Induction-Recursion and Initial Algebras

Peter Dybjer*and Anton Setzer†

Abstract

Induction-recursion is a powerful definition method in intuitionistic type theory. It extends (generalized) inductive definitions and allows us to define all standard sets of Martin-Löf type theory as well as a large collection of commonly occurring inductive data structures. It also includes a variety of universes which are constructive analogues of inaccessibles and other large cardinals below the first Mahlo cardinal. In this article we give a new compact formalization of inductive-recursive definitions by modeling them as initial algebras in slice categories. We give generic formation, introduction, elimination, and equality rules generalizing the usual rules of type theory. Moreover, we prove that the elimination and equality rules are equivalent to the principle of the existence of initial algebras for certain endofunctors. We also show the equivalence of the current formulation with the formulation of induction-recursion as a reflection principle given in [12]. Finally, we discuss two type-theoretic analogues of Mahlo cardinals in set theory: an external Mahlo universe which is defined by induction-recursion and captured by our formalization, and an internal Mahlo universe, which goes beyond induction-recursion. We show that the external Mahlo universe, and therefore also the theory of inductive-recursive definitions, have proof-theoretical strength of at least Rathjen's theory KPM.

1 Introduction

Induction-recursion is a powerful definition method in intuitionistic type theory in the sense of Scott ("Constructive Validity") [31] and Martin-Löf [17, 18, 19].

The first occurrence of formal induction-recursion is Martin-Löf's definition of a *universe à la Tarski* [19], which consists of a set U_0 of codes for small sets together with a decoding function T_0 which maps a code to the small set it denotes. U_0 is inductively generated at the same time as T_0 is defined by recursion on the elements of U_0, and the introduction rules for U_0 refer to T_0. It is called universe "à la Tarski" because of the similarity with Tarski's truth definition: U_0 is a generalized syntax of "formulas" and T_0 maps each formula to its "meaning". In earlier formulations of Martin-Löf type theory [17, 18] universes are formulated "à la Russell", where there is no syntactic distinction between an element of U_0 and the set it denotes. Therefore there is no need for a decoding function and hence there is no (explicit) induction-recursion.

Intuitionistic type theory with inductive-recursive definitions is also a suitable metalanguage for *intuitionistic metamathematics*. For example, in Martin-Löf's

*Department of Computing Science, Chalmers University of Technology, SE-412 96 Gteborg, Sweden. Email: `peterd@cs.chalmers.se`

†Supported by the Nuffield Foundation, grant No. NAL/00303/G; Department of Computer Science, University of Wales Swansea, Singleton Park, Swansea SA2 8PP, UK, Email: `a.g.setzer@swan.ac.uk`

proof of normalization of an early version of his type theory [20] he introduces Tait-style computability predicates for dependent types. Whereas Tait defines a family of computability predicates indexed by the types of the simply typed lambda calculus, Martin-Löf's computability predicates are indexed by those types which themselves are computable. This gives rise to a situation where the computable types are inductively generated at the same time as the computability predicate on terms of such a type is defined, and where the definition of a computable type refers to the notion of a computable term. Martin-Löf presumably considered this definition intuitionistically valid, but did not provide an explicit discussion of why this is so.

It is a non-trivial problem to give classical mathematical meaning to Martin-Löf's computability predicates. One approach is due to Aczel [1] for the closely related construction of a Frege structure. Other approaches have been proposed by Allen [2] and by Löfwall and Sjödin [16].

Although Martin-Löf's computability predicates nowadays can be regarded as an informal example of an inductive-recursive definition and therefore as a precursor of the concept of induction-recursion, its inductive-recursive nature is not explicit: instead of "computable type" Martin-Löf states when the notion of computability for a certain type "has been defined" and there is no explicit notion of proof for the fact that computability has been defined. In order to obtain an explicit inductive-recursive definition one has to formalize the metalanguage. It is an example of *indexed* induction-recursion [11, 13], since we are defining computability *predicates* and thus by Curry-Howard *indexed families of sets*.

More examples of formal induction-recursion occur in recent work on large universes in type theory. These are constructive analogues of large cardinals in set theory. For example, Martin-Löf's universes are analogues of inaccessible cardinals; Palmgren's superuniverse [23] is an analogue of a hyperinaccessible cardinal. Rathjen, Griffor and Palmgren's quantifier universes [30] are analogues of Mahlo's π-numbers; Palmgren's higher order universes [25] go even further and are generally conjectured to reach the strength of Rathjen's theory KPM; in Section 6 we will describe a weak version of Setzer's Mahlo universe [36, 34, 35], which is still inductive-recursive, and show that it has at least the strength of Rathjen's theory KPM [28]. Setzer's original Mahlo universe is an example of a universe which goes beyond induction-recursion.

