \vec{a})}(t, \vec{t}, M) : \phi(t, \vec{t})}$$

  By the induction hypothesis, $M' \Vdash \phi'(0, \overrightarrow{t[\rho]}) \land \forall n.\ \phi'(n, \overrightarrow{t[\rho]}) \to \phi'(S(n), \overrightarrow{t[\rho]})$. Therefore, $M' \downarrow \langle M_1, M_2 \rangle$, $M_1 \Vdash \phi'(0, \overrightarrow{t[\rho]})$ and $M_2 \Vdash \forall n.\ \phi'(n, \overrightarrow{t[\rho]}) \to \phi'(S(n), \overrightarrow{t[\rho]})$. By Lemma 2.2.24 therefore also $\text{fst}(M) \Vdash \phi'(0, \overrightarrow{t[\rho]})$ and $\text{snd}(M) \Vdash \forall n.\ \phi'(n, \overrightarrow{t[\rho]}) \to \phi'(S(n), \overrightarrow{t[\rho]})$. Since $\rho \models \Gamma \vdash \text{ind}_{n, \vec{a}.\ \phi(n, \vec{a})}(t, \overrightarrow{t[\rho]}, M) : \phi(t, \overrightarrow{t[\rho]})$, $t[\rho]$ is closed, so it is either a numeral or it reduces to a numeral.

By Lemma 2.2.24, it suffices to show that for all numerals m, $\mathrm{ind}(\overline{n}, M') \Vdash \phi'(\overline{n}, \vec{t[\rho]})$. We proceed by induction on m:

– If $m = 0$, $\mathrm{ind}(m, \overrightarrow{t[\rho]}, M') \to \mathrm{fst}(M')$. Since $\mathrm{fst}(M') \Vdash \phi'(0) \wedge \forall n.\ \phi'(n, \overrightarrow{t[\rho]}) \to \phi'(S(n), \overrightarrow{t[\rho]})$, Lemma 2.2.24 shows the claim.

– If $m = S(k)$, $\mathrm{ind}(m, \overrightarrow{t[\rho]}, M') \to \mathrm{snd}(M')\ k\ \mathrm{ind}(k, \overrightarrow{t[\rho]}, M')$. We know that $\mathrm{snd}(M') \downarrow \lambda n.\ O$ and $O[n := k] \Vdash \phi'(k, \overrightarrow{t[\rho]}) \to \phi'(S(\overline{k}), \overrightarrow{t[\rho]})$. By the induction hypothesis $\mathrm{ind}(\overline{k}, \overrightarrow{t[\rho]}, M') \Vdash \phi'(\overline{k}, \overrightarrow{t[\rho]})$, so by Lemma 2.2.25 $O[n := \overline{k}]\ \mathrm{ind}(\overline{k}, \overrightarrow{t[\rho]}, M') \Vdash \phi'(S(\overline{k}), \overrightarrow{t[\rho]})$. Lemma 2.2.24 applied again shows the claim.

-
$$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash M : \psi}\ \phi \leftrightarrow_\omega \psi$$

Straightforward by Lemma 2.2.26. ■

Corollary 2.2.29 (Normalization) *If* $\vdash M : \phi$*, then* $M \downarrow$*.*

Proof Take ρ mapping all free first-order variables of M, ϕ to themselves. Then $\rho \models\vdash M : \phi$. By Theorem 2.2.28, $\overline{M}[\rho] \Vdash \phi[\rho]$, so by the choice of ρ, $\overline{M} \Vdash \phi$. Lemmas 2.2.23 and 2.2.20 show the claim. ■

By Corollary 2.2.19, we have just shown consistency of HA. As HA is equiconsistent with PA [Göd65], by Gödel's Second Theorem we must have used more power than available in PA. However, it might seem that it is not the case; after all, the proof amounts to simple induction and verification of atomic formulas.

The answer lies in the induction axiom. In PA, it is impossible to formalize the proof of all instances of the schema at the same time, because there is no way to carry out the inductive proof in the normalization theorem "uniformly" for all formulas. However, for any finite number of formulas, the proof can be formalized, so PA can prove consistency of any finite fragment of its constructive counterpart.

The Disjunction Property (DP) still holds, for closed formulas. A technical lemma is useful:

Lemma 2.2.30 *If $\vdash_{HA} M : \phi$ and ϕ is closed, then there is a term N such that $\vdash_{HA} N : \phi$ and $FV_F(N) = \emptyset$.*

Proof Let $FV_F(M) = \vec{a}$. By ϕ being closed and Lemma 2.2.13, $\vdash_{HA} M[\vec{a} := \vec{0}] : \phi$. ■

Corollary 2.2.31 (Disjunction Property) *If $\vdash_{HA} \phi \vee \psi$ and ϕ, ψ are closed, then either $\vdash_{HA} \phi$ or $\vdash_{HA} \psi$.*

Proof By Lemma 2.2.10, there is a lambda term M such that $\vdash M : \phi \vee \psi$. By Lemma 2.2.30, we may assume that $FV_F(M) = \emptyset$. Thus we can use Lemma 2.2.18 and the proof of the Disjunction Property from the previous section. ■

From the point of view of computational information, the more important property is Numerical Existence Property (NEP):

Corollary 2.2.32 (Numerical Existence Property) *If $\vdash_{HA} \exists a.\ \phi$ and $\exists a.\ \phi$ is closed, then there is a natural number n such that $\vdash_{HA} \phi[a := n_n]$.*

Proof By Lemma 2.2.10, there is a lambda term M such that $\vdash M : \exists a.\ \phi$. By Lemma 2.2.30, we may assume that $FV_F(M) = \emptyset$. By Corollary 2.2.18, $M \downarrow v$, $\vdash v : \exists a.\ \phi$ and v is closed. By Canonical Forms, $v = [t, N]$. By Inversion, $\vdash N : \phi_1[a := t]$ and $\phi_1 =_\omega \phi$. By Lemma 2.2.6, $\vdash N : \phi[a := t_n]$, so by Lemma 2.2.10, $HA \vdash \phi[a := [\![t]\!]_n]$. ■

2.2.5 Computation in HA

We have finally arrived at the stage where we can present a realistic example of computation hidden in proofs. Consider a simple example — a function, which given any natural number returns 0 if the number is even and 1 otherwise. Let us first write the specification:

$$\begin{array}{rcl} \phi(n,m) & \equiv & (m = 0 \vee m = 1) \wedge \\ & & ((m = 0 \rightarrow \exists o.\ n = o + o) \wedge \\ & & (m = 1 \rightarrow \exists o.\ n = S(o + o))) \\ \text{Specification} & : & \forall n \exists m.\ \phi(n,m) \end{array}$$

We prove this simple theorem by induction on n. For $n = 0$, take $m = 0$. For $n = S(k)$, take m from the induction hypothesis. If m is 0, return 1, otherwise return 0. The correctness follows easily.

By the Curry-Howard isomorphism, there is a lambda term M such that $\vdash M : \forall n \exists m.\ \phi(n,m)$. A proof assistant, such as Nuprl or Coq, could produce M automatically during the process of proving $\forall n \exists m.\ \phi$. We show only the part of M which is relevant for the computation, leaving the reconstruction of the omitted terms (denoted by the "_" character), types and of the proof tree $\vdash M : \forall n \exists m.\ \phi(n,m)$ to the reader.

$$\begin{array}{rcl} M & \equiv & \lambda n.\ \mathrm{ind}_{n,m.\ \phi(n,m)}(n, \langle M_1, M_2 \rangle) \\ M_1 & \equiv & [0, \langle \mathrm{inl}(\mathrm{eqReflRep}(0)), \_ \rangle] \\ M_2 & \equiv & \lambda n.\ \lambda x.\ \mathrm{let}\ [m, y] := x\ \mathrm{in}\ M_3 \\ M_3 & \equiv & \mathrm{case}(\mathrm{fst}(y), x.\ [1, \langle \mathrm{inr}(\mathrm{eqReflRep}(1)), \_ \rangle], x.\ [0, \langle \mathrm{inl}(\mathrm{eqReflRep}(0)), \_ \rangle]) \end{array}$$

We want to use M as a program P which, given any n, produces a number m such that $m = n \bmod 2$. The program P, given n, works as follows:

- It constructs $M\ n$, which is of the type $\exists m.\ \phi(n,m)$.
- By Normalization, $M\ n \downarrow v$. By Lemma 2.2.18, $v : \exists m.\ \phi(n,m)$. By Canonical Forms, $v = [m, N]$ for some natural number m and term N.

- It therefore normalizes $M\ n$ to $[m, N]$ and returns m.

Let us denote by $P(n)$ the result of P on a number n. By the definition of P and the properties of λH, for any natural number n we have $HA \vdash \phi(n, P(n))$. Thus, the program P satisfies its specification.

To make this account a bit less abstract, let us see the computation of $P(0)$ and $P(1)$. For $P(0)$, we have:

$$
\begin{aligned}
&M\ 0 \equiv \\
&(\lambda n.\ \mathrm{ind}_{n,m.\ \phi(n,m)}(n, \langle M_1, M_2 \rangle))\ 0 \to \mathrm{ind}_{n,m.\ \phi(n,m)}(0, \langle M_1[n := 0], M_2[n := 0\rangle)) \to \\
&\mathrm{fst}(\langle M_1[n := 0], M_2[n := 0]\rangle)) \to M_1[n := 0] \equiv \\
&[0, \langle \mathrm{inl}(\mathrm{eqReflRep}(0)), \_\rangle]
\end{aligned}
$$

Therefore, $P(0)$ returns 0.

The reduction sequence for $P(1)$ is a bit longer. We abbreviate eqReflRep by eqRR.

$M\ 1 \equiv$

$(\lambda n.\ \mathrm{ind}_{n,m.\ \phi(n,m)}(n, \langle M_1, M_2\rangle))\ S(0) \rightarrow$

$\mathrm{ind}_{n,m.\ \phi(n,m)}(S(0), \langle M_1[n := S(0)], M_2[n := S(0)]\rangle)) \rightarrow$

$\mathrm{snd}(\langle M_1[n := S(0)], M_2[n := S(0)]\rangle))\ 0\ \mathrm{ind}_{n,m.\ \phi(n,m)}(0,$

$\quad \langle M_1[n := S(0)], M_2[n := S(0)]\rangle)) \rightarrow$

$M_2[n := S(0)]\ 0\ \mathrm{ind}_{n,m.\ \phi(n,m)}(0, \langle M_1[n := S(0)], M_2[n := S(0)]\rangle)) \equiv$

$(\lambda n.\ \lambda x.\ \mathrm{let}\ [m, y] := x\ \mathrm{in}\ M_3)\ 0\ \mathrm{ind}_{\phi(n,m)}(0, \langle M_1[n := S(0)], M_2[n := S(0)]\rangle)) \rightarrow$

$\lambda x.\ \mathrm{let}\ [m, y] := x\ \mathrm{in}\ M_3)\ 0\ \mathrm{ind}_{\phi(n,m)}(0, \langle M_1[n := S(0)], M_2[n := S(0)]\rangle)) \rightarrow$

$\mathrm{let}\ [m, y] := \mathrm{ind}_{n,m.\ \phi(n,m)}(0, \langle M_1[n := S(0)], M_2[n := S(0)]\rangle)\ \mathrm{in}\ M_3 \rightarrow^*$

$\mathrm{let}\ [m, y] := [0, \langle \mathrm{inl}(\mathrm{eRR}(0)), \_\rangle]\ \mathrm{in}\ M_3 \rightarrow$

$M_3[m := 0][y := \langle \mathrm{inl}(\mathrm{eRR}(0)), \_\rangle] \equiv$

$\mathrm{case}(\mathrm{fst}(\langle \mathrm{inl}(\mathrm{eRR}(0)), \_\rangle), x.\ [1, \langle \mathrm{inr}(\mathrm{eRR}(1)), \_\rangle], x.\ [0, \langle \mathrm{inl}(\mathrm{eRR}(0)), \_\rangle]) \rightarrow$

$\mathrm{case}(\mathrm{inl}(\mathrm{eRR}(0)), x.\ [1, \langle \mathrm{inr}(\mathrm{eRR}(1)), \_\rangle], x.\ [0, \langle \mathrm{inl}(\mathrm{eRR}(0)), \_\rangle]) \rightarrow$

$[1, \langle \mathrm{inr}(\mathrm{eRR}(1)), \_\rangle]$

Therefore, $P(1)$ returns 1.

The programs that can be extracted from HA proofs include all functions provably recursive in PA. This class includes all primitive recursive functions, Ackermann's function and much more. See [AZ97] for an example of a function which is *not* provably recursive in HA.

CHAPTER 3

TOWARDS COMPUTATIONAL UNDERSTANDING OF SET THEORY II : SETS

Having defined the isomorphism for simple systems, we can now move towards more powerful and abstract systems. Whereas in the previous chapter the main role was played by the natural numbers, one of the most concrete mathematical objects in existence, this chapter will be dominated by much more abstract entities: sets. While we will consider only sets of natural numbers in Section 3.1, the full set theory will enter in Section 3.2.

3.1 Second-order arithmetic

The formulation of arithmetic presented in the previous chapter was developed carefully by logicians trying to capture intuitions about numbers using first-order logic. However, the original Peano axiomatization is *not* first-order. The induction axiom in [Pea89] is "Any *set* containing 0 and closed under the successor operation contains all natural numbers". Thus, an abstract notion of a set of natural numbers was deemed necessary to talk about natural numbers.

Although first-order axiomatization is more elementary and useful for various purposes, a second-order[1] axiomatization is closer to Peano's original ideas, much more expressive and, most importantly from our point of view, it is a significant step in our journey toward understanding computation in set theory. Thus in this section we investigate second-order Heyting arithmetic, which we call HAS. More information on second- and higher-order logics can be found in [Lei94].

The *intuitionistic second-order logic*, underlying HAS, is an extension of IFOL. Informally, the extension consists of adding sets of individuals to the language,

[1]"second" refers to the fact that a logic encompasses sets of individuals.

together with the capability of quantifying over them and discussing whether a number is a member of a set.

Formally, we first fix a countable set of *set variables* $SVar$. Elements of $SVar$ will be usually denoted by letters X, Y, Z. We will sometimes also call them *second-order variables*. Furthermore, we extend the definition of terms and formulas. One significant difference from IFOL is that in second-order logic terms and formulas are defined together, by mutual induction. More specifically, there is a new syntactic kind of terms, which we call *set terms*. The set terms are generated by the following abstract grammar:

$$A ::= X \mid \{a \mid \phi\},$$

where ϕ is an arbitrary formula with the first-order variable a bound inside. Intuitively, the term $\{a \mid \phi\}$ denotes the set of all n such that $\phi[a := n]$ holds. Thus, the only sets HAS can mention are sets of natural numbers. We will call the $\{a \mid \phi\}$ terms *comprehension terms* or *set terms*. We will use letters A, B, C exclusively in this section to denote set terms. The set of all set terms will be denoted by $STms$ and the set of all closed set terms by $STms_c$.

We extend the definition of formulas to incorporate reasoning about set terms. The extension consists of adding one new atomic formula and two new quantifiers:

$$\phi ::= \ldots \mid t \in A \mid \forall X.\ \phi \mid \exists X.\ \phi$$

The variable X in the definition binds in ϕ. Formulas can therefore have two kinds of variables — first- and second-order. We denote the free first-order variables in a formula ϕ by $FV_F(\phi)$, the free second-order variables by $FV_S(\phi)$ and all free variables by $FV(\phi)$. These definitions are extended to contexts in a natural way.

The need for mutual induction in the definition should now be clear — formulas can mention set terms, while set terms may contain formulas.

$$
\begin{array}{rcl}
t & ::= & a \mid 0 \mid S(t) \mid t+t \mid t*t \\
A & ::= & X \mid \{a \mid \phi\} \\
\phi & ::= & t=t \mid \bot \mid \phi \to \psi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \forall n.\ \phi \mid \exists n.\ \phi \\
 & & t \in A \mid \forall X.\ \phi \mid \exists X.\ \phi
\end{array}
$$

$$
\frac{}{\Gamma, \phi \vdash \phi} \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash \phi} \qquad \frac{\Gamma, \phi \vdash \psi}{\Gamma \vdash \phi \to \psi} \qquad \frac{\Gamma \vdash \phi \to \psi \quad \Gamma \vdash \phi}{\Gamma \vdash \psi}
$$

$$
\frac{\Gamma \vdash \phi}{\Gamma \vdash \phi \vee \psi} \qquad \frac{\Gamma \vdash \psi}{\Gamma \vdash \phi \vee \psi} \qquad \frac{\Gamma \vdash \phi \vee \psi \quad \Gamma, \phi \vdash \vartheta \quad \Gamma, \psi \vdash \vartheta}{\Gamma \vdash \vartheta}
$$

$$
\frac{\Gamma \vdash \phi \quad \Gamma \vdash \psi}{\Gamma \vdash \phi \wedge \psi} \qquad \frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \phi} \qquad \frac{\Gamma \vdash \phi \wedge \psi}{\Gamma \vdash \psi}
$$

$$
\frac{\Gamma \vdash \phi}{\Gamma \vdash \forall a.\ \phi}\ a \notin FV_F(\Gamma) \qquad \frac{\Gamma \vdash \forall a.\ \phi}{\Gamma \vdash \phi[a := t]}
$$

$$
\frac{\Gamma \vdash \phi[a := t]}{\Gamma \vdash \exists a.\ \phi} \qquad \frac{\Gamma \vdash \exists a.\ \phi \quad \Gamma, \phi \vdash \psi}{\Gamma \vdash \psi}\ a \notin FV_F(\Gamma) \cup FV_F(\psi)
$$

$$
\frac{\Gamma \vdash \phi[a := t]}{\Gamma \vdash t \in \{a \mid \phi\}} \qquad \frac{\Gamma \vdash t \in \{a \mid \phi\}}{\Gamma \vdash \phi[a := t]}
$$

$$
\frac{\Gamma \vdash \phi}{\Gamma \vdash \forall X.\ \phi}\ X \notin FV_S(\Gamma) \qquad \frac{\Gamma \vdash \forall X.\ \phi}{\Gamma \vdash \phi[X := A]}
$$

$$
\frac{\Gamma \vdash \phi[X := A]}{\Gamma \vdash \exists X.\ \phi} \qquad \frac{\Gamma \vdash \exists X.\ \phi \quad \Gamma, \phi \vdash \psi}{\Gamma \vdash \psi}\ X \notin FV_S(\Gamma, \psi)
$$

(eqRefl)	$\forall n.\ n = n$	(eqSymm)	$\forall n, m.\ n = m \to m = n$
(eqTrans)	$\forall n, m, o.\ n = m \wedge m = o \to n = o$	(eqS)	$\forall n, m.\ n = m \to S(n) = S(m)$
(P3)	$\forall n.\ S(n) = 0 \to \bot$	(P4)	$\forall n, m.\ S(n) = S(m) \to n = m$
(plusZ)	$\forall n.\ n + 0 = 0$	(plusS)	$\forall n, m.\ n + S(m) = S(n + m)$
(mulZ)	$\forall n.\ n * 0 = 0$	(mulS)	$\forall n, m.\ n * S(m) = n * m + m$

(IND) $\forall X.\ 0 \in X \to (\forall n.\ n \in X \to S(n) \in X) \to \forall n.\ n \in X$

Figure 3.1: Second-order Heyting Arithmetic (HAS)

The logic rules are extended to take into account new formulas in the following way:

$$\frac{\Gamma \vdash \phi[a := t]}{\Gamma \vdash t \in \{a \mid \phi\}} \qquad \frac{\Gamma \vdash t \in \{a \mid \phi\}}{\Gamma \vdash \phi[a := t]}$$

$$\frac{\Gamma \vdash \phi}{\Gamma \vdash \forall X.\ \phi}\ X \notin FV_S(\Gamma) \qquad \frac{\Gamma \vdash \forall X.\ \phi}{\Gamma \vdash \phi[X := A]}$$

$$\frac{\Gamma \vdash \phi[X := A]}{\Gamma \vdash \exists X.\ \phi} \qquad \frac{\Gamma \vdash \exists X.\ \phi \quad \Gamma, \phi \vdash \psi}{\Gamma \vdash \psi}\ X \notin FV_S(\Gamma, \psi)$$

The intuitive justification of new rules should be clear. Note the symmetry between the first two rules; it will be crucial for developing the lambda calculus corresponding to HAS proofs.

HAS is deeply impredicative. Consider, for example, the set $D \equiv \{n \mid \forall X.\ n \in X\}$. This is a set of natural numbers and as such, it can be used to instantiate any second-order universal quantifier. Thus, since in order to determine whether $n \in D$ we need to know whether $\forall X.\ n \in X$, in particular we should know already whether $n \in D$. This perceived circularity is still disquieting to some mathematicians who subscribe to the view that mathematics should be predicative. See [Fef05] for more information.

Having extended the logic, we proceed to amend the axioms of HA. The amendment amounts to axiomatizing induction in the spirit closer to the Peano's original definition:

Definition 3.1.1 *HAS arises by replacing the induction axiom schema in HA by one axiom:*

$$\text{(IND) } \forall X.\ 0 \in X \to (\forall n.\ n \in X \to S(n) \in X) \to \forall n.\ n \in X$$

Note that HAS contains HA:

Lemma 3.1.2 $HAS \vdash HA$.

Proof It suffices to show that all instances of the induction axiom schema are provable. Recall that the schema has the following form:

$$(\mathrm{ind}_{\phi(n,\vec{a})})\ \forall \vec{n}.\ \phi(0,\vec{n}) \to (\forall n.\ \phi(n,\vec{n}) \to \phi(S(n),\vec{n})) \to \forall n.\ \phi(n,\vec{n})$$

Take any $\vec{n}$, suppose $\phi(0,\vec{n})$ and $\forall n.\ \phi(n,\vec{n}) \to \phi(S(n),\vec{n})$ and take any n. We need to show $\phi(n,\vec{n})$. Take $X = \{a \mid \phi(a,\overline{n})\}$. Then $0 \in X$ and $\forall m.\ m \in X \to S(m) \in X$, so $\forall m.\ m \in X$. In particular, $n \in X$ so $\phi(n,\vec{n})$. ■

3.1.1 λS calculus

We now present a lambda calculus λS for HAS. We add to λH the new syntactic category of set terms, which are exactly the set terms of HAS, generated by the same grammar:

$$A ::= X \mid \{a \mid \phi\}$$

We eliminate the induction terms from HA and extend the resulting grammar by the following clauses:

$$\begin{aligned} M \quad ::= \quad & M\ A \mid \lambda X.\ M \mid [A, M] \mid \mathrm{lets}\ [X, x : \phi] := M \text{ in } N \mid \\ & \mathrm{sepRep}(t, M) \mid \mathrm{sepProp}(t, M) \mid \mathrm{ind}(t, M) \end{aligned}$$

The first four new terms mirror their counterparts in the first-order logic. The $\mathrm{ind}(t, M)$ term corresponds to the induction axiom. The $\mathrm{sepRep}(t, M)$ and $\mathrm{sepProp}(t, M)$ terms correspond to the proof rules governing the comprehension terms. Informally, if M is a proof of $\phi(t)$, then $\mathrm{sepRep}(t, M)$ is a proof of $t \in \{n \mid \phi(n)\}$. Symmetrically, if M is a proof of $t \in \{n \mid \phi(n)\}$, then $\mathrm{sepProp}(t, M)$ is a proof of $\phi(t)$. This symmetry is crucial for the construction of λS.

The reduction system is extended by the following reduction relations:

$$(\lambda X.\ M)\ A \to M[X := A] \qquad \mathrm{lets}\ [X, x : \phi] := [A, M] \text{ in } N \to N[X := A][x := M]$$

$$\mathrm{ind}(0, M) \to \mathrm{fst}(M) \qquad \mathrm{ind}(S(t), M) \to \mathrm{snd}(M)\ t\ \mathrm{ind}(t, M), \text{where } t \text{ is closed}$$

$$\mathrm{ind}(t, M) \to \mathrm{ind}(t_n, M), \text{where } t \text{ is a closed non-numeral}$$

$$\mathrm{sepProp}(t, \mathrm{sepRep}(u, M)) \to M$$

The last reduction rule is significantly different from all rules we encountered so far. It provides a way to eliminate possible redundancies introduced by the comprehension axiom. For example, the proof "$0 = 0$, so $0 \in \{a|\ 0 = 0\}$, so $0 = 0$" is clearly redundant and the rule would reduce it to "$0 = 0$". We retain the $\to_\omega$ relation unchanged.

The new evaluation contexts are:

$$[\circ] \quad ::= \quad \ldots \mid [\circ]\ A \mid \mathrm{lets}\ [a, x : \phi] := [\circ] \text{ in } N \mid \mathrm{ind}([\circ], M)$$

The new values in the calculus are:

$$V \quad ::= \quad \ldots \mid \mathrm{sepRep}(t, M) \mid \lambda X.\ M \mid [A, M]$$

The typing system of λH is extended by the following typing rules:

$$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash \lambda X.\ M : \forall X.\ \phi}\ X \notin FV_S(\Gamma) \qquad \frac{\Gamma \vdash M : \forall X.\ \phi}{\Gamma \vdash M\ A : \phi[X := A]}$$

$$\frac{\Gamma \vdash M : \phi[X := A]}{\Gamma \vdash [A, M] : \exists X.\ \phi} \qquad \frac{\Gamma \vdash M : \exists X.\ \phi \quad \Gamma, x : \phi \vdash N : \psi}{\Gamma \vdash \mathrm{lets}\ [X, x : \phi] := M \text{ in } N : \psi}\ X \notin FV_S(\Gamma, \psi)$$

$$\frac{\Gamma \vdash M : 0 \in A \land \forall n.\ n \in A \to S(n) \in A}{\Gamma \vdash \mathrm{ind}(t, M) : t \in A}$$

$$\frac{\Gamma \vdash M : \phi[a := t]}{\Gamma \vdash \mathrm{sepRep}(t, M) : t \in \{a \mid \phi\}} \qquad \frac{\Gamma \vdash M : t \in \{a \mid \phi\}}{\Gamma \vdash \mathrm{sepProp}(t, M) : \phi[a := t]}$$

Note the similarity between the first four rules and the rules in the first-order case. There is no significant difference; this will make our proofs of the standard lemmas much shorter.

The correspondence continues to hold:

Lemma 3.1.3 (Curry-Howard isomorphism) *If $\Gamma \vdash O : \phi$ then $rg(\Gamma) \vdash_{HAS} \phi$. If $\Gamma \vdash_{HAS} \phi$, then there exists a term M such that $\overline{\Gamma} \vdash M : \phi$, where $\overline{\Gamma} = \{(x_\phi, \phi) \mid \phi \in \Gamma\}$.*

Proof Straightforward induction on the proof tree $\Gamma \vdash O : \phi$. ∎

3.1.2 Properties of λS

The properties are extended mostly in a predictable way.

Lemma 3.1.4 (Inversion) *Suppose $\Gamma \vdash Q : \Psi$. Suppose Q is of the form:*

- $\lambda X.\ M$. *Then $\Psi =_\omega \forall X.\ \phi$ and $\Gamma \vdash M : \phi$.*
- $M\ A$. *Then $\Gamma \vdash M : \forall X.\ \phi$ and $\Psi =_\omega \phi[X := A]$.*
- $[A, M]$. *Then $\Gamma \vdash M : \phi[X := A]$ and $\Psi =_\omega \exists X.\ \phi$.*
- $\text{lets } [a, x : \phi] := M \text{ in } N$. *Then $\Gamma \vdash M : \exists X.\ \phi$ and $\Gamma, x : \phi \vdash N : \Psi_1$ and $\Psi =_\omega \Psi_1$.*
- $\text{sepRep}(t, M)$. *Then for some ϕ, $\Psi =_\omega t \in \{a \mid \phi\}$ and $\Gamma \vdash M : \phi[a := t]$.*
- $\text{sepProp}(t, M)$. *Then for some ϕ, $\Psi =_\omega \phi[a := t]$ and $\Gamma \vdash M : t \in \{a \mid \phi\}$.*
- $\text{ind}(t, M)$. *Then for some A, $\Psi =_\omega t \in A$ and $\Gamma \vdash M : 0 \in A \land \forall n.\ n \in A \to S(n) \in A$.*

Lemma 3.1.5 (Canonical Forms) *Suppose Q is a value, $\vdash Q : \Psi$ and Ψ is of the form:*

- $t \in \{a \mid \phi\}$. *Then $Q = \text{sepRep}(t_1, M)$, $\vdash M : \phi_1[a := t_1]$, $\phi_1 =_\omega \phi$ and $t_1 =_\omega t$.*
- $\forall X.\ \phi$. *Then $Q = \lambda X.\ M$, $\vdash M : \phi_1$ and $\phi_1 =_\omega \phi$.*

- $\exists X.\ \phi$. *Then* $Q = [A, M]$, $M : \phi_1[X := A]$ *and* $\phi_1 =_\omega \phi$.

The substitution lemmas are proved with no new difficulties:

Lemma 3.1.6 *If* $\Gamma, x : \phi \vdash M : \psi$ *and* $\Gamma \vdash N : \phi$, *then* $\Gamma \vdash M[x := N] : \psi$.

Lemma 3.1.7 *If* $\Gamma \vdash Q : \Psi$, *then for any first-order variable* b *and term* u, $\Gamma[b := u] \vdash Q[b := u] : \Psi[b := u]$.

We need a new substitution lemma, for second-order variables:

Lemma 3.1.8 *If* $\Gamma \vdash Q : \Psi$, *then* $\Gamma[X := A] \vdash Q[X := A] : \Psi[X := A]$.

Proof Straightforward induction on $\Gamma \vdash Q : \Psi$. The proof is very similar to the proof of Lemma 3.1.7. ■

Lemma 3.1.9 (Subject Reduction, Preservation) *If* $\Gamma \vdash O : \Psi$ *and* $O \to P$, *then* $\Gamma \vdash P : \Psi$.

Proof By induction on the definition of $O \to P$. The proofs of most of the new cases mirror the first-order cases in the proof in Section 2.2. We present therefore the only significantly new case. Case $O \to P$ of:

- $$\mathrm{sepProp}(t, \mathrm{sepRep}(u, M)) \to M$$

  By Inversion, for some ϕ, $\Psi =_\omega \phi[a := t]$ and $\Gamma \vdash \mathrm{sepRep}(u, M) : t \in \{a \mid \phi\}$. By Inversion again, for some ϕ_1, $t =_\omega u$, $\phi_1 =_\omega \phi$ and $\Gamma \vdash M : \phi_1[a := u]$. Thus also $\Gamma \vdash M : \phi[a := t]$ and consequently $\Gamma \vdash M : \Psi$. ■

Lemma 3.1.10 (Progress) *If* $\vdash P : \Psi$ *and* $FV_F(P) = \emptyset$ *then either* P *is a value or there is* Q *such that* $P \to Q$.

Proof By induction on the length of P. The proofs of most of the new cases mirror the first-order cases in Section 2.2. We present the remaining case:

- $P = \mathrm{sepProp}(t, M)$. By Inversion, for some ϕ, $\Gamma \vdash M : t \in \{a \mid \phi\}$. By the induction hypothesis, either M is a value or $M \to M'$. In the former case, by Canonical Forms for some t_1, $M = \mathrm{sepRep}(t_1, N)$ and since $\mathrm{sepProp}(t, \mathrm{sepRep}(t_1, M)) \to M$, the claim follows. In the latter case, $\mathrm{sepProp}(t, M) \to \mathrm{sepProp}(t, M')$. ■

Note the importance of the reduction rule $\mathrm{sepProp}(t, \mathrm{sepRep}(u, M)) \to M$ to the proof; had it not been for this rule, the reductions might stop well before reaching the values.

3.1.3 Realizability for HAS

In HAS, providing a BHK interpretation becomes challenging. What should the construction for $\exists X.\ \phi$ be? Presumably a set of natural numbers A along with the construction of $\phi[X := A]$. However, such an approach would presume independent existence of sets of natural numbers; entities not at all intuitive and acceptable from a constructive point of view. Although some substitutes exist, such as the set of all functions from $\mathbb{N}$ to a two-element set, in a constructive world they do not reach full generality of arbitrary subsets of natural numbers.

We therefore at this point leave BHK interpretations behind, providing only realizability relations instead. We are not worried about the existence of sets of natural numbers, nor even arbitrarily large sets as we shall see later. We remind the reader that in our view the theories we consider are constructive cores of classical theories, providing information about their computational content. Their philosophical justification is of no concern to us. What is important is the method to tackle impredicativity, generalizing smoothly to our final goal: the set theory.

As before, we start with the erasure map, erasing λS to $\lambda\overline{S}$. A important feature of λS is that the second-order terms play no role in reductions; contrast this with the first-order terms, which are necessary for the reduction rule corresponding to the induction axiom. The erasure map takes this into account and replaces all the second-order terms by the term $\emptyset \equiv \{n \mid \bot\}$. We present the representative cases.

$$\overline{M\ A} = \overline{M}\ \emptyset \qquad \overline{\lambda X.\ M} = \lambda X.\ \overline{M} \qquad \overline{[A, M]} = [\emptyset, \overline{M}]$$

$$\overline{\mathrm{lets}\ [X, x : \phi] := M\ \mathrm{in}\ N} = \mathrm{lets}\ [X, x] := \overline{M}\ \mathrm{in}\ \overline{N}$$

$$\overline{\mathrm{sepRep}(t, M)} = \mathrm{sepRep}(\overline{M}) \qquad \overline{\mathrm{sepProp}(t, M)} = \mathrm{sepProp}(\overline{M})$$

The main lemma is still valid:

Lemma 3.1.11 *If $\overline{M} \downarrow$ then $M \downarrow$.*

The moment we try to write a realizability relation in a naive way, we encounter the following problem. The meaning of a term $\{a \mid \phi\}$ should somehow be related to whether ϕ is realizable or not. On the other hand, looking at realizability for HA, we would probably like to have $M \Vdash \forall X.\ \phi$ if for all terms A, $M \Vdash \phi[X := A]$. As A may be arbitrary, in particular containing $\forall X.\ \phi$ again, it seems that some circularity is involved. And indeed, there is no way to write a definition in this manner. This should not come as a surprise, as HAS is much more powerful than HA. Since realizability can be used as a tool to prove the normalization of λS and consequently the consistency of HAS, simple induction formalizable in HA cannot suffice.

We need to utilize in an essential way the impredicativity of "real" natural numbers. For this purpose, we introduce a notion of λ-*set*:

Definition 3.1.12 *A λ-set is a set of pairs (M, n), where M is a closed term of $\lambda\overline{S}$ and n is a natural number.*

The (uncountable) set of all λ-sets will be denoted by H^λ. More formally:

$$H^\lambda \equiv P(\lambda\overline{S}_c \times \mathbb{N}),$$

where P denotes the power set operation and $\lambda\overline{S}_c$ denotes the set of all closed $\lambda\overline{S}$ terms.

H^λ gives us the necessary leverage to define realizability for HAS. The elements of H^λ will be denoted by letters D, E, F.

We will use the sets from H^λ in the realizability relation. For this purpose, we introduce environments to our realizability relations as well. There will be no confusion between these environments and the ones used for normalization proofs; the context will always make the situation clear.

Definition 3.1.13 *An* environment *is a finite function mapping second-order variables to H^λ.*

We will use the letter ρ to denote environments.

From now on, realizability relations will be parameterized by environments. We will write $M \Vdash_\rho \phi$ and read "M realizes ϕ in an environment ρ". We incorporate the definition of a meaning of a term into the realizability relation. As before, we define realizability only on formulas ϕ such that $FV_F(\phi) = \emptyset$. However, ϕ can have free second-order variables; their meaning will be fixed by the environment. As before, *realizers* are closed terms of $\lambda\overline{S}$.

Definition 3.1.14 *The realizability relation for HAS, written as $M \Vdash_\rho \phi$, relates realizers with formulas ϕ such that $FV_F(\phi) = \emptyset$ for environments ρ defined on $FV_S(\phi)$. The meaning of a closed term T in an environment ρ is denoted by $[\![T]\!]_\rho$.*

They are defined by the following clauses:

$$
\begin{array}{lcl}
[\![0]\!]_\rho & \equiv & 0 \\
[\![S(t)]\!]_\rho & \equiv & [\![t]\!]_\rho + 1 \\
[\![t+s]\!]_\rho & \equiv & [\![t]\!]_\rho + [\![s]\!]_\rho \\
[\![t*s]\!]_\rho & \equiv & [\![t]\!]_\rho * [\![s]\!]_\rho \\
[\![X]\!]_\rho & \equiv & \rho(X) \\
[\![\{a \mid \phi\}]\!]_\rho & \equiv & \{(\mathrm{sepRep}(M), m) \mid M \Vdash_\rho \phi[a := m_n]\} \\
M \Vdash_\rho t = s & \equiv & M \downarrow \wedge [\![t]\!]_\rho = [\![s]\!]_\rho \\
M \Vdash_\rho t \in A & \equiv & M \downarrow v \wedge (v, [\![t]\!]_\rho) \in [\![A]\!]_\rho \\
M \Vdash_\rho \bot & \equiv & \bot \\
M \Vdash_\rho \phi \wedge \psi & \equiv & M \downarrow \langle M_1, M_2\rangle \wedge (M_1 \Vdash_\rho \phi) \wedge (M_2 \Vdash_\rho \psi) \\
M \Vdash_\rho \phi \vee \psi & \equiv & (M \downarrow \mathrm{inl}(M_1) \wedge M_1 \Vdash_\rho \phi) \vee (M \downarrow \mathrm{inr}(M_1) \wedge M_1 \Vdash_\rho \psi) \\
M \Vdash_\rho \phi \to \psi & \equiv & (M \downarrow \lambda x.\ M_1) \wedge \forall N.\ (N \Vdash_\rho \phi) \to (M_1[x := N] \Vdash_\rho \psi) \\
M \Vdash_\rho \exists a.\ \phi & \equiv & M \downarrow [t, N] \wedge N \Vdash_\rho \phi[a := t] \\
M \Vdash_\rho \forall a.\ \phi & \equiv & M \downarrow \lambda a.\ N \wedge \forall t \in Tms_c.\ N[a := t] \Vdash_\rho \phi[a := t] \\
M \Vdash_\rho \exists X.\ \phi & \equiv & M \downarrow [\emptyset, N] \wedge \exists D \in H^\lambda.\ N \Vdash_{\rho[X:=D]} \phi \\
M \Vdash_\rho \forall X.\ \phi & \equiv & M \downarrow \lambda X.\ N \wedge \forall D \in H^\lambda.\ \ N[X := \emptyset] \Vdash_{\rho[X:=D]} \phi
\end{array}
$$

Note that this definition is deeply impredicative. The meaning C of a comprehension term $\{a \mid \phi\}$ is a set from H^λ. In order to determine the members of C, we may need to quantify over all sets in H^λ, including C. The two guilty clauses in the definition are:

$$
\begin{array}{lcl}
M \Vdash_\rho t \in A & \equiv & M \downarrow v \wedge (v, [\![t]\!]_\rho) \in [\![A]\!]_\rho \\
M \Vdash_\rho \forall X.\ \phi & \equiv & M \downarrow \lambda X.\ N \wedge \forall D \in H^\lambda.\ \ N[X := \emptyset] \Vdash_{\rho[X:=D]} \phi
\end{array}
$$

For example, taking again $D \equiv \{n \mid \forall X.\ n \in X\}$, we see that $[\![D]\!]_\rho \equiv \{(\mathrm{sepRep}(M), m) \mid M \Vdash_\rho \forall X.\ m_n \in X\}$ which is equal to $\{(\mathrm{sepRep}(M), m) \mid \forall C \in H^\lambda.\ M[X := \emptyset] \Vdash_\rho m_n \in C\}$. Since $[\![D]\!]_\rho \in H^\lambda$, in order to determine whether

$(\mathrm{sepRep}(M), m) \in [\![D]\!]_\rho$, it seems that we already need to have the information about members of $[\![D]\!]_\rho$. Despite this fact, the definition is not circular:

Lemma 3.1.15 *The definition of realizability is well-founded.*

Proof Use the measure function m which takes a clause in the definition and returns an element of $\mathbb{N}^2$ with the lexicographical order:

$$\begin{aligned} m(M \Vdash_\rho \phi) &= (\text{“the number of comprehension terms in } \phi, \\ &\quad\ \text{“structural complexity of } \phi\text{”}) \\ m([\![t]\!]_\rho) &= (\text{“the number of comprehension terms in } t, 0) \end{aligned}$$

Then the measure of the definiendum is always greater than the measure of the definiens. Note that in the clauses for formulas the structural complexity decreases, while the number of comprehension terms does not grow larger. Moreover, in the definition of $[\![\{a \mid \phi\}]\!]_\rho$, one comprehension term disappears. ∎

It is easy to see that the standard lemmas continue to hold:

Lemma 3.1.16 *If $M \Vdash_\rho \phi$, then $M \downarrow$.*

Lemma 3.1.17 *If $M \to^* N$, then $M \Vdash_\rho \phi$ iff $N \Vdash_\rho \phi$.*

There is one new Lemma, stating that environments "commute" with substitutions:

Lemma 3.1.18 $[\![s[a := t]]\!]_\rho = [\![s[a := ([\![t]\!]_\rho)_n]]\!]$ *and* $M \Vdash_\rho \phi[a := t]$ *iff* $M \Vdash_\rho \phi[a := ([\![t]\!]_\rho)_n]$. *Also,* $[\![A[X := B]]\!]_\rho = [\![A]\!]_{\rho[X:=[\![B]\!]_\rho]}$ *and* $M \Vdash_\rho \phi[X := B]$ *iff* $M \Vdash_{\rho[X:=[\![B]\!]_\rho]} \phi$

Proof Straightforward induction on the definition of realizability. ∎

3.1.4 Normalization of λS

The normalization proof is not changed much compared to λH. We need to extend the notion of an environment.

Definition 3.1.19 *An* environment*, denoted by ρ, is a finite partial function from $Var \cup FVar \cup SVar$ to $\lambda\overline{S} \cup Tms_c \cup H^\lambda$ such that $\rho^{\rightarrow}(Var) \subseteq \lambda\overline{S}_c, \rho^{\rightarrow}(FVar) \subseteq Tms_c$ and $\rho^{\rightarrow}(SVar) \subseteq H^\lambda$.*

We define $M[\rho]$ and $\phi[\rho]$ essentially as before:

$$\begin{aligned} M[\rho] &\equiv M[\vec{x} := \overrightarrow{\rho(x)}, \vec{a} := \overrightarrow{\rho(a)}] \\ \phi[\rho] &\equiv \phi[\vec{a} := \overrightarrow{\rho(a)}] \end{aligned}$$

Note that any ρ can be used as a realizability environment, by considering only its part mapping second-order variables to sets from H^λ. Therefore we will be using the notation $\Vdash_\rho$ also for these environments ρ.

We write $\rho \models \Gamma \vdash M : \phi$ if for all $(x, \phi) \in \Gamma$, $\rho(x) \Vdash_\rho \phi[\rho]$ and ρ is defined on $FV(\Gamma, M, \phi)$. It is easy to see that if $\rho \models \Gamma \vdash M : \phi$, then $\overline{M}[\rho] \Vdash_\rho \phi[\rho]$ is defined.

Lemma 3.1.20 (Normalization) *If $\Gamma \vdash O : \Psi$, then for all $\rho \models \Gamma \vdash O : \Psi$, $\overline{M}[\rho] \Vdash_\rho \Psi[\rho]$.*

Proof For any lambda term M, M' in the proof denotes $\overline{M}[\rho]$. We proceed by induction on the proof and show the new cases compared to HA. Case $\Gamma \vdash O : \Psi$ of:

- $$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash \lambda X.\ M : \forall X.\ \phi}\ X \notin FV_S(\Gamma)$$

  Take any $\rho \models \Gamma \vdash \lambda X.\ M : \forall X.\ \phi$. We need to show that for all D, $\overline{M}[\rho][X := \emptyset] \Vdash_{\rho[X:=D]} \phi[\rho]$. Take any such D and let $\rho' = \rho[X := D]$.

Then $\rho' \models \Gamma \vdash M : \phi$, so by the induction hypothesis, $\overline{M}[\rho'] \Vdash_{\rho'} \phi[\rho']$. As $\overline{M}[\rho][X := \emptyset] = \overline{M}[\rho']$ and $\phi[\rho'] = \phi[\rho]$, the claim follows.

- $$\frac{\Gamma \vdash M : \forall X.\ \phi}{\Gamma \vdash M\ A : \phi[X := A]}$$

  Take any $\rho \models \Gamma \vdash \lambda X.\ M : \forall X.\ \phi$. Then also $\rho \models \Gamma \vdash M : \forall X.\ \phi$. By the induction hypothesis, $M' \downarrow \lambda X.\ N$ and for all D, $N[X := \emptyset] \Vdash_{\rho[X:=D]} \phi$. In particular, taking $D = [\![A]\!]_\rho$, $N[X := \emptyset] \Vdash_{\rho[X:=[\![A]\!]_\rho]} \phi$, so by Lemma 3.1.18, $N[X := \emptyset] \Vdash_\rho \phi[X := A]$. Since $\overline{M\ A}[\rho] = M'\ \emptyset \to^* (\lambda X.\ N)\ \emptyset \to N[X := \emptyset]$, Lemma 2.2.24 shows the claim.

- $$\frac{\Gamma \vdash M : \phi[X := A]}{\Gamma \vdash [A, M] : \exists X.\ \phi}$$

  Take any $\rho \models \Gamma \vdash [A, M] : \exists X.\ \phi$. It suffices to show that there is D such that $M' \Vdash_{\rho[X:=D]} \phi$. By the induction hypothesis, $M' \Vdash_\rho \phi[X := A]$. Taking $D = [\![A]\!]_\rho$ and applying Lemma 3.1.18, we get the claim.

- $$\frac{\Gamma \vdash M : \exists X.\ \phi \quad \Gamma, x : \phi \vdash N : \psi}{\Gamma \vdash \text{lets } [X, x : \phi] := M \text{ in } N : \psi}\ X \notin FV_S(\Gamma, \psi)$$

  Take any $\rho \models \Gamma \vdash \text{lets } [X, x : \phi] := M \text{ in } N : \psi$. By the induction hypothesis, $M' \Vdash_\rho \exists X.\ \phi[\rho]$, so $M' \downarrow [\emptyset, M_1]$ and there is some D such that $M_1 \Vdash_{\rho[X:=D]} \phi$. Since $\overline{\text{lets } [X, x] := M \text{ in } N} \to^* \text{lets } [X, x] := [\emptyset, M_1] \text{ in } N' \to N'[x := M_1]$, by Lemma 3.1.17 it suffices to show that $N'[x := M_1] \Vdash_\rho \psi$. Let $\rho' \equiv \rho[X := D, x := M_1]$. Note that $\rho' \models \Gamma, x : \phi \vdash N : \psi$, so by the induction hypothesis $N[\rho'] \Vdash_{\rho'} \psi[\rho']$. As $X \notin FV_S(\psi)$, it is easy to see that $N[\rho'] \Vdash_\rho \psi[\rho]$. As $N[\rho'] = N'[x := M_1]$, we get the claim.

- $$\frac{\Gamma \vdash M : 0 \in A \land \forall n.\ n \in A \to S(n) \in A}{\Gamma \vdash \text{ind}(t, M) : t \in A}$$

The proof follows the steps of the proof of the corresponding case in λH.

- $$\frac{\Gamma \vdash M : \phi[a := t]}{\Gamma \vdash \mathrm{sepRep}(t, M) : t \in \{a \mid \phi\}}$$

  Take any $\rho \models \Gamma \vdash \mathrm{sepRep}(t, M) : t \in \{a \mid \phi\}$. By the induction hypothesis, $M' \Vdash_\rho \phi'[a := t[\rho]]$. We need to show $M' \Vdash_\rho \phi'[a := ([\![t[\rho]]\!]_\rho)_n]$. Lemma 3.1.18 shows the claim.

- $$\frac{\Gamma \vdash M : t \in \{a \mid \phi\}}{\Gamma \vdash \mathrm{sepProp}(t, M) : \phi[a := t]}$$

  Take any $\rho \models \Gamma \vdash \mathrm{sepProp}(t, M) : \phi[a := t]$. By the induction hypothesis, $M' \downarrow v$ and $(v, [\![t[\rho]]\!]_\rho) \in [\![\{a \mid \phi\}]\!]_\rho$. This means that for some N, $v = \mathrm{sepRep}(N)$ and $N \Vdash_\rho \phi[a := ([\![t[\rho]]\!]_\rho)_n]$. Since $\mathrm{sepProp}(M') \to^* \mathrm{sepProp}(\mathrm{sepRep}(N)) \to N$, Lemma 3.1.18 shows the claim. ■

Corollary 3.1.21 (Normalization) *If* $\vdash M : \phi$, *then* $M \downarrow$.

Corollary 3.1.22 *HAS has the Disjunction Property and the Numerical Existence Property.*

As in case of HA, normalization provides the access to the computational content of HAS proofs. Second-order quantification enhances the specification language significantly, as for example real and complex numbers can be easily encoded as sets of natural numbers. There are many classical examples of such encodings presented for example in [Sim99]. Constructively, coming up with a "correct" definition of real numbers is more difficult. See [Lub07] for the description of the issues involved.

At this point, our adventures with weak systems are over. We showed normalization of propositional logic, using realizability. Later, we extended our methods to the first-order Heyting arithmetic, incorporating quantification to our proofs. Finally, in this section, we showed how to tackle systems involving impredicativity using our method. We are now ready to approach the strongest and the most expressive formal system ever invented — ZFC set theory.

3.2 Set theory

In the introduction, we described the origins of ZFC, Zermelo-Fraenkel set theory with Choice, now widely accepted as the foundation of mathematics. In the previous sections we showed how to find computation in the constructive cores of propositional logic, first and second-order arithmetic. We are now ready to expose the computational content of the constructive core of ZFC. We assume basic familiarity with set theory, found for example in the first few chapters of [Kun80] or [Jec03]. The results of this section can also be found in [Moc06a].

The constructive core of ZFC is called IZF, Intuitionistic Zermelo-Fraenkel. It was first introduced by Myhill [Myh73]. It is essentially what remains of ZFC after the excluded middle is carefully taken away. More specifically, apart from eliminating the excluded middle, we also need to make two changes to the axioms of ZFC. First, as Choice is inherently nonconstructive, it needs to be taken away. Second, surprisingly, Foundation in the standard formulation also implies the Excluded Middle, so in IZF it is reformulated as $\in$-Induction.

An important decision to make is whether to use the Replacement or the Collection axiom schema. We will call the version with Collection IZF_C and the version with Replacement IZF_R. In the literature, IZF usually denotes IZF_C. Both theories extended with excluded middle are equivalent to ZF [Fri73] and thus proof-theoretically as strong as ZFC. We will have more to say about the differences in Section 4.2.

Both versions have been investigated thoroughly. Results up to 1985 are presented in [Bee85, Š85]. Later research concentrated on weaker subsystems [AR01, Lub02]. A predicative constructive set theory CZF has attracted particular interest. [AR01] describes the set-theoretic apparatus available in CZF and provides further references.

3.2.1 The axioms of $\mathbf{IZF_R}$

IZF is a first-order theory. There are three binary relational symbols: $\in_I, \in, =$. The first one might look unfamiliar to the reader. It denotes an intensional membership relation and it is used as a low-level mechanism to tackle the extensional nature of set theory — see Lemmas 3.2.6, 3.2.7 and 3.2.8. In this section, we will focus on IZF_R, as Collection is much more difficult to tackle using our methods. However, it is possible to apply our methods also to IZF_C, as we show in Section 4.2.

Similarly to HAS, some terms of IZF_R are parameterized by formulas. We therefore define terms and formulas at the same time, by mutual induction and the following abstract grammar:

$$\begin{aligned} t &::= a \mid \emptyset \mid \{t, t\} \mid \omega \mid \bigcup t \mid P(t) \mid S_{a,\vec{f}.\ \phi(a,\vec{f})}(t, \vec{t}) \mid R_{a,b,\vec{f}.\ \phi(a,b,\vec{f})}(t, \vec{t}) \\ \phi &::= \bot \mid t \in t \mid t = t \mid t \in_I t \mid \phi \vee \phi \mid \phi \wedge \phi \mid \phi \to \phi \mid \forall a.\ \phi \mid \exists a.\ \phi \end{aligned}$$

The terms $S_{a,\vec{f}.\ \phi(a,\vec{f})}(t, \vec{t})$ and $R_{a,b,\vec{f}.\ \phi(a,b,\vec{f})}(t, \vec{t})$ could be displayed more familiarly as $\{c \in t \mid \phi(c, \vec{t})\}$ and $\{c \mid (\forall x \in t \exists! y.\ \phi(x, y, \vec{t})) \wedge (\exists x \in t.\ \phi(x, c, \vec{t}))\}$, respectively, where $\exists! y.\ \phi$ is an abbrevation for "there exists exactly one y such that ϕ".

The rules of the first-order logic remain unchanged. The axioms follow, with $S(t)$ abbreviating $t \cup \{t, t\}$.

- (IN) $\forall a, b.\ a \in b \leftrightarrow \exists c.\ c \in_I b \wedge a = c$
- (EQ) $\forall a, b.\ a = b \leftrightarrow \forall d.\ (d \in_I a \to d \in b) \wedge (d \in_I b \to d = a)$

- (EMPTY) $\forall c.\ c \in_I \emptyset \leftrightarrow \bot$
- (PAIR) $\forall a, b \forall c.\ c \in_I \{a, b\} \leftrightarrow c = a \vee c = b$
- (INF) $\forall c.\ c \in_I \omega \leftrightarrow c = 0 \vee \exists b \in \omega.\ c = S(b)$
- (SEP$_{\phi(a,\vec{f})}$) $\forall \vec{f} \forall a \forall c.\ c \in_I S_{a,\vec{f}.\ \phi(a,\vec{f})}(a, \vec{f}) \leftrightarrow c \in a \wedge \phi(c, \vec{f})$
- (UNION): $\forall a \forall c.\ c \in_I \bigcup a \leftrightarrow \exists b \in a.\ c \in b$
- (POWER) $\forall a \forall c.\ c \in_I P(a) \leftrightarrow \forall b.\ b \in c \to b \in a$
- (REPL$_{\phi(a,b,\vec{f})}$) $\forall \vec{f}, a \forall c.\ c \in_I R_{a,b,\vec{f}.\ \phi(a,b,\vec{f})}(a, \vec{f}) \leftrightarrow (\forall x \in a \exists ! y.\ \phi(x, y, \vec{f})) \wedge (\exists x \in a.\ \phi(x, c, \vec{f}))$
- (IND$_{\phi(a,\vec{f})}$) $\forall \vec{f}.(\forall a.(\forall b \in_I a.\ \phi(b, \vec{f})) \to \phi(a, \vec{f})) \to \forall a.\ \phi(a, \vec{f})$

Extensionality and IZF$_R$

There are two axioms seemingly missing from IZF$_R$. The first one is the Leibniz axiom schema:

$$(\mathrm{L}_\phi)\ \forall a, b, \vec{f}.\ a = b \wedge \phi(a, \vec{f}) \to \phi(b, \vec{f})$$

The second is Extensionality:

$$(\mathrm{EXT})\ \forall a, b.\ a = b \leftrightarrow \forall c.\ c \in a \leftrightarrow c \in b$$

Their presence in set theories is the reason for incorporation of the relational symbol $\in_I$ in our logic along with (IN) and (EQ) axioms. We will now show that both Leibniz and Extensionality axioms can be derived. Moreover, we shall see that the rest of the axioms of IZF$_R$ hold with $\in_I$ replaced by the more familiar $\in$. Therefore, a user of our presentation of IZF$_R$ does not need to worry about $\in_I$, as the standard presentation is derivable.

From now on in this section, we work in IZF$_R$. The following sequence of lemmas establishes that equality and membership behave in the correct way. Statements similar in spirit are also proved in the context of Boolean-valued models.

Our treatment slightly simplifies the standard presentation by avoiding the need for mutual induction.

Lemma 3.2.1 *For all a, $a = a$.*

Proof By $\in$-induction on a. Take any $b \in_I a$. By the induction hypothesis, $b = b$, so also $b \in a$. ■

Corollary 3.2.2 *If $a \in_I b$, then $a \in b$.*

Lemma 3.2.3 *For all a, b, if $a = b$, then $b = a$.*

Proof Straightforward. ■

Lemma 3.2.4 *For all b, a, c, if $a = b$ and $b = c$, then $a = c$.*

Proof By $\in$-induction on b. First take any $d \in_I a$. By $a = b$, $d \in b$, so there is $e \in_I b$ such that $d = e$. By $b = c$, $e \in c$, so there is $f \in_I c$ such that $e = f$. By the induction hypothesis for e, $d = f$, so $d \in c$.

The other direction is symmetric and proceeds from c to a. Take any $d \in_I c$. By $b = c$, $d \in b$, so there is $e \in_I b$ such that $d = e$. By $a = b$, $e \in a$, so there is $f \in_I a$ such that $e = f$. The induction hypothesis gives the claim. ■

Lemma 3.2.5 *For all a, b, c, if $a \in c$ and $a = b$, then $b \in c$.*

Proof Since $a \in c$, there is $d \in_I c$ such that $a = d$. By previous lemmas we also have $b = d$, so $b \in c$. ■

Lemma 3.2.6 *For all a, b, d, if $a = b$ and $d \in a$, then $d \in b$.*

Proof Suppose $d \in a$, then there is e such that $e \in_I a$ and $d = e$. By $a = b$, $e \in b$. By Lemma 3.2.5, $d \in b$. ■

Lemma 3.2.7 (Extensionality) *If for all d, $d \in a$ iff $d \in b$, then $a = b$.*

Proof Take any $d \in_I a$. By Corollary 3.2.2 $d \in a$, so by Lemma 3.2.6 also $d \in b$. The other direction is symmetric. ■

All the lemmas above have been mechanically verified, with the help of a toy prover we wrote to experiment with IZF_R.

Lemma 3.2.8 (The Leibniz axiom) *For any term $t(a, \vec{f})$ and formula $\phi(a, \vec{f})$ not containing $\in_I$, if $a = b$, then $t(a, \vec{f}) = t(b, \vec{f})$ and $\phi(a, \vec{f}) \leftrightarrow \phi(b, \vec{f})$.*

Proof Straightforward mutual induction on generation of t and ϕ. We show some representative cases. Case t or ϕ of:

- $\bigcup t_1(a)$. If $c \in_I \bigcup t_1(a)$, then for some d, $c \in d \in t_1(a)$. By the induction hypothesis $t_!(a) = t_1(b)$, so by Lemma 3.2.6 $d \in t_1(b)$, so $c \in_I \bigcup t_1(b)$ and by Corollary 3.2.2 also $c \in \bigcup t_1(b)$. The other direction is symmetric and by the (EQ) axiom we get $t(a) = t(b)$.
- $S_{a,\vec{f}}.\ \phi(a, \vec{f})(t_1(a), \vec{u}(a))$. If $c \in_I S_{a,\vec{f}}.\ \phi(a, \vec{f})(t_1(a), \vec{u}(a))$, then $c \in t_1(a)$ and $\phi(c, \vec{u}(a))$. By the induction hypothesis, $t_1(a) = t_1(b)$, $\vec{u}(a) = \vec{u}(b)$, and thus $\phi(c, \vec{u}(b))$ and $c \in t_1(b)$, so $c \in_I S_{a,\vec{f}}.\ \phi(a, \vec{f})(t_1(b), \vec{u}(b))$ and also $c \in S_{a,\vec{f}}.\ \phi(a, \vec{f})(t_1(b), \vec{u}(b))$.
- $t(a) \in s(a)$. By the induction hypothesis, $t(a) = t(b)$ and $s(a) = s(b)$. Thus by Lemma 3.2.6 $t(a) \in s(b)$ and by Lemma 3.2.5 $t(b) \in s(b)$.
- $\forall c.\ \phi(c, a, \vec{f})$. Take any c, we have $\phi(c, a, \vec{f})$, so by induction hypothesis $\phi(c, b, \vec{f})$, so $\forall c.\ \phi(c, b, \vec{f})$. ■

Lemma 3.2.9 *For any term $t_A(\vec{a})$, $c \in t_A(\vec{a})$ iff $\phi_A(c, \vec{a})$.*

Proof The right-to-left direction follows immediately by Corollary 3.2.2. For the left-to-right direction, suppose $c \in t_A(\vec{a})$. Then there is d such that $d \in_I t_A(\vec{a})$ and $c = d$. Therefore $\phi_A(d, \vec{a})$ holds and by the Leibniz axiom we also get $\phi_A(c, \vec{a})$. ■

Lemma 3.2.10 *For any axiom A of IZF_R, $IZF_R \vdash A[\in_I := \in]$.*

Proof Lemma 3.2.9 shows the claim for all the axioms apart from $\in$-Induction. So suppose $\forall a.\ (\forall b \in a.\ \phi(b, \vec{f})) \to \phi(a, \vec{f})$. We need to show $\forall a.\ \phi(a, \vec{f})$. We proceed by $\in_I$-induction on a. It suffices to show $\forall c.\ (\forall d \in_I c.\ \phi(d, \vec{f})) \to \phi(c, \vec{f})$. Take any c and suppose $\forall d \in_I c.\ \phi(d, \vec{f})$. We need to show $\phi(c, \vec{f})$. Take a to be c in the assumption, so it suffices to show that $\forall b \in c.\ \phi(b, \vec{f})$. Take any $b \in c$. Then there is $e \in_I c$ such that $e = b$. By the induction hypothesis $\phi(e, \vec{f})$ holds and hence by the Leibniz axiom we get $\phi(b, \vec{f})$, which shows the claim. ■

Therefore a user of IZF$_R$ can ignore the $\in_I$ symbol and thanks to Lemma 3.2.10 use the $\in_I$-free axiomatization.

The Replacement axiom

A more familiar formulation of Replacement could be: "For all $\vec{F}, A$, if for all $x \in A$ there is exactly one y such that $\phi(x, y, \vec{F})$ holds, then there is a set D such that $\forall x \in A \exists y \in D.\ \phi(x, y, \vec{F})$ and for all $d \in D$ there is $x \in A$ such that $\phi(x, d, \vec{F})$". Let this formulation of Replacement be called (REPL0$_\phi$), let (R_ϕ) be the term-free statement of our Replacement axiom, that is:

$$(R_\phi) \equiv \forall \vec{f}, a \exists ! d.\ \forall c.\ c \in d \leftrightarrow (\forall x \in a \exists ! y.\ \phi(x, y, \vec{f})) \land (\exists x \in a.\ \phi(x, c, \vec{f}))$$

and let IZ denote IZF$_R$ without the Replacement axiom and corresponding function symbols. To justify our definition of Replacement, we prove the following two lemmas:

Lemma 3.2.11 *$IZ \vdash (R_\phi) \to (REPL0_\phi)$.*

Proof Assume (R$_\phi$), take any $\vec{F}, A$ and suppose that for all $x \in A$ there is exactly one y such that $\phi(x, y, \vec{F})$. Let D be the set we get by applying (R_ϕ). Take any

$x \in A$, then there is y such that $\phi(x, y, \vec{F})$, so $y \in D$. Moreover, if $d \in D$ then there is $x \in A$ such that $\phi(x, d, \vec{F})$. This shows (REPL0$_\phi$). ■

Lemma 3.2.12 *IZ* $\vdash$ *(REPL0$_\phi$)* $\to$*(R$_\phi$).*

Proof Assume (REPL0$_\phi$), take any $\vec{F}, A$ and consider the set

$$B \equiv \{a \in A \mid \forall x \in A \exists ! y.\ \phi(x, y, \vec{F})\}.$$

Then for all $b \in B$ there is exactly one y such that $\phi(b, y, \vec{F})$. Use (REPL0$_\phi$) to get a set D. Then D is the set we are looking for. Indeed, if $d \in D$, then there is $b \in B$ such that $\phi(b, d, \vec{F})$ and so by the definition of B, $\forall x \in A \exists ! y.\ \phi(x, y, \vec{F})$ and $b \in A$. On the other hand, take any d and suppose that $\forall x \in A \exists ! y.\ \phi(x, y, \vec{F})$ and there is $x \in A$ such that $\phi(x, d, \vec{F})$. Then $x \in B$, so there is $y' \in D$ such that $\phi(x, y', \vec{F})$. But y' must be equal to d, so $d \in D$. As it is trivial to see that D is unique, the claim follows. ■

The terms of IZF$_R$

The original presentation of IZF with Replacement presented in [Myh73] is term-free. Let us call it IZF$_{R0}$. We will now show that IZF$_R$ is a definitional extension of IZF$_{R0}$.

In IZF$_{R0}$ for each axiom (A) among the Empty Set, Pairing, Infinity, Separation, Replacement, Union and Power Set axioms, we can derive $\forall \vec{a} \exists ! d \forall c.\ c \in d \leftrightarrow \phi_A(c, \vec{a})$, using Lemma 3.2.12 in case of the Replacement axiom. We therefore definitionally extend IZF$_{R0}$, by introducing for each such (A) the corresponding new function symbol $t_A(\vec{a})$ along with the defining axiom $\forall \vec{a} \forall c.\ c \in t_A(\vec{a}) \leftrightarrow \phi_A(c, \vec{a})$.

We then need to provide the Separation and Replacement function symbols $R_{a,b,\vec{f}.\ \phi(a,b,\vec{f})}$ and $S_{a,\vec{f}.\ \phi(a,\vec{f})}$, where ϕ may contain the new terms. To fix our attention, consider the Separation axiom. For some function symbol $S_{a,\vec{f}.\ \phi(a,\vec{f})}$, we need to have:

$$\forall \vec{f}, a \forall c.\ c \in S_{a,\vec{f}.\ \phi(a,\vec{f})}(a, \vec{f}) \leftrightarrow c \in a \wedge \phi(c, \vec{f})$$

As all terms present in ϕ were introduced via a definitional extension of IZF$_{R0}$, there is a term-free formula ϕ' equivalent to ϕ. We therefore have:

$$\forall \vec{f}, a \forall c.\ c \in S_{a,\vec{f}.\ \phi'(a,\vec{f})}(a, \vec{f}) \leftrightarrow c \in a \wedge \phi'(c, \vec{f})$$

and consequently:

$$\forall \vec{f}, a \forall c.\ c \in S_{a,\vec{f}.\ \phi'(a,\vec{f})}(a, \vec{f}) \leftrightarrow c \in a \wedge \phi(c, \vec{f})$$

We define $S_{a,\vec{f}.\ \phi(a,\vec{f})}$ to be $S_{a,\vec{f}.\ \phi'(a,\vec{f})}$. Similarly, we can define $R_{a,b,\vec{f}.\ \phi(a,b,\vec{f})}$ to be $R_{a,b,\vec{f}.\ \phi'(a,b,\vec{f})}$. After iterating this process ω-many times, we obtain all instances of terms and axioms (A) present in IZF$_R$.

In order to finish the demonstration that IZF$_R$ is a definitional extension of IZF$_{R0}$, it remains to justify the instances of the Leibniz and $\in$-Induction axioms, when the parameterizing formula contains terms. For the Leibniz axiom, take any $A, B, \vec{F}$ and suppose $A = B$ and $\phi(A, \vec{F})$. Then there is a term-free formula ϕ' equivalent to ϕ, so also $\phi'(A, \vec{F})$. By the Leibniz axiom in IZF$_{R0}$, $\phi'(B, \vec{F})$, so also $\phi(B, \vec{F})$.

For the $\in$-Induction axiom, take any $\vec{F}$ and suppose:

$$\forall a.\ (\forall b \in a.\ \phi(b, \vec{F})) \to \phi(a, \vec{F})$$

Taking ϕ' to be the term-free formula equivalent to ϕ, we get:

$$\forall a.\ (\forall b \in a.\ \phi'(b, \vec{F})) \to \phi'(a, \vec{F})$$

By $\in$-Induction in IZF$_{R0}$, we get $\forall a.\ \phi'(a, \vec{F})$, thus also $\forall a.\ \phi(a, \vec{F})$.

3.2.2 λZ calculus

The lambda calculus λZ for IZF_R can be seen as a generalization of ideas underlying λH and λS. The first-order part of the calculus is the same as the first-order part of λH. As the axioms of IZF_R have a similar form to the comprehension axiom in HAS, similar Prop and Rep terms are used. Finally, the terms corresponding to the induction axiom behave similarly to the induction terms in λH.

We proceed with the formal definitions. As this calculus is the culmination of developments in the previous sections, we present it in its entirety.

The lambda terms in λZ will be denoted by letters M, N, O, P. There are two kinds of lambda abstraction in λZ, one corresponding to the proofs of implication, the other to the proofs of universal quantification. We use separate sets of variables for these abstractions and call them *proof-* and *first-order* variables, respectively. Letters x, y, z will be used for the proof variables and letters a, b, c for the first-order variables. Letters t, s, u are reserved for IZF_R terms. The types in the system are IZF_R formulas. The terms are generated by the following abstract grammar. The first group of terms is standard and comes from the first-order logic:

$$\begin{aligned} M \quad ::= \quad & x \mid M\ N \mid \lambda a.\ M \mid \lambda x : \phi.\ M \mid \mathrm{inl}(M) \mid \mathrm{inr}(M) \mid \mathrm{fst}(M) \mid \mathrm{snd}(M) \\ & [t, M] \mid M\ t \mid \langle M, N \rangle \mid \mathrm{case}(M, x : \phi.\ N, x : \psi.\ O) \\ & \mathrm{magic}(M) \mid \mathrm{let}\ [a, x : \phi] := M\ \mathrm{in}\ N \end{aligned}$$

The second group of terms corresponds to the IZF_R axioms:

$$\mathrm{inProp}(t, u, M) \mid \mathrm{inRep}(t, u, M)$$

$$\mathrm{eqProp}(t, u, M) \mid \mathrm{eqRep}(t, u, M)$$

$$\mathrm{pairProp}(t, u_1, u_2, M) \mid \mathrm{pairRep}(t, u_1, u_2, M)$$

$$\mathrm{unionProp}(t, u, M) \mid \mathrm{unionRep}(t, u, M)$$

$$\mathrm{sep}_{a,\vec{f}.\ \phi(a,\vec{f})}\mathrm{Prop}(t,u,\vec{u},M) \mid \mathrm{sep}_{a,\vec{f}.\ \phi(a,\vec{f})}\mathrm{Rep}(t,u,\vec{u},M)$$

$$\mathrm{powerProp}(t,u,M) \mid \mathrm{powerRep}(t,u,M)$$

$$\mathrm{infProp}(t,M) \mid \mathrm{infRep}(t,M)$$

$$\mathrm{repl}_{a,b,\vec{f}.\ \phi(a,b,\vec{f})}\mathrm{Prop}(t,u,\vec{u},M) \mid \mathrm{repl}_{a,b,\vec{f}.\ \phi(a,b,\vec{f})}\mathrm{Rep}(t,u,\vec{u},M)$$

$$\mathrm{ind}_{a,\vec{f}.\ \phi(a,\vec{f})}(M,\vec{t})$$

The ind terms correspond to the (IND) axiom and Prop and Rep terms correspond to the respective axioms of IZF_R. As in case of HAS, we adopt a convention of using axRep and axProp terms to tacitly mean all Rep and Prop terms, for ax being one of in, eq, pair, union, sep, power, inf, repl, unless we list some of them separately. With this convention in mind, we can summarize the definition of the Prop and Rep terms as:

$$\mathrm{axProp}(t,\vec{u},M) \mid \mathrm{axRep}(t,\vec{u},M),$$

where the number of terms in the sequence $\vec{u}$ depends on the particular axiom.

The variables in λ, case and let terms bind respective terms. We denote all free variables of a term M by $FV(M)$ and the free first-order variables of a term by $FV_F(M)$. The free (first-order) variables of a context Γ are denoted by $FV(\Gamma)$ ($FV_F(\Gamma)$) and defined in a natural way.

The deterministic reduction relation $\to$ arises from the following reduction rules and evaluation contexts:

$$(\lambda x : \phi.\ M)\ N \to M[x := N] \qquad (\lambda a.\ M)\ t \to M[a := t]$$

$$\mathrm{fst}(\langle M, N\rangle) \to M \qquad \mathrm{snd}(\langle M, N\rangle) \to N$$

$$\mathrm{case}(\mathrm{inl}(M), x : \phi.\ N, x : \psi.\ O) \to N[x := M]$$

$$\mathrm{case}(\mathrm{inr}(M), x : \phi.\ N, x : \psi.\ O) \to O[x := M]$$

$$\text{let } [a, x : \phi] := [t, M] \text{ in } N \to N[a := t][x := M]$$

$$\text{axProp}(t, \vec{u}, \text{axRep}(t, \vec{u}, M)) \to M$$

$$\text{ind}_{a,\vec{f}.\ \phi(a,\vec{f})}(M, \vec{t}) \to \lambda c.\ M\ c\ (\lambda b.\lambda x : b \in_I c.\ \text{ind}_{a,\vec{f}.\ \phi(a,\vec{f})}(M, \vec{t})\ b)$$

In the reduction rules for ind terms, the variable x is new.

The evaluation contexts continue to describe the lazy evaluation order:

$$\begin{array}{lll} [\circ] & ::= & \text{fst}([\circ]) \mid \text{snd}([\circ]) \mid \text{case}([\circ], x.N, x.O) \\ & & \text{axProp}(t, \vec{u}, [\circ]) \mid \text{let } [a, x : \phi] := [\circ] \text{ in } N \mid [\circ]\ M \mid \text{magic}([\circ]) \end{array}$$

The values of λZ are generated by the following abstract grammar, where M is an arbitrary term.

$$V ::= \lambda a.\ M \mid \lambda x : \phi.\ M \mid \text{inr}(M) \mid \text{inl}(M) \mid [t, M] \mid \langle M, N \rangle \mid \text{axRep}(t, \vec{u}, M)$$

The type system for λZ follows. Types are IZF_R formulas, and terms are λZ terms. Contexts Γ are finite sets of pairs (x_i, ϕ_i). The first set of rules corresponds to the first-order logic.

$$\frac{}{\Gamma, x : \phi \vdash x : \phi} \qquad \frac{\Gamma \vdash M : \phi \to \psi \quad \Gamma \vdash N : \phi}{\Gamma \vdash M\ N : \psi} \qquad \frac{\Gamma, x : \phi \vdash M : \psi}{\Gamma \vdash \lambda x : \phi.\ M : \phi \to \psi}$$

$$\frac{\Gamma \vdash M : \phi \quad \Gamma \vdash N : \psi}{\Gamma \vdash \langle M, N \rangle : \phi \wedge \psi} \qquad \frac{\Gamma \vdash M : \phi \wedge \psi}{\Gamma \vdash \text{fst}(M) : \phi} \qquad \frac{\Gamma \vdash M : \phi \wedge \psi}{\Gamma \vdash \text{snd}(M) : \psi}$$

$$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash \text{inl}(M) : \phi \vee \psi} \qquad \frac{\Gamma \vdash M : \psi}{\Gamma \vdash \text{inr}(M) : \phi \vee \psi}$$

$$\frac{\Gamma \vdash M : \phi \vee \psi \quad \Gamma, x : \phi \vdash N : \vartheta \quad \Gamma, x : \psi \vdash O : \vartheta}{\Gamma \vdash \text{case}(M, x : \phi.\ N, x : \psi.\ O) : \vartheta}$$

$$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash \lambda a.\ M : \forall a.\ \phi}\ a \notin FV_F(\Gamma) \qquad \frac{\Gamma \vdash M : \forall a.\ \phi}{\Gamma \vdash M\ t : \phi[a := t]} \qquad \frac{\Gamma \vdash M : \phi[a := t]}{\Gamma \vdash [t, M] : \exists a.\ \phi}$$

$$\frac{\Gamma \vdash M : \bot}{\Gamma \vdash \text{magic}(M) : \phi} \qquad \frac{\Gamma \vdash M : \exists a.\ \phi \quad \Gamma, x : \phi \vdash N : \psi}{\Gamma \vdash \text{let } [a, x : \phi] := M \text{ in } N : \psi}\ a \notin FV_F(\Gamma, \psi)$$

The rest of the rules correspond to IZF_R axioms:

$$\frac{\Gamma \vdash M : \forall d.\ (d \in_I t \to d \in u) \land (d \in_I u \to d \in t)}{\Gamma \vdash \mathrm{eqRep}(t, u, M) : t = u}$$

$$\frac{\Gamma \vdash M : t = u}{\Gamma \vdash \mathrm{eqProp}(t, u, M) : \forall d.\ (d \in_I t \to d \in u) \land (d \in_I u \to d \in t)}$$

$$\frac{\Gamma \vdash M : \exists c.\ c \in_I u \land t = c}{\Gamma \vdash \mathrm{inRep}(t, u, M) : t \in u} \qquad \frac{\Gamma \vdash t \in u}{\Gamma \vdash \mathrm{inProp}(t, u, M) : \exists c.\ c \in_I u \land t = c}$$

$$\frac{\Gamma \vdash M : \phi_A(t, \vec{u})}{\Gamma \vdash \mathrm{axRep}(t, \vec{u}, M) : t \in_I t_A(\vec{u})} \qquad \frac{\Gamma \vdash M : t \in_I t_A(\vec{u})}{\Gamma \vdash \mathrm{axProp}(t, \vec{u}, M) : \phi_A(t, \vec{u})}$$

$$\frac{\Gamma \vdash M : \forall c.\ (\forall b.\ b \in_I c \to \phi(b, \vec{t})) \to \phi(c, \vec{t})}{\Gamma \vdash \mathrm{ind}_{a,\vec{f}.\ \phi(a,\vec{f})}(M, \vec{t}) : \forall a.\ \phi(a, \vec{t})}$$

As expected, we have the correspondence between IZF_R and λZ:

Lemma 3.2.13 *If $\Gamma \vdash O : \phi$ then IZF_R+$rg(\Gamma) \vdash \phi$, where $rg(\Gamma) = \{\phi \mid (x, \phi) \in \Gamma\}$. If IZF_R+$\Gamma \vdash \phi$, then there exists a term M such that $\overline{\Gamma} \vdash M : \phi$, where $\overline{\Gamma} = \{(x_\phi, \phi) \mid \phi \in \Gamma\}$.*

Proof Both parts follow by easy induction on the proof. The first part is straightforward, to get the claim simply erase the lambda terms from the proof tree. For the second part, we show terms and trees corresponding to IZF_R axioms:

- Let ϕ be one of the IZF_R axioms apart from $\in$-Induction. Then $\phi = \forall \vec{a}.\ \forall c.\ c \in_I t_A(\vec{a}) \leftrightarrow \phi_A(c, \vec{a})$ for the axiom (A). Recall that $\phi_1 \leftrightarrow \phi_2$ is an abbreviation for $(\phi_1 \to \phi_2) \land (\phi_2 \to \phi_1)$. Let T be the following proof tree:

$$\dfrac{\dfrac{\Gamma, x : \phi_A(c, \vec{a}) \vdash x : \phi_A(c, \vec{a})}{\Gamma, x : \phi_A(c, \vec{a}) \vdash \mathrm{axRep}(c, \vec{a}, x) : c \in_I t_A(\vec{a})}}{\Gamma \vdash \lambda x : \phi_A(c, \vec{a}).\ \mathrm{axRep}(c, \vec{a}, x) : \phi_A(c, \vec{a}) \to c \in_I t_A(\vec{a})}$$

  Let $M \equiv \langle \lambda x : c \in_I t_A(\vec{a}).\ \mathrm{axProp}(c, \vec{a}, x), \lambda x : \phi_A(c, \vec{a}).\ \mathrm{axRep}(c, \vec{a}, x)\rangle$. Then the following proof tree shows the claim:

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, x : c \in_I t_A(\vec{a}) \vdash x : c \in_I t_A(\vec{a})}{\Gamma, x : c \in_I t_A(\vec{a}) \vdash \mathrm{axProp}(c, \vec{a}, x) : \phi_A(c, \vec{a})}}{\Gamma \vdash \lambda x : c \in_I t_A(\vec{a}).\ \mathrm{axProp}(c, \vec{a}, x) : c \in_I t_A(\vec{a}) \to \phi_A(c, \vec{a})} \qquad T}{\Gamma \vdash M : c \in_I t_A(\vec{a}) \leftrightarrow \phi_A(c, \vec{a})}}{\Gamma \vdash \lambda \vec{a} \lambda c. M : \forall \vec{a}.\ \forall c.\ c \in_I t_A(\vec{a}) \leftrightarrow \phi_A(c, \vec{a})}$$

- Let ϕ be the $\in$-induction axiom. Let $M = \lambda \vec{f} \lambda x : (\forall a.(\forall b.\ b \in a \to \psi(b, \vec{f})) \to \psi(a, \vec{f})).\ \mathrm{ind}_{\psi(a,\vec{f})}(\vec{f}, x)$. The following proof tree shows the claim:

$$\dfrac{\dfrac{\Gamma, x : \forall a.(\forall b.\ b \in_I a \to \psi(b, \vec{f})) \to \psi(a, \vec{f}) \vdash x : \forall a.(\forall b.\ b \in_I a \to \psi(b, \vec{f})) \to \psi(a, \vec{f})}{\Gamma, x : \forall a.(\forall b.\ b \in_I a \to \phi(b, \vec{f})) \to \psi(a, \vec{f}) \vdash \mathrm{ind}_{\psi(a,\vec{f})}(\vec{f}, x) : \forall a.\ \psi(a, \vec{f})}}{\Gamma \vdash M : \forall \vec{f}.(\forall a.(\forall b.\ b \in_I a \to \psi(b, \vec{f})) \to \psi(a, \vec{f})) \to \forall a.\ \psi(a, \vec{f})}$$

■

3.2.3 Properties of λZ

The properties are extended from λH in an entirely predictable way. Note that we do not have to worry about the $=_\omega$ relation anymore, because there are no number terms and corresponding reduction rules in the logic.