Induction-recursion as a general unifying principle for definitions of this kind was identified by Dybjer [8, 11], who presents an external schema for their syntactic form. This schema extends earlier schemata for inductive definitions in type theory [6, 7, 9, 26]. Dybjer and Setzer [12] give a finite axiomatization of induction-recursion as a very general reflection principle. They also show the consistency of their axiomatization by building a model in classical set theory extended by a Mahlo cardinal. In this model function spaces are interpreted as full classical function spaces.

Models for inductive-recursive definitions of a set U with decoding function T are obtained as inductive definitions of the graph of T and captured formally as the least fixed point of a monotone operator on the lattice of subsets of a sufficiently large base set, see Dybjer [11] and the above-mentioned model by Dybjer and Setzer [12]. This should explain why it has not been natural to isolate the concept of induction-recursion in set theory. It also makes it difficult to trace the history of the concept.

In this paper we give a new compact finite axiomatization $\mathbf{IR_{elim}}$ of inductive-recursive definitions based on the idea of modeling them as initial algebras in slice categories. Such a categorical model is an abstraction of the above-mentioned set theoretic semantics. It thus provides an alternative view of inductive-recursive definitions to the axiomatization $\mathbf{IR_{refl}}$ given by Dybjer and Setzer [12].

The two axiomatizations highlight different aspects of inductive-recursive definitions. $\mathbf{IR_{refl}}$ is based on the idea that induction-recursion is a reflection principle. In the paradigmatic example operations on sets, such as Π and Σ, are reflected as operations $\widehat{\Pi}$ and $\widehat{\Sigma}$ on a particular set, the first universe U_0. Induction-recursion generalizes this idea to operations on arbitrary types. Moreover, $\mathbf{IR_{refl}}$ is based on a commuting square generalizing the usual initial algebra diagram used for modelling inductive definitions with the following correspondences:

inductively defined set	initial algebra
recursively defined function	initial arrow

$\mathbf{IR_{elim}}$ on the other hand coincides with a natural understanding of induction-recursion: elements of a set U are introduced by a constructor, and for every such element the value of the decoding function T is determined. Therefore it is closer to the standard set theoretic model. It also leads in a natural way to the functors on slice categories associated with the codes for inductive-recursive definition. (We show that the introduction/elimination rules in $\mathbf{IR_{elim}}$ are equivalent to the principle that these functors have initial algebras.) The theory $\mathbf{IR_{elim}}$ is shorter than $\mathbf{IR_{refl}}$; it has fewer rules and concepts. Moreover, it is easier to construct codes for inductive-recursive definitions in $\mathbf{IR_{elim}}$. As a consequence it is more suitable as a basis for an implementation of induction-recursion. However, each of the formulations has conceptual advantages and is of importance for metamathematical investigations.

Plan of the paper. In Section 2 we introduce the logical framework for intuitionistic type theory. In Section 3 we introduce $\mathbf{IR_{elim}}$. In Section 4 we introduce $\mathbf{IR^{ext}_{init}}$, which axiomatizes closure under certain initial algebras in slice categories, and prove the equivalence of $\mathbf{IR^{ext}_{init}}$ and $\mathbf{IR^{ext}_{elim}}$ (the extensional version of $\mathbf{IR_{elim}}$). In Section 5 we recall the theory $\mathbf{IR_{refl}}$ from Dybjer and Setzer [12] and show that it is equivalent to the two other theories under certain assumptions. Finally, in Section 6 we discuss Setzer's Mahlo universes which are type-theoretic analogues of Mahlo cardinals in set theory. There are two versions. One is an external Mahlo universe which is defined by induction-recursion and can be formalized in $\mathbf{IR_{elim}}$. We also determine a lower bound for its strength and therefore for $\mathbf{IR_{elim}}$, $\mathbf{IR^{ext}_{init}}$ and $\mathbf{IR_{refl}}$. The other version is Setzer's original internal Mahlo universe, which goes beyond induction-recursion.

2 A Logical Framework for Type Theory

2.1 Basic Rules

In the most recent versions, Martin-Löf type theory is presented in two stages:

- The first stage contains the most basic rules for dependent types. This is often referred to as the "logical framework" or "theory of types".
- The second stage contains the formation, introduction, elimination, and equality rules for a number of set formers such as $\mathrm{N}_n, \mathrm{N}, +, \Sigma, \ldots$. This is sometimes referred to as the "theory of sets" and is about the basic notion of set in Martin-Löf type theory, that is, sets as inductively defined data types. It is important to distinguish between this notion of set and the notion of iterative set in the sense of set theory.