Lemma 3.2.14 (Inversion) *Suppose $\Gamma \vdash Q : \Psi$. Suppose Q is of the form:*

- $\mathrm{inRep}(t, u, M)$. *Then $\Psi = t \in u$ and $\Gamma \vdash M : \exists c.\ c \in_I u \land t = c$.*
- $\mathrm{inProp}(t, u, M)$. *Then $\Psi = \exists c.\ c \in_I u \land t = c$ and $\Gamma \vdash M : t \in u$.*
- $\mathrm{eqRep}(t, u, M)$. *Then $\Psi = t = u$ and $\Gamma \vdash M : \forall d.\ (d \in_I t \to d \in u) \land (d \in_I u \to d \in t)$.*
- $\mathrm{eqProp}(t, u, M)$. *Then $\Psi = \forall d.\ (d \in_I t \to d \in u) \land (d \in_I u \to d \in t)$ and $\Gamma \vdash M : \forall d.\ (d \in_I t \to d \in u) \land (d \in_I u \to d \in t)$.*
- $\mathrm{axRep}(t, \vec{u}, M)$. *Then $\Psi = t \in_I t_A(\vec{u})$ and $\Gamma \vdash M : \phi_A(t, \vec{u})$.*
- $\mathrm{axProp}(t, \vec{u}, M)$. *Then $\Psi = \phi_A(t, \vec{u})$ and $\Gamma \vdash M : t \in_I t_A(\vec{u})$.*
- $\mathrm{ind}_{a,\vec{f}.\ \phi(a,\vec{f})}(M, \vec{t})$. *Then $\Psi = \forall a.\ \phi(a, \vec{t})$ and $\Gamma \vdash M : \forall c.\ (\forall b.\ b \in_I c \to \phi(b, \vec{t})) \to \phi(c, \vec{t})$.*

Lemma 3.2.15 (Canonical forms) *Suppose Q is a value, $\vdash Q : \Psi$ and Ψ is of the form:*

- $t \in_I t_A(\vec{u})$. *Then* $Q = \text{axRep}(t, \vec{u}, M)$.
- $t \in t_A(\vec{u})$. *Then* $Q = \text{inRep}(t, \vec{u}, M)$.
- $t = u$. *Then* $Q = \text{eqRep}(t, u, M)$.

The propositional and first-order Substitution Lemmas continue to hold, together with Weakening.

Lemma 3.2.16 (Subject Reduction, Preservation) *If* $\Gamma \vdash P : \Psi$ *and* $P \to Q$, *then* $\Gamma \vdash Q : \Psi$.

Proof By induction on the definition of $O \to P$. As usual, we show the new cases. Case $O \to P$ of:

- $\text{axProp}(t, \vec{u}, \text{axRep}(t, \vec{u}, M_1)) \to M_1$. The proof tree must end with:
$$\frac{\dfrac{\Gamma \vdash M_1 : \phi_A(t, \vec{u})}{\Gamma \vdash \text{axRep}(t, \vec{u}, M_1)) : t \in_I t_A(\vec{u})}}{\Gamma \vdash \text{axProp}(t, \vec{u}, \text{axRep}(t, \vec{u}, M_1)) : \phi_A(t, \vec{u})}$$

  The claim follows immediately.

- $\text{ind}_{a,\vec{f}.\ \psi(a,\vec{f})}(M_1, \vec{t}) \to \lambda c.\ M_1\ c\ (\lambda b.\lambda x : b \in_I c.\ \text{ind}_{a,\vec{f}.\ \psi(a,\vec{b})}(M_1, \vec{t})\ b)$. The proof tree must end with:
$$\frac{\Gamma \vdash M_1 : \forall c.\ (\forall b.\ b \in_I c \to \psi(b, \vec{t})) \to \psi(c, \vec{t})}{\Gamma \vdash \text{ind}_{a,\vec{f}.\ \psi(a,\vec{f})}(M_1, \vec{t}) : \forall a.\ \psi(a, \vec{t})}$$

  We choose b, c, x to be fresh. By applying α-conversion we can also obtain a proof tree of $\Gamma \vdash M_1 : \forall e.\ (\forall d.\ d \in_I e \to \psi(d, \vec{t})) \to \psi(e, \vec{t})$, where $\{d, e\} \cap \{b, c\} = \emptyset$. Then by Weakening we get $\Gamma, x : b \in_I c \vdash M_1 : \forall e.\ (\forall d.\ d \in_I e \to \psi(d, \vec{t})) \to \psi(e, \vec{t})$, so also $\Gamma, x : b \in_I c \vdash \text{ind}_{a,\vec{f}.\ \psi(a,\vec{b})}(M_1, \vec{t}) : \forall a.\ \psi(a, \vec{t})$. Let the proof tree T be defined as:
$$\frac{\dfrac{\dfrac{\Gamma, x : b \in_I c \vdash \text{ind}_{a,\vec{f}.\ \psi(a,\vec{b})}(M_1, \vec{t}) : \forall a.\ \psi(a, \vec{t})}{\Gamma, x : b \in_I c \vdash \text{ind}_{a,\vec{f}.\ \psi(a,\vec{b})}(M_1, \vec{t})\ b : \psi(b, \vec{t})}}{\Gamma \vdash \lambda x : b \in_I c.\ \text{ind}_{a,\vec{f}.\ \psi(a,\vec{b})}(M_1, \vec{t})\ b : b \in_I c \to \psi(b, \vec{t})}}{\Gamma \vdash \lambda b.\lambda x : b \in_I c.\ \text{ind}_{a,\vec{f}.\ \psi(a,\vec{b})}(M_1, \vec{t})\ b : \forall b.\ b \in_I c \to \psi(b, \vec{t})}$$

Then the following proof tree shows the claim:

$$\dfrac{\dfrac{\dfrac{\Gamma \vdash M_1 : \forall c.\ (\forall b.\ b \in_I c \to \psi(b, \vec{t})) \to \psi(c, \vec{t})}{\Gamma \vdash M_1\ c : (\forall b.\ b \in_I c \to \psi(b, \vec{t})) \to \psi(c, \vec{t})} \quad T}{\Gamma \vdash M_1\ c\ (\lambda b.\lambda x : b \in_I c.\ \mathrm{ind}_{a,\vec{f}.\ \psi(a,\vec{b})}(M_1, \vec{t})\ b) : \psi(c, \vec{t})}}{\Gamma \vdash \lambda c.\ M_1\ c\ (\lambda b.\lambda x : b \in_I c.\ \mathrm{ind}_{a,\vec{f}.\ \psi(a,\vec{b})}(M_1, \vec{t})\ b) : \forall c.\ \psi(c, \vec{t})}$$

■

Lemma 3.2.17 (Progress) *If $\vdash P : \Psi$ then either P is a value or there is Q such that $P \to Q$.*

Proof Induction on the length of P. Case P of:

- $\mathrm{axProp}(t, \vec{u}, O)$. By Inversion, the proof tree ends with:

  $$\dfrac{\vdash O : t \in_I t_A(\vec{u})}{\vdash \mathrm{axProp}(t, \vec{u}, O) : \phi_A(t, \vec{u})}$$

  By the induction hypothesis, either O is a value or there is O_1 such that $O \to O_1$. In the former case, by Canonical Forms, $O = \mathrm{axRep}(t, \vec{u}, P)$ and $M \to P$. In the latter, by the evaluation rules $\mathrm{axProp}(t, \vec{u}, O) \to \mathrm{axProp}(t, \vec{u}, O_1)$.
- $\mathrm{axRep}(t, \vec{u}, O)$ is a value.
- The cases where P is a term corresponding to the equality and membership axioms work in the same way.
- The ind terms always reduce. ■

3.2.4 Realizability for $\mathbf{IZF_R}$

Realizability for IZF was born and at the same time used for an amazing variety of applications in the Ph. D. thesis of David McCarty [McC84], written under the direction of Dana Scott. The definition and the thesis were a major inspiration for our proof of normalization of λZ.

McCarty's realizability relation is presented in a conventional way. It relates natural numbers, interpreted as pairs, Turing machine indices and other constructs, with set-theoretic formulas. Our definition, as usual, uses lambda terms as vehicles for computational content. It seems that the definitions behave mostly similarly from the computational point of view. We conjecture, however, that if a formal correspondence were to be defined and stated, some differences would show up in the treatment of existential quantifier.

As usual, we start with the erasure map. In λZ, reductions are *set-oblivious* — the set terms do not play any role in reductions. We therefore erase them to $\emptyset$. The erasure map is induced by the following cases:

$$\overline{\mathrm{axRep}(t, \vec{u}, M)} = \mathrm{axRep}(\overline{M}) \qquad \overline{\mathrm{axProp}(t, \vec{u}, M)} = \mathrm{axProp}(\overline{M})$$

$$\overline{\mathrm{ind}_{a, \vec{f}.\ \phi(a, \vec{f})}(M, \vec{t})} = \mathrm{ind}(\overline{M})$$

$$\overline{\lambda x : \phi.\ M} = \lambda x.\ \overline{M} \qquad \overline{\mathrm{let}\ [a, x : \phi] := M \ \mathrm{in}\ N} = \mathrm{let}\ [a, x] := \overline{M} \ \mathrm{in}\ \overline{N}$$

$$\overline{\mathrm{case}(M, x : \phi.\ N, x : \psi.\ O)} = \mathrm{case}(\overline{M}, x.\overline{N}, x.\overline{O})$$

The erasure on the rest of the terms is defined in a natural way, for example $\overline{\langle M, N \rangle} = \langle \overline{M}, \overline{N} \rangle$, $\overline{[t, M]} = [\emptyset, \overline{M}]$ and $\overline{M\ t} = \overline{M}\ \emptyset$. As before, we call the closed terms of $\lambda\overline{Z}$ *realizers*. The set of all realizers will be denoted by $\lambda\overline{Z}_c$ and the set of all $\lambda\overline{Z}$ values which are realizers will be denoted by $\lambda\overline{Z}_{vc}$.

Lemma 3.2.18 *If $\overline{M}$ normalizes, so does M.*

Having defined realizers, we proceed to define the realizability relation. Just as we used λ-sets in HAS, we use λ-*names* in IZF_R:

Definition 3.2.19 *A set A is a λ-name iff A is a set of pairs (v, B) such that $v \in \lambda\overline{Z}_{vc}$ and B is a λ-name.*

In other words, λ-names are sets hereditarily labelled by $\lambda\overline{Z}$ values which are realizers.

Definition 3.2.20 *The class of λ-names is denoted by V^λ.*

Thus, any λ-name has the form:

$$\{(v, A) \mid v \in \lambda\overline{Z}_{vc} \wedge A \in V^\lambda\}$$

Formally, V^λ is generated by the following transfinite inductive definition on ordinals:

$$V^\lambda_\alpha = \bigcup_{\beta<\alpha} P(\lambda\overline{Z}_{vc} \times V^\lambda_\beta) \qquad V^\lambda = \bigcup_{\alpha\in\mathrm{ORD}} V^\lambda_\alpha$$

Definition 3.2.21 *The* λ-rank *of a λ-name A, denoted by $\lambda rk(A)$, is the smallest α such that $A \in V^\lambda_\alpha$.*

We now define three auxiliary relations between realizers and pairs of sets in V^λ, which we write as $M \Vdash A \in_I B$, $M \Vdash A \in B$, $M \Vdash A = B$. These relations are a prelude to the definition of realizability.

$$\begin{aligned}
M \Vdash A \in_I B \quad &\equiv \quad M \downarrow v \wedge (v, A) \in B \\
M \Vdash A \in B \quad &\equiv \quad M \downarrow \mathrm{inRep}(N) \wedge N \downarrow [\emptyset, O] \wedge \exists C \in V^\lambda.\ O \downarrow \langle O_1, O_2\rangle \wedge \\
& \qquad O_1 \Vdash C \in_I B \wedge O_2 \Vdash A = C \\
M \Vdash A = B \quad &\equiv \quad M \downarrow \mathrm{eqRep}(M_0) \wedge M_0 \downarrow \lambda a.\ M_1 \wedge \forall D \in V^\lambda.\ M_1[a := \emptyset] \downarrow \langle O, P\rangle \wedge \\
& \qquad O \downarrow \lambda x.\ O_1 \wedge \forall N.\ (N \Vdash D \in_I A) \to O_1[x := N] \Vdash D \in B \wedge \\
& \qquad P \downarrow \lambda x.\ P_1 \wedge \forall N.\ (N \Vdash D \in_I B) \to P_1[x := N] \Vdash D \in A
\end{aligned}$$

The relations $M \Vdash A \in B$ and $M \Vdash A = B$ are defined together in a standard way by transfinite recursion. See for example [Rat05a] for more details.

Definition 3.2.22 *For any set $C \in V^\lambda$, C^+ denotes $\{(M, A) \mid M \Vdash A \in C\}$.*

In realizability for HAS, the environments were used to store "semantic" objects — the elements of H^λ. This is also going to be the case in IZF_R; however, in order to give a smooth presentation and make the account closer to the standard accounts of realizability for constructive set theories [McC84, Rat04, Rat06], we make it possible for the formulas to mention constants from V^λ as well. Formally, we extend the first-order language of IZF_R in the following way:

Definition 3.2.23 *A (class-sized) first-order language $L(V^\lambda)$ arises from enriching the IZF_R signature with constants for all λ-names.*

We leave as an exercise a reformulation of our development not using $L(V^\lambda)$.

From now on until the end of this section, symbols M, N, O, P range exclusively over realizers, letters a, b, c vary over first-order variables in the language, letters A, B, C vary over λ-names. Environments are finite partial functions from first-order variables in $L(V^\lambda)$ to V^λ.

Definition 3.2.24 *For any formula ϕ of $L(V^\lambda)$, any term t of $L(V^\lambda)$, ρ defined on all free variables of ϕ and t, any realizer M, we define by metalevel induction a realizability relation $M \Vdash_\rho \phi$ in an environment ρ and a meaning of a term $[\![t]\!]_\rho$ in an environment ρ:*

1. $[\![a]\!]_\rho \equiv \rho(a)$
2. $[\![A]\!]_\rho \equiv A$
3. $[\![\omega]\!]_\rho \equiv \omega'$, *where ω' is defined by the means of inductive definition: ω' is the smallest set such that:*
   - $(\mathrm{infRep}(N), A) \in \omega'$ *if* $N \downarrow \mathrm{inl}(O)$, $O \Vdash_\rho A = 0$ *and* $A \in V^\lambda_\omega$.
   - *If* $(M, B) \in \omega'^{+}$, *then* $(\mathrm{infRep}(N), A) \in \omega'$ *if* $N \downarrow \mathrm{inr}(N_1)$, $N_1 \downarrow [\emptyset, O]$, $O \downarrow \langle M, P\rangle$, $P \Vdash_\rho A = S(B)$ *and* $A \in V^\lambda_\omega$.

Note that if $(M, B) \in \omega'^{+}$, then there is a finite ordinal α such that $B \in V^{\lambda}_{\alpha}$.

4. $[\![t_A(\vec{u})]\!]_\rho \equiv \{(\mathrm{axRep}(N), B) \in \lambda\overline{Z}_{vc} \times V^{\lambda}_{\gamma} \mid N \Vdash_\rho \phi_A(B, [\![\vec{u}]\!]_\rho)\}$. *The ordinal γ will be defined below.*

5. $M \Vdash_\rho \bot \equiv \bot$

6. $M \Vdash_\rho t \in_I s \equiv M \Vdash [\![t]\!]_\rho \in_I [\![s]\!]_\rho$

7. $M \Vdash_\rho t \in s \equiv M \Vdash [\![t]\!]_\rho \in [\![s]\!]_\rho$

8. $M \Vdash_\rho t = s \equiv M \Vdash [\![t]\!]_\rho = [\![s]\!]_\rho$

9. $M \Vdash_\rho \phi \wedge \psi \equiv M \downarrow \langle M_1, M_2\rangle \wedge (M_1 \Vdash_\rho \phi) \wedge (M_2 \Vdash_\rho \psi)$

10. $M \Vdash_\rho \phi \vee \psi \equiv (M \downarrow \mathrm{inl}(M_1) \wedge M_1 \Vdash_\rho \phi) \vee (M \downarrow \mathrm{inr}(M_1) \wedge M_1 \Vdash_\rho \psi)$

11. $M \Vdash_\rho \phi \to \psi \equiv (M \downarrow \lambda x.\ M_1) \wedge \forall N.\ (N \Vdash_\rho \phi) \to (M_1[x := N] \Vdash_\rho \psi)$

12. $M \Vdash_\rho \exists a.\ \phi \equiv M \downarrow [\emptyset, N] \wedge \exists A \in V^{\lambda}.\ N \Vdash_\rho \phi[a := A]$

13. $M \Vdash_\rho \forall a.\ \phi \equiv M \downarrow \lambda a.\ N \wedge \forall A \in V^{\lambda}.\ N[a := \emptyset] \Vdash \phi[a := A]$

The definition of the ordinal γ in item 4 depends on $t_A(\vec{u})$. This ordinal is close to the rank of the set denoted by $t_A(\vec{u})$ and is chosen so that Lemma 3.2.38 can be proved. Let $\vec{\alpha} = \overrightarrow{\lambda rk([\![u]\!]_\rho)}$ and let $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)$. Case $t_A(\vec{u})$ of:

- $\{u_1, u_2\}$ — $\gamma = max(\alpha_1, \alpha_2)$
- $P(u)$ — $\gamma = \alpha_1 + 1$.
- $\bigcup u$ — $\gamma = \alpha_1$.
- $S_{a,\vec{f}.\ \phi(a,\vec{f})}(u, \vec{u})$ — $\gamma = \alpha_1$.
- $R_{a,b,\vec{f}.\ \phi(a,b,\vec{f})}(u, \vec{u})$. This case is more complicated. The names, such as N_{21}, are chosen to match the corresponding clause in the proof of Lemma 3.2.38. Let $G = \{(N_1, (N_{21}, B)) \in \lambda\overline{Z}_c \times [\![u]\!]^{+}_\rho \mid \exists d \in V^{\lambda}.\ \psi(N_1, N_{21}, B, d)\}$, where $\psi(N_1, N_{21}, B, d) \equiv (N_1 \downarrow \lambda a.\ N_{11}) \wedge (N_{11}[a := \emptyset] \downarrow \lambda x.\ O) \wedge (O[x :=$

$N_{21}] \downarrow [\emptyset, O_1]) \wedge O_1 \Vdash_\rho \phi(B, d, \overrightarrow{[\![u]\!]_\rho}) \wedge \forall e.\ \phi(B, e, \overrightarrow{[\![u]\!]_\rho}) \to e = d)$. Then for all $g \in G$ there is D and $(N_1, (N_{21}, B))$ such that $g = (N_1, (N_{21}, B))$ and $\psi(N_1, N_{21}, B, D)$. Use Collection to collect these D's in one set H, so that for all $g \in G$ there is $D \in H$ such that the property holds. Apply Replacement to H to get the set of λ-ranks of sets in H. Then $\beta \equiv \bigcup H$ is an ordinal and for any $D \in H$, $\lambda rk(D) < \beta$. Therefore for all $g \in G$ there is $D \in V^\lambda_\beta$ and $(N_1, (N_{21}, B))$ such that $g = (N_1, (N_{21}, B))$ and $\psi(N_1, N_{21}, B, D)$ holds. Set $\gamma = \beta + 1$.

The proof of well-foundedness of the realizability definition is just a little bit more difficult than the similar proof for HAS.

Definition 3.2.25 *For any closed term s, we define number of occurrences of s in any term t and formula ϕ, denoted by $Occ(s, t)$ and $Occ(s, \phi)$, respectively, by induction on the definition of terms and formulas. We show representative clauses of the definition:*

- $Occ(s, s) = 1$.
- $Occ(s, a) = 0$, *where a is a variable.*
- $Occ(s, t_A(\vec{u})) = Occ(s, u_1) + \ldots + Occ(s, u_n)$.
- $Occ(s, S_\phi(t, \vec{u})) = Occ(s, \phi) + Occ(s, t) + Occ(s, u_1) + \ldots + Occ(s, u_n)$.
- $Occ(s, t \in u) = Occ(s, t) + Occ(s, u)$.
- $Occ(s, \phi \wedge \psi) = Occ(s, \phi) + Occ(s, \psi)$.

In a similar manner we define the number of function symbols FS in a term and formula.

Lemma 3.2.26 *The definition of realizability is well-founded.*

Proof Use the measure function m which takes a clause in the definition and returns an element of $\mathbb{N}^3$ with the lexicographical order:

$$\begin{aligned} m(M \Vdash_\rho \phi) &= (Occ(\omega, \phi), FS(\phi), \text{“structural complexity of } \phi\text{”}) \\ m(\llbracket t \rrbracket_\rho) &= (Occ(\omega, t), FS(t), 0) \end{aligned}$$

Then the measure of the definiendum is always greater than the measure of the definiens — in the clauses for formulas the structural complexity goes down, while the rest of parameters do not grow larger. In the definition of $\llbracket \omega \rrbracket_\rho$, one ω disappears. Finally, in the definition of $\llbracket t_A(\vec{u}) \rrbracket_\rho$, the topmost t_A disappears, while no new V_i's and ω's appear. ■

The three critical clauses of our realizability definition are:

- $\llbracket t_A(\vec{u}) \rrbracket_\rho \equiv \{(\text{axRep}(N), B) \in \lambda \overline{Z}_{vc} \times V^\lambda_\gamma \mid N \Vdash_\rho \phi_A(B, \llbracket \vec{u} \rrbracket_\rho)\}$
- $M \Vdash_\rho t \in_I s \equiv M \Vdash \llbracket t \rrbracket_\rho \in_I \llbracket s \rrbracket_\rho$
- $M \Vdash_\rho \forall a.\ \phi \equiv M \downarrow \lambda a.\ N \wedge \forall A \in V^\lambda.\ N[a := \emptyset] \Vdash \phi[a := A]$

These clauses exhibit deep impredicativity of set theory. The trick we use to tackle the impredicativity is essentially the same as the one we applied for HAS. To define the realizability clause for the universal quantifier, we quantify over all sets in V^λ. Since the meaning of any term is a member of V^λ, the construction makes sense, as we just saw in Lemma 3.2.26.

As expected, realizability "commutes" with substitution:

Lemma 3.2.27 $\llbracket t[a := s] \rrbracket_\rho = \llbracket t[a := \llbracket s \rrbracket_\rho] \rrbracket_\rho = \llbracket t \rrbracket_{\rho[a := \llbracket s \rrbracket_\rho]}$ *and* $M \Vdash_\rho \phi[a := s]$ *iff* $M \Vdash_\rho \phi[a := \llbracket s \rrbracket_\rho]$ *iff* $M \Vdash_{\rho[a := \llbracket s \rrbracket_\rho]} \phi$.

Proof By induction on the definition of realizability. We show representative cases. Case t of:

- A — then $\llbracket t[a := s] \rrbracket_\rho = \llbracket t[a := \llbracket s \rrbracket_\rho] \rrbracket_\rho = \llbracket t \rrbracket_{\rho[a := \llbracket s \rrbracket_\rho]} = A$.

- a — then $[\![t[a := s]]\!]_\rho = [\![s]\!]_\rho$, $[\![t[a := [\![s]\!]_\rho]]\!]_\rho = [\![[\![s]\!]_\rho]\!]_\rho = [\![s]\!]_\rho$ and also $[\![t]\!]_{\rho[a:=[\![s]\!]_\rho]} = [\![s]\!]_\rho$.
- $t_A(\vec{u})$. Then $[\![t[a := s]]\!]_\rho = \{(\mathrm{axRep}(N), A) \mid N \Vdash_\rho \phi_A(A, \vec{u}[a := s])\}$. By the induction hypothesis, this is equal to $\{(\mathrm{axRep}(N), A) \mid N \Vdash_{\rho[a:=[\![s]\!]_\rho]} \phi_A(A, \vec{u})\} = [\![t]\!]_{\rho[a:=[\![s]\!]_\rho]}$ and also to $\{(\mathrm{axRep}(N), A) \mid N \Vdash_\rho \phi_A(A, \vec{u}[a := [\![s]\!]_\rho])\}$ and thus to $[\![t[a := [\![s]\!]_\rho]]\!]_\rho$.

For formulas, the atomic cases follow by the proof above and the non-atomic cases follow immediately by application of the induction hypothesis. ■

The standard lemmas continue to hold:

Lemma 3.2.28 *If $(M \Vdash_\rho \phi)$ then $M \downarrow$.*

Lemma 3.2.29 *If $M \to^* M'$ then $M' \Vdash_\rho \phi$ iff $M \Vdash_\rho \phi$.*

Lemma 3.2.30 *If $M \Vdash_\rho \phi \to \psi$ and $N \Vdash_\rho \psi$, then $M\ N \Vdash_\rho \psi$.*

The following Lemma is strictly technical:

Lemma 3.2.31 *If ρ agrees with ρ' on $FV(\phi)$, then $M \Vdash_\rho \phi$ iff $M \Vdash_{\rho'} \phi$. In particular, if $a \notin FV(\phi)$, then $M \Vdash_\rho \phi$ iff $M \Vdash_{\rho[a:=A]} \phi$.*

Proof Straightforward induction on the definition of realizability — the environment is used only to provide the meaning of the free variables of terms in a formula. ■

We now establish several properties of the realizability relation, which mostly state that the truth in the realizability universe is not far from the truth in the real world, as far as ranks of sets are concerned. Several lemmas mirror similar facts from McCarty's thesis [McC84].

Lemma 3.2.32 *If $A \in V^\lambda_\alpha$, then there is $\beta < \alpha$ such that for all B, if $M \Vdash_\rho B \in A$, then $B \in V^\lambda_\beta$. If $M \Vdash_\rho B = A$, then $B \in V^\lambda_\alpha$. If $M \Vdash_\rho B \in_I A$, then $\lambda rk(B) < \lambda rk(A)$.*

Proof By induction on α. Take any $A \in V^\lambda_\alpha$. By the definition of V^λ_α, there is $\beta < \alpha$ such that $A \subseteq \lambda\overline{Z}_{vc} \times V^\lambda_\beta$. Suppose $M \Vdash_\rho B \in A$. Then $M \downarrow \mathrm{inRep}(N)$, $N \downarrow [\emptyset, O]$, $O \downarrow \langle O_1, O_2\rangle$ and there is C such that $O_1 \Vdash C \in_I A$ and $O_2 \Vdash B = C$. Therefore, $O_1 \downarrow v$ and $(v, C) \in A$. Thus $C \in V^\lambda_\beta$, so by the induction hypothesis also $B \in V^\lambda_\beta$ and we get the claim of the first part of the lemma.

For the second part, suppose $M \Vdash_\rho B = A$. This means that $M \downarrow \mathrm{eqRep}(M_0)$, $M_0 \downarrow \lambda a.\ M_1$ and for all D, $M_1[a := \emptyset] \downarrow \langle O, P\rangle$. Moreover, $O \downarrow \lambda x.\ O_1$ and for all $N \Vdash_\rho D \in_I B$ we have $O_1[x := N] \Vdash_\rho D \in A$. In particular, if $(v, D) \in B$, then $O_1[x := v] \Vdash_\rho D \in A$. By the first part of the lemma, any such D is in V^λ_β for some $\beta < \alpha$, so $B \in V^\lambda_\alpha$.

The third part is trivial. ■

Lemma 3.2.33 *$M \Vdash_\rho A = B$ iff $M \downarrow \mathrm{eqRep}(N)$ and $N \Vdash_\rho \forall d.\ (d \in_I A \to d \in B) \land (d \in_I B \to d \in A)$. Also, $M \Vdash_\rho A \in B$ iff $M \downarrow \mathrm{inRep}(N)$ and $N \Vdash_\rho \exists c.\ c \in_I B \land A = c$.*

Proof Simply expand what it means for M to realize respective formulas. ■

The following two lemmas will be used for the treatment of ω in Lemma 3.2.38.

Lemma 3.2.34 *If $A, B \in V^\lambda_\alpha$, then $[\![\{A, B\}]\!]_\rho \in V^\lambda_{\alpha+1}$.*

Proof Take any $(M, C) \in [\![\{A, B\}]\!]_\rho$. By the definition of $[\![\{A, B\}]\!]_\rho$, any such C is in V^λ_α, so $[\![\{A, B\}]\!]_\rho \in V^\lambda_{\alpha+1}$. ■

Lemma 3.2.35 *If $A \in V^\lambda_\alpha$ and $M \Vdash_\rho B = S(A)$, then $B \in V^\lambda_{\alpha+3}$.*

Proof $M \Vdash_\rho B = S(A)$ means $M \Vdash_\rho B = \bigcup\{A, \{A, A\}\}$. By Lemma 3.2.32, it suffices to show that $[\![\bigcup\{A, \{A, A\}\}]\!]_\rho \in V^\lambda_{\alpha+3}$. Applying Lemma 3.2.34 twice, we find that $[\![\{A, \{A, A\}\}]\!]_\rho \in V^\lambda_{\alpha+2}$. By the definition of $[\![\bigcup\{A, \{A, A\}\}]\!]_\rho$, if $(M, C) \in [\![\bigcup\{A, \{A, A\}\}]\!]_\rho$, then $C \in V_{\lambda rk([\![\bigcup\{A,\{A,A\}\}]\!]_\rho)}$, so $C \in V^\lambda_{\alpha+2}$. Therefore $[\![\bigcup\{A, \{A, A\}\}]\!]_\rho \in V^\lambda_{\alpha+3}$ which shows the claim. ■

Lemma 3.2.36 *If $A, B \in V^\lambda_\alpha$ and $M \Vdash_\rho C = (A, B)$, then $C \in V^\lambda_{\alpha+2}$.*

Proof Similar to the proof of Lemma 3.2.35, utilizing Lemmas 3.2.34 and 3.2.32. ■

We will need one helpful realizer, mirroring the proof of Lemma 3.2.1:

Lemma 3.2.37 *There is a term* eqRefl *such that* eqRefl $\Vdash_\rho \forall a.\ a = a$.

Proof Take the term eqRefl $\equiv$ ind(M), where $M = \lambda c.\ \lambda x.\ \text{eqRep}(\lambda d.\ \langle N, N\rangle)$ and $N = \lambda y.\ \text{inRep}([\emptyset, \langle y, x\ \emptyset\ y\rangle])$. Then eqRefl $\to \lambda a.\ M\ a\ (\lambda e.\ \lambda z.\ \text{ind}(M)\ e)$. It suffices to show that for any A, $M\ \emptyset\ (\lambda e.\ \lambda z.\ \text{ind}(M)\ e) \Vdash_\rho A = A$. We proceed by induction on λ-rank of A. We have $M\ \emptyset\ (\lambda e.\ \lambda z.\ \text{ind}(M)\ e) \downarrow \text{eqRep}(\lambda d.\ \langle N, N\rangle[x := \lambda e.\ \lambda z.\ \text{ind}(M)\ e])$. It suffices to show that for all $D \in V^\lambda$, for all $O \Vdash_\rho D \in_I A$, $\text{inRep}([\emptyset, \langle O, (\lambda e.\ \lambda z.\ \text{ind}(M)\ e)\ \emptyset\ O\rangle]) \Vdash_\rho D \in A$. Take any D and $O \Vdash_\rho D \in_I A$. By Lemma 3.2.32, $\lambda rk(D) < \lambda rk(A)$. We need to show the existence of C such that $O \Vdash_\rho C \in_I A$ and $(\lambda e.\ \lambda z.\ \text{ind}(M)\ e)\ \emptyset\ O \Vdash_\rho D = C$. Taking $C \equiv D$, the first part follows trivially. Since $(\lambda e.\ \lambda z.\ \text{ind}(M)\ e)\ \emptyset\ O \to^* \text{ind}(M)\ \emptyset \to M\ \emptyset\ (\lambda e.\ \lambda z.\ \text{ind}(M)\ \emptyset)$, we get the claim by Lemma 4.2.11 and the induction hypothesis. ■

The following lemma states the crucial property of our realizability relation.

Lemma 3.2.38 *$(M, C) \in [\![t_A(\vec{u})]\!]_\rho$ iff $M = \text{axRep}(N)$ and $N \Vdash_\rho \phi_A(C, \overrightarrow{[\![u]\!]_\rho})$.*

Proof The proof proceeds by case analysis on $t_A(\vec{u})$. We first do the proof for all terms apart from ω, then we show the claim for ω.

For all terms, save ω, the left-to-right direction is immediate. For the right-to-left direction, suppose $N \Vdash_\rho \phi_A(C, \overrightarrow{[\![u]\!]_\rho})$ and $M = \text{axRep}(N)$. To show that $(M, C) \in [\![t_A(\vec{u})]\!]_\rho$, we need to show that $C \in V^\lambda_\gamma$. Let $\vec{\alpha} = \overrightarrow{\lambda rk([\![u]\!]_\rho)} = (\alpha_1, \ldots, \alpha_n)$. Case $t_A(\vec{u})$ of:

- $\{u_1, u_2\}$. Suppose that $N \Vdash_\rho C = [\![u_1]\!]_\rho \lor C = [\![u_2]\!]_\rho$. Then either $N \downarrow \text{inl}(N_1) \land N_1 \Vdash_\rho C = [\![u_1]\!]_\rho$ or $N \downarrow \text{inr}(N_1) \land N_1 \Vdash_\rho C = [\![u_2]\!]_\rho$. By Lemma 3.2.32, in the former case $C \in V^\lambda_{\alpha_1}$, in the latter $C \in V^\lambda_{\alpha_2}$, so $C \in V^\lambda_{max(\alpha_1,\alpha_2)}$.
- $P(u)$. Suppose that $N \Vdash_\rho \forall d.\ d \in C \to d \in [\![u]\!]_\rho$. Then $N \downarrow \lambda a.\ N_1$ and $\forall D.\ N_1[a := \emptyset] \Vdash_\rho D \in C \to D \in [\![u]\!]_\rho$, so $\forall D.\ N_1[a := \emptyset] \downarrow \lambda x.\ N_2$ and for all O, if $O \Vdash D \in C$ then $N_2[x := O] \Vdash_\rho D \in [\![u]\!]_\rho$. Take any $(v, B) \in C$. Then $\text{inRep}([\emptyset, \langle v, \text{eqRefl } \emptyset\rangle]) \Vdash_\rho B \in C$, so $N_2[x := \text{inRep}([\emptyset, \langle v, \text{eqRefl } \emptyset\rangle]] \Vdash_\rho B \in [\![u]\!]_\rho$. Thus by Lemma 3.2.32 any such B is in $V^\lambda_{\alpha_1}$, so $C \in V^\lambda_{\alpha_1+1}$.
- $\bigcup u$. Suppose $N \Vdash_\rho \exists c.\ c \in [\![u]\!]_\rho \land C \in c$. Then $N \downarrow [\emptyset, N_1]$ and there is B such that $N_1 \Vdash_\rho B \in [\![u]\!]_\rho \land C \in B$. Thus $N_1 \downarrow \langle N_1, N_2\rangle$, $N_1 \Vdash_\rho B \in [\![u]\!]_\rho$, $N_2 \Vdash_\rho C \in B$. By Lemma 3.2.32, any such B is in $V^\lambda_{\alpha_1}$, so also $C \in V^\lambda_{\alpha_1}$.
- $S_{a,\vec{f}.\ \phi(a,\vec{f})}(u, \vec{u})$. Suppose $N \Vdash_\rho C \in [\![u]\!]_\rho \land \phi(C, \overrightarrow{[\![u]\!]_\rho})$. Then $N \downarrow \langle N_1, N_2\rangle$ and $N_1 \Vdash_\rho C \in [\![u]\!]_\rho$. Thus $C \in V^\lambda_{\alpha_1}$.
- $R_{a,b,\vec{f}.\ \phi(a,b,\vec{f})}(u, \vec{u})$. Suppose $N \Vdash_\rho (\forall x \in [\![u]\!]_\rho \exists! y.\ \phi(x, y, \overrightarrow{[\![u]\!]_\rho})) \land \exists x \in [\![u]\!]_\rho.\ \phi(x, C, \overrightarrow{[\![u]\!]_\rho})$. Then $N \downarrow \langle N_1, N_2\rangle$ and $N_2 \Vdash_\rho \exists x \in [\![u]\!]_\rho.\ \phi(x, C, [\![\vec{u}]\!]_\rho)$. Thus $N_2 \downarrow [\emptyset, N_{20}]$, $N_{20} \downarrow \langle N_{21}, N_{22}\rangle$ and there is B such that $N_{21} \Vdash_\rho B \in [\![u]\!]_\rho$ and $N_{22} \Vdash_\rho \phi(B, C, [\![\vec{u}]\!]_\rho)$. We also have $N_1 \Vdash_\rho \forall x \in [\![u]\!]_\rho \exists! y.\ \phi(x, y, [\![\vec{u}]\!]_\rho)$, so $N_1 \downarrow \lambda a.\ N_{11}$ and for all C, $N_{11}[a := \emptyset] \downarrow \lambda x.\ O$ and for all $P \Vdash_\rho C \in [\![u]\!]_\rho$, $O[x := P] \Vdash_\rho \exists! y.\ \phi(C, y, \overrightarrow{[\![u]\!]_\rho})$. So taking $C = B$ and $P = N_{21}$, there is D such that $N_1 \downarrow \lambda a.\ N_{11}$, $N_{11}[a := \emptyset] \downarrow \lambda x.\ O$ and $O[x := N_{21}] \downarrow$

$[\emptyset, O_1]$ and $O_1 \Vdash_\rho \phi(B, D, \overrightarrow{[\![u]\!]_\rho}) \land \forall e.\ \phi(B, e, \overrightarrow{[\![u]\!]_\rho}) \to e = D$. Therefore $(N_1, (N_{21}, B)) \in G$ from the definition of γ, so there is $D \in V^\lambda_\gamma$ such that $N_1 \downarrow \lambda a.\ N_{11}$, $N_{11}[a := \emptyset] \downarrow \lambda x.\ O$, $O[x := N_{21}] \downarrow [\emptyset, O_1]$ and $O_1 \Vdash_\rho \phi(B, D, [\![\vec{u}]\!]_\rho) \land \forall e.\ \phi(B, e, [\![\vec{u}]\!]_\rho) \to e = D$. So $O_1 \downarrow \langle O_{11}, O_{12}\rangle$ and $O_{12} \Vdash_\rho \forall e.\ \phi(B, e, [\![\vec{u}]\!]_\rho) \to e = D$. Therefore, $O_{12} \downarrow \lambda a.\ Q$, $Q[a := \emptyset] \downarrow \lambda x.\ Q_1$ and $Q_1[x := N_{22}] \Vdash_\rho C = D$. By Lemma 3.2.32, $C \in V^\lambda_\gamma$.