All sets introduced at the second stage can be defined by induction-recursion and the remaining sections of this paper provide a complete definition of this "theory of sets".

In this section we will define the "theory of types". This will contain the rules for such a theory in Nordström et al [22], but also some new rules. We shall here give an informal introduction and refer the reader to Appendix A for the complete collection of rules.

The logical framework has four forms of judgement:

- $A : \mathsf{type}$,
- $A = B : \mathsf{type}$,
- $a : A$,
- $a = b : A$.

Each of these judgements can be hypothetical, that is, depend on a context Γ of the form $x_1 : A_1, \ldots, x_n : A_n$, which specifies the types of the free variables x_i of the judgement. The empty context ($n = 0$) is denoted by $\emptyset$.

For the treatment of contexts we need a fifth judgement

- Γ $\mathsf{context}$.

A hypothetical judgement is written $\Gamma \Rightarrow A : \mathsf{type}$, etc. When presenting inference rules we shall often simplify rules by omitting uniformly appearing contexts (see Appendix A for details).

As usual, we have a type set, but we also add a new type stype of "small types". This contains all sets and is closed under $\mathbf{0}$, $\mathbf{1}$, $\mathbf{2}$, dependent product and dependent function space. (All these constructions are introduced below. Synonyms of "dependent product type" are "disjoint union of a family of types" and "Σ-type", and synonyms of "dependent function space" are "Cartesian products of a family of types" and "Π-type"). However set itself is not an element of stype. The reason for the need for stype is discussed in Section 3.2 and [11]. [1]

We have the following rules:

$$\mathsf{set} : \mathsf{type} \qquad\qquad \mathsf{stype} : \mathsf{type}$$

$$\frac{A : \mathsf{set}}{A : \mathsf{stype}} \qquad\qquad \frac{A : \mathsf{stype}}{A : \mathsf{type}}$$

$$\frac{A = B : \mathsf{set}}{A = B : \mathsf{stype}} \qquad\qquad \frac{A = B : \mathsf{stype}}{A = B : \mathsf{type}}$$

$\mathbf{0}$, $\mathbf{1}$ and $\mathbf{2}$ are stypes with 0, 1, 2 elements respectively. In the case of $\mathbf{1}$ we add the η-rule. This has the effect that for any set A, the functions $f_0 := (x, y)x : A \to (\mathbf{1} \to A)$ and $f_1 := (x)x(*) : (\mathbf{1} \to A) \to A$ are inverses with respect to definitional equality, that is, we have definitionally $f_0 \circ f_1 = \mathrm{id}$ and $f_1 \circ f_0 = \mathrm{id}$ (with $\mathrm{id} := (x)x$). The same holds for $g_0 := (x)\langle x, *\rangle : A \to (A \times \mathbf{1})$ and $g_1 := (x)\pi_0(x) : (A \times \mathbf{1}) \to A$ (where π_0 is left projection). The stype $\mathbf{0}$ is added for systematic reasons. If we omitted it we could still define the empty set as the set N_0 with only one constructor of type $\mathrm{N}_0 \to \mathrm{N}_0$, see p. 13.

[1] In the proof assistant Agda for type theory (developed by C. Coquand and T. Coquand [5]) the logical framework has been modified so that the type set is closed under the dependent product and dependent function space of the logical framework. If we formulated induction-recursion based on that version of the logical framework there would be no need to distinguish between stype and set.

The rules for $\mathbf{0}$, $\mathbf{1}$, $\mathbf{2}$ are:

$$\mathbf{0} : \mathsf{stype} \qquad \frac{a : \mathbf{0} \qquad x : \mathbf{0} \Rightarrow A : \mathsf{type}}{\mathrm{case}_0(a) : A}$$

$$\mathbf{1} : \mathsf{stype} \qquad * : \mathbf{1} \qquad \frac{a : \mathbf{1}}{a = * : \mathbf{1}}$$

$$\mathbf{2} : \mathsf{stype} \qquad *_0 : \mathbf{2} \qquad *_1 : \mathbf{2}$$

$$\frac{x : \mathbf{2} \Rightarrow A : \mathsf{type} \qquad a : \mathbf{2} \qquad b : A[x := *_0] \qquad c : A[x := *_1]}{\mathrm{case}_2((x)A, a, b, c) : A[x := a]}$$

$$\frac{x : \mathbf{2} \Rightarrow A : \mathsf{type} \qquad b : A[x := *_0] \qquad c : A[x := *_1]}{\mathrm{case}_2((x)A, *_0, b, c) = b : A[x := *_0]}$$

$$\frac{x : \mathbf{2} \Rightarrow A : \mathsf{type} \qquad b : A[x := *_0] \qquad c : A[x := *_1]}{\mathrm{case}_2((x)A, *_1, b, c) = c : A[x := *_1]}$$

Both type and stype are closed under dependent function types written as $(x : A) \to B$. Function abstraction is written as $(x : A)a$ and application as $a(b)$. They are related by both the β- and the η-rule. Further type and stype are closed under dependent products written as $(x : A) \times B$. Pairs are written as $\langle a, b\rangle$ and the left and right projection of a is written as $\pi_0(a)$ and $\pi_1(a)$. Again, we have analogues of β and η (surjective pairing).

We also use some abbreviations. We omit the type in an abstraction, that is, write $(x)a$ instead of $(x : A)a$. We sometimes write curried function types as $(x_1 : A_1, \ldots, x_n : A_n) \to A$ instead of $(x_1 : A_1) \to \ldots \to (x_n : A_n) \to A$, and omit variables which are not used. $A \to B := (A) \to B$. We write repeated application as $a(b_1, \ldots, b_n)$ instead of $a(b_1) \cdots (b_n)$, and repeated abstraction as $(x_1, \ldots, x_n)a$ instead of $(x_1) \cdots (x_n)a$. Furthermore, if we apply an expression $f(a_1, \ldots, a_n)$ introduced in this form by a rule to arguments $b_1, \ldots, b_k$, we write $f(a_1, \ldots, a_n, b_1, \ldots, b_k)$ instead of $f(a_1, \ldots, a_n)(b_1, \ldots, b_k)$.

We will in the following not mention equality versions of the rules. Moreover, we will omit types and premises in equality judgements and use "bracket notations" like $E[t]$ as usual, see General Assumption A.0.3 in Appendix A for details.

We introduce furthermore the following notation for the definition of a function from one type into a product type from its two projections: Assume
Assume the following: $A : \mathsf{type}$; $x : A \Rightarrow B[x] : \mathsf{type}$; $x : A, y : B[x] \Rightarrow C[x, y] : \mathsf{type}$; $f : A \to B[x]$ and $g : (x : A) \to C[x, f(x)]$. Then

$$\langle f, g\rangle^{\mathrm{fun}} := (x)\langle f(x), g(x)\rangle : A \to ((y : B[x]) \times C[x, y]) \ .$$

2.2 Extensions of the Logical Framework

In the subsequent three sections we shall give three different formalizations of inductive-recursive definitions in type theory: $\mathbf{IR_{elim}}$ (Section 3), $\mathbf{IR^{ext}_{init}}$ (Section 4) and $\mathbf{IR_{refl}}$ (Section 5). We shall also prove the equivalence of these three theories extended by further rules and will therefore introduce additional theories.

Each of these theories consists of the rules of the logical framework together with some (yet to be specified) rules for inductive-recursive definitions. A *rule* r is here an $n+1$-tuple $\Gamma_1 \Rightarrow \theta_1, \ldots, \Gamma_{n+1} \Rightarrow \theta_{n+1}$ of dependent judgements in the language of type theory with respect to a certain collection of constructors (for a full formalization of the language of type theory see for instance Setzer [33], chapter 2). If R is a collection of rules we introduce the type theory $\mathrm{TT}(R)$. We use the notation $R \vdash \Gamma \Rightarrow \theta$ to make explicit that the judgement θ is derivable in the context Γ by using the rules of the logical framework (without extensionality) and by applying rules in R: If r is as above and $R \vdash \Gamma_i \Rightarrow \theta_i$ $(i = 1, \ldots, n)$, then $R \vdash \Gamma_{n+1} \Rightarrow \theta_{n+1}$. We will as usual suppress R when writing down the judgements of type theory and often also keep the context Γ implicit.

2.3 Extensional Equality

Some of the rules of the theory $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{init}}$ will only be typeable if we assume certain rules of *extensional equality*. These rules are similar to those of Martin-Löf's extensional type theory [18, 19] but are here formulated for the types and stypes of the logical framework. (Martin-Löf's extensional type theory was formulated without a logical framework.)

$$\frac{A : \mathsf{type} \qquad a : A \qquad b : A}{a =_A b : \mathsf{type}} \qquad \frac{A : \mathsf{stype} \qquad a : A \qquad b : A}{a =_A b : \mathsf{stype}}$$

$$\frac{A : \mathsf{type} \qquad a : A}{\mathrm{ref} : a =_A a}$$

$$\frac{A : \mathsf{type} \qquad a : A \qquad b : A \qquad r : a =_A b}{a = b : A}$$

$$\frac{A : \mathsf{type} \qquad a : A \qquad b : A \qquad r : a =_A b}{r = \mathrm{ref} : a =_A b}$$

We define for $A : \mathsf{type}$, $x : A \Rightarrow B[x] : \mathsf{type}$

$$(f =^{\mathrm{fun}}_{(x:A) \to B[x]} g) := (x : A) \to (f(x) =_{B[x]} g(x)) \enspace ,$$

which is equivalent to $f =_{(x:A) \to B} g$, but has proof objects which can be used more directly.