Now we tackle ω. For the left-to-right direction, obviously $M = \mathrm{infRep}(N)$. For the claim about N we proceed by induction on the definition of ω':

- The base case. Then $N \downarrow \mathrm{inl}(O)$ and $O \Vdash_\rho A = 0$, so $N \Vdash_\rho A = 0 \lor \exists y \in \omega'.\ A = S(y)$.
- Inductive step. Then $N \downarrow \mathrm{inr}(N_1)$, $N_1 \downarrow [\emptyset, O]$, $O \downarrow \langle M', P\rangle$, $(M', B) \in \omega'^+$, $P \Vdash_\rho A = S(B)$. Therefore, there is C (namely B) such that $M' \Vdash_\rho C \in \omega'$ and $P \Vdash_\rho A = S(C)$. Thus $[\emptyset, O] \Vdash_\rho \exists y.\ y \in \omega' \land A = S(y)$, so $N \Vdash_\rho A = 0 \lor \exists y \in \omega'.\ A = S(y)$.

For the right-to-left direction, suppose $N \Vdash_\rho A = 0 \lor (\exists y.\ y \in \omega' \land A = S(y))$. Then either $N \downarrow \mathrm{inl}(N_1)$ or $N \downarrow \mathrm{inr}(N_1)$. In the former case, $N_1 \Vdash_\rho A = 0$, so by Lemma 3.2.32 $A \in V^\lambda_\omega$. In the latter, $N_1 \Vdash_\rho \exists y.\ y \in \omega' \land A = S(y)$. Thus $N_1 \downarrow [\emptyset, O]$ and there is B such that $O \Vdash_\rho B \in \omega' \land A = S(B)$. So $O \downarrow \langle M', P\rangle$, $(M', B) \in \omega'^+$ and $P \Vdash_\rho A = S(B)$. This is exactly the inductive step of the definition of ω', so it remains to show that $A \in V^\lambda_\omega$. Since $(M', B) \in \omega'^+$, there is a finite ordinal α such that $B \in V^\lambda_\alpha$. By Lemma 3.2.35, $A \in V^\lambda_{\alpha+3}$, so also $A \in V^\lambda_\omega$ and we get the claim. ■

3.2.5 Normalization of λZ

In this section, environments ρ are finite partial functions mapping propositional variables to realizers and first-order variables to V^λ. In other words, $\rho : Var \cup FVar \to \lambda\overline{Z}_c \cup V^\lambda$, $\rho^{\to}(Var) \subseteq \lambda\overline{Z}_c$ and $\rho^{\to}(FVar) \subseteq V^\lambda$. As before, any ρ can be used as a realizability environment by considering only the mapping of first-order variables to V^λ. Therefore we will be using the notation $\Vdash_\rho$ also for these environments ρ.

Definition 3.2.39 *For a sequent $\Gamma \vdash M : \phi$, $\rho \models \Gamma \vdash M : \phi$ means that ρ is defined on $FV(\Gamma, M, \phi)$ and for all $(x_i, \phi_i) \in \Gamma$, $\rho(x_i) \Vdash_\rho \phi_i$.*

Note that if $\rho \models \Gamma \vdash M : \phi$, then for any term t in Γ, ϕ, $[\![t]\!]_\rho$ is defined and so is the realizability relation $M \Vdash_\rho \phi$.

Definition 3.2.40 *As usual, $M[\rho] \equiv M[x_1 := \rho(x_1), \ldots, x_n := \rho(x_n)]$, where $FV(M) = \{x_1, \ldots, x_n\}$.*

Theorem 3.2.41 (Normalization) *If $\Gamma \vdash M : \vartheta$ then for all $\rho \models \Gamma \vdash M : \vartheta$, $\overline{M}[\rho] \Vdash_\rho \vartheta$.*

Proof For any λZ term M, M' in the proof denotes $\overline{M}[\rho]$. We proceed by metalevel induction on $\Gamma \vdash M : \vartheta$ and show some cases where the treatment significantly differs from the normalization proof for λH. Case $\Gamma \vdash M : \vartheta$ of:

- $$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash \lambda a.\ M : \forall a.\ \phi}$$

  By the induction hypothesis, for all $\rho \models \Gamma \vdash M : \phi$, $\overline{M}[\rho] \Vdash \phi$. We need to show that for all $\rho \models \Gamma \vdash \lambda a.\ M : \forall a.\ \phi$, $\overline{(\lambda a.\ M)}[\rho] \Vdash_\rho \forall a.\ \phi$. This is equivalent to $\lambda a.\ \overline{M}[\rho] \Vdash_\rho \forall a.\ \phi$. Take any such ρ. We need to show that

$\forall A.\ \overline{M}[\rho][a := \emptyset] \Vdash_\rho \phi[a := A]$. Take any A. Since $\rho[a := A] \models \Gamma \vdash M : \phi$ and $\overline{M}[\rho][a := \emptyset] = \overline{M}[\rho[a := A]][a := \emptyset]$, we get the claim by the induction hypothesis.

- $$\frac{\Gamma \vdash M : \forall a.\ \phi}{\Gamma \vdash M\ t : \phi[a := t]}$$

  By the induction hypothesis, $M' \Vdash_\rho \forall a.\ \phi$, so $M' \downarrow \lambda a.\ N$ and $\forall A.\ N[a := \emptyset] \Vdash_\rho \phi[a := A]$. In particular $N[a := \emptyset] \Vdash_\rho \phi[a := [\![t]\!]_\rho]$. By Lemma 3.2.27, $N[a := \emptyset] \Vdash_\rho \phi[a := t]$. Since $\overline{M\ t}[\rho] = M'\ \emptyset \to^* (\lambda a.\ N)\ \emptyset \to N[a := \emptyset]$, Lemma 3.2.29 gives us the claim.

- $$\frac{\Gamma \vdash M : \phi[a := t]}{\Gamma \vdash [t, M] : \exists a.\ \phi}$$

  By the induction hypothesis, $M' \Vdash_\rho \phi[a := t]$, so by Lemma 3.2.27, $M' \Vdash_\rho \phi[a := [\![t]\!]_\rho]$. Thus, there is a lambda-name A, namely $[\![t]\!]_\rho$, such that $M' \Vdash_\rho \phi[a := A]$. Thus, $\overline{[t, M]}[\rho] = [\emptyset, M'] \Vdash_\rho \exists a. \phi$, which is what we want.

- $$\frac{\Gamma \vdash M : \exists a.\ \phi \quad \Gamma, x : \phi \vdash N : \psi}{\Gamma \vdash \text{let } [a, x : \phi] := M \text{ in } N : \psi}\ a \notin FV(\Gamma, \psi)$$

  Let $\rho \models \Gamma \vdash \text{let } [a, x : \phi] := M \text{ in } N : \psi$. We need to show that $\overline{\text{let } [a, x : \phi] := M \text{ in } N}[\rho] = \text{let } [a, x] := M' \text{ in } \overline{N}[\rho] \Vdash_\rho \psi$. By the induction hypothesis, $M' \Vdash_\rho \exists a.\ \phi$, so $M' \downarrow [\emptyset, M_1]$ and for some A, $M_1 \Vdash_\rho \phi[a := A]$. By the induction hypothesis again, for any $\rho' \models \Gamma, x : \phi \vdash N : \psi$ we have $\overline{N}[\rho'] \Vdash_{\rho'} \psi$. Take $\rho' = \rho[x := M_1, a := A]$. Since $a \notin FV(\psi)$, by Lemma 3.2.31 $\overline{N}[\rho'] \Vdash_\rho \psi$. Now, $\text{let } [a, x : \phi] := M' \text{ in } \overline{N}[\rho] \to^* \text{let } [a, x] := [\emptyset, M_1] \text{ in } \overline{N}[\rho] \to \overline{N}[\rho][a := \emptyset][x := M_1] = \overline{N}[\rho']$. Lemma 3.2.29 gives us the claim.

- $$\frac{\Gamma \vdash M : \forall d.\ (d \in_I t \to d \in u) \land (d \in_I u \to d \in t)}{\Gamma \vdash \text{eqRep}(t, u, M) : t = u}$$

  By the induction hypothesis, $M' \Vdash_\rho \forall d.\ (d \in_I t \to d \in u) \land (d \in_I u \to d \in t)$. By Lemma 3.2.27, $M' \Vdash_\rho \forall d.\ (d \in_I [\![t]\!]_\rho \to d \in [\![u]\!]_\rho) \land (d \in_I [\![u]\!]_\rho \to d \in [\![t]\!]_\rho)$. By Lemma 3.2.33, $\text{eqRep}(M') \Vdash_\rho [\![t]\!]_\rho = [\![u]\!]_\rho$. Lemma 3.2.27 applied again gives us the claim.

- $$\frac{\Gamma \vdash M : t = u}{\Gamma \vdash \text{eqProp}(t, u, M) : \forall d.\ (d \in_I t \to d \in u) \land (d \in_I u \to d \in t)}$$

  By the induction hypothesis, $M' \Vdash_\rho t = u$. By Lemma 3.2.27, $M' \Vdash_\rho [\![t]\!]_\rho = [\![u]\!]_\rho$. By Lemma 3.2.33, $M' \downarrow \text{eqRep}(N)$ and $N \Vdash_\rho \forall d.\ (d \in_I [\![t]\!]_\rho \to d \in [\![u]\!]_\rho) \land (d \in_I [\![u]\!]_\rho \to d \in [\![t]\!]_\rho)$. Since $\overline{\text{eqProp}(t, u, M)} = \text{eqProp}(M') \to^* \text{eqProp}(\text{eqRep}(N)) \to N$, by Lemma 3.2.29 $\overline{\text{eqProp}(t, u, M)} \Vdash_\rho \forall d.\ (d \in_I [\![t]\!]_\rho \to d \in [\![u]\!]_\rho) \land (d \in_I [\![u]\!]_\rho \to d \in [\![t]\!]_\rho)$. Lemma 3.2.27 applied once again gives us the claim.

- For inProp and inRep, the proof is similar to the two previous cases.

- $$\frac{\Gamma \vdash M : \phi_A(t, \vec{u})}{\Gamma \vdash \text{axRep}(t, \vec{u}, M) : t \in_I t_A(\vec{u})}$$

  By the induction hypothesis, $M' \Vdash_\rho \phi_A(t, \vec{u})$. By Lemma 3.2.27 this is equivalent to $M' \Vdash_\rho \phi_A([\![t]\!]_\rho, \overrightarrow{[\![u]\!]_\rho})$. By Lemma 3.2.38 $(\text{axRep}(M'), [\![t]\!]_\rho) \in [\![t_A(\vec{u})]\!]_\rho$, so $\text{axRep}(M') \Vdash_\rho t \in_I t_A(\vec{u})$.

- $$\frac{\Gamma \vdash M : t \in_I t_A(\vec{u})}{\Gamma \vdash \text{axProp}(t, \vec{u}, M) : \phi_A(t, \vec{u})}$$

  By the induction hypothesis, $M' \Vdash_\rho t \in_I t_A(\vec{u})$. This means that $M' \downarrow v$ and $(v, [\![t]\!]_\rho) \in [\![t_A(\vec{u})]\!]_\rho$. By Lemma 3.2.38, $v = \text{axRep}(N)$ and $N \Vdash_\rho$

$\phi_A(\llbracket t \rrbracket_\rho, \overrightarrow{\llbracket u \rrbracket_\rho})$. By Lemma 3.2.27, $N \Vdash_\rho \phi_A(t, \vec{u})$. Moreover, $\overline{\mathrm{axProp}(t, \vec{u}, M)} = \mathrm{axProp}(M') \to^* \mathrm{axProp}(\mathrm{axRep}(N)) \to N$. Lemma 3.2.29 gives us the claim.

- $$\frac{\Gamma \vdash M : \forall c.\ (\forall b.\ b \in_I c \to \phi(b, \vec{t})) \to \phi(c, \vec{t})}{\Gamma \vdash \mathrm{ind}(M, \vec{t}) : \forall a.\ \phi(a, \vec{t})}$$

  Since $\mathrm{ind}(M')$ reduces to $\lambda c.\ M'\ c\ (\lambda b.\ \lambda x.\ \mathrm{ind}(M')\ b)$, by Lemma 3.2.29 it suffices to show that for all C, $M'\ \emptyset\ (\lambda b.\ \lambda x.\ \mathrm{ind}(M')\ b) \Vdash_\rho \phi(C, \vec{t})$. We proceed by induction on λ-rank of C. Take any C. By the induction hypothesis, $M' \Vdash_\rho \forall c.\ (\forall b.\ b \in_I c \to \phi(b, \vec{t})) \to \phi(c, \vec{t})$, so $M' \downarrow \lambda c.\ N$ and $N[c := \emptyset] \Vdash_\rho \forall b.\ b \in_I C \to \phi(b, \vec{t})$. By Lemma 3.2.30, it suffices to show that $\lambda b.\ \lambda x.\ \mathrm{ind}(M')\ b \Vdash_\rho \forall b.\ b \in_I C \to \phi(b, \vec{t})$. Take any B and $O \Vdash_\rho B \in_I C$, we need to show that $\mathrm{ind}(M')[x := O]\ \emptyset \Vdash_\rho \phi(B, \vec{t})$. As $x \notin FV(M')$, it suffices to show that $\mathrm{ind}(M')\ \emptyset \Vdash_\rho \phi(B, \vec{t})$, which, by Lemma 3.2.29, is equivalent to $M'\ \emptyset\ (\lambda b.\ \lambda x.\ \mathrm{ind}(M')\ b) \Vdash_\rho \phi(B, \vec{t})$. As $O \Vdash_\rho B \in_I C$, the λ-rank of B is less than the λ-rank of C and we get the claim by the induction hypothesis.

  ■

Corollary 3.2.42 (Normalization) *If $\vdash M : \phi$, then $M \downarrow$.*

Proof Take ρ mapping all free propositional variables of M to themselves, and all free first-order variables a of M to $\emptyset$. Then $\rho \models\vdash M : \phi$. By Theorem 3.2.41, $\overline{M}[\rho]$ normalizes. By the definition of ρ, $\overline{M}[\rho] = \overline{M}$. By Lemma 3.2.18, M normalizes.

■

Normalization properties of set theories

All reduction systems we have considered so far are deterministic — the evaluation contexts uniquely determine the place in the term where one of the reduction rules is applied. However, it is possible to imagine extended systems, where base reduction rules can be applied anywhere in the term in a nondeterministic fashion.

In such extended systems, which are actually more common in the world of applied type theories, such as Calculus of (Inductive) Constructions or Extended Calculus of Constructions, the normalization has two aspects: weak and strong:

Definition 3.2.43 *A calculus* weakly normalizes *if there is a reduction path terminating in a value. It* strongly normalizes *if all reduction paths terminate in a value.*

Obviously, strong normalization implies weak normalization. The other direction needs not hold; however, the gap between weak and strong normalization for the calculi used in proof assistants seems minimal. Apart from Martin-Löf's type theory, all of them strongly normalize. Many of them can be specified in the framework of *Pure Type Systems* and it has been conjectured by Barendregt, Geuvers and Klop that for Pure Type Systems, weak normalization entails strong normalization. For more about Pure Type Systems see [Bar92]. Reasonable calculi, such as Calculus of Inductive Constructions or Extended Calculus of Constructions, have also the property that their inconsistency implies the existence of a non-normalizing term, which violates even weak normalization of the calculus.

Suppose we extend our reduction systems to enable applications of base reductions anywhere in lambda terms. Then $\lambda^{\rightarrow}, \lambda H$ and λS strongly normalize. However, most surprisingly, λZ does not. One trivial reason are the ind terms. However, even without them, the system would not strongly normalize, as the following counterexample, invented by Marcel Crabbé and adapted to our framework shows. The original counterexample was not a part of a published paper. Although it is available on the author's website [Cra], our presentation is as comprehensive as any other.

Theorem 3.2.44 (Crabbé's counterexample) *There is a formula ϕ and term M such that $\vdash M : \phi$ and M does not strongly normalize.*

Proof Recall that in set theory $0 \equiv \emptyset$. Let $t = \{x \in 0 \mid x \in x \to \bot\}$. Consider the terms:

$$N \equiv \lambda y : t \in t.\, \mathrm{snd}(\mathrm{sepProp}(t, 0, y))\, y \qquad M \equiv \lambda x : t \in 0.\, N\, (\mathrm{sepRep}(t, 0, \langle x, N\rangle))$$

We first show that these terms can be typed. Let T denote the following proof tree:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{y : t \in t \vdash y : t \in \{x \in 0 \mid x \in x \to \bot\}}}{y : t \in t \vdash \mathrm{sepProp}(t, 0, y)) : t \in 0 \land t \in t \to \bot}}{y : t \in t \vdash \mathrm{snd}(\mathrm{sepProp}(t, 0, y)) : t \in t \to \bot} \qquad \dfrac{}{y : t \in t \vdash y : t \in t}}{y : t \in t \vdash \mathrm{snd}(\mathrm{sepProp}(t, 0, y))\ y : \bot}}{\vdash \lambda y : t \in t.\ \mathrm{snd}(\mathrm{sepProp}(t, 0, y))\ y : t \in t \to \bot}$$

By Weakening, we can also obtain a tree T_1 showing that $x : t \in 0 \vdash N : t \in t \to \bot$. The following proof tree shows that $\vdash M : t \in 0 \to \bot$:

$$\dfrac{\dfrac{\dfrac{T_1}{x : t \in 0 \vdash N : t \in t \to \bot} \qquad \dfrac{\dfrac{\dfrac{}{x : t \in 0 \vdash x : t \in 0} \quad \dfrac{T_1}{x : t \in 0 \vdash N : t \in t \to \bot}}{x : t \in 0 \vdash \langle x, N\rangle : t \in 0 \land t \in t \to \bot}}{x : t \in 0 \vdash \mathrm{sepRep}(t, 0, \langle x, N\rangle) : t \in t}}{x : t \in 0 \vdash N\ (\mathrm{sepRep}(t, 0, \langle x, N\rangle)) : \bot}}{\vdash \lambda x : t \in 0.\ N\ (\mathrm{sepRep}(t, 0, \langle x, N\rangle)) : t \in 0 \to \bot}$$

We now exhibit an infinite reduction sequence starting from M:

$$\begin{array}{ll}
M = \lambda x : t \in 0.\ N\ (\mathrm{sepRep}(t, 0, \langle x, N\rangle)) & = \\
\lambda x : t \in 0.\ (\lambda y : t \in t.\ \mathrm{snd}(\mathrm{sepProp}(t, 0, y))\ y)\ (\mathrm{sepRep}(t, 0, \langle x, N\rangle)) & \to \\
\lambda x : t \in 0.\ \mathrm{snd}(\mathrm{sepProp}(t, 0, (\mathrm{sepRep}(t, 0, \langle x, N\rangle))))\ (\mathrm{sepRep}(t, 0, \langle x, N\rangle)) & \to \\
\lambda x : t \in 0.\ \mathrm{snd}(\langle x, N\rangle)\ (\mathrm{sepRep}(t, 0, \langle x, N\rangle)) & \to \\
\lambda x : t \in 0.\ N\ (\mathrm{sepRep}(t, 0, \langle x, N\rangle)) = M & \to \ldots
\end{array}$$

■

Moreover, a slight (from the semantic point of view) modification to IZF_R, namely making it non-well-founded, results in a system which is not even weakly normalizing. A very small fragment is sufficient for this effect to arise. Let T be an intuitionistic set theory consisting of 2 axioms:

- (C) $\forall a.\ a \in c \leftrightarrow a = c$
- (D) $\forall a.\ a \in d \leftrightarrow a \in c \land a \in a \to a \in a.$

The constant c denotes a non-well-founded set. The existence of d can be derived from the separation axiom: $d = \{a \in c \mid a \in a \to a \in a\}$. The lambda calculus corresponding to T is defined just as for IZF_R.

Lemma 3.2.45 $T \vdash d \in c$

Proof It suffices to show that $d = c$. Take any $e \in d$, then $e \in c$. On the other hand, suppose $e \in c$. Since obviously $e \in e \to e \in e$, we also get $e \in d$. ■

Theorem 3.2.46 *There is a formula ϕ and a term M such that $\vdash_T M : \phi$ and M does not weakly normalize.*

Proof Let N be the lambda term corresponding to the proof of Lemma 3.2.45 along with the proof tree T_N. Take $\phi = d \in d \to d \in d$. Consider the terms:

$$O \equiv \lambda x : d \in d.\ \mathrm{snd}(\mathrm{dProp}(d, c, x))\ x \qquad M \equiv O\ (\mathrm{dRep}(d, c, \langle N, O\rangle)).$$

Again, we first show that these terms are typable. Let S be the following proof tree, showing that $\vdash O : d \in d \to d \in d$:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{x : d \in d \vdash x : d \in d}}{x : d \in d \vdash \mathrm{dProp}(d, c, x)) : d \in c \land d \in d \to d \in d}}{x : d \in d \vdash \mathrm{snd}(\mathrm{dProp}(d, c, x)) : d \in d \to d \in d} \qquad \dfrac{}{x : d \in d \vdash x : d \in d}}{\dfrac{x : d \in d \vdash \mathrm{snd}(\mathrm{dProp}(d, c, x))\ x : d \in d}{\vdash \lambda x : d \in d.\ \mathrm{snd}(\mathrm{dProp}(d, c, x))\ x : d \in d \to d \in d}}$$

Then the following proof tree shows that M is typable:

$$\dfrac{\dfrac{S}{\vdash O : d \in d \to d \in d} \quad \dfrac{\dfrac{\dfrac{T_N}{\vdash N : d \in c} \quad \dfrac{S}{\vdash O : d \in d \to d \in d}}{\vdash \langle N, O\rangle : d \in c \land d \in d \to d \in d}}{\vdash \mathrm{dRep}(d, c, \langle N, O\rangle) : d \in d}}{\vdash O\ (\mathrm{dRep}(d, c, \langle N, O\rangle)) : d \in d}$$

Finally, we exhibit the only reduction sequence starting from M:

$$\begin{array}{ll}
M = O\ (\mathrm{dRep}(d, c, \langle N, O\rangle)) & = \\
(\lambda x : d \in d.\ \mathrm{snd}(\mathrm{dProp}(d, c, x))\ x)\ (\mathrm{dRep}(d, c, \langle N, O\rangle)) & \to \\
\mathrm{snd}(\mathrm{dProp}(d, c, \mathrm{dRep}(d, c, \langle N, O\rangle)))\ (\mathrm{dRep}(d, c, \langle N, O\rangle)) & \to \\
\mathrm{snd}(\langle N, O\rangle)\ (\mathrm{dRep}(d, c, \langle N, O\rangle)) & \to \\
O\ (\mathrm{dRep}(d, c, \langle N, O\rangle)) = M & \to \ldots
\end{array}$$

■

Therefore constructive set theories dwell on a precarious border between normalization and lack of thereof. An exciting theoretical challenge, which we leave open, is to provide a more detailed map of this border.

Applications

As before, we can derive the standard properties of constructive theories:

Corollary 3.2.47 (Disjunction Property) *If IZF$_R$$\vdash \phi \lor \psi$, then IZF$_R$$\vdash \phi$ or IZF$_R$$\vdash \psi$.*

Proof Standard. ■

To show Numerical Existence Property, we first define an extraction function F which takes a proof $\vdash M : t \in \omega$ and returns a natural number n. F works as follows:

It applies Lemma 3.2.9 to obtain a proof $\vdash N : t = 0 \lor \exists y \in \omega.\ t = S(y)$. F then normalizes N to either $\mathrm{inl}(O)$ or $\mathrm{inr}(O)$. In the former case, F returns 0.

In the latter, $\vdash O : \exists y.\ y \in \omega \wedge t = S(y)$. Normalizing O it gets $[t_1, P]$, where $\vdash P : t_1 \in \omega \wedge t = S(t_1)$. Normalizing P it obtains Q such that $\vdash Q : t_1 \in \omega$. Then F returns $F(\vdash Q : t_1 \in \omega) + 1$.

To show that F terminates for all its arguments, consider the sequence of terms $t, t_1, t_2, \ldots$ obtained throughout the execution of F. We have $\mathrm{IZF}_R \vdash t \in \omega$, $\mathrm{IZF}_R \vdash t = S(t_1)$, $\mathrm{IZF}_R \vdash t_1 = S(t_2)$ and so on. The length of the sequence is therefore exactly the natural number denoted by t.

Corollary 3.2.48 (Numerical Existence Property) *If $IZF_R \vdash \exists x \in \omega.\ \phi(x)$, then there is a natural number n and term t such that $IZF_R \vdash \phi(\overline{n})$, where $\overline{n}$ denotes the IZF_R numeral corresponding to n.*

Proof As before, use the Curry-Howard isomorphism to get a value $[t, M]$ such that $\vdash [t, M] : \exists x.\ x \in \omega \wedge \phi(x)$. Thus $\vdash M : t \in \omega \wedge \phi(t)$, so $M \downarrow \langle M_1, M_2 \rangle$ and $\vdash M_1 : t \in \omega$. Take $n = F(\vdash M_1 : t \in \omega)$. By patching together the proofs $\mathrm{IZF}_R \vdash t = S(t_1)$, $\mathrm{IZF}_R \vdash t_1 = S(t_2)$, $\ldots$,$\mathrm{IZF}_R \vdash t_n = 0$ obtained throughout the execution of F, we get $\mathrm{IZF}_R \vdash t = \overline{n}$. By the Leibniz axiom, $\mathrm{IZF}_R \vdash \phi(\overline{n})$. ■

There are also two properties characteristic of constructive set theories:

Corollary 3.2.49 (Term Existence Property) *If $IZF_R \vdash \exists x.\ \phi(x)$, then there is a term t such that $IZF_R \vdash \phi(t)$.*

Proof By the Curry-Howard isomorphism, there is a λZ-term M such that $\vdash M : \exists x.\ \phi$. By normalizing M and applying Canonical Forms, we get $[t, N]$ such that $\vdash N : \phi(t)$ and thus by the Curry-Howard isomorphism $\mathrm{IZF}_R \vdash \phi(t)$. ■

Corollary 3.2.50 (Set Existence Property) *If $IZF_R \vdash \exists x.\ \phi(x)$ and $\phi(x)$ is term-free, then there is a term-free formula $\psi(x)$ such that $IZF_R \vdash \exists! x.\ \phi(x) \wedge \psi(x)$.*

Proof Take t from Term Existence Property, so that $\mathrm{IZF}_R \vdash \phi(t)$. We showed before that IZF_R is a definitional extension of its term-free version, so there is a term-free formula $\psi(x)$ defining t such that $\mathrm{IZF}_R \vdash (\exists! x.\ \psi(x)) \land \psi(t)$. Then $\mathrm{IZF}_R \vdash \exists! x.\ \phi(x) \land \psi(x)$ can be easily derived. ■

We could now present a concrete example of program extraction in IZF_R, just as we did for HA and λH. However, such an example would be prohibitively large, given the amount of prerequisite development in set theory necessary to define even an addition function. Instead, we show a general way of program extraction from IZF_R, independent of the underlying lambda calculus and using only DP, NEP and TEP.

3.3 Program extraction

We now describe a generic procedure of extraction from IZF_R proofs. The results of this section are also available in [CM06]. To facilitate the description, we will use a very simple fragment of type theory, which we call TT^0.

The *types* of TT^0 are generated by the following abstract grammar.

$$\tau ::= * \mid P_\phi \mid \mathrm{nat} \mid \mathrm{bool} \mid (\tau, \tau) \mid \tau + \tau \mid \tau \to \tau$$

We associate with each type τ of TT^0, a set of its elements, which are finitistic objects. The set of elements of τ is denoted by $El(\tau)$ and defined by structural induction on τ:

- $El(*) = \{*\}$.
- $El(P_\phi)$ is the set of all IZF_R proofs of formula ϕ.
- $El(\mathrm{nat}) = \mathbb{N}$, the set of natural numbers.
- $El(\mathrm{bool})$ contains two objects: $\mathrm{true}, \mathrm{false}$.

- $M \in El((\tau_1, \tau_2))$ is the set consisting of all pairs (M, N) such that M is in $El(\tau_1)$ and N is in $El(\tau_2)$.
- $M \in El(\tau_1 + \tau_2)$ iff either $M = \mathrm{inl}(M_1)$ and $M_1 \in El(\tau_1)$ or $M = \mathrm{inr}(M_1)$ and $M_1 \in El(\tau_2)$.
- $M \in El(\tau_1 \to \tau_2)$ iff M is a method which given any element of $El(\tau_1)$ returns an element of $El(\tau_2)$.

In the last clause, we use an abstract notion of "method". It will not be necessary to formalize it, but for the interested reader, all "methods" we use are functions provably recursive in $ZF + Con(ZF)$. This is because the normalization theorem can be formalized given a model of IZF_C and ZF is equiconsistent with IZF_C [Fri73].

The notation $M : \tau$ means that $M \in El(\tau)$.

We call a TT^0 type *pure* if it does not contain $*$ and P_ϕ. There is a natural mapping of pure types TT^0 to sets $\tau \to [\![\tau]\!]$, defined as follows:

- $[\![\mathrm{nat}]\!] = \mathbb{N}$.
- $[\![\mathrm{bool}]\!] = 2$.
- $[\![(\tau, \sigma)]\!] = [\![\tau]\!] \times [\![\sigma]\!]$.
- $[\![\tau + \sigma]\!] = [\![\tau]\!] + [\![\sigma]\!]$, the disjoint union of $[\![\tau]\!]$ and $[\![\sigma]\!]$.
- $[\![\tau \to \sigma]\!] = [\![\tau]\!] \to [\![\sigma]\!]$.

If a set (represented by an IZF_R term) is in a codomain of the map above, we call it *type-like*. If a set A is type-like, then there is a unique pure type τ such that $[\![\tau]\!] = A$. We denote this type $Type(A)$.

Before we proceed further, let us extend TT^0 with a new type Q_τ, where τ is any pure type of TT^0. The members of $El(Q_\tau)$ are pairs $(t, \mathcal{P})$ such that $\mathcal{P} \vdash t \in [\![\tau]\!]$ ($\mathcal{P}$ is an IZF_R proof of $t \in [\![\tau]\!]$). Note that any natural number n can be injected into Q_{nat}.

We first define a helper function T, which takes a pure type τ and returns another type. Intuitively, $T(\tau)$ is the type of the extract from a statement $\exists x.\ x \in [\![\tau]\!]$. T is defined by induction on τ:

- $T(\text{bool}) = \text{bool}$.
- $T(\text{nat}) = \text{nat}$.
- $T((\tau, \sigma)) = (T(\tau), T(\sigma))$
- $T(\tau + \sigma) = T(\tau) + T(\sigma)$.
- $T(\tau \to \sigma) = Q_\tau \to T(\sigma)$ (in order to utilize an IZF$_R$ function from $[\![\tau]\!]$ to $[\![\sigma]\!]$ we need to supply an element of a set $[\![\tau]\!]$, that is an element of Q_τ)

Now we assign to each formula ϕ of IZF$_R$ a TT^0 type $\overline{\phi}$, which intuitively describes the *computational content* of an IZF$_R$ proof of ϕ. We will use the type $*$ to mark parts of the proofs we are not interested in. We do it by induction on ϕ:

- $\overline{a \in b} = *$.
- $\overline{a = b} = *$ (atomic formulas carry no useful computational content).
- $\overline{\phi_1 \vee \phi_2} = \overline{\phi_1} + \overline{\phi_2}$.
- $\overline{\phi_1 \wedge \phi_2} = (\overline{\phi_1}, \overline{\phi_2})$.
- $\overline{\phi_1 \to \phi_2} = P_{\phi_1} \to \overline{\phi_2}$.
- $\overline{\exists a \in A.\ \phi_1} = (T(Type(A)), \overline{\phi_1})$, if A is type-like.
- $\overline{\exists a \in A.\ \phi_1} = *$, if A is not type-like.
- $\overline{\exists a.\ \phi_1} = *$.
- $\overline{\forall a \in A.\ \phi_1} = Q_{Type(A)} \to \overline{\phi_1}$, if A is type-like.
- $\overline{\forall a \in A.\ \phi_1} = *$, if A is not type-like.
- $\overline{\forall a.\ \phi_1} = *$.

The definition is tailored for the developments in Section 5, but it can easily be extended to allow meaningful extraction from a larger class of formulas, for example we could extract a term from $\exists a.\ \phi_1$ using TEP. For now, we present some natural examples of our translation in action:

1. $\overline{\exists x \in \mathbb{N}.\ x = x} = \langle \text{nat}, * \rangle$.
2. $\overline{\forall x \in \mathbb{N} \exists y \in \mathbb{N}.\ \phi} = Q_{\text{nat}} \to \langle \text{nat}, \overline{\phi} \rangle$.
3. $\overline{\forall f \in \mathbb{N} \to \mathbb{N} \exists x \in \mathbb{N}.\ f(x) = 0} = Q_{\text{nat} \to \text{nat}} \to \langle \text{nat}, * \rangle$.

The extra $*$ can be easily discarded from types (and extracts).

Lemma 3.3.1 *For any term t, which is not type-like, $\overline{\phi[a := t]} = \overline{\phi}$.*

Proof Straightforward induction on ϕ. ■

Lemma 3.3.2 ($\mathbf{IZF}_R$) $(\exists a \in 2.\ \phi(a))$ *iff* $\phi(0) \vee \phi(1)$.

Proof Suppose there is $a \in 2$ such that $\phi(a)$. Then either $a = 0$ or $a = 1$. In the former case $\phi(0)$, in the latter $\phi(1)$. In any case $\phi(0) \vee \phi(1)$. ■

We are now ready to describe the extraction function E, which takes an IZF_R proof $\mathcal{P}$ of a formula ϕ and returns an object of TT^0 type $\overline{\phi}$. We do it by induction on ϕ, checking on the way that the returned object is of type $\overline{\phi}$. Recall that DP, TEP and NEP denote Disjunction, Term and Numerical Existence Property, respectively. Case ϕ of:

- $a \in b$ — return $*$. We have $* : *$.
- $a = b$ — return $*$. We have $* : *$, too.
- $\phi_1 \vee \phi_2$. Apply DP to $\mathcal{P}$ to get a proof $\mathcal{P}_1$ of either ϕ_1 or ϕ_2. In the former case return $\text{inl}(E(\mathcal{P}_1))$, in the latter return $\text{inr}(E(\mathcal{P}_1))$. By the induction hypothesis, $E(\mathcal{P}_1) : \overline{\phi_1}$ (or $E(\mathcal{P}_1) : \overline{\phi_2}$), so $E(\mathcal{P}) : \overline{\phi}$ follows.

- $\phi_1 \land \phi_2$. Then there are proofs $\mathcal{P}_1$ and $\mathcal{P}_2$ such that $\mathcal{P}_1 \vdash \phi_1$ and $\mathcal{P}_2 \vdash \phi_2$. Return a pair $(E(\mathcal{P}_1), E(\mathcal{P}_2))$. By the induction hypothesis, $E(\mathcal{P}_1) : \overline{\phi_1}$ and $E(\mathcal{P}_2) : \overline{\phi_2}$, so $(E(\mathcal{P}_1), E(\mathcal{P}_2)) : \overline{\phi_1 \land \phi_2}$.
- $\phi_1 \to \phi_2$. Return a function G which takes an IZF_R proof $\mathcal{Q}$ of ϕ_1, applies $\mathcal{P}$ to $\mathcal{Q}$ (using the modus-ponens rule of the first-order logic) to get a proof $\mathcal{R}$ of ϕ_2 and returns $E(\mathcal{R})$. By the induction hypothesis, any such $E(\mathcal{R})$ is in $El(\overline{\phi_2})$, so $G : P_{\phi_1} \to \overline{\phi_2}$.
- $\exists a \in A.\ \phi_1(a)$, where A is type-like. Let $T = Type(A)$. We proceed by induction on T, case T of:
  - bool. By Lemma 3.3.2, we have $\phi_1(0) \lor \phi_1(1)$. Apply DP to get a proof $\mathcal{Q}$ of either $\phi_1(0)$ or $\phi_1(1)$. Let b be false or true, respectively. Return a pair $(b, E(\mathcal{Q}))$. By the induction hypothesis, $E(\mathcal{Q}) : \overline{\phi_1(\llbracket b \rrbracket)}$. By Lemma 3.3.1, $E(\mathcal{Q}) : \overline{\phi_1}$.
  - nat. Apply NEP to $\mathcal{P}$ to get a natural number n and a proof $\mathcal{Q}$ of $\phi_1(\overline{n})$. Return a pair $(n, E(\mathcal{Q}))$. By the induction hypothesis, $E(\mathcal{Q}) : \overline{\phi_1(\overline{n})}$, by Lemma 3.3.1, $E(\mathcal{Q}) : \overline{\phi_1}$, so $(n, E(\mathcal{Q})) : (\mathrm{nat}, \overline{\phi_1})$.
  - (τ, σ). Construct a proof $\mathcal{Q}$ of $\exists a_1 \in \llbracket \tau \rrbracket \exists a_2 \in \llbracket \sigma \rrbracket.\ a = \langle a_1, a_2 \rangle \land \phi_1$. Let $M = E(\mathcal{Q})$. By the induction hypothesis M is a pair $\langle M_1, M_2 \rangle$ such that $M_1 : T(\tau)$ and $M_2 : \overline{\exists a_2 \in \llbracket \sigma \rrbracket.\ a = \langle a_1, a_2 \rangle \land \phi_1}$. Therefore M_2 is a pair $\langle M_{21}, M_{22} \rangle$, $M_{21} : T(\sigma)$ and $M_{22} : \overline{a = \langle a_1, a_2 \rangle \land \phi_1}$. Therefore M_{22} is a pair $\langle N, O \rangle$, where $O : \overline{\phi_1}$. Therefore $\langle M_1, M_{21} \rangle : T((\tau, \sigma))$, so $\langle \langle M_1, M_{21} \rangle, O \rangle : (T((\tau, \sigma)), \overline{\phi_1})$ and we are justified to return $\langle \langle M_1, M_{21} \rangle, O \rangle$.
  - $\tau + \sigma$. Construct a proof $\mathcal{Q}$ of $(\exists a_1 \in \llbracket \tau \rrbracket.\ \phi_1) \lor (\exists a_1 \in \llbracket \sigma \rrbracket.\ \phi_1)$. Apply DP to get the proof $\mathcal{Q}_1$ of (without loss of generality) $\exists a_1 \in \llbracket \tau \rrbracket.\ \phi_1$. Let

$M = E(\mathcal{Q}_1)$. By the induction hypothesis, $M = \langle M_1, M_2 \rangle$, where $M_1 : T(\tau)$ and $M_2 : \overline{\phi_1}$. Return $\langle \mathrm{inl}(M_1), M_2 \rangle$, which is of type $(T(\tau+\sigma), \overline{\phi_1})$.

  - $\tau \to \sigma$. Use TEP to get a term f such that $(f \in [\![\tau]\!] \to [\![\sigma]\!]) \land \phi_1(f)$. Construct proofs $\mathcal{Q}_1$ of $\forall x \in [\![\tau]\!] \exists y \in [\![\sigma]\!]. f(x) = y$ and $\mathcal{Q}_2$ of $\phi_1(f)$. By the induction hypothesis and Lemma 3.3.1, $E(\mathcal{Q}_2) : \overline{\phi_1}$. Let G be a function which works as follows: G takes a pair $t, \mathcal{R}$ such that $\mathcal{R} \vdash t \in [\![\tau]\!]$, applies $\mathcal{Q}_1$ to $t, \mathcal{R}$ to get a proof $\mathcal{R}_1$ of $\exists y \in [\![\sigma]\!].\ f(t) = y$ and calls $E(\mathcal{R}_1)$ to get a term M. By the induction hypothesis, $M : \exists y \in [\![\sigma]\!].\ f(t) = y$, so $M = \langle M_1, M_2 \rangle$, where $M_1 : T(\sigma)$. G returns M_1. Our extraction procedure $E(\mathcal{P})$ returns $\langle G, E(\mathcal{Q}_2) \rangle$. The type of $\langle G, E(\mathcal{Q}_2) \rangle$ is $\langle \mathcal{Q}_\tau \to T(\sigma), \overline{\phi_1} \rangle$ which is equal to $\langle T(\tau \to \sigma), \overline{\phi_1} \rangle$.