We will also need to add the extensionality rules to $\mathbf{IR}_{\mathbf{elim}}$ and $\mathbf{IR}_{\mathbf{refl}}$ (yielding $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{elim}}$ and $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{refl}}$) in order to prove their equivalence to $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{init}}$.

We want to emphasize that the rules for extensional equality are not needed when formulating $\mathbf{IR}_{\mathbf{elim}}$ and $\mathbf{IR}_{\mathbf{refl}}$: the formation, introduction, elimination, and equality rules are all typeable in intensional type theory, and it is therefore possible to consider them in that setting as well. However, our experience suggests that some constructions in intuitionistic metamathematics are difficult and perhaps even impossible to perform in intensional type theory, and that intensional type theory is sometimes counterintuitive − proofs which one informally believes are correct turn out to be incorrect when type checking them formally. It seems therefore that the extensional versions $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{elim}}$ or $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{refl}}$ are closer to our mathematical intuition and therefore more natural. On the other hand it is possible to construct non-normalizing terms in extensional type theory so decidability of type-checking is lost. It is this property that is used crucially in the elegant implementations of proof assistants for intensional type theory.

2.4 The Category of Types

All three formalizations are based to a lesser or greater extent on category-theoretic ideas. However the definitions in the category theoretical part can be done only in the presence of extensional equality. We shall introduce for each collection of rules R, which contains the rules of extensionality, and each context Γ the category $\mathbf{Type}_R(\Gamma)$. Its objects are A such that $R \vdash \Gamma \Rightarrow A : \mathsf{type}$ and we identify objects A, B such that $R \vdash \Gamma \Rightarrow A = B : \mathsf{type}$. Arrows from A to B are functions f for which we can prove $R \vdash \Gamma \Rightarrow f : A \to B$ (and again we identify f, g such that $R \vdash \Gamma \Rightarrow f = g : A \to B$).

We shall also usually suppress R and Γ in $\mathbf{Type}_R(\Gamma)$ and simply write $\mathbf{Type}$.

It is only $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{init}}$ which pursues the categorical approach fully. This theory is obtained by postulating, in the language of type theory, that for each type D and each code γ (for an inductive-recursive definition with decoding into D) there exists a certain endofunctor $\mathbb{F}_\gamma$ on the slice category $\mathbf{Type}/D$ (more precisely $\mathbf{Type}_{\mathbf{IR}^{\mathbf{ext}}_{\mathbf{init}}}(\Gamma)/D$) and that this endofunctor has an initial algebra.

Here we recall that the objects of the slice category $\mathbf{Type}/D$ are pairs $\langle U, T\rangle$ such that U is an object and $T : U \to D$ in $\mathbf{Type}$. Arrows from $\langle U, T\rangle$ to $\langle U', T'\rangle$ are pairs $\langle f_0, f_1\rangle$ of reindexing functions functions $f_0 : U \to U'$ and proofs $f_1 : T' \circ f_0 =^{\mathrm{fun}}_{U \to D} T$ (in R under assumption Γ). f_1 is therefore a proof that a certain diagram commutes, a fact indicated as follows:

$$\begin{tikzcd} U \arrow[rr, "f_0"] \arrow[dr, "T"'] & {} \arrow[d, phantom, "f_1"] & U' \arrow[dl, "T'"] \\ & D & \end{tikzcd}$$

Again we identify $\langle f_0, f_1\rangle$ and $\langle f_0', f_1'\rangle$, such that f_0 and f_0' are equal as arrows in $\mathbf{Type}/D$ (which implies by uniqueness of equality proofs that we can prove in R and under assumption Γ $f_1 = f_1'$.)

Both $\mathbf{IR}_{\mathbf{elim}}$ and $\mathbf{IR}_{\mathbf{refl}}$ also use categorical ideas to some extent, but in both cases the main guideline has been to formulate the rules in a way which is natural from a type-theoretic perspective and which recovers the usual rules of type theory without undue coding. So it will not be appropriate to investigate the categorical properties of $\mathbf{Type}_{\mathbf{IR}_{\mathbf{elim}}}$ and $\mathbf{Type}_{\mathbf{IR}_{\mathbf{refl}}}$ – in fact one needs to modify the definition of $\mathbf{Type}$ first in order to obtain categories at all, which then still lack many of the expected properties. However, in the presence of extensional equality and induction on the collection of strictly positive functors these theories will be equivalent to $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{init}}$.

3 A New Formalization of Induction-Recursion

In this section we shall give the formal rules of the theory $\mathbf{IR}_{\mathbf{elim}}$. To motivate these rules we shall begin with some informal discussion. There are two issues:

- the correspondence between certain endofunctors on slice categories and the rules for inductive-recursive definitions;
- which endofunctors arise in this way.

3.1 Algebras in Slice Categories

Consider again the first universe à la Tarski U_0 with decoding $\mathrm{T}_0 : \mathrm{U}_0 \to \mathsf{set}$. This universe is closed under the formation of Σ-types, so there is a constructor[2]

$$\hat{\Sigma} : ((a : \mathrm{U}_0) \times (\mathrm{T}_0(a) \to \mathrm{U}_0)) \to \mathrm{U}_0 \ ,$$

[2]Note that we here use the uncurried version of the constructor, whereas the usual logical framework based version of type theory employs the curried version.

which is decoded as

$$\mathrm{T}_0(\hat{\Sigma}(\langle a, b\rangle)) = \Sigma(\mathrm{T}_0(a), \mathrm{T}_0 \circ b) \ .$$

Observe the occurrence of T_0 in the introduction rule for U_0.

We can draw the defining equation for T_0 as a commuting diagram

$$\begin{tikzcd}
(a : \mathrm{U}_0) \times (\mathrm{T}_0(a) \to \mathrm{U}_0) \arrow[rr, "\hat{\Sigma}"] \arrow[dr, "\langle a,b\rangle \mapsto \Sigma(\mathrm{T}_0(a), \mathrm{T}_0 \circ b)"'] & & \mathrm{U}_0 \arrow[dl, "\mathrm{T}_0"] \\
& \mathsf{set} &
\end{tikzcd}$$

Consider now the above under the additional assumption of extensionality in the slice category **Type**/set. We note that the above diagram expresses that we have an $\mathbb{F}_0$-algebra for a certain endofunctor $\mathbb{F}_0$ on **Type**/set the object part of which has the following two components:

$$\begin{array}{rcl}
\mathbb{F}_0^{\mathrm{U}}(U, T) & = & (a : U) \times (T(a) \to U) \ , \\
\mathbb{F}_0^{\mathrm{T}}(U, T, \langle a, b\rangle) & = & \Sigma(T(a), T \circ b) \ .
\end{array}$$

(To avoid confusion, note that U in italic font is a variable ranging over arbitrary sets, whereas U_0 (and later U) in Roman font are constant sets defined by induction-recursion, and similarly for italic T and Roman T_0 and T.) When formalized in type theory, this $\mathbb{F}_0$-algebra becomes a quadruple

$$\langle \mathrm{U}_0, \mathrm{T}_0, \hat{\Sigma}, \mathrm{eq}_0 \rangle$$

(where eq_0 is a proof of the commutativity of the above mentioned diagram).

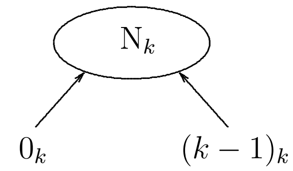
In $\mathbf{IR}_{\mathbf{elim}}$ we will also have a principle of *universe-elimination* for U_0 motivated by syntactic considerations. In $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{init}}$ however, universe-elimination is replaced by the principle that $\langle \mathrm{U}_0, \mathrm{T}_0, \hat{\Sigma}, \mathrm{eq}_0 \rangle$ is an *initial* $\mathbb{F}_0$*-algebra*. Furthermore we shall show in 4.4 that these principles are equivalent assuming rules of extensionality.

3.2 Inductive and Non-inductive Arguments

To get a complete formalization we must specify which endofunctors give rise to type-theoretically justifiable constructions. To motivate this specification we shall analyze the structure of a number of constructors (or introduction rules) for sets in type theory, and in particular emphasize the distinction between *inductive* and *non-inductive* arguments of a constructor (or premises of an introduction rule).

We shall first look at the case of inductive definitions, that is, the special case of inductive-recursive definitions where a recursively defined function does not participate in the inductive generation process.

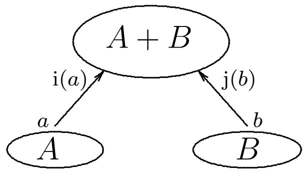
The simplest inductive definition is that of a finite set N_k with constructors $i_k : \mathrm{N}_k$[3] $(i = 0, \ldots, k-1)$. The constructors have no arguments at all.