- $\exists a \in A.\ \phi_1(a)$, where A is not type-like. Return $*$.
- $\exists a.\ \phi_1(a)$, $\forall a.\ \phi_1(a)$. Return $*$.
- $\forall a \in A.\ \phi_1(a)$, where A is type-like. Return a function G which takes an element $(t, \mathcal{Q})$ of $Q_{Type(A)}$, applies $\mathcal{P}$ to t and $\mathcal{Q}$ to get a proof R of $\phi_1(t)$, and returns $E(\mathcal{R})$. By the induction hypothesis and Lemma 3.3.1, $E(\mathcal{R}) : \overline{\phi_1}$, so $G : Q_{Type(A)} \to \overline{\phi_1}$.
- $\forall a \in A.\ \phi_1(a)$, where A is not type-like. Return $*$.

We have described a general method of extracting programs from IZF_R proofs. The salient feature of the method is that the only information it utilizes about IZF_R is that it possesses DP, NEP and TEP. Moreover, it can easily be seen that foregoing the capability of higher-order program extraction, we could use only DP and NEP. The method can therefore be applied to provide the extraction mechanism from *any* constructive theory with these properties, for example CZF [Rat05a]. We believe an implementation of our ideas should not be very tedious.

3.4 Historical context

While Brouwer created the philosophy of intuitionism, it was Arend Heyting who formalized IPC and IFOL (against the spirit of Brouwer's view of mathematics, we might add). Heyting Arithmetic was defined not much later [Hey31]. Troelstra's book [Tro73] gives an excellent overview of research on constructive arithmetics. For the description of constructivism and various constructive formal systems, we recommend [Bee85] and [TvD88]. Gödel's System T [Göd58] exhibited computational content of arithmetic in a style close to modern lambda calculi.

Realizability originated with Kleene's paper [Kle45], where he applied the technique to HA. Since then, it has been generalized and applied to a variety of systems [Tro98]. The most impressive application from our point of view, developed by David McCarty [McC84], is of course set theory.

The Curry-Howard isomorphism can be traced back to the remark of Curry [CFC58] which we cited in the introduction to this chapter. However, quoting [SU06]: "*Brouwer - Heyting - Kolmogorov - Schönfinkel - Curry - Meredith - Kleene - Feys - Gödel - Läuchli - Kreisel - Tait - Lawvere - Howard - de Bruijn - Scott - Martin-Löf - Girard - Reynolds - Stenlund - Constable - Coquand - Huet - . . . isomorphism* might be a more appropriate name, still not including all the contributions.". We strongly recommend [SU06] as a textbook and a source of further references regarding the isomorphism.

IZF in its version with Replacement was introduced by Myhill [Myh73]. In that paper he showed DP, NEP, TEP and SEP for the theory, using a complicated, nonconstructive method. Not much later, Friedman showed that IZF_C is equiconsistent with ZFC [Fri73] and that it has the same set of provably recursive functions [Fri78]. In a joint paper with Ščedrov [Fv85], they showed that IZF_R is weaker that IZF_C; the exact relation between these theories is still unknown.

McCarty, a student of Dana Scott, wrote his thesis [McC84], summarized nicely by Lipton [Lip95], in 1985. A good description of the results up to 1985 can be found in [Bee85, Š85]. After that, the research on IZF slowed down. Before our first paper [Moc06a] appeared, further research on constructive set theories was concentrated on weaker subtheories, such as Aczel's CZF [Acz78, AR01] or Intuitionistic Kripke-Platek [Lub02]. One notable exception was Lubarsky's investigation of intuitionistic L [Lub93]. Recently, Rathjen [Rat06] used realizability-with-truth to show DP and NEP for extensions of IZF_C with various choice principles.

Several normalization results for impredicative constructive set theories much weaker than IZF_R exist. Bailin [Bai88] proved strong normalization of a constructive set theory without the induction and replacement axioms. Miquel interpreted a theory of similar strength in a Pure Type System [Miq04]. In [Miq03] he also defined a strongly normalizing lambda calculus with types based on $F\omega.2$, capable of interpreting IZF_C without the $\in$-induction axiom. This result was later extended — Dowek and Miquel [DM06] interpreted a version of constructive Zermelo set theory in a strongly normalizing deduction-modulo system.

Krivine [LK01] defined realizability using lambda calculus for classical set theory conservative over ZF. The types for the calculus were defined. However, it seems that the types correspond more to the truth in the realizability model than to provable statements in the theory. Moreover, the calculus does not even weakly normalize.

CHAPTER 4

BEYOND IZF

In this section we investigate some extensions to IZF_R. The first one, inaccessible sets, extends the theory with a capability of providing constructive set-theoretic semantics for popular constructive theorem provers based on type theory. The second one extends the logic of IZF_R with features characteristic of dependent set theories. The results of this chapter can also be found in [Moc06b] and [Moc07], respectively.

4.1 Inaccessible sets

Since the advent of the Curry-Howard isomorphism, many systems exploiting the isomorphism, with program extraction capability, have been built. They include Agda/Alfa [Coq, Hal], Coq [The04], Lego [LP92], Minlog [BBS+98], Nuprl [CAB+86] — to name a few. Some are quite powerful — for example Coq can interpret an intuitionistic version of Zermelo's set theory [Wer97]. With such power at hand, these systems have the potential of becoming very useful tools.

There is, however, one problem they all share, namely their foundational basis. In order to use Coq or Nuprl, one has to master the ways of types, a setting quite different from set theory, the standard framework for doing mathematics. A newcomer to this world, presented even with Π and Σ types emulating familiar universal and existential quantifiers, is likely to become confused. The fact that the consistency of the systems is usually justified by a normalization theorem in one form or other, does not make matters easier. Even when set-theoretic semantics is provided, it does not help much, given that the translation of "the statement $\forall x : \mathrm{nat}, \phi(x)$ is provable" is "the set $\Pi_{n \in \mathbb{N}} [\![\phi[x := n_n]]\!]$ is inhabited", instead of the expected "for all $x \in \mathbb{N}$, $\phi(x)$ holds". The systems which are not based on type theory share the problem of unfamiliar foundations. This is a serious shortcoming preventing the systems from becoming widely used, as the initial barrier to cross is quite high.

The work presented in the previous chapter can be seen as a first step toward a solution to this problem, as IZF_R enables extraction of programs from proofs, while using the standard, natural language of set theory. However, even though IZF_R is quite powerful, it is unclear if it is as strong as type theories underlying the systems of Coq and LEGO, Calculus of Inductive Constructions (CIC) and Extended Calculus of Constructions (ECC), as all known set-theoretical interpretations of these theories use ω-many strongly inaccessible cardinals [Wer97, Acz99], which are very strong set-theoretic constructs.

We therefore extend IZF_R to incorporate ω-many inaccessible sets, which we call *$IZF_{R\omega}$*. Our axiomatization uses an inductive definition of inaccessible sets. $\mathrm{IZF}_{R\omega}$ extended with excluded middle is equivalent to ZF with ω-many strong inaccessible cardinals.

In a constructive setting inaccessible sets perform a similar function to strongly inaccessible cardinals in the classical world and universes in type theories. They are "large" sets/types, closed under certain operations ensuring that they give rise to models of set/type theories. The closure conditions largely coincide in both worlds and an inaccessible can be used to provide a set-theoretic intepretation of a universe [Wer97, Acz99]. Both CIC and ECC have ω-many universes. By results of Aczel [Acz99], $\mathrm{IZF}_{R\omega}$ is strong enough to interpret ECC. It is reasonable to expect that CIC could be interpreted too, as the inductive types in CIC need to satisfy positivity conditions, and sufficiently strong inductive definitions are available in $\mathrm{IZF}_{R\omega}$ due to the presence of the Power Set and unrestricted Separation axioms. Indeed, Werner's set-theoretic interpretation [Wer97] of a large fragment of CIC uses only the existence of inductively-defined sets in the set-theoretic universe to interpret inductively-defined types.

Our normalization result makes it possible to extract programs from proofs. Thus $\mathrm{IZF}_{R\omega}$ has all the proof-theoretic power of LEGO and likely Coq, uses familiar set-theoretic language and enables program extraction from proofs. This makes it an attractive basis for a powerful and easy to use theorem prover.

To extend IZF_R with inaccessible sets, we add a family of axioms (INAC_i) for $i > 0$. We call the resulting theory $\mathrm{IZF}_{R\omega}$. The axiom (INAC_i) asserts the existence of the i-th inaccessible set, denoted by a new constant symbol V_i, and is defined as follows:

$$(\mathrm{INAC}_i)\ \forall c.\ c \in_I V_i \leftrightarrow \phi_1^i(c, V_i) \land \forall d.\ \phi_2^i(d) \to c \in d$$

Following the conventions set up for IZF_R, $\phi_{INAC_i}(c)$ is $\phi_1^i(c, V_i) \land \forall d.\ \phi_2^i(d) \to c \in d$. The formula $\phi_1^i(c, d)$ intuitively sets up conditions for c being a member of V_i, while $\phi_2^i(d)$ says what it means for d to be inaccessible. To streamline the definition, we set V_0 to denote ω.

Definition 4.1.1 *The formula $\phi_1^i(c, V_i)$ for $i > 0$ is a disjunction of the following five clauses:*

1. $c = V_{i-1}$
2. *there is $a \in V_i$ such that $c \in a$.*
3. *there is $a \in V_i$ such that c is a union of a.*
4. *there is $a \in V_i$ such that c is a power set of a.*
5. *there is $a \in V_i$ such that c is a function from a to V_i.*

Definition 4.1.2 *The formula $\phi_2^i(d)$ for $i > 0$ is a conjunction of the following five clauses:*

1. $V_{i-1} \in d$.

2. $\forall e, f.\ e \in d \wedge f \in e \to f \in d$.

3. $\forall e \in d.\ \bigcup e \in d$.

4. $\forall e \in d.\ P(e) \in d$.

5. $\forall e \in d.\ \forall f \in e \to d.\ f \in d$*, where* $e \to d$ *denotes the set of all functions from* e *to* d.

Briefly, the i-th inaccessible set is the smallest transitive set containing V_{i-1} as an element and closed under unions, power sets and taking functions from its elements into itself. As in case of IZF_R, we can derive the $\in_I$-free version of the axiom:

Lemma 4.1.3 $\forall c.\ c \in V_i \leftrightarrow \phi_1^i(c, V_i) \wedge \forall d.\ \phi_2^i(d) \to c \in d$.

Proof The right-to-left direction is immediate. For the left-to-right direction, suppose $c \in V_i$. Then there is $e \in_I V_i$ such that $c = e$. First we show that $\phi_1^i(C, V_i)$ holds. We have five possible situations:

- $e = V_{i-1}$. Then also $c = V_{i-1}$.
- There is $a \in V_i$ such that $e \in a$. By the Leibniz axiom, $c \in a$ as well.
- There is $a \in V_i$ such that e is a union of a. Then also $c = \bigcup a$.
- There is $a \in V_i$ such that e is a power set of a. Then also $c = P(a)$.
- There is $a \in V_i$ such that e is a function from a to V_i. This means that for all $x \in a$ there is exactly one $y \in V_i$ such that $(x, y) \in e$ and for all $z \in e$ there is $x \in a$ and $y \in V_i$ such that $z = (x, y)$. By $c = e$ and Extensionality we also have for all $x \in a$ there is exactly one $y \in V_i$ such that $(x, y) \in c$. If $z \in c$, then also $z \in e$, so we get x and y such that $z = (x, y)$, which shows the claim.

For the second part of the claim, note that if e is a member of every set d satisfying $\phi_2^i(d)$, then by the Leibniz axiom so is c. This ends the proof. ■

It is easy to see that IZF$_{R\omega}$+ EM is equivalent to ZF with ω-many strongly inaccessible cardinals. For a theory T, let $M(T)$ denote a sentence "T has a model". To show that the set V_i defined by (INAC$_i$) behaves as an inaccessible set in IZF$_{R\omega}$ we prove:

Theorem 4.1.4 (IZF$_{R\omega}$) *For all $i > 0$, $V_i \models$IZF$_R$+ M(IZF$_R$) + M(IZF$_R$+ M(IZF$_R$)) + ... (i times).*

Proof By Clause 2 in the Definition 4.1.1, V_1 is transitive, so the equality and membership relations are absolute. Clause 1 gives us $\omega \in V_1$ and since its definition is Δ_0, $V_1 \models$(INF). Clauses 3 and 4 provide the (UNION) and (POWER) axioms. Transitivity then gives (SEP) and (PAIR), while Clause 5, thanks to Lemma 3.2.12, gives (REPL$_\phi$). The existence of the empty set follows by (INF) and (SEP). For the Induction axiom, we need to show:

$$\forall \vec{f} \in V_i.\ (\forall a \in V_i.\ (\forall b \in V_i.\ b \in a \to \phi^{V_i}(b, \vec{f})) \to \phi^{V_i}(a, \vec{f})) \to \forall a \in V_i.\ \phi^{V_i}(a, \vec{f})$$

Take any $\vec{F} \in V_i$. It suffices to show that:

$$(\forall a.\ a \in V_i \to (\forall b.\ b \in V_i \to b \in a \to \phi^{V_i}(b, \vec{F})) \to \phi^{V_i}(a, \vec{F})) \to \forall a.\ a \in V_i \to \phi^{V_i}(a, \vec{F})$$

This is equivalent to:

$$(\forall a.\ (\forall b.\ b \in a \to b \in V_i \to \phi^{V_i}(b, \vec{F})) \to a \in V_i \to \phi^{V_i}(a, \vec{F})) \to \forall a.\ a \in V_i \to \phi^{V_i}(a, \vec{F})$$

But this is the instance of the induction axiom for the formula $a \in V_i \to \phi^{V_i}(a, \vec{f})$.

Thus $V_1 \models$IZF$_R$. Since $V_1 \in V_2$, $V_2 \models$ IZF$_R$+ M(IZF$_R$). Since $V_2 \in V_3$, $V_3 \models$IZF$_R$+ M(IZF$_R$+ M(IZF$_R$)). Proceeding in this manner by induction we get the claim. ■

We now extend λZ to the calculus λZ_ω corresponding to $\mathrm{IZF}_{R\omega}$. The new terms are:

$$\mathrm{inac}_i\mathrm{Prop}(t, M) \mid \mathrm{inac}_i\mathrm{Rep}(t, M),$$

together with the obvious reduction rule:

$$\mathrm{inac}_i\mathrm{Prop}(t, \mathrm{inac}_i\mathrm{Rep}(t, M)) \to M$$

Thus, inaccessibles fit neatly in the framework of Prop and Rep terms. It is easy to see that the proofs of all properties proved for IZF_R and λZ transfer unchanged to $\mathrm{IZF}_{R\omega}$ and λZ_ω. We therefore pass without further ado to the definition of realizability. The calculus $\lambda\overline{Z_\omega}$ and realizers are defined just as for IZF_R.

As IZF_R has ω-many inaccessibles, it is not surprising that we need to work in a meta-theory with ω-many inaccessibles. We denote the i-th inaccesible by Γ_i and choose them so that $\Gamma_i \in \Gamma_{i+1}$.

We need to extend the definition of realizability by the meaning of inaccessible terms. We set $[\![V_i]\!]_\rho \equiv U_i$, where U_i is defined as follows. Recall first that the axiom (INAC_i) has the following form:

$$(\mathrm{INAC}_i)\ \forall c.\ c \in_I V_i \leftrightarrow \phi_1^i(c, V_i) \land \forall d.\ \phi_2^i(d) \to c \in d.$$

We define a monotonic operator F on sets as:

$$F(A) = A \cup \{(\mathrm{inac}_i\mathrm{Rep}(N), C) \in \lambda\overline{Z}_{\omega vc} \times V^\lambda_{\Gamma_i} \mid N \Vdash_\rho \phi_1^i(C, A) \land \forall d.\ \phi_2^i(d) \to C \in d\}.$$

We set U_i to be the smallest fixpoint of F. Formally, U_i is generated by the following transfinite inductive definition on ordinals:

$$U_{i,\gamma} = F(\bigcup_{\beta<\gamma} U_{i,\beta}) \qquad U_i = \bigcup_{\gamma \in \mathrm{ORD}} U_{i,\gamma}$$

Since F adds only elements from $\lambda\overline{Z}_{\omega vc} \times V^\lambda_{\Gamma_i}$, any element of U_i is in $\lambda\overline{Z}_{\omega vc} \times V^\lambda_{\Gamma_i}$, so $U_i \in V^\lambda_{\Gamma_{i+1}}$.

The proof that the resulting realizability relation is not circular must be slightly modified. We need one extra definition:

Definition 4.1.5 *Let $M(\mathbb{N})$ denote the set of all multisets over $\mathbb{N}$. Formally, a member A of $M(\mathbb{N})$ is a function from $\mathbb{N}$ to $\mathbb{N}$, returning for any n the number of copies of n in A. We impose the standard well-founded ordering on $M(\mathbb{N})$. Recall from Definition 3.2.25 that $Occ(V_i, x)$ is the number of occurences of V_i in x. We define a function V taking terms and formulas into $M(\mathbb{N})$: $V(x)$ for any number i returns $Occ(V_i, x)$, for x being either a term or a formula.*

Lemma 4.1.6 *The definition of realizability is well-founded.*

Proof Use the measure function m which takes a clause in the definition and returns an element of $M(\mathbb{N}) \times \mathbb{N}^3$ with the lexicographical order:

$$\begin{aligned} m(M \Vdash_\rho \phi) &= (V(\phi), Occ(\omega, \phi), FS(\phi), \text{“structural complexity of } \phi\text{”}) \\ m([\![t]\!]_\rho) &= (V(t), Occ(\omega, t), FS(t), 0) \end{aligned}$$

Then the measure of the definiendum is always greater than the measure of the definiens — in the clauses for formulas the structural complexity goes down, while the rest of parameters do not grow larger. In the definition of $[\![V_i]\!]_\rho$, one V_i disappears replaced by two V_{i-1}'s. In the definition of $[\![\omega]\!]_\rho$, one ω disappears. Finally, in the definition of $[\![t_A(\vec{u})]\!]_\rho$, the topmost t_A disappears, while no new V_i's and ω's appear. ■

We need a few more technical lemmas to reach Lemma 4.1.14. They amount to tedious computations of ranks, necessary to show the new case in Lemma 4.1.14.

First, we fix one extra realizer:

Lemma 4.1.7 *There is a realizer* lei *such that* lei $\Vdash_\rho \forall a, b, c.\ a \in c \wedge a = b \to b \in c$.

Proof Follows by Lemmas 3.2.5, 3.2.13 and Theorem 3.2.41. ■

Lemma 4.1.8 $\lambda rk(C) \leq rk(C^+) + \omega$.

Proof If $(M, A) \in C$, then $M \Vdash_\rho A \in_I C$. We have $\mathrm{inRep}([\emptyset, \langle M, \mathrm{eqRefl}\ \emptyset\rangle]) \Vdash_\rho A \in C$, so $(\mathrm{inRep}([\emptyset\langle M, \mathrm{eqRefl}\ \emptyset\rangle]), A) \in C^+$. The extra ω is there to deal with possible difficulties with finite C's, as we do not know a priori the rank of set-theoretic encoding of $\mathrm{inRep}([\emptyset, \langle M, \mathrm{eqRefl}\ \emptyset\rangle]$. ∎

Lemma 4.1.9 *If $N \Vdash_\rho \forall x \in A.\ \phi$ then for all $(O, X) \in A^+$, $N \downarrow \lambda a.\ N_1$ and $N_1[a := \emptyset] \downarrow \lambda x.\ N_2$ and $N_2[x := O] \Vdash_\rho \phi[x := X]$. Also, if $N \Vdash_\rho \exists x \in A.\ \phi$ then there is $(O, X) \in A^+$ such that $N \downarrow [\emptyset, N_1]$, $N_1 \downarrow \langle O, N_2\rangle$ and $N_2 \Vdash_\rho \phi[x := X]$.*

Proof If $N \Vdash_\rho \forall x \in A.\ \phi$ then $N \downarrow \lambda a.\ N_1$ and for all X, $N_1[a := \emptyset] \Vdash_\rho X \in A \to \phi$, so $N_1[a := \emptyset] \downarrow \lambda x.\ N_2$ and for all O such that $O \Vdash_\rho X \in A$, $N_2[x := O] \Vdash_\rho \phi[x := X]$. This implies that for all X, for all O, if $O \Vdash_\rho X \in A$, then $N \downarrow \lambda a.\ N_1$, $N_1[a := \emptyset] \downarrow \lambda x.\ N_2$ and $N_2[x := O] \Vdash_\rho \phi[x := X]$, which proves the first part of the claim.

If $N \Vdash_\rho \exists x \in A.\ \phi$, then $N \downarrow [\emptyset, N_1]$ and there is X such that $N_1 \downarrow \langle O, N_2\rangle$, $O \Vdash_\rho X \in A$ and $N_2 \Vdash_\rho \phi[x := X]$, so there is $(O, X) \in A^+$ such that $N \downarrow [\emptyset, N_1]$, $N_1 \downarrow \langle O, N_2\rangle$ and $N_2 \Vdash_\rho \phi[x := X]$. ∎

Lemma 4.1.10 *Suppose $A \in U_i$ and $N \Vdash_\rho$"C is a function from A into V_i". Then $C \in V^\lambda_{\Gamma_i}$.*

Proof First let us write formally the statement "C is a function from A into V_i". This means "for all $x \in A$ there is exactly one $y \in V_i$ such that $(x, y) \in C$ and for all $z \in C$ there is $x \in A$ and $y \in V_i$ such that $z = (x, y)$". Thus $N \downarrow \langle N_1, N_2\rangle$, $N_1 \Vdash_\rho \forall x \in A \exists! y \in V_i.\ (x, y) \in C$ and $N_2 \Vdash_\rho \forall z \in C \exists x \in A \exists y \in V_i.\ z = (x, y)$. So $N_1 \Vdash_\rho \forall x \in A \exists y \in V_i.\ (x, y) \in C \land \forall z.\ (x, z) \in C \to z = y$. By Lemma 4.1.9, for all $(O, X) \in A^+$ there is $(P, Y) \in U_i^+$ such that $\phi(O, X, P, Y)$ holds, where

$\phi(O,X,P,Y)$ is defined as:

$$\begin{aligned}\phi(O,X,P,Y) \quad \equiv \quad & (N_1 \downarrow \lambda a.\ N_{11}) \wedge (N_{11}[a := \emptyset] \downarrow \lambda x.\ N_{12}) \wedge \\ & (N_{12}[x := O] \downarrow [\emptyset, N_{13}]) \wedge (N_{13} \downarrow \langle P, Q\rangle) \wedge (Q \downarrow \langle Q_1, Q_2\rangle) \wedge \\ & (Q_1 \Vdash_\rho (X,Y) \in C) \wedge (Q_2 \Vdash_\rho \forall z.\ (X,z) \in C \to z = Y)\end{aligned}$$

Let $\psi(O,X,P,Y)$ be defined as:

$$\psi(O,X,P,Y) \equiv \exists Q_1, Q_2.\ (Q_1 \Vdash_\rho (X,Y) \in C) \wedge (Q_2 \Vdash_\rho \forall z.\ (X,z) \in C \to z = Y)$$

Obviously, if $\phi(O,X,P,Y)$ then $\psi(O,X,P,Y)$. So for all $(O,X) \in A^+$ there is $(P,Y) \in U_i^+$ such that $\psi(O,X,P,Y)$ holds.

Define a function F which takes $(O,X) \in A^+$ and returns the set $\{(P,Y) \in U_i^+ \mid \psi(O,X,P,Y)\}$. Suppose $(P_1,Y_1),(P_2,Y_2) \in F((O,X))$. Then there are Q_{11}, Q_{12}, Q_{21} such that $Q_{11} \Vdash_\rho (X,Y_1) \in C$, $Q_{12} \Vdash_\rho \forall z.\ (X,z) \in C \to z = Y_1$, $Q_{21} \Vdash_\rho (X,Y_2) \in C$. By Lemma 4.1.9, $Q_{12} \downarrow \lambda a.\ R_1$, $R_1[a := \emptyset] \downarrow \lambda x.\ R_2$ and $R_2[x := Q_{21}] \Vdash_\rho Y_2 = Y_1$. Since eqSymm $\emptyset$ $\emptyset$ $R_2[x := Q_{21}] \Vdash_\rho Y_1 = Y_2$, by Lemma 3.2.32 the λ-ranks of Y_1, Y_2 are the same and, since any such (P,Y) is a member of U_i^+, they are smaller than Γ_i. Also, for any $(O,X) \in A^+$, $F(O,X)$ is inhabited.

Furthermore, define a function G from A^+ to Γ_i, which takes $(O,X) \in A^+$ and returns $\bigcup\{\lambda rk((P,Y)) \mid (P,Y) \in F(O,X) \wedge \psi(O,X,P,Y)\}$. Then for any $(O,X) \in A^+$, $G(O,X)$ is an ordinal smaller than Γ_i and if $(P,Y) \in U_i^+$ and $\psi(O,X,P,Y)$, then $(P,Y) \in V^\lambda_{G(O,X)}$. Moreover, as Γ_i is inaccessible, $G \in R(\Gamma_i)$, where $R(\Gamma_i)$ denotes the Γ_i-th element of the standard cumulative hierarchy. Therefore $\bigcup ran(G)$ is also an ordinal smaller than Γ_i. We define an ordinal β to be $\max(\lambda rk(A), \bigcup ran(G))$.

Now take any $(M,B) \in C^+$, so $M \Vdash_\rho B \in C$. Then, by the definition of N_2 and Lemma 4.1.9 there is $(O,X) \in A^+$ and $(O_1,Z) \in U_i^+$ such that $N_2 \downarrow \lambda a.\ N_{21}$, $N_{21}[a := \emptyset] \downarrow \lambda x.\ N_{22}$, $N_{22}[x := M] \downarrow [\emptyset, N_{23}]$, $N_{23} \downarrow \langle O, N_{24}\rangle$, $N_{24} \downarrow [\emptyset, N_{25}]$,

$N_{25} \downarrow \langle O_1, R\rangle$ and $R \Vdash_\rho B = (X, Z)$. Let $M_1 = \mathrm{lei}\ \emptyset\ \emptyset\ \emptyset\ \langle M, R\rangle$, then $M_1 \Vdash_\rho (X, Z) \in C$. Take any element $(P, Y) \in F(O, X)$ and accompanying Q_1, Q_2. Then $Q_2 \downarrow \lambda a.\ Q_3$, $Q_3[a := \emptyset] \downarrow \lambda x.\ Q_4$ and $Q_4[x := M_1] \Vdash_\rho Z = Y$. By Lemma 3.2.32, $\lambda rk(Z) \leq \lambda rk(Y)$ and thus $\lambda rk(Z) \leq \beta$. Since $(O, X) \in A^+$, $\lambda rk(X) \leq \beta$, too. By Lemma 3.2.36, $\lambda rk(B) \leq \beta + 2$. By Lemma 4.1.8, $rk(B) \leq \beta + \omega$, so $rk(C^+) \leq \beta + \omega + 1$. By Lemma 4.1.8 again, $\lambda rk(C) \leq \beta + 2\omega$. Since $\beta + 2\omega$ is still smaller than Γ_i, we get the claim. ∎

Lemma 4.1.11 *If $M \Vdash_\rho A \in U_{i,\gamma}$, then $M \Vdash_\rho A \in V_i$.*

Proof If $M \Vdash_\rho A \in U_{i,\gamma}$, then $M \downarrow \mathrm{inRep}(N)$, $N \downarrow [\emptyset, O]$, $O \downarrow \langle O_1, O_2\rangle$ and there is C such that $O_1 \downarrow v$, $(v, C) \in U_{i,\gamma}$, $O_2 \Vdash_\rho C = A$. Then also $(v, C) \in U_i$, so $O_1 \Vdash_\rho C \in_I V_i$, so also $M \Vdash_\rho A \in V_i$. ∎

Lemma 4.1.12 *If $N \Vdash_\rho \psi_i(C, U_{i,\gamma})$, where ψ_i is one of the five clauses defining $\phi_1^i(C, U_{i,\gamma})$ in the Definition 4.1.1, then $N \Vdash_\rho \psi_i(C, V_i)$.*

Proof There are five cases to consider:

- $N \Vdash_\rho C = V_{i-1}$. This case is trivial.
- $N \Vdash_\rho \exists a.\ a \in U_{i,\gamma} \wedge c \in a$. Then there is A such that $N \downarrow [\emptyset, O]$, $O \downarrow \langle O_1, O_2\rangle$, $O_1 \Vdash_\rho A \in U_{i,\gamma}$, $O_2 \Vdash_\rho C \in A$. By Lemma 4.1.11, $O_1 \Vdash_\rho A \in V_i$, so also $N \Vdash_\rho \exists a.\ a \in V_i \wedge c \in a$.
- $N \Vdash_\rho \exists a.\ a \in U_{i,\gamma} \wedge c = \bigcup a$. Then there is A such that $N \downarrow [\emptyset, O]$, $O \downarrow \langle O_1, O_2\rangle$, $O_1 \Vdash_\rho A \in U_{i,\gamma}$, $O_2 \Vdash_\rho C = \bigcup A$. Thus by Lemma 4.1.11 $O_1 \Vdash_\rho A \in V_i$ and we get the claim in the same way as in the previous case.
- $N \Vdash_\rho \exists a.\ a \in U_{i,\gamma} \wedge C = P(a)$. Similar to the previous case.
- $N \Vdash_\rho \exists a.\ a \in U_{i,\gamma} \wedge C \in a \to U_{i,\gamma}$. Then there is A such that $N \downarrow [\emptyset, O]$, $O \downarrow \langle O_1, O_2\rangle$, $O_1 \Vdash_\rho A \in U_{i,\gamma}$, $O_2 \Vdash_\rho$ "C is a function from A into $U_{i,\gamma}$".

By Lemma 4.1.11, $O_1 \Vdash_\rho A \in V_i$. Expanding the second part, we have $O_2 \downarrow \langle P_1, P_2 \rangle$, $P_1 \Vdash_\rho \forall x \in A \exists! y \in U_{i,\gamma}.\ (x, y) \in C$ and $P_2 \Vdash_\rho \forall z \in C \exists x \in A \exists y \in U_{i,\gamma}.\ z = (x, y)$. We will tackle P_1 and P_2 separately.

– For P_1, we have for all X, $P_1 \downarrow \lambda a.\ P_{11}$, $P_{11}[a := \emptyset] \downarrow \lambda x.\ Q$ and for all $R \Vdash_\rho X \in A$ there is Y such that $Q[x := R] \downarrow [\emptyset, Q_0]$, $Q_0 \downarrow \langle Q_1, Q_2 \rangle$, $Q_1 \Vdash_\rho Y \in U_{i,\gamma}$ and $Q_2 \Vdash_\rho (X, Y) \in C \wedge \forall z.\ (X, z) \in C \to z = Y$. By Lemma 4.1.11 we also have $Q_1 \Vdash_\rho Y \in V_i$, so also $P_1 \Vdash_\rho \forall x \in a \exists! y.\ y \in V_i \wedge (x, y) \in C$.

– For P_2, we have for all Z, $P_2 \downarrow \lambda a.\ P_{11}$, $P_{11}[a := \emptyset] \downarrow \lambda x.\ Q$ and for all $R \Vdash_\rho Z \in C$ there are X, Y such that $Q[x := R] \downarrow [t_1, Q_0]$, $Q_0 \downarrow \langle Q_1, Q_2 \rangle$ and $Q_1 \Vdash_\rho X \in A$. Moreover, $Q_2 \downarrow [\emptyset, S_0]$, $S_0 \downarrow \langle S_1, S_2 \rangle$ and $S_1 \Vdash_\rho Y \in U_{i,\gamma}$. By Lemma 4.1.11 we also have $S_1 \Vdash_\rho Y \in V_i$, so also $P_2 \Vdash_\rho \forall z \in C \to \exists x \in A \exists y \in V_i.\ z = (x, y)$.

Therefore also $O_2 \Vdash_\rho$ "C is a function from A into V_i" and in the end $N \Vdash_\rho \exists a.\ a \in V_i \wedge C \in a \to V_i$. ■

Corollary 4.1.13 *If $M \Vdash_\rho \phi_1^i(C, U_{i,\gamma})$, then $M \Vdash_\rho \phi_1^i(C, V_i)$.*

Now we can prove the new case in the main Lemma:

Lemma 4.1.14 $(M, C) \in [\![t_A(\vec{u})]\!]_\rho$ *iff* $M = \mathrm{axRep}(N)$ *and* $N \Vdash_\rho \phi_A(C, \overrightarrow{[\![u]\!]_\rho})$.

Proof We first show the left-to-right direction. Suppose $(M, A) \in U_i$, then $M = \mathrm{inac_iRep}(N)$. We must have $N \Vdash_\rho \phi_1^i(A, U_{i,\gamma}) \wedge \forall d.\ \phi_2^i(d) \to A \in d$ for some ordinal γ. Then $N \downarrow \langle N_1, N_2 \rangle$, $N_1 \Vdash_\rho \phi_1^i(A, U_{i,\gamma})$, $N_2 \Vdash_\rho \forall d.\ \phi_2^i(d) \to A \in d$. Corollary 4.1.13 gives us $N_1 \Vdash_\rho \phi_1^i(A, V_i)$, so $N \Vdash_\rho \phi_1^i(A, V_i) \wedge \forall d.\ \phi_2^i(d) \to A \in d$, which is what we want.

For the right-to-left direction, suppose $N \Vdash_\rho \phi_1^i(C, V_i) \land \forall d.\ \phi_2^i(d) \to C \in d$. We need to show that $(\mathrm{inac_i Rep(N)}, C) \in U_i$. By the definition of U_i it suffices to show that $C \in V_{\Gamma_i}$. We have $N \downarrow \langle N_1, N_2 \rangle$ and $N_1 \Vdash_\rho$ "C is equal to V_{i-1} or there is $A \in V_i$ such that C is a powerset/union/member of A, or C is a function from A into V_i.". The proof splits into corresponding five cases. The first four are easy to prove using Lemma 3.2.32 and the definition of the ordinal γ in the clause 4 in the definition of realizability. The last one follows by Lemma 4.1.10. ∎

The normalization theorem is proved in exactly the same way as for IZF_R. The same applies to the properties DP, NEP, EP and TEP. Our developments in Section 3.3 thus provide a mechanism to extract programs from $\mathrm{IZF}_{R\omega}$ proofs. We have therefore defined an impredicative constructive set theory with inaccessible sets, with program extraction capability, which at the same time can be used to provide semantics for important type theories such as the Calculus of Inductive Constructions. These developments demonstrate the power and adaptability of our approach; note how little was needed from the conceptual point of view to extend our framework for IZF_R to $\mathrm{IZF}_{R\omega}$. A natural question, which we leave open, is whether it can be extended with even stronger axioms, such as the existence of a Mahlo set.

4.2 Dependent set theory

In the introduction, we stated that type theories form a basis for computation and that they are used extensively in computer science. In the previous section, we remarked on the problem of understanding type theories. What is so difficult to understand about them is that the underlying logics are very different from all the logics we presented so far in this thesis.

One particular difference, present in all type theories used in practice, is the treatment of quantifiers, justified in the end by the BHK interpretation. Recall that in IFOL and λH, the typing rules regulating quantifiers are:

$$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash \lambda a.\ M : \forall a.\ \phi}\ a \notin FV_F(\Gamma) \qquad \frac{\Gamma \vdash M : \forall a.\ \phi}{\Gamma \vdash M\ t : \phi[a := t]}$$

$$\frac{\Gamma \vdash M : \phi[a := t]}{\Gamma \vdash [t, M] : \exists a.\ \phi} \qquad \frac{\Gamma \vdash M : \exists a.\ \phi \quad \Gamma, x : \phi \vdash N : \psi}{\Gamma \vdash \text{let } [a, x : \phi] := M \text{ in } N : \psi}\ a \notin FV_F(\Gamma, \psi)$$

The corresponding clauses in the BHK interpretation are:

- The construction of $\exists a.\ \phi$ consists of an object t and a construction of $\phi[a := t]$.
- The construction of $\forall a.\ \phi$ is a method which transforms any object t to a construction of $\phi[a := t]$.

While the typing rules above do realize the interpretation, a different reading of the interpretation is possible. If a construction of $\exists a.\ \phi$ consists of an object t and a construction of $\phi[a := t]$, maybe it should be possible to put our hands on this object and the corresponding construction. Thus, we can construct alternative rules for the existential quantifier:

$$\frac{\Gamma \vdash M : \phi[a := t]}{\Gamma \vdash [t, M] : \exists a.\ \phi} \qquad \frac{\Gamma \vdash M : \exists a.\ \phi}{\Gamma \vdash \pi_2^{a.\phi}(M) : \phi[a := \pi_1^{a.\phi}(M)]}$$

Note the difference. There are new object terms in the logic: $\pi_1^{a.\phi}(M)$, providing witnesses to existential statements. Thus in particular such a theory automatically possesses TEP, as $\pi_1^{a.\phi}(M)$ witnesses any formula $\exists a.\ \phi$ such that $\vdash M : \exists a.\ \phi$.

However, some questions must be addressed. For example, what exactly is a formula in such a system? Note that now proof terms can appear in formulas. Thus formulas can *depend* on proofs; this is the reason why type theories which include this interpretation of the existential quantifier are called *dependent* type theories.

As a dependent theory can be seen as an extension of first-order logic with proof terms (the standard rules for the existential quantifier are derivable in type theories), natural questions arise: can we extend a set theory to include dependent features? And would such an extension be of any use?

In this section, we provide answers to these questions. More specifically, we extend IZF_R and λZ to incorporate several features typical of type theories — dependent implications, conjunctions and what we call restricted Σ-types. We call the resulting "dependent" set theory IZF_D and the underlying lambda calculus λD.

There are several attractive properties of IZF_D. First of all, λD still normalizes. We use a strong version of the Axiom of Choice to provide the interpretation of new set terms. The normalization result makes it possible to extract programs from IZF_D proofs, as explained in Section 3.3.

Second, we show that the combination of dependent features in the logic and Replacement axiom significantly increases the power of a set theory, by proving that IZF_D can prove the axioms of IZF with Collection (IZF_C). As known since the results of Friedman and Ščedrov [Fv85], Replacement and Collection are not equivalent in the constructive setting. While the proof-theoretic power of IZF_C equals that of ZFC [Fri73], IZF_R is weaker: it has less provably recursive functions [Fv85]. It is conjectured in [Fv85] that its consistency can be proved in ZF. Moreover, Collection is a very useful tool in the development of mathematics in constructive set theories; most notably in the treatment of inductive definitions [AR01, Rat05b]. Thus, IZF_D is a remarkably strong set theory, having the proof-theoretic power of ZFC and all the benefits of Collection at its disposal.

The importance of consistency results in this area cannot be overestimated, as theories tend to dwell close to inconsistency. This is one reason for the restriction of Σ-types we adopt, which amounts to disallowing the standard reduction rule

$\pi_1([t, M]) \to t$. Although IZF$_D$ with unrestricted Σ-types enjoys useful proof-theoretic properties, such as Subject Reduction, we show that it is also inconsistent.

We proceed to present IZF$_D$. However, as common with dependent theories, a separate presentation of formulas, terms and lambda terms is impossible. IZF$_D$ is one large lambda calculus with types, whose judgments include provable statements, valid terms and formulas.

4.2.1 IZF$_\mathbf{D}$

The theory IZF$_D$ is a dependent version of IZF$_R$. It arises by extending the constructive first-order logic of IZF$_R$ with dependent features. As any detailed account of a theory based on dependent logic involves a large amount of syntax, we postpone the formal treatment to the next section and first describe the theory informally.

Intuitively, the axioms of IZF$_D$ are: Empty Set, Pairing, Infinity, Power Set, $\in$-Induction and dependent Separation and Replacement. The underlying logic is an extension of IFOL by dependent implications, conjunctions and restricted Σ-types. Formally, IZF$_D$ does not have any axioms in the traditional sense; it is a logic powerful enough to *derive* all the formulas listed in Figure 4.1. However, these formulas are helpful in defining and understanding IZF$_D$.

As in IZF$_R$, the axioms (IN) and (EQ) along with the intensional membership symbol $\in_I$ form a backbone of the Leibniz ($\forall \vec{f}, a, b.\ a = b \to \phi(a, \vec{f}) \to \phi(b, \vec{f})$) and Extensionality ($\forall a, b.\ (\forall c.\ c \in a \leftrightarrow c \in b) \to a = b$) axioms, which are derivable in our axiomatization. Similarly, IZF$_D$ can prove all the axioms with $\in_I$ replaced by $\in$. A closer look at the corresponding proofs in Section 3.2 reveals that these statements are true only for first-order formulas. See the discussion in Section 4.2.2.