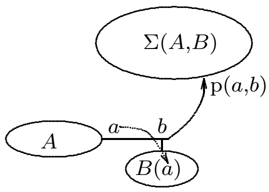
[3]Note that i_k is the ith element of the set N_k and should not be confused with the notation for sequences as used in mathematics.

In this diagram, the arrow represents the constructor.

The next example is the disjoint union of two sets $A + B$ with constructors $\mathrm{i} : A \to A + B$ and $\mathrm{j} : B \to A + B$. A and B are arbitrary previously defined sets, which we refer to as *parameters* of the definition.

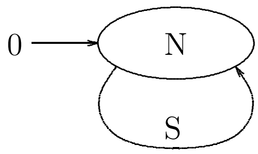


The next example is the set $\Sigma(A, B)$ with constructor $\mathrm{p} : (x : A, B(x)) \to \Sigma(A, B)$: it has two arguments, where the index set $B(x)$ of the second one depends on the first argument A. Again this can be parameterized by $A : \mathsf{set}$ and $B : (x : A) \to \mathsf{set}$.

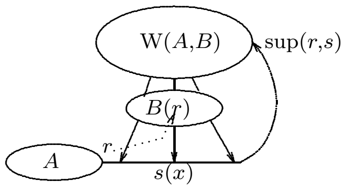


In this diagram, the dotted arrows denote dependencies of later arguments of the constructor on previous arguments.

The set N of natural numbers has constructors $0 : \mathrm{N}$ and $\mathrm{S} : \mathrm{N} \to \mathrm{N}$. Here, the type of the argument of S is N itself. We call such an argument "inductive".[4] In contrast, all previous arguments are "non-inductive", since their types only refer to previously defined sets.

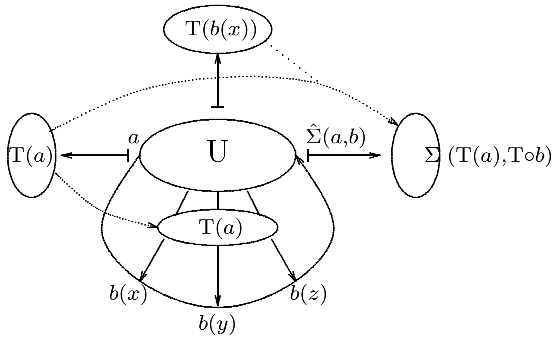


If we look at the set $\mathrm{W}(A, B)$ with constructor $\sup : (r : A, s : B(r) \to \mathrm{W}(A, B)) \to \mathrm{W}(A, B)$ we see that we have one non-inductive argument r and a family of inductive arguments s indexed by the set $B(r)$:



The primary example of proper induction-recursion is the definition of the universe à la Tarski. We note that U_0 is defined inductively, and while defining it, we simultaneously define recursively for every element $a : \mathrm{U}_0$ a set $\mathrm{T}_0(a) : \mathsf{set}$. Consider the constructor $\hat{\Sigma} : (a : \mathrm{U}_0, b : \mathrm{T}(a) \to \mathrm{U}_0) \to \mathrm{U}_0$ with $\mathrm{T}_0(\hat{\Sigma}(a, b))= \Sigma(\mathrm{T}_0(a), (x)\mathrm{T}_0(b(x))) : \mathsf{set}$. $\hat{\Sigma}$ has two inductive arguments, where the second depends on the first. This dependence is not direct (since U_0 is not defined yet), but indirect via T_0, that is, using the recursion-hypothesis for T_0. Finally, the definition of $\mathrm{T}_0(\hat{\Sigma}(a, b))$ refers to the value of T_0 for all inductive arguments:

[4]In [11] the terminology "recursive argument" was used, but "inductive argument" seems to be better in connection with induction-recursion, since it primarily has to do with the inductively defined set and only indirectly with the recursively defined function.



In this diagram $\mapsto$ represents the function mapping the constructed element to the recursively defined result. The dotted arrows express dependencies of later arguments of the constructor and of the recursively defined result of the constructor on previous arguments.

We shall analyze the common structure of the examples above. First the arguments are classified as either inductive or non-inductive.

The type of a non-inductive argument is an stype. The typical case is that it is a set, for instance the two arguments of the constructor p for $\Sigma(A, B)$ are elements of the sets A and $B(a)$. In the case of $\Pi(A, B)$ we have one constructor $\lambda : (f : (x : A) \to B) \to \Pi(A, B)$ and the non-inductive argument f is indexed over the stype $(x : A) \to B$. This type is not a set. In fact, this is why we cannot simply require that the type of a non-inductive argument is a set: we want to follow Martin-Löf and define Π-sets as inductively generated by λ.