The underlying logic includes dependent implications and conjunctions, de-

- (IN) $\forall a, b.\ a \in b \leftrightarrow \exists c.\ c \in_I b \wedge a = c$
- (EQ) $\forall a, b.\ a = b \leftrightarrow \forall d.\ (d \in_I a \to d \in b) \wedge (d \in_I b \to d \in a)$
- (EMPTY) $\forall c.\ c \in_I \emptyset \leftrightarrow \bot$
- (PAIR) $\forall a, b \forall c.\ c \in_I \{a, b\} \leftrightarrow c = a \vee c = b$
- (INF) $\forall c.\ c \in_I \omega \leftrightarrow c = 0 \vee \exists b \in \omega.\ c = S(b)$
- ($\mathrm{SEP}_{p,a,\vec{f}.\ \phi)}$) $\forall \vec{f}, a \forall c.\ c \in_I S_{a,\vec{f}.\ \phi}(a, \vec{f}) \leftrightarrow (p : c \in a) \wedge \phi[a := c]$
- (UNION) $\forall a \forall c.\ c \in_I \bigcup a \leftrightarrow \exists b \in a.\ c \in b$
- (POWER) $\forall a \forall c.\ c \in_I P(a) \leftrightarrow \forall b.\ b \in c \to b \in a$
- ($\mathrm{REPL}_{p,a,b,\vec{f}.\ \phi}$) $\forall \vec{f}, a \forall c.\ c \in_I R_{p,a,b\vec{f}.\ \phi}(a, \vec{f}) \leftrightarrow (\forall x.\ (p : x \in a) \to \exists! y.\ \phi[a, b := x, y]) \wedge (\exists x.\ (p : x \in a) \wedge \phi[a, b := x, c])$
- ($\mathrm{IND}_{\phi(a,\vec{f})}$) $\forall \vec{f}.\ (\forall a.\ (\forall b.\ (b \in_I a) \to \phi(b, \vec{f})) \to \phi(a, \vec{f})) \to \forall a.\ \phi(a, \vec{f})$

Figure 4.1: The axioms of $\mathrm{IZF_D}$

noted by $(p : \phi) \to \psi$ and $(p : \phi) \wedge \psi$. These can be found in the Separation and Replacement axioms. Their parameterizing formulas can depend on proofs, denoted by p. Intuitively, in IZF_D proofs are a valid subject of discourse. This is the main feature which distinguishes the axioms of IZF_D from traditional axiomatizations. In particular, note that the axioms of IZF_R are precisely what remains if the schemas are restricted to purely first-order formulas.

4.2.2 The terms of λD

The terms of λD are divided into three syntactic categories, encompassing proof terms, set terms and formulas, respectively. We will generally use letters M, N, O, P for proof terms, s, t, u for set terms, ϕ, ψ, ϑ for formulas and T, S for arbitrary terms. Thus, whenever one of these symbols is encountered in the text, the reader should assume that it has been generated by the corresponding part of the grammar. There are two kinds of variables. The first one, denoted by letters p, q, x, y, z, as usual corresponds to the propositional implications. The second one, denoted usually by letters a, b, c, intuitively corresponds to the first-order quantification. We call them *proof* and *set* variables, respectively. The following abstract grammar defines the terms of λD. The first part generates the proof terms:

$$M ::= x \mid M\ N \mid \lambda a.\ M \mid \lambda x : \phi.\ M \mid \mathrm{inl}(M) \mid \mathrm{inr}(M) \mid$$

$$\mathrm{fst}(M) \mid \mathrm{snd}(M) \mid [t, M] \mid M\ t \mid \langle M, N \rangle \mid$$

$$\mathrm{case}(M, x : \phi.\ N, x : \psi.\ O) \mid \mathrm{magic}(M) \mid \pi_2^{a.\phi}(M)$$

$$\mathrm{ind}_{a,\vec{f}.\ \phi(a,\vec{f})}(M, \vec{t})$$

$$\mathrm{inProp}(t, u, M) \mid \mathrm{inRep}(t, u, M)$$

$$\mathrm{eqProp}(t, u, M) \mid \mathrm{eqRep}(t, u, M)$$

$$\mathrm{pairProp}(t, u_1, u_2, M) \mid \mathrm{pairRep}(t, u_1, u_2, M)$$

$$\mathrm{unionProp}(t, u, M) \mid \mathrm{unionRep}(t, u, M)$$

$$\mathrm{sep}_{p,a,\vec{f}.\phi}\mathrm{Prop}(t, u, \vec{u}, M) \mid \mathrm{sep}_{p,a,\vec{f}.\phi}\mathrm{Rep}(t, u, \vec{u}, M)$$

$$\mathrm{powerProp}(t, u, M) \mid \mathrm{powerRep}(t, u, M)$$

$$\mathrm{infProp}(t, M) \mid \mathrm{infRep}(t, M)$$

$$\mathrm{repl}_{p,a,b,\vec{f}.\phi}\mathrm{Prop}(t, u, \vec{u}, M) \mid \mathrm{repl}_{p,a,b,\vec{f}.\phi}\mathrm{Rep}(t, u, \vec{u}, M)$$

As before, we adopt the convention of using axRep and axProp terms to tacitly mean all Rep and Prop terms, for ax being one of in, eq, pair, union, sep, power, inf and repl.

The second part of the grammar generates the set terms:

$$\begin{aligned} t \quad ::= \quad & a \mid \pi_1^{a.\phi}(M) \mid \emptyset \mid \{t_1, t_2\} \mid \omega \mid P(t) \mid \bigcup t \mid \\ & S_{p,a,\vec{f}.\phi}(t, \vec{t}) \mid R_{p,a,b,\vec{f}.\phi}(t, \vec{t}) \end{aligned}$$

The term $S_{p,a,\vec{f}.\phi}(t, \vec{t})$ intuitively corresponds to the set $\{(p : a \in t) \mid \phi\}$. The term $R_{p,a,b,\vec{f}.\phi}(t, \vec{t})$ intuitively corresponds to the set $\{y \mid (\forall (p : x \in t) \exists ! y.\ \phi[a, b, \vec{f} := x, y, \vec{t}]) \wedge (\exists p : x \in t.\ \phi[a, b, \vec{f} := x, y, \vec{t}])\}$. The term $\pi_1^{a.\phi}(M)$ can be thought of as a dependent version of the Hilbert's epsilon operator $\epsilon a.\ \phi$. These intuitions are justified by the typing system in Section 4.2.2.

The third part generates the formulas of IZF_D:

$$\phi \ ::= \ \perp \mid (x : \phi) \to \psi \mid (x : \phi) \wedge \psi \mid \phi \vee \psi \mid \forall a.\ \phi \mid \exists a.\ \phi$$

The formulas $(x : \phi) \to \psi$ and $(x : \phi) \wedge \psi$ are dependent versions of implication and conjunction. The variable x binds in ψ, which can mention x (inside of $\pi_1^{a.\phi}$ terms). Traditional formulas $\phi \to \psi$ and $\phi \wedge \psi$ are defined as abbreviations for $(x : \phi) \to \psi$ and $(x : \phi) \wedge \psi$, where x is fresh.

Definition 4.2.1 *A* lambda term *is a term generated by the first part of the grammar. A* set term *is a term generated by the second part of the grammar. A* formula *is a term generated by the third part of the grammar.*

The free variables of a term M are denoted by $FV(M)$. The definition of $FV(M)$, as well as the definition of the (capture-avoiding) substitution, follows the grammar in a natural way, taking into account the formulas appearing in subscripts and superscripts of terms. We show two representative cases of the definition:

$$\begin{aligned} FV(\pi_1^{a.\phi}(M)) &= (FV(\phi) \setminus \{a\}) \cup FV(M) \\ FV(\mathrm{sep}_{p,a,\vec{f}.\phi}\mathrm{Rep}(u, \vec{u}, M)) &= (FV(\phi) \setminus \{p, a, \vec{f}\}) \cup \\ & \quad FV(u) \cup FV(\vec{u}) \cup FV(M) \end{aligned}$$

The reduction relation, denoted by $\to$, is deterministic and defined on lambda terms. It arises from the following reduction rules and evaluation contexts:

$$(\lambda x : \phi.\ M)\ N \to M[x := N] \qquad (\lambda a.\ M)\ t \to M[a := t]$$

$$\mathrm{fst}(\langle M, N\rangle) \to M \qquad \mathrm{snd}(\langle M, N\rangle) \to N \qquad \pi_2^{a.\phi}([t, M]) \to M$$

$$\mathrm{case}(\mathrm{inl}(M), x : \phi.\ N, x : \psi.\ O) \to N[x := M]$$

$$\mathrm{case}(\mathrm{inr}(M), x : \phi.\ N, x : \psi.\ O) \to O[x := M]$$

$$\mathrm{axProp}(t, \vec{u}, \mathrm{axRep}(t, \vec{u}, M)) \to M$$

$$\mathrm{ind}_{a,\vec{f}.\ \phi}(M, \vec{t}) \to \lambda c.\ M\ c\ (\lambda b.\lambda x : b \in_I c.\ \mathrm{ind}_{a,\vec{f}.\ \phi}(M, \vec{t})\ b)$$

The evaluation contexts still describe call-by-need (lazy) evaluation order:

$$\begin{aligned} [\circ] \ ::= \ & \mathrm{fst}([\circ]) \mid \mathrm{snd}([\circ]) \mid \mathrm{case}([\circ], x : \phi.\ N, x : \psi.\ O) \mid \\ & \pi_2^{a.\phi}([\circ]) \mid \mathrm{axProp}(t, \vec{u}, [\circ]) \mid [\circ]\ M \mid \mathrm{magic}([\circ]) \end{aligned}$$

The standard reduction rule $\pi_1^{a.\phi}([t, M]) \to t$ is not present. The reasons for this omission will become clear in Section 4.2.2.

The set of λD-values will be denoted by λD_v. In the definition, $t, \vec{u}, \phi, M, N$ are arbitrary terms.

$$V ::= \lambda a.\ M \mid \lambda x : \phi.\ M \mid \mathrm{inr}(M) \mid \mathrm{inl}(M) \mid [t, M] \mid \langle M, N\rangle \mid \mathrm{axRep}(t, \vec{u}, M)$$

We now introduce a type system for λD. Contexts, denoted by Γ, are finite sequences of pairs (z, T), where z is a variable and T is either a formula or a string Set. The *domain* of a context $\Gamma = z_1 : T_1, \ldots, z_n : T_n$, denoted by $\mathrm{dom}(\Gamma)$, is the sequence $z_1, \ldots, z_n$, treated as set when convenient. There are three kinds of typing judgments:

- $\Gamma \vdash t : \mathrm{Set}$, read as "$t$ is a set term in the context Γ".
- $\Gamma \vdash \phi : \mathrm{Form}$, read as "$\phi$ is a formula in the context Γ".
- $\Gamma \vdash M : \phi$, read as: "M is a proof of the formula ϕ in the context Γ".

We therefore incorporate the definition of terms and formulas in the typing system. This is necessary in order to ensure that the arguments of $\pi_1^{a.\phi}(M)$ terms are valid proofs of $\exists a.\ \phi$. We start with rules for terms:

$$\frac{}{\Gamma, a : \mathrm{Set} \vdash a : \mathrm{Set}}\ a \notin \mathrm{dom}(\Gamma)$$

$$\frac{\Gamma \vdash \vec{u} : \mathrm{Set}}{\Gamma \vdash t_A(\vec{u}) : \mathrm{Set}}$$

$$\frac{\Gamma \vdash t, \vec{t} : \mathrm{Set} \quad \Gamma, a, \vec{f} : \mathrm{Set}, p : a \in t \vdash \phi : \mathrm{Form}}{\Gamma \vdash S_{p,a,\vec{f}.\ \phi}(t, \vec{t}) : \mathrm{Set}}$$

$$\frac{\Gamma \vdash t, \vec{t} : \mathrm{Set} \quad \Gamma, a, b, \vec{f} : \mathrm{Set}, p : a \in t \vdash \phi : \mathrm{Form}}{\Gamma \vdash R_{p,a,b,\vec{f}.\ \phi}(t, \vec{t}) : \mathrm{Set}}$$

Furthermore, we define the formulas:

$$\frac{\Gamma \vdash \phi : \mathrm{Form}}{\Gamma, x : \phi \vdash x : \phi}\ x \notin \mathrm{dom}(\Gamma)$$

$$\frac{}{\Gamma \vdash \bot : \mathrm{Form}} \qquad \frac{\Gamma \vdash t : \mathrm{Set} \quad \Gamma \vdash u : \mathrm{Set}}{\Gamma \vdash t \circ u : \mathrm{Form}}\ \circ \in \{\in_I, =, \in\}$$

$$\frac{\Gamma \vdash \phi : \mathrm{Form} \quad \Gamma \vdash \psi : \mathrm{Form}}{\Gamma \vdash \phi \vee \psi : \mathrm{Form}} \qquad \frac{\Gamma \vdash \phi : \mathrm{Form} \quad \Gamma, x : \phi \vdash \psi : \mathrm{Form}}{\Gamma \vdash (x : \phi) \circ \psi : \mathrm{Form}}\ \circ \in \{\to, \wedge\}$$

$$\frac{\Gamma, a : \mathrm{Set} \vdash \phi : \mathrm{Form}}{\Gamma \vdash Qa.\ \phi : \mathrm{Form}}\ Q \in \{\forall, \exists\}$$

And finally, we define the proofs. First, the rules governing the dependent logic. We need to incorporate Weakening into the formal system, as it is no longer provable.

$$\frac{\Gamma \vdash S : T}{\Gamma, a : \mathrm{Set} \vdash S : T}\ a \notin \mathrm{dom}(\Gamma) \qquad \frac{\Gamma \vdash S : T \quad \Gamma \vdash \phi : \mathrm{Form}}{\Gamma, x : \phi \vdash S : T}\ x \notin \mathrm{dom}(\Gamma)$$

$$\frac{\Gamma, x : \phi \vdash M : \psi}{\Gamma \vdash \lambda x : \phi.\ M : (x : \phi) \to \psi} \qquad \frac{\Gamma, a : \mathrm{Set} \vdash M : \phi}{\Gamma \vdash \lambda a.\ M : \forall a.\ \phi}$$

$$\frac{\Gamma \vdash M : (x : \phi) \to \psi \quad \Gamma \vdash N : \phi}{\Gamma \vdash M\ N : \psi[x := N]} \qquad \frac{\Gamma \vdash M : \forall a.\ \phi \quad \Gamma \vdash t : \mathrm{Set}}{\Gamma \vdash M\ t : \phi[a := t]}$$

$$\frac{\Gamma \vdash M : \phi \quad \Gamma \vdash N : \psi[x := M]}{\Gamma \vdash \langle M, N \rangle : (x : \phi) \wedge \psi}$$

$$\frac{\Gamma \vdash M : (x : \phi) \wedge \psi}{\Gamma \vdash \mathrm{fst}(M) : \phi} \qquad \frac{\Gamma \vdash M : (x : \phi) \wedge \psi}{\Gamma \vdash \mathrm{snd}(M) : \psi[x := \mathrm{fst}(M)]}$$

$$\frac{\Gamma \vdash t : \mathrm{Set} \quad \Gamma \vdash M : \phi[x := t]}{\Gamma \vdash [t, M] : \exists a.\ \phi}$$

$$\frac{\Gamma \vdash M : \exists a.\ \phi}{\Gamma \vdash \pi_1^{a.\phi}(M) : \mathrm{Set}} \qquad \frac{\Gamma \vdash M : \exists a.\ \phi}{\Gamma \vdash \pi_2^{a.\phi}(M) : \phi[a := \pi_1^{a.\phi}(M)]}$$

$$\frac{\Gamma \vdash M : \phi}{\Gamma \vdash \mathrm{inl}(M) : \phi \vee \psi} \qquad \frac{\Gamma \vdash M : \psi}{\Gamma \vdash \mathrm{inr}(M) : \phi \vee \psi}$$

$$\frac{\Gamma \vdash M : \phi \vee \psi \quad \Gamma, x : \phi \vdash N : \vartheta \quad \Gamma, x : \psi \vdash O : \vartheta}{\Gamma \vdash \mathrm{case}(M, x : \phi.\ N, x : \psi.\ O) : \vartheta}$$

$$\frac{\Gamma \vdash M : \bot}{\Gamma \vdash \mathrm{magic}(M) : \phi}$$

Second, we present the rules corresponding to set theory.

$$\frac{\Gamma \vdash M : \forall c.\ (\forall b.\ b \in c \to \phi[a, \vec{f} := b, \vec{t}]) \to \phi[a, \vec{f} := c, \vec{t}] \quad \Gamma \vdash \vec{t} : \mathrm{Set}}{\Gamma \vdash \mathrm{ind}_{a, \vec{f}.\ \phi}(M, \vec{t}) : \forall a.\ \phi[\vec{f} := \vec{t}]}$$

$$\frac{\Gamma \vdash M : \phi_A(t, \vec{u}) \quad \Gamma \vdash t, \vec{u} : \mathrm{Set}}{\Gamma \vdash \mathrm{axRep}(t, \vec{u}, M) : t \in_I t_A(\vec{u})} \qquad \frac{\Gamma \vdash M : t \in_I t_A(\vec{u})}{\Gamma \vdash \mathrm{axProp}(t, \vec{u}, M) : \phi_A(t, \vec{u})}$$

$$\frac{\Gamma \vdash M : \exists c.\ c \in_I u \wedge t = c}{\Gamma \vdash \mathrm{inRep}(t, u, M) : t \in u} \qquad \frac{\Gamma \vdash M : t \in u}{\Gamma \vdash \mathrm{inProp}(t, u, M) : \exists c.\ c \in_I u \wedge t = c}$$

$$\frac{\Gamma \vdash M : \forall d.\ (d \in_I t \to d \in u) \wedge (d \in_I u \to d \in t)}{\Gamma \vdash \mathrm{eqRep}(t, u, M) : t = u}$$

$$\frac{\Gamma \vdash M : t = u}{\Gamma \vdash \mathrm{eqProp}(t, u, M) : \forall d.\ (d \in_I t \to d \in u) \wedge (d \in_I u \to d \in t)}$$

We write $\Gamma \vdash T : S$, when this judgment can be derived using the typing rules. The theory IZF_D arises from the typing system, by considering the formulas ϕ such that $\vdash M : \phi$ for some term M, to be provable in IZF_D.

Most of the rules are standard. The typing system incorporates the definition of formulas and terms of set theory. The term $\pi_1^{a.\phi}$ can be thought of as a version of the Hilbert's epsilon operator, as it provides a witness to any provable existential quantifier. For example, if $\vdash M : \exists a.\ a = P(\omega)$, then $\pi_1^{a.\phi}(M)$ is "the" A such that $A = P(\omega)$: $\vdash \pi_2^{a.\phi}(M) : \pi_1^{a.\phi}(M) = P(\omega)$. In fact, a dependent version of the Hilbert's axiom is provable, as it is easy to see that $\vdash \lambda x : \exists a.\ \phi.\ \pi_2^{a.\phi}(x) : (x : \exists a.\ \phi) \to \phi[a := \pi_1^{a.\phi}(x)]$.

Non-extensionality of $\pi_1^{a.\phi}$ operator.

The $\pi_1^{a.\phi}$ operator is non-extensional — from the facts that $M : \exists a.\ \phi$, $N : \exists a.\ \psi$ and $O : \forall a.\ \phi \leftrightarrow \psi$ we cannot derive $\pi_1^{a.\phi}(M) = \pi_1^{a.\psi}(N)$. For this reason, there are instances of the Leibniz axiom not provable in IZF_D, such as $(*)$ $a = b \to \pi_1^{c.\phi(a)}(M) \in e \to \pi_1^{c.\phi(b)}(M) \in e$. However, IZF_D does show $\forall a, b.\ a = b \to \phi(a) \to \phi(b)$ for all formulas not mentioning $\pi_1^{a.\phi}$ terms. Moreover, the formula (*) does not correspond to reasoning in mathematical practice. We hope to investigate this topic further in the future.

Although IZF_D might seem formidable at the first sight, we remark that its complexity does not surpass that of other formal systems intended for general use [The04, Muz93, Kre02].

Sadly, IZF_D does not possess the nice proof-theoretic properties we are so used to. In particular, Subject Reduction and Progress *do not* hold. The reasons are mostly explained below — an extension of IZF_D making it possible to prove the standard properties makes the theory inconsistent. From the practical point of view, this should not be considered a shortcoming. See the discussion in Section 4.2.6.

Unrestricted Sigma-types

Recall that $=_\to$ denotes the smallest equivalence relation extending $\to$. There are two natural rules missing from λD: the reduction rule $\pi_1^{a.\phi}([t, M]) \to t$ and the typing rule:

$$\frac{\Gamma \vdash M : \psi}{\Gamma \vdash M : \phi}\ \phi =_\to \phi \qquad (*)$$

Let IZF_D^Σ denote IZF_D extended with these rules. Unlike IZF_D, IZF_D^Σ enjoys nice proof-theoretic properties, such as Subject Reduction. However, as the following theorem shows, it also suffers the property of being inconsistent.

Theorem 4.2.2 *IZF_D^{Σ} is inconsistent.*

Proof Recall first that in set theories, $0 = \emptyset, 1 = \{\emptyset\}$. For the informal proof, consider the set $B = \{x \in 1 \mid \exists a.\ a = a\}$. We can show that for any p proving $x \in B$, there is exactly one y which witnesses the formula $\exists a.\ a = a$, namely the set A used for proving p. Formally, we set $y = \pi_1^{a.\ a=a}(\mathrm{snd}(\mathrm{sep}_{\exists \mathrm{a.\ a=a}}\mathrm{Prop}(x, 1, p)))$. By the Replacement axiom, all these y's can be collected in one set C. Now take any set D and use it to show that $\exists a.\ a = a$ and furthermore that $0 \in B$. Applying (*) to the y corresponding to this proof, we easily find that $D \in C$. Therefore C contains all sets and thus is subject to Russell's paradox.[1]

For the formal proof, we only present the relevant terms and provable judgments. Let eqRefl denote the term corresponding to the proof of $\forall a.\ a = a$, let 0in1 denote the term corresponding to the proof of $0 \in 1$ and let russ denote the proof term corresponding to the proof of $\forall a.\ (\forall b.\ b \in a) \to \bot$. The terms are

[1] Russell's paradox is not necessary to derive contradiction, as $\in$-induction together with $C \in C$ is also contradictory.

probably best read in a bottom-up fashion.

$$\begin{array}{rcl}
B & \equiv & S_{\exists a.\ a=a}(1) \\
t & \equiv & \pi_1^{a.\ a=a}(\mathrm{snd}(\mathrm{sep}_{\exists \mathrm{a.\ a=a}}\mathrm{Prop}(x, 1, p))) \\
M & \equiv & \langle \mathrm{eqRefl}\ t, \lambda z.\ \lambda q : z = t.\ q \rangle \\
N & \equiv & \lambda x.\ \lambda p : x \in B.\ [t, M] \\
\vdash N & : & \forall x.\ (p : x \in B) \to \exists! y.\ y = t \\
C & \equiv & R_{p,x,y.\ y=t}(B) \\
P & \equiv & \mathrm{sep}_{\exists \mathrm{a.\ a=a}}\mathrm{Rep}(0, 1, \langle 0\mathrm{in}1, [a, \mathrm{eqRefl}\ a] \rangle) \\
a : \mathrm{Set} \vdash P & : & 0 \in B \\
Q & \equiv & \lambda a.\ \mathrm{repl}_{\mathrm{p,x,y.\ y=t}}\mathrm{Rep}(a, B, \langle N, [0, \langle P, \mathrm{eqRefl}\ a \rangle] \rangle) \\
\vdash Q & : & \forall a.\ a \in C \\
\vdash \mathrm{russ}\ C\ Q & : & \bot \qquad\qquad \blacksquare
\end{array}$$

4.2.3 Realizability

As we mentioned earlier, we need to use a strong version of the Axiom of Choice to define the realizability relation. Let ZFO be the Zermelo-Fraenkel set theory extended with the binary relational symbol $<$ and the axiom stating that $<$ well-orders the universe. In this section we work in ZFO. Although ZFO might seem excessive as a metatheory for the purpose of proving normalization of a constructive system, we remark that with a bit more effort and slightly more obscure presentation, we could carry out the proof in ZF (relativizing all statements to L, a naturally well-ordered universe). Moreover, we conjecture that the proof could be formalized in IZF_D. Thus, if the conjecture is true, IZF_D is capable of selfvalidating itself. The reason for our belief in the conjecture is that the following definition, crucial for the interpretation of $\pi_1^{a.\phi}(M)$ terms, essentially *defines* the $\pi_1^{a.\phi}(M)$ term in the first-order set-theoretic setting:

Definition 4.2.3 *If $\phi(a)$ is a ZFO formula, then* "the first a such that ϕ" *is defined to be:*

- *The empty set, if there is no A such that $\phi(A)$.*
- *The smallest set A in the ordering $<$ such that $\phi(A)$ holds, otherwise.*

As usually, we employ the erasure map from λD to $\lambda\overline{D}$. The reductions are still first-order ignorant, so we replace the first-order terms by $\emptyset$. The map is defined inductively on all terms in an obvious way. We show several representative cases:

$$\overline{x} = x \qquad \overline{a} = \emptyset \qquad \overline{M\ N} = \overline{M}\ \overline{N} \qquad \overline{(\lambda a.\ M)} = \lambda a.\ \overline{M}$$

$$\overline{\pi_1^{a.\phi}(M)} = \emptyset \qquad \overline{(t_A(\vec{u}))} = \emptyset \qquad \overline{\lambda x : \phi.\ M} = \lambda x.\ \overline{M}$$

$$\overline{\pi_2^{a.\phi}(M)} = \pi_2^{a.\overline{\phi}}(\overline{M}) \quad \overline{\mathrm{axRep}(t, \vec{u}, M)} = \mathrm{axRep}(\overline{M})$$

Definition 4.2.4 *A* realizer *is any closed term of $\lambda\overline{D}$.*

The standard Lemma continues to hold:

Lemma 4.2.5 *If $\overline{M}$ normalizes, then so does M.*

We proceed to define the realizability relation $M \Vdash_\rho \phi$, read as "M realizes ϕ", where M is a realizer and ϕ comes from the extended language L defined below. The class of λ-names is defined as usual:

Definition 4.2.6 *A set A is a λ-name iff A is a set of pairs (v, B) such that $v \in \lambda\overline{D}_{vc}$ and B is a λ-name.*

Definition 4.2.7 *The class of λ-names is denoted by V^λ.*

As usual, we now extend the language of IZF_D to encompass the λ-names. We also restrict the formulas by allowing only realizers R as arguments of $\pi_1^{a.\phi}([\circ])$ terms. We call the resulting class-sized language L. Thus, the grammar is extended and modified by:

$$t ::= A \mid \pi_1^{a.\phi}(R) \mid \ldots$$

From now on until the end of this section, symbols M, N, O, P range exclusively over realizers, letters a, b, c vary over set variables in the language, letters A, B, C vary over λ-names, letters ϕ, ψ over formulas in L. Environments are finite partial functions from set variables to V^λ.

Definition 4.2.8 *For any formula ϕ of L, any set term t of L and ρ defined on all free variables of ϕ and t, we define by metalevel induction a realizability relation $M \Vdash_\rho \phi$ in an environment ρ and a meaning of a term $[\![t]\!]_\rho$ in an environment ρ. We show the new cases in the definition compared with IZF_R:*

- $[\![\pi_1^{a.\phi}(M)]\!]_\rho$ *is the first A such that $M \downarrow [\emptyset, N]$ and $N \Vdash_\rho \phi[a := A]$.*
- $[\![t_A(\vec{u})]\!]_\rho \equiv \{(\mathrm{axRep}(\emptyset, \vec{\emptyset}, N), B) \in R \times V_\gamma^\lambda \mid N \Vdash_\rho \phi_A(B, [\![\vec{u}]\!]_\rho)\}$. *The ordinal γ is defined below.*
- $M \Vdash_\rho (x : \phi) \wedge \psi \equiv M \downarrow \langle M_1, M_2 \rangle \wedge (M_1 \Vdash_\rho \phi) \wedge (M_2 \Vdash_\rho \psi[x := M_1])$
- $M \Vdash_\rho (x : \phi) \to \psi \equiv (M \downarrow \lambda x.\ \ M_1) \wedge \forall N.\ (N \Vdash_\rho \phi) \to (M_1[x := N] \Vdash_\rho \psi[x := N])$

The definition of the ordinal γ in item 4 stays exactly the same for all terms, apart from the Replacement term, where the definition is slightly changed: Let $\vec{\alpha} = \overrightarrow{\lambda rk([\![u]\!]_\rho)}$ and let $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)$. Case $t_A(\vec{u})$ of:

- $R_{p,a,b,\vec{f}.\ \phi}(u,\vec{u})$. To make the account clearer, we set:

$$\phi(M,A,B,\vec{F}) \equiv \phi[p,a,b,\vec{f} := M,A,B,\vec{F}]$$

  Let $G = \{(N_1,(N_{21},B)) \in \lambda\overline{D}_c \times [\![u]\!]^+_\rho \mid \exists d \in V^\lambda.\ \psi(N_1,N_{21},B,d)\}$, where $\psi(N_1,N_{21},B,d) \equiv (N_1 \downarrow \lambda a.\ N_{11}) \wedge (N_{11}[a := \emptyset] \downarrow \lambda x.\ O) \wedge (O[x := N_{21}] \downarrow [\emptyset, O_1]) \wedge O_1 \Vdash_\rho \phi(N_{21},B,d,\overrightarrow{[\![u]\!]_\rho}) \wedge \forall e.\ \phi(N_{21},B,e,\overrightarrow{[\![u]\!]_\rho}) \to e = d)$. Then for all $g \in G$ there is D and $(N_1,(N_{21},B))$ such that $g = (N_1,(N_{21},B))$ and $\psi(N_1,N_{21},B,D)$. Use Collection to collect these D's in one set H, so that for all $g \in G$ there is $D \in H$ such that the property holds. Apply Replacement to H to get the set of λ-ranks of sets in H. Then $\beta \equiv \bigcup H$ is an ordinal and for any $D \in H$, $\lambda rk(D) < \beta$. Therefore for all $g \in G$ there is $D \in V^\lambda_\beta$ and $(N_1,(N_{21},B))$ such that $g = (N_1,(N_{21},B))$ and $\psi(N_1,N_{21},B,D)$ holds. Set $\gamma = \beta + 1$.

It is easy to see that this definition of realizability is also well-founded. The standard lemmas continue to hold:

Lemma 4.2.9 $[\![t[a := s]]\!]_\rho = [\![t[a := [\![s]\!]_\rho]]\!]_\rho = [\![t]\!]_{\rho[a:=[\![s]\!]_\rho]}$ *and* $M \Vdash_\rho \phi[a := s]$ *iff* $M \Vdash_\rho \phi[a := [\![s]\!]_\rho]$ *iff* $M \Vdash_{\rho[a:=[\![s]\!]_\rho]} \phi$.

Proof We show the new case, that is $t = \pi_1^{b.\phi}(M)$. In this situation, $[\![t[a := s]]\!]_\rho = [\![\pi_1^{b.\phi[a:=s]}(M[a := s])]\!]_\rho$. Since M is a realizer, $M[a := s] = M$. This is thus the first A such that $M \downarrow [\emptyset, N]$ and $N \Vdash_\rho \phi[a := s][b := A]$. Furthermore $[\![t[a := [\![s]\!]_\rho]]\!]_\rho = [\![\pi_1^{b.\phi[a:=[\![s]\!]_\rho]}(M)]\!]_\rho$, which is the first A such that $M \downarrow [\emptyset, N]$ and $N \Vdash_\rho \phi[a := [\![s]\!]_\rho][b := A]$. By the induction hypothesis for ϕ, this is equivalent to $N \Vdash_\rho \phi[a := s][b := A]$. Finally, $[\![t]\!]_{\rho[a:=[\![s]\!]_\rho]} = [\![\pi_1^{b.\phi}(M)]\!]_{\rho[a:=[\![s]\!]_\rho]}$, which is the first A such that $M \downarrow [\emptyset, N]$ and $N \Vdash_{\rho[a:=[\![s]\!]_\rho]} \phi[b := A]$. By the induction hypothesis for ϕ, this is equivalent to $N \Vdash_\rho \phi[b := A][a := s]$, which is equivalent to $N \Vdash_\rho \phi[a := s][b := A]$. ■

The standard lemmas continue to hold:

Lemma 4.2.10 *If* $(M \Vdash_\rho \phi)$ *then* $M \downarrow$.

Lemma 4.2.11 *If* $M \to^* M'$ *then* $M' \Vdash_\rho \phi$ *iff* $M \Vdash_\rho \phi$.

Realizability is also invariant with respect to reductions of lambda terms inside of set terms and formulas:

Lemma 4.2.12 *If* $M \to^* N$, *then* $[\![t[x := M]]\!]_\rho = [\![t[x := N]]\!]_\rho$ *and* $O \Vdash_\rho \phi[x := M]$ *iff* $O \Vdash_\rho \phi[x := N]$.

Proof Straightforward induction on the definition of realizability. ■

The keystone in the normalization proof is proved similarly as before:

Lemma 4.2.13 $(M, C) \in [\![t_A(\vec{u})]\!]_\rho$ *iff* $M = \mathrm{axRep}(N)$ *and* $N \Vdash_\rho \phi_A(C, \overrightarrow{[\![u]\!]_\rho})$.

Proof We only show the right-to-left part for the two terms where the proof differs from the proof of Lemma 3.2.38. Suppose $N \Vdash_\rho \phi_A(A, \overrightarrow{[\![u]\!]_\rho})$ and $M = \mathrm{axRep}(N)$. To show that $(M, A) \in [\![t_A(\vec{u})]\!]_\rho$, we need to show that $A \in V^\lambda_\gamma$. Let $\vec{\alpha} = \overrightarrow{\lambda rk([\![u]\!]_\rho)}$. Case $t_A(\vec{u})$ of:

- $S_{p,a,\vec{f}.\ \phi}(u, \vec{u})$. Suppose $N \Vdash_\rho p : A \in [\![u]\!]_\rho \land \phi[a, \vec{f} := A, \overrightarrow{[\![u]\!]_\rho})$. Then $N \downarrow \langle N_1, N_2\rangle$ and $N_1 \Vdash_\rho A \in [\![u]\!]_\rho$. Lemma 3.2.32 shows the claim.

- $R_{p,a,b,\vec{f}.\ \phi}(u, \vec{u})$. As before, to make the account clearer, we set:

  $$\phi(M, A, B, \vec{F}) \equiv \phi[p, a, b, \vec{f} := M, A, B, \vec{F}]$$

  Suppose $N \Vdash_\rho (\forall p : x \in [\![u]\!]_\rho \exists ! y.\ \phi(p, x, y, \overrightarrow{[\![u]\!]_\rho})) \land \exists p : x \in [\![u]\!]_\rho.\ \phi(p, x, A, \overrightarrow{[\![u]\!]_\rho})$. Then $N \downarrow \langle N_1, N_2\rangle$ and $N_2 \Vdash_\rho \exists x.\ (p : x \in [\![u]\!]_\rho) \land \phi(p, x, A, [\![\vec{u}]\!]_\rho)$. Thus $N_2 \downarrow [\emptyset, N_{20}]$ and there is B such that $N_{20} \downarrow \langle N_{21}, N_{22}\rangle$ and $N_{21} \Vdash_\rho B \in [\![u]\!]_\rho$ and $N_{22} \Vdash_\rho \phi(N_{21}, B, A, \overrightarrow{[\![u]\!]_\rho})$. We also have $N_1 \Vdash_\rho \forall x.\ (p : x \in [\![u]\!]_\rho) \to$

$\exists! y.\ \phi(p, x, y, \overrightarrow{[\![u]\!]_\rho})$, so $N_1 \downarrow \lambda a.\ N_{11}$ and for all C, $N_{11}[a := \emptyset] \downarrow \lambda x.\ O$ and for all $P \Vdash_\rho C \in [\![u]\!]_\rho$, $O[x := P] \Vdash_\rho \exists! y.\ \phi(P, C, y, \overrightarrow{[\![u]\!]_\rho})$. So taking $C = B$, and $P = N_{21}$, there is D such that $N_1 \downarrow \lambda a.\ N_{11}$, $N_{11}[a := \emptyset] \downarrow \lambda x.\ O$, $O[x := N_{21}] \downarrow [\emptyset, O_1]$ and $O_1 \Vdash_\rho \phi(N_{21}, B, D, \overrightarrow{[\![u]\!]_\rho}) \wedge \forall e.\ \phi(N_{21}, B, e, \overrightarrow{[\![u]\!]_\rho}) \to e = D$. Therefore $(N_1, (N_{21}, B)) \in G$ from the definition of γ, so there is $D \in V^\lambda_\gamma$ such that $N_1 \downarrow \lambda a.\ N_{11}$, $N_{11} \downarrow \lambda x.O$, $O[x := N_{21}] \downarrow [\emptyset, O_1]$ and $O_1 \Vdash_\rho \phi(N_{21}, B, D, \overrightarrow{[\![u]\!]_\rho}) \wedge \forall e.\ \phi(N_{21}, B, e, \overrightarrow{[\![u]\!]_\rho}) \to e = D$. So $O_1 \downarrow \langle O_{11}, O_{12} \rangle$ and $O_{12} \Vdash_\rho \forall e.\ \phi(N_{21}, B, e, \overrightarrow{[\![u]\!]_\rho}) \to e = D$. Therefore, $O_{12} \downarrow \lambda a.\ Q$, $Q[a := \emptyset] \downarrow \lambda x.\ Q_1$ and $Q_1[x := N_{22}] \Vdash_\rho A = D$. By Lemma 3.2.32, $A \in V^\lambda_\gamma$. ■

4.2.4 Normalization

We are now ready to prove that λD normalizes, thus enabling program extraction from IZF_D proofs. The environments in this section are finite partial functions which map set variables to V^λ and proof variables to realizers. As usual, any such environment can be used as a realizability environment by ignoring the mapping of proof variables.

Definition 4.2.14 *For any term T with free proof variables $x_1, \ldots, x_n$ and ρ defined on $x_1, \ldots, x_n$, $T[\rho]$ denotes $T[x_1 := \rho(x_i), \ldots, x_n := \rho(x_n)]$.*

Definition 4.2.15 *For a sequent $\Gamma \vdash M : \phi$, $\rho \models \Gamma$ means that ρ is defined on* $\mathrm{dom}(\Gamma)$*, for all $(a_i, \mathrm{Set}) \in \mathrm{dom}(\Gamma)$, $\rho(a_i) \in V^\lambda$ and for all $(x_i, \phi_i) \in \Gamma$, $\rho(x_i) \Vdash_\rho$* $\overline{\phi_i}[\rho]$.

Theorem 4.2.16 (Normalization) *If $\Gamma \vdash O : \vartheta$ then for all $\rho \models \Gamma$, $\overline{O}[\rho] \Vdash_\rho \overline{\vartheta}[\rho]$.*

Proof We proceed by metalevel induction on $\Gamma \vdash O : \vartheta$. As usual, we write O' to denote $\overline{O}[\rho]$, where ρ is clear from the context. Note first that O' is a realizer. We only show the new cases in the proof. Case $\Gamma \vdash O : \vartheta$ of:

- $$\frac{\Gamma \vdash M : \exists a.\ \phi}{\Gamma \vdash \pi_2^{a.\phi}(M) : \phi[a := \pi_1^{a.\phi}(M)]}$$

  Take any $\rho \models \Gamma$. Note that $(\phi[a := \pi_1^{a.\phi}(M)])' = \phi'[a := \pi_1^{a.\phi'}(M')]$. By the induction hypothesis, $M' \Vdash_\rho \exists a.\ \phi'$, so $M' \downarrow [\emptyset, N]$ and there is some A such that $N \Vdash_\rho \phi'[a := A]$. Furthermore, $[\![\pi_1^{a.\phi'}(M')]\!]_\rho$ is the first A such that $M' \downarrow [\emptyset, Q]$ and $Q \Vdash_\rho \phi'[a := A]$, so also $N \Vdash_\rho \phi'[a := [\![\pi_1^{a.\phi'}(M')]\!]_\rho]$. By Lemma 4.2.9, $N \Vdash_\rho \phi'[a := \pi_1^{a.\phi'}(M')]$. Since $\pi_2^{a.\phi'}(M') \to^* N$, by Lemma 4.2.11 $\pi_2^{a.\phi'}(M') \Vdash_\rho \phi'[a := \pi_1^{a.\phi'}(M')]$, which shows the claim.

- $$\frac{\Gamma \vdash M : \phi \quad \Gamma \vdash N : \psi[x := M]}{\Gamma \vdash \langle M, N \rangle : (x : \phi) \land \psi}$$

  Take any $\rho \models \Gamma$. By the induction hypothesis, $M' \Vdash_\rho \phi'$ and $N' \Vdash_\rho \psi'[x := M']$, which is precisely what needs to be shown.

- $$\frac{\Gamma \vdash M : (x : \phi) \land \psi}{\Gamma \vdash \mathrm{fst}(M) : \phi}$$

  The proof is the same as in case of λZ.

- $$\frac{\Gamma \vdash M : (x : \phi) \land \psi}{\Gamma \vdash \mathrm{snd}(M) : \psi[x := \mathrm{fst}(M)]}$$

  Take any $\rho \models \Gamma$. By the induction hypothesis, $M' \downarrow \langle M_1, M_2 \rangle$ and $M_2 \Vdash_\rho \psi'[x := M_1]$. As $\mathrm{snd}(M') \to^* M_2$, by Lemma 4.2.11 it suffices to show that $M_2 \Vdash_\rho (\psi[x := \mathrm{fst}(M)])'$, which is equivalent to $M_2 \Vdash_\rho \psi'[x := \mathrm{fst}(M')]$. Since $\mathrm{fst}(M') \to^* M_1$ and $(\psi[x := \mathrm{fst}(M)])' = \psi'[x := \mathrm{fst}(M')]$, Lemma 4.2.12 shows the claim.

- $$\frac{\Gamma \vdash M : (x : \phi) \to \psi \quad \Gamma \vdash N : \phi}{\Gamma \vdash M\ N : \psi[x := N]}$$

  Take any $\rho \models \Gamma$. By the induction hypothesis, for some ϕ_1, $M' \downarrow \lambda x.\ M_1$, $N' \Vdash_\rho \phi'$ and for all $P \Vdash_\rho \phi'$, $M_1[x := P] \Vdash_\rho \psi'[x := P]$. Thus in particular $M_1[x := N'] \Vdash_\rho \psi'[x := N']$. As $(M\ N)' = M'\ N' \to^* (\lambda x.\ M_1)\ N' \to M_1[x := N']$ and $(\psi[x := N])' = \psi'[x := N']$, Lemma 4.2.11 shows the claim.

- $$\frac{\Gamma, x : \phi \vdash M : \psi}{\Gamma \vdash \lambda x : \phi.\ M : (x : \phi) \to \psi}$$

  Take any $\rho \models \Gamma$. We need to show that for any $N \Vdash_\rho \phi'$, $M'[x := N] \Vdash_\rho \psi'[x := N]$. Take any such N. Since $\rho[x := N] \models \Gamma, x : \phi$, by the induction hypothesis $\overline{M}[\rho[x := N]] \Vdash_\rho \overline{\psi}[\rho[x := N]]$. It is easy to see that this is equivalent to $M'[x := N] \Vdash_\rho \psi'[x := N]$. ■

Corollary 4.2.17 (Normalization) *If $\vdash M : \phi$, then $M' \downarrow$ and thus also $M \downarrow$.*

Corollary 4.2.18 *IZF_D is consistent.*

As we mentioned, λD does not have the Subject Reduction property. Thus, IZF$_D$ does not necessarily have any of the usual properties: DP, NEP and TEP. This is the price we pay for avoiding inconsistency of unrestricted Σ-types. The benefits are presented in the following section.

4.2.5 The properties of λD

In this section, we relate IZF$_D$ and its classical counterpart to well-known first-order set theories.

Theorem 4.2.19 *IZF_D proves IZF_C.*

Proof The precise formulation of the claim is: if $\mathrm{IZF}_C \vdash \phi$, then for some term M, $\mathrm{IZF}_D \vdash M : \phi$. We formulate IZF_C as IZF_R extended with the Collection axiom schema:

$$\forall \vec{f}.\ \forall a.\ (\forall x \in a \exists y.\ \phi) \to \exists b.\ \forall x \in a \exists y \in b.\ \phi$$

To show that IZF_D interprets IZF_R, we first need to prove that it interprets the rules of IFOL. Most of them are present in the type system of λD as special cases when dependencies are not used. The only missing rule is elimination of the existential quantifier.

$$\frac{\Gamma \vdash \exists a.\ \phi \quad \Gamma \vdash \forall a.\ \phi \to \psi}{\Gamma \vdash \psi}\ a \notin FV(\psi)$$

It is easy to show that in IZF_D the following rule is admissible, that is if assumptions are derivable, then so is the conclusion:

$$\frac{\Gamma \vdash M : \exists a.\ \phi \quad \Gamma \vdash N : \forall a.\ \phi \to \psi}{\Gamma \vdash N\ (\pi_1^{a.\phi}(M))\ (\pi_2^{a.\phi}(M)) : \psi}\ a \notin FV(\psi)$$

Second, we need to provide the interpretation of IZF_R terms in IZF_D and show that they satisfy the respective axioms. This is straightforward, as it suffices to add extraneous binders for Separation and Replacement terms. For example, we interpret $\{x \in a \mid \phi\}$ as $\{p : x \in a \mid \phi\}$, where p is fresh.

The only nontrivial thing left is the interpretation of the Collection axiom. Intuitively, it follows from Replacement, as using dependent implication and $\pi_1^{a.\phi}$ terms, we can transform a proof p of $\forall x \in a \exists y.\ \phi$ into $\forall q : x \in a \exists ! y.\ \phi \land y = \pi_1^{a.\phi}(p\ x\ q)$. Formally, we exhibit the proof terms. To increase readability, we display $\forall x.\ (p : x \in a) \to \phi$ as $\forall p : x \in a.\ \phi$, $\exists x.\ (p : x \in a) \land \phi$ as $\exists p : x \in a.\ \phi$ and $R_{p,x,y,\vec{f}.\ \phi}(t, \vec{t})$ as $\{y \mid (\forall p : x \in t \exists ! y.\ \phi[\vec{f} := \vec{t}]) \land (\exists p : x \in a.\ \phi[\vec{f} := \vec{t}])\}$.

$$
\begin{array}{rcl}
M_1 & \equiv & \langle \pi_2^{y.\phi}(p\ x\ q), \mathrm{eqRefl}\ \pi_1^{y.\phi}(p\ x\ q)\rangle \\
\vdash M_1 & : & \psi \\
M_2 & \equiv & \lambda z.\ \lambda r : \phi[y := z] \land z = \pi_1^{y.\phi}(p\ x\ q).\ \mathrm{snd}(r) \\
\vdash M_2 & : & \forall z.\ \phi[y := z] \land z = \pi_1^{y.\phi}(p\ x\ q) \to z = \pi_1^{y.\phi}(p\ x\ q) \\
M_3 & \equiv & \lambda q : x \in a.\ [\pi_1^{y.\phi}(p\ x\ q), \langle M_1, M_2\rangle] \\
\vdash M_3 & : & \forall q : x \in a \exists! y.\ \psi \\
M_4 & \equiv & [x, \langle q, \langle \pi_2^{y.\phi}(p\ x\ q), \mathrm{eqRefl}\ \pi_1^{y.\phi}(p\ x\ q)\rangle\rangle] \\
\vdash M_4 & : & \exists q : x \in a.\ \psi \\
M_5 & \equiv & \mathrm{repl}_{\mathrm{q,x,y,\tilde{f}}.\psi}\mathrm{Rep}(\pi_1^{y.\phi}(p\ x\ q), a, \vec{f}, \langle M_3, M_4\rangle) \\
\vdash M_5 & : & \pi_1^{y.\phi}(p\ x\ q) \in t \\
M_6 & \equiv & \lambda x.\ \lambda q : x \in a.\ [\pi_1^{y.\phi}(p\ x\ q), \langle M_5, \pi_2^{y.\phi}(p\ x\ q)\rangle] \\
\vdash M_6 & : & \forall x \in a.\ \exists y \in t.\ \phi \\
\psi & \equiv & \phi \land y = \pi_1^{y.\phi}(p\ x\ q) \\
t & \equiv & \{y \mid (\forall q : x \in a \exists! y.\ \psi) \land \exists q : x \in a.\ \psi\} \\
N & \equiv & \lambda \vec{f}.\ \lambda p : \forall x \in a \exists y.\ \phi.\ [t, M_6] \\
\vdash N & : & \forall \vec{f}.\ (\forall x \in a \exists y.\ \phi) \to \exists b.\ \forall x \in a \exists y \in b.\ \phi
\end{array}
\qquad \blacksquare
$$

Therefore, by the results of [Fv85], the proof-theoretic strength of IZF_D equals that of ZFC.

We now consider a classical version of IZF_D. Let ZF_D be IZF_D extended with the excluded middle axiom EM. We show that ZF_D is consistent. For this purpose, take a formulation of ZFO with set terms, such as IZF_R + EM + "the universe is well-ordered by $<$". Define an erasure map on formulas and set terms of λD, which returns formulas and set terms of ZFO. The representative cases of the definition follow, where $\iota a.\phi$ denotes "the first a such that ϕ":

$$\overline{a} = a \quad \overline{\pi_1^{a.\phi}(M)} = \iota a.\phi \quad \overline{\emptyset} = \emptyset \quad \overline{\{t_1, t_2\}} = \{\overline{t_1}, \overline{t_2}\} \quad \overline{\omega} = \omega$$

$$\overline{(p : \phi) \to \psi} = \overline{\phi} \to \overline{\psi} \quad \overline{S_{p,a,\vec{f}.\phi}(t, \vec{t})} = \{a \in t \mid \overline{\phi(a, \vec{t})}\}$$

$$\overline{R_{p,a,b,\vec{f}.\phi}(t, \vec{t})} = \{y \mid \forall x \in \vec{t}\exists! y.\ \overline{\phi}(x, y, \vec{t}) \land \exists x \in \vec{t}.\ \overline{\phi}(x, y, \vec{t})\}$$

With the map at hand, we can easily prove by induction on the proof the consistency result:

Theorem 4.2.20 *If* $ZF_D \vdash t : \mathrm{Set}$, *then* t *is a term of ZFO. If* $ZF_D \vdash M : \phi$, *then* $ZFO \vdash \overline{\phi}$. *Thus* ZF_D *is consistent.*

Theorem 4.2.21 ZF_D *interprets ZF.*

Proof By Theorem 4.2.19, IZF_D interprets IZF_C. Since $\mathrm{ZF}=\mathrm{IZF}_C + \mathrm{EM}$, the claim follows. ■

4.2.6 Program extraction

The program extraction method we described in Section 3.3 is based on DP, NEP and TEP. Given that IZF_D does not possess Subject Reduction property, we cannot use our standard tools to prove these properties for IZF_D. However, it is easy to derive them for the realizability model, in the sense we are about to define.

Definition 4.2.22 *The formula* ϕ *is* true *in the realizability model if there is a realizer* M *such that* $M \Vdash_\emptyset \phi$.

We write $V^\lambda \models \phi$, when ϕ is true in the realizability model.

Lemma 4.2.23 V^λ *has the DP, NEP and TEP properties.*

Proof The claim amounts to replacing provability in the standard formulations by realizability.

To see that DP holds, assume $V^\lambda \models \phi \vee \psi$. Then there is a realizer M such that $M \Vdash_\emptyset \phi \vee \psi$, so either $M \downarrow \mathrm{inl}(N)$ and $N \Vdash_\emptyset \phi$ or $M \downarrow \mathrm{inr}(N)$ and $N \Vdash_\emptyset \psi$. In the former case $V^\lambda \models \phi$, in the latter $V^\lambda \models \psi$.

NEP is proved in exactly the same manner — mirroring the proof for the previous lambda calculi we described. For TEP, assume $V^\lambda \models \exists a.\ \phi$. Then there is a realizer M such that $M \Vdash_\emptyset \exists a.\ \phi$, so $\pi_2^{a.\phi}(M) \Vdash_\emptyset \phi[a := \pi_1^{a.\phi}(M)]$, so $V^\lambda \models \phi[a := \pi_1^{a.\phi}(M)]$. ■

Thus, using techniques from Section 3.3, we can extract programs from IZF_D proofs via the realizability model. If $\mathrm{IZF}_D \vdash M : \phi$ then also $\overline{M} \Vdash_\rho \phi$, for appropriate ρ and we can now apply Lemma 4.2.23 to obtain necessary properties. The downside of this approach, compared to our previous developments, is that the extracted programs are correct with respect to the realizability model, instead of the original theory. Thus, if we extract a natural number n from a proof $\mathrm{IZF}_D \vdash M : \exists a \in \omega.\ \phi$, then we can only assert that $V^\lambda \models \phi[a := n_n]$. This distinction would likely be of no significant concern in applications, as thanks to Theorem 4.2.16 we know that the truth in the realizability world is consistent with statements provable in IZF_D. Moreover, it is easy to make the problem disappear, by *axiomatizing* Subject Reduction in IZF_D, that is adding the rule:

$$\frac{\Gamma \vdash N : \phi}{\Gamma \vdash M : \phi}\ (M \to N) \vee (N \to M)$$

This is actually the approach adopted in Nuprl.

4.2.7 Conclusion

We presented a combination of two worlds we investigate — sets and types. By adding features typical for dependent type theories to the first-order logic underlying IZF_R, we gained new capabilities and proof-theoretic strength. The price we pay is the loss of Subject Reduction property, attractive from the theoretical point of view. Given Theorem 4.2.2, it is unclear if there is a simple method to restore the property back.

4.3 Historical context

Inaccessible sets, called cardinals in a classical setting, are the first "large" objects, whose existence cannot be proved in ZF/IZF. [Kan03] provides a detailed account of "Higher Infinite" in the world of sets. In the constructive context, powerful large set axioms (including the existence of class-many inaccessibles) were added to IZF_C by Friedman and Ščedrov [Fv84]. The notion of an inaccessible set they use differs from ours, as their inaccessibles must also model the Collection axiom. We do not know if these two notions coincide. Both DP and NEP were shown for the resulting theories, but we do not think that SEP and TEP could be proved with their technique.

Inaccessible sets were also investigated in the context of weaker, predicative CZF. Crosilla and Rathjen [CR02] showed that the power of inaccessible set axioms might be closely linked to the $\in$-induction axiom. They proved that inaccessible sets added to CZF with $\in$-induction taken away do not add any proof-theoretical power. Setzer and Rathjen also investigated Mahlo axioms in the context of constructive set and type theories [Rat03, Set00].

There is a significant amount of research on connections between type and set theories. Aczel [Acz78, Acz82, Acz86, Acz99] described mutual interpretations of variants of CZF and Martin-Löf type theory. Werner [Wer97] did the same thing for Zermelo set theory and Calculus of Constructions. Miquel [Miq03, Miq04] investigated embeddings of impredicative set theories without $\in$-induction axiom schema in type theories. Howe [How96] investigated an extension of the set theoretic universe with type-theoretical constructs in order to validate the type theory of Nuprl.

Modifications of logic underlying set theory were investigated before. Agerholm and Gordon [AG95, Gor96] studied classical higher-order set theory HOL-ST. They did not find a clear advantage of HOL-ST over first-order ZF. A map theory [Gru92] provides a unified framework for sets and computation. An ongoing research on algebraic set theory [MP02] investigates set theories based on category theory. There are also set theories based on linear logics [Shi94, Ter04].

CHAPTER 5

PROGRAM-EXTRACTING SEMANTICS

We will now present an application of the frameworks we set up in the previous chapter. Namely, we show that standard set-theoretic semantics for simple type theories using our IZF_R can be used as a basis for program extracting capability.

Church's Higher-Order logic [Chu40] has been remarkably successful at capturing the intuitive reasoning of mathematicians. It was distilled from *Principia Mathematica*, and is sometimes called the Simple Theory of Types based on that legacy. It incorporates the lambda calculus as its notation for functions, including propositional functions, thus interfacing well with computer science.

One of the reasons Higher-Order logic is successful is that its axiomatic basis is very small, and it has a clean set-theoretic semantics at a low level of the cummulative hierarchy of sets (up to $\omega + \omega$) and can thus be formalized in a small fragment of ZFC set theory . This means it interfaces well with standard mathematics and provides a strong basis for trust. Moreover, the set theory semantics is the basis for many extensions of the core logic; for example, it is straightforward to add arrays, recursive data types, and records to the logic.

Church's theory is the logical basis of two of the most successful interactive provers used in hardware and software verification, HOL and PVS. This is due in part to the two characteristics mentioned above in addition to its elegant automation based on Milner's tactic mechanism and its elegant formulation in the ML metalanguage.

Until recently, one of the few drawbacks of HOL was that its logical base did not allow a way to express a constructive subset of the logic. This issue was considered by Harrison for HOL-light [Har96], and recently Berghofer implemented a constructive version of HOL in the Isabelle implementation [Ber04, BN02] in large part to enable the extraction of programs from constructive proofs. This raises the question of finding a semantics for HOL that justifies this intuitively sound extraction.

The standard justification for program extraction is based on logics that embedded extraction deeply into their semantics; this is the case for the Calculus of Inductive Constructions (CIC) [CPM90, BC04], Minlog [BBS+98], Computational Type Theory (CTT) [ABC+06, CAB+86] or the closely related Intuitionistic Type Theory (ITT) [ML82, NPS90]. The mechanism of extraction is built deeply into the logic and the provers based on it, e.g. Agda [ACN90] is based on ITT, Coq [The04] on CIC, MetaPRL [HNC+03] and Nuprl [ACE+00] on CTT.

In this section we show that there is a way to provide a clean set-theoretic semantics for HOL and at the same stroke use it to semantically justify program extraction. The idea is to first factor HOL into its constructive core, say Constructive HOL, plus the axioms of excluded middle and choice. The semantics for this language can be given in ZFC set theory, and in this semantics, IZF_R provides the semantics for Constructive HOL. We furthermore use our developments in Section 3.3 to provide a program extraction from CHOL proofs via the semantics.

Our set-theoretic semantics for HOL has the following properties:

- It is as simple as the standard semantics, presented in Gordon and Melham's [GM93].
- It works in constructive set theory.
- It provides a semantic basis for program extraction.
- It can be applied to the constructive version of HOL recently implemented in Isabelle-HOL as a means of using constructive HOL proofs as programs.

This chapter is organized as follows. In section 5.1 we present a constructive version of HOL. In section 5.2 we define set-theoretic semantics. We show how to use the semantics for program extraction in section 5.3.

5.1 Higher-order logic

In this section, we present higher-order logic in detail. There are two syntactic categories: *terms* and *types*. The types are generated by the following abstract grammar:

$$\tau ::= \text{nat} \mid \text{bool} \mid \text{prop} \mid \tau \to \tau \mid (\tau, \tau)$$

The distinction between bool and prop corresponds to the distinction between the two-element type and the type of propositions in type theory, or between the two-element object and the subobject classifier in category theory or, as we shall see, between 2 and the set of all subsets of 1 in constructive set theory.

The terms of HOL are generated by the following abstract grammar:

$$t ::= x_\tau \mid c_\tau \mid (t_{\tau\to\sigma}\ u_\tau)_\sigma \mid (\lambda x_\tau.\ t_\sigma)_{\tau\to\sigma} \mid (t_\tau, s_\sigma)_{(\tau,\sigma)}$$

Thus each term t_α in HOL is annotated with a type α, which we call *the type of* t. We will often skip annotating terms with types; this practice should not lead to confusion, as the implicit type system is very simple. Terms of type prop are called *formulas*.

The free variables of a term t are denoted by $FV(t)$ and defined as usual. We consider α-equivalent terms equal.

Definition 5.1.1 *A* formula *is a term of type* prop.

Our version of HOL has a set of built-in constants. To increase readability, we write $c : \tau$ instead of c_τ to provide the information about the type of c. If the type of a constant involves α, it is a constant *schema*: there is one constant for each type τ substituted for α. There are thus constants $=_{\text{bool}}$, $=_{\text{nat}}$ and so on.

$$\bot : \text{prop} \qquad \top : \text{prop} \qquad =_\alpha : (\alpha, \alpha) \to \text{prop}$$

$$\Rightarrow: (\text{prop}, \text{prop}) \to \text{prop} \qquad \wedge : (\text{prop}, \text{prop}) \to \text{prop} \qquad \vee : (\text{prop}, \text{prop}) \to \text{prop}$$

$$\forall_\alpha : (\alpha \to \text{prop}) \to \text{prop} \qquad \exists_\alpha : (\alpha \to \text{prop}) \to \text{prop} \qquad \varepsilon_\alpha : (\alpha \to \text{prop}) \to \alpha$$

$$0 : \text{nat} \qquad S : \text{nat} \to \text{nat} \qquad \text{false} : \text{bool} \qquad \text{true} : \text{bool}$$

We present the proof rules for HOL in a sequent-based natural deduction style. A *sequent* is a pair (Γ, t), where Γ is a list of formulas and t is a formula. Free variables of a context are the free variables of all its formulas. A sequent (Γ, t) is written as $\Gamma \vdash t$. We write binary constants (equality, implication, etc.) using infix notation. We use standard abbreviations for quantifiers: $\forall a : \tau.\ \phi$ abbreviates $\forall_\tau(\lambda a_\tau.\ \phi)$, similarly with $\exists a : \tau.\ \phi$. The proof rules for HOL are:

$$\frac{}{\Gamma \vdash t}\ t \in \Gamma \qquad \frac{}{\Gamma \vdash t = t} \qquad \frac{\Gamma \vdash t = s}{\Gamma \vdash \lambda x_\tau.\ t = \lambda x_\tau.\ s}\ x_\tau \notin FV(\Gamma)$$

$$\frac{\Gamma \vdash t \quad \Gamma \vdash s}{\Gamma \vdash t \wedge s} \qquad \frac{\Gamma \vdash t \wedge s}{\Gamma \vdash t} \qquad \frac{\Gamma \vdash t \wedge s}{\Gamma \vdash s} \qquad \frac{}{\Gamma \vdash \top}$$

$$\frac{\Gamma \vdash t}{\Gamma \vdash t \vee s} \qquad \frac{\Gamma \vdash s}{\Gamma \vdash t \vee s} \qquad \frac{\Gamma \vdash t \vee s \quad \Gamma, t \vdash u \quad \Gamma, s \vdash u}{\Gamma \vdash u}$$

$$\frac{\Gamma, t \vdash s}{\Gamma \vdash t \to s} \qquad \frac{\Gamma \vdash s \to t \quad \Gamma \vdash s}{\Gamma \vdash t} \qquad \frac{\Gamma \vdash s = u \quad \Gamma \vdash t[u]}{\Gamma \vdash t[s]}$$

$$\frac{\Gamma \vdash f_{\alpha \to \text{prop}}\ t_\alpha}{\Gamma \vdash \exists_\alpha(f_{\alpha \to \text{prop}})} \qquad \frac{\Gamma \vdash \exists_\alpha(f_{\alpha \to \text{prop}}) \quad \Gamma, f_{\alpha \to \text{prop}}\ x_\alpha \vdash u}{\Gamma \vdash u}\ x_\alpha \text{ new}$$

Finally, we list HOL axioms.

1. (FALSE) $\bot = \forall b : \text{prop}.\ b$.
2. (FALSENOTTRUE) $\text{false} = \text{true} \to \bot$.
3. (BETA) $(\lambda x_\tau.\ t_\sigma)s_\tau = t_\sigma[x_\tau := s_\tau]$.
4. (ETA) $(\lambda x_\tau.\ f_{\tau \to \sigma}\ x_\tau) = f_{\tau \to \sigma}$.
5. (FORALL) $\forall_\alpha = \lambda P_{\alpha \to \text{prop}}.\ (P = \lambda x_\alpha.\ \top)$.
6. (P3) $\forall n : \text{nat}.\ (0 = S(n)) \to \bot$.
7. (P4) $\forall n, m : \text{nat}.\ S(n) = S(m) \to n = m$.

8. (P5) $\forall P : \text{nat} \to \text{prop}.\ P(0) \wedge (\forall n : \text{nat}.\ P(n) \to P(S(n))) \to \forall n : \text{nat}.\ P(n)$.

9. (BOOL) $\forall x : \text{bool}.\ (x = \text{false}) \vee (x = \text{true})$.

10. (EM) $\forall x : \text{prop}.\ (x = \bot) \vee (x = \top)$.

11. (CHOICE) $\forall P : \alpha \to \text{prop}.\ \forall x : \alpha.\ P\ x \to P(\varepsilon_{(\alpha \to \text{prop}) \to \alpha}(P))$.

Our choice of rules and axioms is redundant. Propositional connectives, for example, could be defined in terms of quantifiers and bool. However, we believe that this makes the account of the semantics clearer and shows how easy it is to define a sound semantics for such system.

The theory CHOL (Constructive HOL) arises by taking away from HOL axioms (CHOICE) and (EM).

We write $\vdash_H \phi$ and $\vdash_C \phi$ to denote that HOL and CHOL, respectively, proves ϕ. We will generally use letters P, Q to denote proof trees. A notation $P \vdash_C \phi$ means that P is a proof tree in CHOL of ϕ.

5.2 Semantics

In this section, we will define set-theoretic semantics for HOL and CHOL. We start by fixing several definitions and proving several easy lemmas in constructive set theory.

5.2.1 Set theory

The set-theoretic semantics needs a small part of the cumulative hierarchy — $R_{\omega+\omega}$ is sufficient to carry out all the constructions. The Axiom of Choice is necessary in order to define the meaning of the ε constant. For this purpose, C will denote a[1] blatantly non-constructive function such that for any $X, Y \in R_{\omega+\omega}$, if X is non-empty, then $C(X, Y) \in X$, and if X is empty then $C(X, Y) = Y$.

Recall that in the world of set theory, $0 = \emptyset$, $1 = \{0\}$ and $2 = \{0, 1\}$. Classically $P(1)$, the set of all subsets of 1, is equal to 2. This is not the case constructively; there is no uniform way of transforming an arbitrary subset A of 1 into an element of 2.

Lemma 5.2.1 *If $P(1) = 2$, then for any ϕ, ϕ or $\neg\phi$.*

Proof Suppose $P(1) = 2$ and take a formula ϕ. Consider $A = \{\_ \in 1 \mid \phi\}$ and $B = \{\_ \in 1 \mid \neg\phi\}$. Since $A \cup B \in P(1)$, $A \cup B \in 2$, so either $A \cup B = 0$ or $A \cup B = 1$. In the former case, $0 \notin A$ and $0 \notin B$. Now, if ϕ, then $0 \in A$ and if $\neg\phi$, then $0 \in B$, therefore we have both ϕ and $\neg\phi$, which is impossible. This means[2]. that $A \cup B = 1$. Therefore $0 \in A \cup B$, so either $0 \in A$ in which case ϕ, or $0 \in B$ in which case $\neg\phi$. So either ϕ or $\neg\phi$. ■

The following helpful lemma, however, does hold in a constructive world:

Lemma 5.2.2 *If $A \in P(1)$, then $A = 1$ iff $0 \in A$.*

Proof Left-to-right direction is immediate. For the right-to-left direction, we have $A \subseteq 1$ and need to show that $1 \subseteq A$. Suppose $B \in 1$, then $B = 0$, but $0 \in A$, so $B \in A$. ■

Let us also define precisely the function application operation in set theory. We borrow the definition from Aczel [Acz99].

$$App(f, x) = \{z \mid \exists y.\ z \in y \wedge (x, y) \in f\}$$

The advantage of using this definition over an intuitive one ("the unique y such that $(x, y) \in f$") is that it is defined for all sets f and x. Partiality of *App* would

[1] Note that if we want to pinpoint C, we need to assume more than AC, as the existence of a definable choice function for $R_{\omega+\omega}$ is not provable in ZFC.

[2] We are using here a bit uninintuitive intuitionistic tautology : $(((\phi \vee \psi) \wedge \neg\phi) \to \psi)$

entail serious problems in constructive settings — see [Moc07] for description of some of the problems.

This definition is equivalent to the standard one when f is a function:

Lemma 5.2.3 *If f is a function from A to B and $x \in A$, then $App(f, x)$ is the unique y such that $(x, y) \in f$.*

Proof Let y be the unique element of B such that $(x, y) \in f$. If $z \in App(f, x)$ then there is y' such that $z \in y'$ and $(x, y') \in f$. Since $y' = y$, $z \in y$. For the other direction, if $z \in y$, then obviously $z \in App(f, x)$. ■

From now on, the notation $f(x)$ means $App(f, x)$. We will use the lambda notation in set theory to define functions: $\lambda x \in A.\, B(x)$ means $\{(x, B(x)) \mid x \in A\}$.

5.2.2 The definition of the semantics

We first define a meaning $[\![\tau]\!]$ of a type τ by structural induction on τ.

- $[\![\text{nat}]\!] = \mathbb{N}$.
- $[\![\text{bool}]\!] = 2$.
- $[\![\text{prop}]\!] = P(1)$.
- $[\![(\tau, \sigma)]\!] = [\![\tau]\!] \times [\![\sigma]\!]$, where $A \times B$ denotes the cartesian product of sets A and B.
- $[\![\tau_1 \to \tau_2]\!] = [\![\tau_1]\!] \to [\![\tau_2]\!]$, where $A \to B$ denotes the set of all functions from A to B.

The meaning of a constant c_α is denoted by $[\![c_\alpha]\!]$ and is defined as follows.

- $[\![=_\alpha]\!] = \lambda(x_1, x_2) \in ([\![\alpha]\!] \times [\![\alpha]\!]).\ \{x \in 1 \mid x_1 = x_2\}$.
- $[\![\to]\!] = \lambda(b_1, b_2) \in [\![\text{prop}]\!] \times [\![\text{prop}]\!].\ \{x \in 1 \mid x \in b_1 \to x \in b_2\}$.

- $[\![\vee]\!] = \lambda(b_1, b_2) \in [\![\text{prop}]\!] \times [\![\text{prop}]\!].\ b_1 \cup b_2$.
- $[\![\wedge]\!] = \lambda(b_1, b_2) \in [\![\text{prop}]\!] \times [\![\text{prop}]\!].\ b_1 \cap b_2$.
- $[\![\text{false}]\!] = [\![\bot]\!] = 0$.
- $[\![\text{true}]\!] = [\![\top]\!] = 1$.
- $[\![\forall_\alpha]\!] = \lambda f \in [\![\alpha]\!] \to [\![\text{prop}]\!].\ \bigcap_{a \in [\![\alpha]\!]} f(a)$.
- $[\![\exists_\alpha]\!] = \lambda f \in [\![\alpha]\!] \to [\![\text{prop}]\!].\ \bigcup_{a \in [\![\alpha]\!]} f(a)$.
- $[\![\varepsilon_\alpha]\!] = \lambda P \in [\![\alpha]\!] \to [\![\text{prop}]\!].\ C(P^{-1}(1), [\![\alpha]\!])$.
- $[\![0]\!] = 0$.
- $[\![S]\!] = \lambda n \in \mathbb{N}.\ n + 1$.

The standard semantics, presented for example by Gordon and Melham [GM93], uses a truth table approach — an implication $\phi \to \psi$ is false iff ϕ is true and ψ is false etc. It is easy to see that with excluded middle, our semantics is equivalent to the standard one.[3]

To present the rest of the semantics, we need to introduce environments. An *environment* is a partial function from HOL variables to sets such that $\rho(x_\tau) \in [\![\tau]\!]$. We will use the symbol ρ exclusively for environments. The meaning $[\![t]\!]_\rho$ of a term t is parameterized by an environment ρ and defined by structural induction on t:

- $[\![c_\tau]\!]_\rho = [\![c_\tau]\!]$.
- $[\![x_\tau]\!]_\rho = \rho(x_\tau)$.
- $[\![s\ u]\!]_\rho = App([\![s]\!]_\rho, [\![u]\!]_\rho)$.
- $[\![\lambda x_\tau.\ u]\!] = \{(a, [\![u]\!]_{\rho[x_\tau := a]}) \mid a \in [\![\tau]\!]\}$.
- $[\![(s, u)]\!]_\rho = ([\![s]\!]_\rho, [\![u]\!]_\rho)$.

[3]For the interested reader, our definition of the meaning of logical constants is essentially a combination of the fact that any complete lattice with pseudo-complements is a model for higher-order logic and that $P(1)$ is a complete lattice with pseudo-complement defined in the clause for $\Rightarrow$. See [RS63] for more information about these notions.

5.2.3 Properties

There are several standard properties of the semantics we have defined. The following two lemmas are proved by induction on t:

Lemma 5.2.4 (Substitution lemma) *For any terms t, s and environments ρ, $[\![t]\!]_{\rho[x:=[\![s]\!]_\rho]} = [\![t[x := s]]\!]_\rho$.*

Proof By induction on t. Case t of:

- c — the claim is obvious.
- x. Then $[\![x]\!]_{\rho[x:=[\![s]\!]_\rho]} = [\![s]\!]_\rho = [\![[x := s]]\!]_\rho$.
- $u\ v$. Then $[\![u\ v]\!]_{\rho[x:=[\![s]\!]]} = \mathit{App}([\![u]\!]_{\rho[x:=[\![s]\!]_\rho]}, [\![v]\!]_{\rho[x:=[\![s]\!]_\rho]})$. By the induction hypothesis, this is equal to $\mathit{App}([\![u[x := s]]\!]_\rho, [\![v[x := s]]\!]_\rho) = [\![u[x := s]\ v[x := s]]\!]_\rho = [\![t[x := s]]\!]_\rho$.
- (u, v). Similar to the previous case.
- $\lambda y_\tau.\ u$. Without loss of generality we may assume that $y \neq x$. Then $[\![t]\!]_{\rho[x:=s]} = \{(a, [\![u]\!]_{\rho[x:=[\![s]\!]_\rho][y:=a]} \mid a \in [\![\tau]\!]\}$. By the induction hypothesis, this is equal to $\{(a, [\![u[x := s]]\!]_{\rho[y:=a]}) \mid a \in [\![\tau]\!]\} = [\![(\lambda y_\tau.\ u[x := s])]\!]_\rho = [\![t[x := s]]\!]_\rho$. ■

Lemma 5.2.5 *For any ρ, $[\![t_\alpha]\!]_\rho \in [\![\alpha]\!]$.*

Proof We proceed by induction on t. Case t of:

- x. The claim follows by properties of environments.
- c_τ. We proceed by case analysis of c. We show the interesting cases.
  - $\forall_\alpha$. The type of c is $(\alpha \to \mathrm{prop}) \to \mathrm{prop}$. We need to show that if f is a function from $[\![\alpha]\!]$ to $P(1)$, then $\bigcap_{a\in[\![\alpha]\!]} f(a)$ is in $P(1)$. But for any $a \in [\![\alpha]\!]$, $f(a) \in P(1)$, and $P(1)$ is closed under intersections.

– $\exists_\alpha$. Similar, follows by the fact that $P(1)$ is closed under unions.

– ε_α. The type of c is $(\alpha \to \text{prop}) \to \text{prop}$. Take any function P from $[\![\alpha]\!]$ to $P(1)$. Then $P^{-1}(1) \subseteq [\![\alpha]\!]$, so the claim follows by the definition of C. ■

In particular, this implies that for any formula t, $[\![t]\!] \subseteq 1$. So if we want to prove that $[\![t]\!] = 1$, then by Lemma 5.2.2 it suffices to show that $0 \in [\![t]\!]$.

5.2.4 Soundness

The soundness theorem establishes validity of the proof rules and axioms with respect to the semantics.

Definition 5.2.6 *$\rho \models \Gamma \vdash t$ means that ρ is defined for $x_\tau \in FV(\Gamma) \cup FV(t)$.*

By the definition of environments, if $\rho \models \Gamma \vdash \bar{t}$, then for all $x_\tau \in FV(\Gamma) \cup FV(t)$, $\rho(x_\tau) \in [\![\tau]\!]$.

Definition 5.2.7 *We write $[\![\Gamma]\!]_\rho = 1$ if $[\![t_1]\!]_\rho = 1, \ldots, [\![t_n]\!]_\rho = 1$, where $\Gamma = t_1, t_2, \ldots, t_n$.*

Theorem 5.2.8 (Soundness) *If $\Gamma \vdash t$, then for all $\rho \models \Gamma \vdash t$, if $[\![\Gamma]\!]_\rho = 1$, then $[\![t]\!] = 1$.*

Proof Straightforward induction on $\Gamma \vdash t$. We show some interesting cases. Case $\Gamma \vdash t$ of:

- $$\frac{}{\Gamma \vdash t}\ t \in \Gamma$$

  The claim is trivial.

- $$\frac{\Gamma \vdash t = s}{\Gamma \vdash \lambda x_\tau.\ t = \lambda x_\tau.\ s}$$

  Take any $\rho \models \Gamma \vdash \lambda x_\tau.\ t = \lambda x_\tau.\ s$. We need to show that $\{(a, [\![t]\!]_{\rho[x_\tau:=a]}) \mid a \in [\![\tau]\!]\} = \{(a, [\![s]\!]_{\rho[x_\tau:=a]}) \mid a \in [\![\tau]\!]\}$. That is, that for any $a \in [\![\tau]\!]$, $[\![t]\!]_{\rho[x_\tau:=a]} = [\![s]\!]_{\rho[x_\tau:=a]}$. Let $\rho' = \rho[x_\tau := a]$. Since $\rho' \models \Gamma \vdash t = s$, by the induction hypothesis we get the claim.

- $$\frac{\Gamma \vdash t \quad \Gamma \vdash s}{\Gamma \vdash t \wedge s}$$

  Suppose $[\![\Gamma]\!]_\rho = 1$. By the induction hypothesis, $0 \in [\![t]\!]_\rho$ and $0 \in [\![s]\!]_\rho$, so $0 \in [\![t]\!]_\rho \cap [\![s]\!]_\rho$.

- $$\frac{\Gamma \vdash t \wedge s}{\Gamma \vdash t} \qquad \frac{\Gamma \vdash t \wedge s}{\Gamma \vdash s}$$

  Reverse the previous case to get the claims.

- $$\frac{\Gamma \vdash t}{\Gamma \vdash t \vee s} \qquad \frac{\Gamma \vdash s}{\Gamma \vdash t \vee s} \qquad \frac{\Gamma \vdash t \vee s \quad \Gamma, t \vdash u \quad \Gamma, s \vdash u}{\Gamma \vdash u}$$

  The first two cases are easy. For the last one, suppose $[\![\Gamma]\!]_\rho = 1$. By the induction hypothesis, we know that $0 \in [\![t]\!]_\rho \cup [\![s]\!]_\rho$, so either $0 \in [\![t]\!]_\rho$ or $0 \in [\![s]\!]_\rho$. In both cases, by the rest of the induction hypothesis, $0 \in [\![u]\!]_\rho$, so we get the claim.

- $$\frac{\Gamma, t \vdash s}{\Gamma \vdash t \Rightarrow s}$$

  Suppose $[\![\Gamma]\!]_\rho = 1$. We need to show that $0 \in \{x \in 1 \mid x \in [\![t]\!]_\rho \to x \in [\![s]\!]_\rho\}$. Since $0 \in 1$, assume $0 \in [\![t]\!]_\rho$. Then $[\![\Gamma, t]\!]_\rho = 1$, so by the induction hypothesis $[\![s]\!]_\rho = 1$ and $0 \in [\![s]\!]_\rho$.

- $$\frac{\Gamma \vdash t \to s \quad \Gamma \vdash t}{\Gamma \vdash s}$$

  Suppose $[\![\Gamma]\!]_\rho = 1$. By the induction hypothesis, $0 \in \{x \in 1 \mid x \in [\![t]\!]_\rho \to x \in [\![s]\!]_\rho\}$ and $0 \in [\![t]\!]_\rho$, so easily $0 \in [\![s]\!]_\rho$.

- $$\frac{\Gamma \vdash s = u \quad \Gamma \vdash t[x := u]}{\Gamma \vdash t[x := s]}$$

  The proof is straightforward, using the Substitution Lemma. Assume $[\![\Gamma]\!]_\rho = 1$. By the induction hypothesis, $[\![s]\!]_\rho = [\![u]\!]_\rho$ and $[\![t[x := u]]\!]_\rho = 1$. So using the Substitution Lemma we get $[\![t[x := u]]\!]_\rho = [\![t]\!]_{\rho[x:=[\![u]\!]_\rho]} = [\![t]\!]_{\rho[x:=[\![s]\!]_\rho]} = [\![t[x := s]]\!]_\rho$.

- $$\frac{\Gamma \vdash f\ t_\alpha}{\Gamma \vdash \exists_\alpha(f_{\alpha\to\mathrm{prop}})}$$

  Assume $[\![\Gamma]\!]_\rho = 1$. We have to show that $0 \in \bigcup_{a\in[\![\alpha]\!]}[\![f]\!]_\rho(a)$, that is that there is $a \in [\![\alpha]\!]$ such that $0 \in f(a)$. By Lemma 5.2.5, $[\![t_\alpha]\!]_\rho \in [\![\alpha]\!]$, so taking $a = [\![t_\alpha]\!]_\rho$ we get the claim by the induction hypothesis.

- $$\frac{\Gamma \vdash \exists_\alpha(f_{\alpha\to\mathrm{prop}}) \quad \Gamma, f\ x_\alpha \vdash u}{\Gamma \vdash u}\ x_\alpha \text{ new}$$

  Suppose $[\![\Gamma]\!]_\rho = 1$. By the induction hypothesis, there is $a \in [\![\alpha]\!]$ such that $0 \in [\![f]\!]_\rho(a)$. Let $\rho' = \rho[x_\alpha := a]$. Then $\rho' \models \Gamma, f\ x_\alpha \vdash u$, so by the induction hypothesis we get $0 \in [\![u]\!]_\rho$, which is what we want.

- $$\frac{}{\Gamma \vdash \forall_\alpha = (\lambda P_{\alpha\to\mathrm{prop}}.\ P = \lambda x_\alpha.\ \top)}$$

  We need to show that $[\![\forall_\alpha]\!]_\rho = [\![\lambda P_{\alpha\to\mathrm{prop}}.\ P = \lambda x_\alpha.\ \top]\!]$. First, $[\![\forall_\alpha]\!]_\rho = \lambda f \in [\![\alpha]\!] \to P(1).\ \bigcap_{a\in[\![\alpha]\!]} f(a)$. Let us denote this function in set theory by F. The domain of F is $[\![\alpha]\!] \to P(1)$.

Second, let $G = [\![(\lambda P_{\alpha\to\mathrm{prop}}.\ P = \lambda x_\alpha.\ \top)]\!]_\rho$. Then we have $G = \{(a, [\![a = \lambda x_\alpha.\ \top]\!]_{\rho[P:=a]}) \mid P \in [\![\alpha]\!] \to P(1)\}$. Note that G is also a function with the domain $[\![\alpha]\!] \to P(1)$. Thus, to show that $F = G$, it suffices to show that for any $P \in [\![\alpha]\!] \to P(1)$, $F(P) = G(P)$. Take any such P. We have $F(P) = \bigcap_{a\in[\![\alpha]\!]} P(a)$ and $G(P) = \{x \in 1 \mid P = \lambda x \in [\![\alpha]\!].\ 1\}$. Now $b \in F(P)$ iff for all $a \in [\![\alpha]\!]$, $b \in P(a)$ iff for all $a \in [\![\alpha]\!]$, $P(a) = 1$ iff $P = \lambda a \in [\![\alpha]\!].\ 1$, iff $b \in \{z \in 1 \mid P = \lambda a \in [\![\alpha]\!].\ 1\}$. ■

Having verified the soundness of the HOL proof rules, we proceed to verify the soundness of the axioms.

Theorem 5.2.9 *For any axiom t of HOL and any environment ρ, $0 \in [\![t]\!]_\rho$.*

Proof We proceed axiom by axiom and sketch the respective proofs.

- (FALSE) $[\![\bot]\!]_\rho = \emptyset = \bigcap_{a\in P(1)} a = [\![\forall b : \mathrm{prop}.\ b]\!]_\rho$. The second equality follows by $0 \in P(1)$.
- (BETA) Follows by the Substitution Lemma. We have $[\![(\lambda x_\tau.\ t_\sigma)s_\tau]\!]_\rho = App([\![\lambda x_\tau.\ t_\sigma]\!]_\rho, [\![s_\tau]\!]_\rho) = App(\{(a, [\![t]\!]_{\rho[x:=a]}) \mid a \in [\![\tau]\!]\}, [\![s_\tau]\!]_\rho) = [\![t]\!]_{\rho[x:=[\![s_\tau]\!]_\rho]} =$ (by the Substitution Lemma) $= [\![t[x := s]]\!]_\rho$.
- (ETA) Follows by the fact that functions in set theory are represented by their graphs. We have:

$$[\![(\lambda x_\tau.\ f_{\tau\to\sigma} x_\tau]\!]_\rho = \{(a, [\![f\ x]\!]_{\rho[x:=a]}) \mid a \in [\![\tau]\!]\} =$$
$$\{(a, App([\![f]\!]_{\rho[x:=a]}, a)) \mid a \in [\![\tau]\!]\} = (\text{since } x \notin FV(f))$$
$$\{(a, [\![f]\!]_\rho(a)) \mid a \in [\![\tau]\!]\} = [\![f]\!]_\rho,$$

since by Lemma 5.2.5, $[\![f]\!]_\rho \in [\![\tau]\!] \to [\![\sigma]\!]$ and functions in set theory are represented by their graphs.

- (FORALL) We have:

$$[\![\forall_\alpha]\!]_\rho = \{(Q, \bigcap_{a \in [\![\alpha]\!]} Q(a)) \mid Q \in [\![\alpha]\!] \to P(1)\}$$

Also:

$$[\![(\lambda Q_{\alpha\to\mathrm{prop}}.\, Q = \lambda x_\alpha.\, \top)]\!]_\rho = \{(Q, \{x \in 1 \mid Q = \lambda x \in [\![\alpha]\!].\, 1\}) \mid Q \in [\![\alpha]\!] \to P(1)\}$$

Take any $Q \in [\![\alpha]\!] \to P(1)$. It suffices to show that $\bigcap_{a\in[\![\alpha]\!]} Q(a) = \{x \in 1 \mid Q = \lambda y \in [\![\alpha]\!].\, 1\}$. But $x \in \bigcap_{a\in[\![\alpha]\!]} Q(a)$ iff for all $a \in [\![\alpha]\!]$, $x \in Q(a)$ and $x = 0$. This happens if and only if $x = 0$ and for all $a \in [\![\alpha]\!]$, $Q(a) = 1$ which is equivalent to $x \in \{x \in 1 \mid Q = \lambda y \in [\![\alpha]\!].\, 1\}$. The sets in question are therefore equal.

- The axioms $P3, P4, P5$ follow by the fact that natural numbers satisfy the respective Peano axioms.

- (BOOL) We need to show that $[\![\forall_{\mathrm{bool}}.\, (\lambda x_{\mathrm{bool}}.\, x = \mathrm{false} \lor x = \mathrm{true})]\!]_\rho = 1$. Unwinding the definition, this is equivalent to $\bigcap_{x\in 2}(\{z \in 1 \mid x = 0\} \cup \{z \in 1 \mid x = 1\}) = 1$. and furthermore to: for all $x \in 2$, $x \in \{z \in 1 \mid x = 0\} \cup \{z \in 1 \mid x = 1\}$. If $x \in 2$, then either $x = 0$ or $x = 1$. In the former case, $0 \in \{z \in 1 \mid x = 0\}$, in the latter $0 \in \{z \in 1 \mid x = 1\}$.

- (EM) We need to show that $[\![\forall_{\mathrm{prop}}.\, (\lambda x_{\mathrm{prop}}.\, x = \bot \lor x = \top)]\!]_\rho = 1$. Reasoning as in the case of (BOOL), we find that this is equivalent to: for all $x \in P(1)$, $x \in \{z \in 1 \mid x = 0\} \cup \{z \in 1 \mid x = 1\}$. Suppose $x \in P(1)$. At this point, it is impossible to proceed further constructively, all we know is that x is a subset of 1, which does not provide enough information to decide whether $x = 0$ or $x = 1$. However, classically, using the rule of excluded middle, $P(1) = 2$ and we proceed as in the previous case.

- (CHOICE) We need to show that:

$$[\![\forall_{\alpha\to\mathrm{prop}}(\lambda P_{\alpha\to\mathrm{prop}}.\, \forall_\alpha(\lambda x_\alpha.\, Px \Rightarrow P(\varepsilon_{(\alpha\to\mathrm{prop})\to\alpha}(P))]\!] = 1$$

We have the following chain of equivalences:

$$
\begin{aligned}
&[\![\forall_{\alpha\to\mathrm{prop}}(\lambda P_{\alpha\to\mathrm{prop}}.\ \forall_\alpha(\lambda x_\alpha.\ Px \Rightarrow P(\varepsilon_{(\alpha\to\mathrm{prop})\to\alpha}(P))]\!] = 1 \leftrightarrow \\
&\bigcap_{P\in[\![\alpha]\!]\to 2} [\![\forall_\alpha(\lambda x_\alpha.\ Px \Rightarrow P(\varepsilon_{(\alpha\to\mathrm{prop})\to\alpha}(P))]\!] = 1 \leftrightarrow \\
&\bigcap_{P\in[\![\alpha]\!]\to 2}\bigcap_{x\in[\![\alpha]\!]} [\![Px \Rightarrow P(\varepsilon_{(\alpha\to\mathrm{prop})\to\alpha}(P))]\!] = 1 \leftrightarrow \\
&\bigcap_{P\in[\![\alpha]\!]\to 2}\bigcap_{x\in[\![\alpha]\!]} [\![Px \Rightarrow P(\varepsilon_{(\alpha\to\mathrm{prop})\to\alpha}(P))]\!] = 1 \leftrightarrow \\
&\bigcap_{P\in[\![\alpha]\!]\to 2}\bigcap_{x\in[\![\alpha]\!]} \{a \in 1 \mid a \in P(x) \to a \in P(C(P^{-1}(\{1\}), [\![\alpha]\!]))\} = 1
\end{aligned}
$$

To show this, it suffices to show that for all $P \in [\![\alpha]\!] \to 2$, for all $x \in [\![\alpha]\!]$, if $0 \in P(x)$ then $0 \in P(C(P^{-1}(\{1\}), [\![\alpha]\!]))$. Take any P and x. Suppose $0 \in P(x)$. Then $P(x) = 1$, so $x \in P^{-1}(\{1\})$. Therefore $C(P^{-1}(\{1\}), [\![\alpha]\!])) \in P^{-1}(\{1\})$, so $P(C(P^{-1}(\{1\}), [\![\alpha]\!]) = 1$, which shows the claim. ■

Corollary 5.2.10 *HOL is consistent: it is not the case that* $\vdash_H \bot$.

Proof Otherwise we would have $[\![\bot]\!] = [\![\top]\!]$, that is $0 = 1$. ■

5.3 Extraction

We will show that the semantics we have defined can serve as a basis for program extraction for proofs. All that is necessary for program extraction from constructive HOL proofs is provided by the semantics and the soundness proof. Therefore, if one wants to provide an extraction mechanism for the constructive part of the logic, it may be sufficient to carefully define set-theoretic semantics, prove the soundness theorem and the extraction mechanism for IZF_R would take care of the rest. We speculate on practical uses of this approach in section 5.4.

As in case of IZF_R , we will show how to do extraction from a subclass of CHOL proofs. The choice of the subclass is largely arbitrary, our choice illustrates the method and can be easily extended.

We say that a CHOL formula is *extractable* if it is generated by the following abstract grammar, where τ varies over pure TT^0 types and $\oplus \in \{\wedge, \vee, \to\}$.

$$\phi ::= \forall x : \tau.\ \phi \mid \exists x : \tau.\ \phi \mid \phi \oplus \phi \mid \bot \mid t = t,$$

We will define extraction for CHOL proofs of extractable formulas. By Theorem 5.2.9, if CHOL $\vdash \phi$, then IZF$_R$$\vdash 0 \in [\![\phi]\!]$. We need to slightly transform this IZF$_R$ proof in order to come up with a valid input to the extraction function E from Section 3.3. To this means, for any extractable ϕ (with possibly free variables) we define a formula ϕ' such that IZF$_R$$\vdash 0 \in [\![\phi]\!] \leftrightarrow \phi'$. The formula ϕ' is essentially ϕ with the type membership information replaced by the set membership information. We define ϕ' by induction on ϕ. The correctness follows trivially in each case. In all the cases we work in IZF$_R$. Case ϕ of:

- $\bot$. Then $\phi' = 0 \in [\![\bot]\!]$.
- $t = s$. Then $\phi' = 0 \in [\![t = s]\!]$.
- $\phi_1 \vee \phi_2$. $0 \in [\![\phi_1 \vee \phi_2]\!]$ iff $0 \in [\![\phi_1]\!]$ or $0 \in [\![\phi_2]\!]$. By the induction hypothesis we get ϕ_1' and ϕ_2' such that $0 \in [\![\phi_1]\!] \leftrightarrow \phi_1'$ and $0 \in [\![\phi_2]\!] \leftrightarrow \phi_2'$. Take $\phi' = \phi_1' \vee \phi_2'$.
- $\phi_1 \wedge \phi_2$. Then $0 \in [\![\phi]\!]$ iff $0 \in [\![\phi_1]\!]$ and $0 \in [\![\phi_2]\!]$. Take ϕ_1' and ϕ_2' from the induction hypothesis and set $\phi' = \phi_1' \wedge \phi_2'$.
- $\phi_1 \to \phi_2$. Then $0 \in [\![\phi_1 \to \phi_2]\!]$ iff $0 \in \{x \in 1 \mid x \in [\![\phi_1]\!] \to x \in [\![\phi_2]\!]\}$ iff $0 \in [\![\phi_2]\!] \to 0 \in [\![\phi_2]\!]$. By the induction hypothesis get ϕ_1' such that $0 \in [\![\phi_1]\!] \leftrightarrow \phi_1'$ and ϕ_2' such that $0 \in [\![\phi_2]\!] \leftrightarrow \phi_2'$. Set $\phi' = \phi_1' \to \phi_2'$.
- $\forall a : \tau.\ \phi_1$. Then $0 \in [\![\phi]\!]$ iff for all $A \in [\![\tau]\!]$, $0 \in App([\![\lambda a : \tau.\ \phi_1]\!], A)$ iff for all $A \in [\![\tau]\!]$, $0 \in App(\{(x, [\![\phi_1]\!]_{\rho[a:=x]}) \mid x \in [\![\tau]\!]\}, A)$ iff for all $A \in [\![\tau]\!]$ $0 \in [\![\phi_1]\!]_{\rho[a:=A]}$ iff, by the Substitution Lemma, for all $A \in [\![\tau]\!]$, $0 \in [\![\phi_1[a := A]]\!]$

iff for all $A \in [\![\tau]\!]$, $0 \in [\![\phi_1]\!]$. Take ϕ_1' from the induction hypothesis and set $\phi' = \forall a \in [\![\tau]\!].\ 0 \in \phi_1'$.

- $\exists a : \tau.\ \phi_1$. Then $0 \in [\![\phi]\!]$ iff $A \in [\![\tau]\!]$ iff $0 \in [\![\phi_1[a := A]]\!]$. Just as in the previous case, get ϕ_1' from the induction hypothesis and set $\phi' = \exists a \in [\![\tau]\!].\ \phi_1'$.

Now we can finally define the extraction process. Suppose CHOL $\vdash \phi$, where ϕ is extractable. Using the soundness theorem, construct an IZF_R proof P that $0 \in [\![\phi]\!]$. Use the definition above to get ϕ' such that $\mathrm{IZF}_R \vdash 0 \in [\![\phi]\!] \leftrightarrow \phi'$ and using P obtain a proof R of ϕ'. Finally, apply the extraction function E to R to get the computational extract.

5.4 Conclusion

We have presented a computational semantics for HOL via the standard interpretation in intuitionistic set theory. The semantics is clean, simple and agrees with the standard one.

The advantage of this approach is that the extraction mechanism is completely external to Constructive HOL. Using only the semantics, we can take any constructive HOL proof and extract from it computational information. No enrichment of the logic in the normalizing proof terms is necessary.

The separation of the extraction mechanism from the logic makes the logic very easily extendable. For example, inductive datatypes and subtyping have clean set-theoretic semantics, so can easily be added to HOL while preserving consistency, as witnessed in PVS. As the semantics would work constructively, the extraction mechanisms from section 5.3 could be easily adapted to incorporate them. Similarly, one could define a set-theoretic semantics for the constructive version of HOL implemented in Isabelle ([Ber04, BN02]) in the same spirit, with the same advantages.

The modularity of our approach and the fact that it is much easier to give set-theoretic semantics for the logic than to prove normalization, could make the development of new trustworthy provers with extraction capabilities much easier and faster.

CHAPTER 6

CONCLUSION

We took the reader on a tour between the static world of sets, which forms the foundation of mathematics, and the dynamic world of types, a possible foundation for computer science. We showed that these worlds are much closer than it seems. Computation can be discovered in the world of sets, and type-theoretic features can be added to set theory with significant advantages. We showed the first applications of our results, by showing that standard semantics for type theories given in computational set theory can provide program-extraction capability from constructive proofs for free.

We would like to close this thesis with three open questions. We believe that answers to these questions will further our understanding of foundations of mathematics and computer science.

- *Is there a strong consistent dependent set theory with unrestricted Σ-types?* In a sense, Aczel's interpretation of CZF in Martin-Löf's type theory [Acz78, Acz82, Acz86] is such a theory. However, it is very weak. Moreover, its dependent nature is only revealed in the type-theoretic model, not in the axiomatization. We hope for a consistency result for a theory constructed along the lines of our IZF_D. One instance of this question is whether IZF_D with standard, non-dependent Replacement and unrestricted Σ-types is consistent.

- *Is there a well-behaved lambda calculus with types which can interpret IZF_C?*

  Although λD does provide an interpretation, the fact that it does not possess the Subject Reduction property makes it an unsatisfying theory from a theoretical point of view. The strongest normalizing lambda calculi in existence with the Subject Reduction property, such as CIC and ECC, can only interpret constructive Zermelo theory [Wer97, Acz99] and their proof-theoretic strength is smaller than ZF.

- *Is there a good characterization of the border area between normalization, lack of thereof and inconsistency?*

  On the side of sets, contradictions are nowadays not that easy to find, apart from Russell's paradox. The land of types is much younger and not as well understood. However, although the original type theory of Martin-Löf turned out to be inconsistent, modern type theories are widely believed to be consistent. Our theories live in the border area — almost contradictory (λD and IZF_D) and almost not normalizing (with the addition of non-well-founded sets). But what exactly tips a type or set theory into losing normalization or consistency?

We are looking forward to seeing answers to these problems.

THE END

BIBLIOGRAPHY

[ABC+06] Stuart Allen, Mark Bickford, Robert Constable, Richard Eaton, Christoph Kreitz, Lori Lorigo, and Evan Moran. Innovations in computational type theory using Nuprl. *Journal of Applied Logic*, 4(4):428–469, 2006.

[Abr96] J.-R. Abrial. *The B-book: assigning programs to meanings*. Cambridge University Press, New York, NY, USA, 1996.

[ACE+00] Stuart Allen, Robert Constable, Richard Eaton, Christoph Kreitz, and Lori Lorigo. The Nuprl open logical environment. In David McAllester, editor, *Proceedings of the 17th International Conference on Automated Deduction*, volume 1831 of *Lecture Notes in Artificial Intelligence*, pages 170–176. Springer Verlag, 2000.

[ACN90] Lennart Augustsson, Thierry Coquand, and Bengt Nordström. A short description of another logical framework. In *Proceedings of the First Annual Workshop on Logical Frameworks*, pages 39–42, Sophia-Antipolis, France, 1990.

[Acz78] Peter Aczel. The type theoretic interpretation of constructive set theory. In A. MacIntyre, L. Pacholski, and J. Paris, editors, *Logic Colloquium '77*, pages 55–66. North Holland, 1978.

[Acz82] Peter Aczel. The type theoretic interpretation of constructive set theory: Choice principles. In S.S. Troelstra and D. van Dalen, editors, *The L.E.J. Brouwer Centenary Symposium*, pages 1–40. North Holland, 1982.

[Acz86] Peter Aczel. The type theoretic interpretation of constructive set theory: Inductive definitions. In *Logic, Methodology and Philosophy of Science VII*, pages 17–49. Elsevier Science Publishers, 1986.

[Acz99] Peter Aczel. On relating type theories and set theories. In T. Altenkirch, W. Naraschewski, and B. Reus, editors, *Types for Proofs and Programs: International Workshop, TYPES '98, Kloster Irsee, Germany, March 1998*, volume 1657 of *LNCS*, pages 1–18, 1999.

[AG95] Sten Agerholm and Michael J. C. Gordon. Experiments with ZF Set Theory in HOL and Isabelle. In *Proc. of the 8th Int. Workshop on Higher Order Logic Theorem Proving and Its Applications*, pages 32–45, London, UK, 1995. Springer-Verlag.

[AR01] Peter Aczel and Michael Rathjen. Notes on constructive set theory. Technical Report 40, Institut Mittag-Leffler (The Royal Swedish Academy of Sciences), 2000/2001.

[AZ97] Z. Adamowicz and P. Zbierski. *Logic of Mathematics: a Modern Course of Classical Logic*. John Wiley & Sons, Inc., 1997.

[Bai88] Sidney C. Bailin. A normalization theorem for set theory. *J. Symb. Log.*, 53(3):673–695, 1988.

[Bar92] Henk P. Barendregt. *Handbook of Logic in Computer Science*, volume 2, chapter Lambda Calculi with Types, pages 118–310. Oxford University Press, 1992.

[BBS+98] H. Benl, U. Berger, H. Schwichtenberg, et al. Proof theory at work: Program development in the Minlog system. In W. Bibel and P. G. Schmitt, editors, *Automated Deduction*, volume II, pages 41–71. Kluwer, 1998.

[BC04] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development; Coq'Art: The Calculus of Inductive Constructions*. Texts in Theoretical Computer Science. Springer-Verlag, 2004.

[Bee85] Michael J. Beeson. *Foundations of Constructive Mathematics*. Springer-Verlag, 1985.

[Ber04] Stefan Berghofer. *Proofs, Programs and Executable Specifications in Higher Order Logic*. PhD thesis, Technische Universität München, 2004.

[BN02] Stefan Berghofer and Tobias Nipkow. Executing Higher Order Logic. In P. Callaghan, Z. Luo, J McKinna, and R. Pollack, editors, *Types for Proofs and Programs: TYPES'2000*, volume 2277 of *LNCS*, pages 24–40. Springer-Verlag, 2002.

[Bou49] N. Bourbaki. Foundations of mathematics for the working mathematician. *J. Symb. Log.*, 14(1):1–8, 1949.

[Bou68a] N. Bourbaki. *Elements of Mathematics, Algebra*, volume 1. Addison-Wesley, Reading, MA, 1968.

[Bou68b] N. Bourbaki. *Elements of Mathematics, Theory of Sets*. Addison-Wesley, Reading, MA, 1968.

[Bro07] L.E.J. Brouwer. *Over de Grondslagen der Wiskunde*. PhD thesis, 1907. English version in [Hey75].

[CAB+86] Robert L. Constable, Stuart F. Allen, H. M. Bromley, W. R. Cleaveland, J. F. Cremer, R. W. Harper, Douglas J. Howe, T. B. Knoblock, N. P. Mendler, P. Panangaden, James T. Sasaki, and Scott F. Smith. *Implementing Mathematics with the Nuprl Proof Development System*. Prentice-Hall, NJ, 1986.

[Can55] Georg Cantor. *Contributions to the Founding of the Theory of Transfinite Numbers*. Dover, 1955.

[CFC58] H. B. Curry, R. Feys, and W. Craig. *Combinatory Logic, Volume I*. Studies in Logic and the Foundations of Mathematics. North-Holland, Amsterdam, 1958.

[Chu40] Alonzo Church. A formulation of the simple theory of types. *The Journal of Symbolic Logic*, 5:55–68, 1940.

[CM06] Robert Constable and Wojciech Moczydłowski. Extracting Programs from Constructive HOL Proofs via IZF Set-Theoretic Semantics. In *Proc. 3rd Int. Joint Conf. on Automated Reasoning (IJCAR 2006)*, volume 4130 of *LNCS*, pages 162–176. Springer, 2006.

[Con71] Robert L. Constable. Constructive mathematics and automatic program writers. In *Proceedings of the IFIP Congress*, pages 229–233. North-Holland, 1971.

[Con98] Robert L. Constable. Types in logic, mathematics and programming. In S. R. Buss, editor, *Handbook of Proof Theory*, chapter X, pages 683–786. Elsevier Science B.V., 1998.

[Coq] Catarina Coquand. Agda. Available from http://www.cs.chalmers.se/˜catarina/agda.

[CPM90] Thierry Coquand and Christine Paulin-Mohring. Inductively defined types, preliminary version. In *COLOG '88, International Conference on Computer Logic*, volume 417 of *LNCS*, pages 50–66. Springer, Berlin, 1990.

[CR02] L. Crosilla and Michael Rathjen. Inaccessible set axioms may have little consistency strength. *Ann. Pure Appl. Logic*, 115(1-3):33–70, 2002.

[Cra] Marcel Crabbé. Non-normalisation de la théorie de Zermelo. Available from http://www.lofs.ucl.ac.be/log/perso/Crabbe/textes/contreexemple.pdf.

[dB70] N. G. de Bruijn. The mathematical language Automath: its usage and some of its extensions. In J. P. Seldin and J. R. Hindley, editors, *Symposium on Automatic Demonstration*, volume 125 of *Lecture Notes in Mathematics*, pages 29–61. Springer-Verlag, 1970.

[DM06] Gilles Dowek and Alexandre Miquel. Cut elimination for Zermelo's set theory. 2006. Manuscript, available from the web pages of the authors.

[Fef05] Solomon Feferman. Predicativity. In Stewart Shapiro, editor, *The Oxford Handbook of the Philosophy of Mathematics and Logic*, pages 590–624. Oxford University Press, 2005.

[Flo67] R. W. Floyd. Assigning meanings to programs. In *Proceedings AMS Symposium Appl. Math.*, pages 19–31, Providence, RI, 1967.

[Fra22] A.A. Fraenkel. Zu den Grundlagen der Cantor-Zermeloschen mengenlehre. *Mathematische Annalen*, 86:230–237, 1922.

[Fre67] Gottlob Frege. Begriffsschrift, a formula language, modeled upon that for arithmetic for pure thought. In van Heijenoort [vH67], pages 1–82.

[Fri73] Harvey Friedman. The consistency of classical set theory relative to a set theory with intuitionistic logic. *The Journal of Symbolic Logic*, pages 315–319, 1973.

[Fri78] Harvey Friedman. Classically and intuitionistically provably recursive functions. In D. S. Scott and G. H. Muller, editors, *Higher Set Theory*, volume 699 of *Lecture Notes in Mathematics*, pages 21–28. Springer-Verlag, 1978.

[Fv84] Harvey Friedman and Andre Ščedrov. Large sets in intuitionistic set theory. *Annals of Pure and Applied Logic*, 27:1–24, 1984.

[Fv85] Harvey Friedman and Andre Ščedrov. The Lack of Definable Witnesses and Provably Recursive Functions in Intuitionistic Set Theories. *Advances in Mathematics*, 57:1–13, 1985.

[Gen69] Gerhard Gentzen. Investigations into logical deduction (1934). In

M. Szalo, editor, *The Collected Paers of Gerhard Gentzen*. North-Holland, Amsterdam, 1969.

[Gir72] Jean-Yves Girard. *Interprétation Fonctionelle et Élimination des Compures de l'Arithmétic d'Ordre Supérieur*. PhD thesis, Université Paris VII, 1972.

[GM93] Michael Gordon and Tom Melham. *Introduction to HOL: A Theorem Proving Environment for Higher-Order Logic*. Cambridge University Press, Cambridge, 1993.

[Göd31] Kurt Gödel. Über formal unentscheidbare sätze der principia mathematica und verwandter systeme I. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931. English version in [vH67].

[Göd58] Kurt Gödel. Über eine bisher noch nicht benüntze Erweiterung des finiten Standpunktes. *Dialectica*, 12(3-4):280–287, 1958.

[Göd65] Kurt Gödel. On intuitionistic arithmetic and number theory. In M. Davis, editor, *The Undecidable*, pages 75–81. Raven Press, 1965.

[Gor96] Mike Gordon. Set Theory, Higher Order Logic or Both? In *TPHOLs '96: Proc. of the 9th Int. Conf. on Theorem Proving in Higher Order Logics*, volume 1125 of *LNCS*, pages 191–202. Springer-Verlag, 1996.

[Gru92] Klaus Grue. Map theory. *Theor. Comput. Sci.*, 102(1):1–133, 1992.

[HA28] David Hilbert and Wilhelm Ackermann. *Grundzüge der theoretischen Logik*. Springer-Verlag, 1928.

[Hal] Thomas Hallgren. Alfa. Available from http://www.cs.chalmers.se/˜hallgren/Alfa.

[Har96] John Harrison. HOLLight: A tutorial introduction. In *Formal Methods in Computer-Aided Design (FMCAD'96)*, volume 1166 of *LNCS*, pages 265–269. Springer, 1996.

[Hey31] Arend Heyting. Die intuitionistische grundlegung der mathematik. *Erkenntnis*, 2:106–115, 1931.

[Hey66] A. Heyting. *Intuitionism, An Introduction*. North-Holland, Amsterdam, 1966.

[Hey75] A. Heyting, editor. *L. E. J. Brouwer. Collected Works*, volume 1. North-Holland, Amsterdam, 1975. (see On the foundations of mathematics 11-98.).

[HNC+03] Jason Hickey, Aleksey Nogin, Robert L. Constable, Brian E. Aydemir, Eli Barzilay, Yegor Bryukhov, Richard Eaton, Adam Granicz, Alexei Kopylov, Christoph Kreitz, Vladimir N. Krupski, Lori Lorigo, Stephan Schmitt, Carl Witty, and Xin Yu. MetaPRL — A modular logical environment. In David Basin and Burkhart Wolff, editors, *Proceedings of the 16th International Conference on Theorem Proving in Higher Order Logics (TPHOLs 2003)*, volume 2758 of *LNCS*, pages 287–303. Springer-Verlag, 2003.

[Hoa69] C. A. R. Hoare. An axiom basis for computer programming. *Communications of the ACM*, 12(10):576–580,583, 1969.

[How80] W. Howard. The formulas-as-types notion of construction. In *To H.B. Curry: Essays on Combinatory Logic, Lambda-Calculus and Formalism*, pages 479–490. Academic Press, NY, 1980.

[How96] Douglas J. Howe. Semantic foundations for embedding HOL in Nuprl. In Martin Wirsing and Maurice Nivat, editors, *Algebraic Methodology and Software Technology*, volume 1101 of *LNCS*, pages 85–101. Springer-Verlag, Berlin, 1996.

[Jec03] Thomas Jech. *Set Theory*. Springer Monographs in Mathematics. Springer-Verlag, 3rd revised edition, 2003.

[Kan03] Akihiro Kanamori. *The Higher Infinite: Large Cardinals in Set Theory from Their Beginnings*. Springer-Verlag, 2nd edition, 2003.

[Kle45] S. C. Kleene. On the interpretation of intuitionistic number theory. *The Journal of Symbolic Logic*, 10(4):109–124, dec 1945.

[Kre02] Christoph Kreitz. *The Nuprl Proof Development System, Version 5: Reference Manual and User's Guide*. Department of Computer Science, Cornell University, December 2002.

[Kun80] Kenneth Kunen. *Set theory: an introduction to independence proofs*. Elsevier, 1980.

[Lau70] H. Lauchli. An abstract notion of realizability for which intuitionistic predicate calculus is complete. In *Intuitionism and Proof Theory*, pages 227–34. North-Holland, Amsterdam, 1970.

[Lei94] Daniel Leivant. Higher order logic. In D. M. Gabbay, C. J. Hogger, and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 2: Deduction Methodologies*, pages 229–321. Clarendon Press, Oxford, 1994.

[Lip95] Jim Lipton. Realizability, set theory and term extraction. In Ph. de Groote, editor, *The Curry-Howard Isomorphism*, volume 8 of *Cahiers du Centre de Logique*, pages 257–364. Université Catholique du Louvain, 1995.

[LK01] Jean Louis Krivine. Typed lambda-calculus in classical Zermelo-Fraeænkel set theory. *Archive for Mathematical Logic*, 40(3):189–205, 2001.

[LP92] Z. Luo and R. Pollack. LEGO proof development system: User's manual. Technical Report ECS-LFCS-92-211, University of Edinburgh, 1992.

[Lub93] Robert S. Lubarsky. Intuitionistic L. In John N. Crossley, Jeffrey B. Remmel, Richard A. Shore, and Moss E. Sweedler, editors, *Logical Methods: In Honor of Anil Nerode's Sixtieth Birthday*, pages 555–571. Birkhauser, 1993.

[Lub02] Robert S. Lubarsky. IKP and Friends. *J. Symb. Log.*, 67(4):1295–1322, 2002.

[Lub07] Robert S. Lubarsky. On the Cauchy Completeness of the Constructive Cauchy Reals. *Electron. Notes Theor. Comput. Sci.*, 167:225–254, 2007.

[McC84] D.C. McCarty. *Realizability and Recursive Mathematics*. D.Phil. Thesis, University of Oxford, 1984.

[Miq03] Alexandre Miquel. A Strongly Normalising Curry-Howard Correspondence for IZF Set Theory. In *Proc. of 12th Ann. Conf. of the EACSL (CSL 2003)*, volume 2803 of *LNCS*, pages 441–454. Springer, 2003.

[Miq04] Alexandre Miquel. Lambda-Z: Zermelo's Set Theory as a PTS with 4 Sorts. In Jean-Christophe Filliâtre, Christine Paulin-Mohring, and

Benjamin Werner, editors, *TYPES*, volume 3839 of *Lecture Notes in Computer Science*, pages 232–251. Springer, 2004.

[Mit96] John C. Mitchell. *Foundations for Programming Languages.* MIT Press, 1996.

[ML73] Per Martin-Löf. An intuitionistic theory of types: Predicative part. In *Logic Colloquium '73*, pages 73–118. North-Holland, Amsterdam, 1973.

[ML82] Per Martin-Löf. Constructive mathematics and computer programming. In *Proceedings of the Sixth International Congress for Logic, Methodology, and Philosophy of Science*, pages 153–175, Amsterdam, 1982. North Holland.

[Moc06a] Wojciech Moczydłowski. Normalization of IZF with Replacement. In *Proc. 15th Ann. Conf. of the EACSL (CSL 2006)*, volume 4207 of *Lecture Notes in Computer Science*. Springer, 2006.

[Moc06b] Wojciech Moczydłowski. A Normalizing Intuitionistic Set Theory with Inaccessible Sets. Technical Report TR2006-2051, Cornell University, 2006.

[Moc07] Wojciech Moczydłowski. A Dependent Set Theory. In *Proceedings of the Twenty-Second Annual IEEE Symposium on Logic in Computer Science*, pages 23–32. LICS, June 2007.

[MP02] Ieke Moerdijk and Erik Palmgren. Type theories, toposes and constructive set theory: predicative aspects of AST. *Annals of Pure and Applied Logic*, 114:155–201, 2002.

[Muz93] Michał Muzalewski. *An Outline of PC Mizar.* Foundation of Logic, Mathematics and Informatics, Mizar User Group, Brussels, 1993.

[Myh73] John Myhill. Some properties of intuitionistic Zermelo-Fraenkel set theory. In *Cambridge Summer School in Mathematical Logic*, volume 29, pages 206–231. Springer, 1973.

[NPS90] Bengt Nordström, Kent Petersson, and Jan M. Smith. *Programming in Martin-Löf's Type Theory.* Oxford Sciences Publication, Oxford, 1990.

[Pea89] Giuseppe Peano. *Arithmetices Principia, Nova Methodo Exposita.* Fratres Bocca, Turin, 1889.

[Pie02] Benjamin C. Pierce. *Types and Programming Languages*. MIT Press, 2002.

[Pra65] D. Prawitz. *Natural Deduction*. Almquist and Wiksell, Stockholm, 1965.

[Rat03] Michael Rathjen. Realizing Mahlo set theory in type theory. *Archive for Mathematical Logic*, 42:89–101, 2003.

[Rat04] Michael Rathjen. Realizability for Constructive Zermelo-Fraenkel set theory. Preprint, 2004.

[Rat05a] Michael Rathjen. The disjunction and related properties for constructive Zermelo-Fraenkel set theory. *Journal of Symbolic Logic*, 70:1233–1254, 2005.

[Rat05b] Michael Rathjen. Generalized inductive definitions in constructive set theory. In Laura Crosilla and Peter Schuster, editors, *From Sets and Types to Topology and Analysis: Towards Practicable Foundations for Constructive Mathematics*. Oxford University Press, 2005.

[Rat06] Michael Rathjen. Metamathematical properties of intuitionistic set theories with choice principles. 2006. Manuscript, available from the web page of the author.

[RS63] Helena Rasiowa and Roman Sikorski. *The Mathematics of Metamathematics*. Number 41 in Moanogrfie Matematyczne. Polish Scientific Publishers, 1963.

[Sco70] D. Scott. Constructive validity. In D. Lacombe M. Laudelt, editor, *Symposium on Automatic Demonstration*, volume 5(3) of *Lecture Notes in Mathematics*, pages 237–275. Springer-Verlag, New York, 1970.

[Set00] Anton Setzer. Extending Martin-Löf type theory by one Mahlo-universe. *Arch. Math. Log.*, 39:155 – 181, 2000.

[Shi94] Masaru Shirahata. *Linear Set Theory*. PhD thesis, 1994.

[Sim99] Stephen G. Simpson. *Subsystems of second order arithmetic*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1999.

[SU06] M.H.B. Sørensen and P. Urzyczyn. *Lectures on the Curry-Howard Isomorphism.* Elsevier, 2006.

[Ter04] Kazushige Terui. Light affine set theory: A naive set theory of polynomial time. *Studia Logica*, 77(1):9–40, 2004.

[The04] The Coq Development Team. *The Coq Proof Assistant Reference Manual – Version V8.0*, April 2004.

[Tro73] Anne Sjerp Troelstra. *Metamathematical Investigation of Intuitionistic Mathematics*, volume 344 of *Lecture Notes in Mathematics*. Springer-Verlag, 1973.

[Tro98] A.S. Troelstra. Realizability. In S.R. Buss, editor, *Handbook of Proof Theory*, volume 137 of *Studies in Logic and the Foundations of Mathematics*, pages 407–473. Elsevier, 1998.

[Tur36] Alan M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.*, 2(42):230–265, 1936.

[TvD88] A.S. Troelstra and D. van Dalen. *Constructivism in Mathematics, An Introduction*, volume I, II. North-Holland, Amsterdam, 1988.

[van99] D. van Dalen. *Mystic, Geometer, and Intuitionist: The Life of L.E.J. Brouwer. The dawning Revolution.*, volume I. Oxford University Press, 1999.

[vH67] J. van Heijenoort, editor. *From Frege to Gödel: A Source Book in Mathematical Logic, 1879–1931.* Harvard University Press, Cambridge, MA, 1967.

[vO02] Jaap van Oosten. Realizability: A historical essay. *Mathematical Structures in Computer Science*, 12(3):239–263, 2002.

[Š85] Andre Ščedrov. Intuitionistic set theory. In *Harvey Friedman's Research on the Foundations of Mathematics*, pages 257–284. Elsevier, 1985.

[vS90] Walter P. van Stigt. *Brouwer's Intuitionism.* North-Holland, Amsterdam, 1990.

[Wer97] Benjamin Werner. Sets in types, types in sets. In *TACS '97: Proc. of* *the 3rd Int. Symposium on Theoretical Aspects of Computer Software*, pages 530–546. Springer-Verlag, 1997.

[Zer08] Ernst Zermelo. Untersuchungen über die Grundlagen der Mengenlehre I. *Mathematische Annalen*, 65:261–281, 1908. English translation in [vH67].