An inductive argument is indexed by an stype. It can be a set, for example the second argument of $\mathrm{W}(A, B)$ is indexed by the set $B(a)$, or it can be an stype, for example the argument of the successor in N is indexed by $\mathbf{1}$. Note that because of the η-rule for $\mathbf{1}$ there are bijections between $\mathbf{1} \to A$ and A and an argument indexed over $\mathbf{1}$ is nothing but a single argument.

The type of later arguments may depend on earlier arguments. The dependency on a non-inductive argument is direct. However, a direct dependency on an inductive argument is not possible, because we cannot make use of the set we are currently defining. However an indirect dependency is possible, namely on the result of the recursively defined function T applied to the elements of the inductively defined set U indexed by the argument.

The result of T for an element introduced by a constructor depends on the arguments of the constructor in the same way as later arguments depend on previous ones.

If a set has several constructors it will be convenient to code them as one constructor with an extra argument indexed by a finite set which selects the chosen constructor. As we show below, the finite sets N_k can be built up successively from $\mathbf{0}$, $\mathbf{1}$, and $\mathbf{2}$ by using inductive definitions.

3.3 The Type OP_D of Operators on Families of D

We want to define a type OP_D of codes γ for all inductive-recursive definitions of sets $\mathrm{U}_\gamma : \mathsf{set}$, $\mathrm{T}_\gamma : \mathrm{U}_\gamma \to D$. U_γ will have one constructor, the arguments of which are elements of an stype which depends on U_γ, T_γ. Further the result of T_γ for an element introduced by a constructor will depend on U_γ, T_γ and the arguments of that constructor. So we will associate with every γ a function

$$\mathbb{F}_\gamma : ((X : \mathsf{set}) \times (X \to D)) \to ((X : \mathsf{stype}) \times (X \to D)) \ ,$$

and $\langle \mathrm{U}_\gamma, \mathrm{T}_\gamma \rangle$ is the least set "closed under $\mathbb{F}_\gamma$", i.e. such that every $a : \pi_0(\mathbb{F}_\gamma(\langle \mathrm{U}_\gamma, \mathrm{T}_\gamma \rangle))$ is represented as a canonical element $\mathrm{intro}_\gamma(a)$ in U_γ with $\mathrm{T}_\gamma(\mathrm{intro}_\gamma(a)) = \pi_1(\mathbb{F}_\gamma(\langle \mathrm{U}_\gamma, \mathrm{T}_\gamma \rangle))(a)$.

The above definition cannot be simplified by defining by induction on γ directly the type of the arguments of the constructor of U_γ: we cannot make use of the induction hypothesis relative to $\mathrm{U}_{\gamma'}, \mathrm{T}_{\gamma'}$.

So elements of OP_D represent primarily functions from set-indexed to stype-indexed families of D. The inductive-recursively defined sets are obtained as the least set-indexed family of D which is "closed under this function". In the theory $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{init}}$, this picture becomes more clear: there OP_D is a type of codes of endofunctors in the slice category $\mathbf{Type}/D$ and U_γ, T_γ together with the constructor and an equality proof form an initial algebra with respect to this functor.

In the following D will be a global parameter. Thus in the rest of the article we shorten our notations as follows:

General assumption 3.3.1 *(a) We assume $\Gamma \Rightarrow D : \mathsf{type}$ (but will usually suppress Γ).*

(b) We suppress a first premise $D : \mathsf{type}$, which has to be added to each rule referring to D.

(c) We write, if some $\gamma : \mathrm{OP}_D$ occurs as a parameter, γ instead of D, γ.

The formation rule for OP is

$$\mathrm{OP}_D : \mathsf{type} \ .$$

To each code γ we associate the two components of $\mathbb{F}_\gamma$ above (note however that OP is an inductive definition, not an inductive-recursive definition on type level, and the inductive definition of OP_D does not refer to $\mathbb{F}^{\mathrm{U}}$, $\mathbb{F}^{\mathrm{T}}$).

$$\frac{\gamma : \mathrm{OP}_D \qquad U : \mathsf{set} \qquad T : U \to D}{\mathbb{F}^{\mathrm{U}}_\gamma(U,T) : \mathsf{stype}}$$

$$\frac{\gamma : \mathrm{OP}_D \qquad U : \mathsf{set} \qquad T : U \to D}{\mathbb{F}^{\mathrm{T}}_\gamma(U,T) : \mathbb{F}^{\mathrm{U}}_\gamma(U,T) \to D}$$

We have the following rules for generating elements of OP_D:

- Addition of a non-inductive argument:

$$\frac{A : \mathsf{stype} \qquad \gamma : A \to \mathrm{OP}_D}{\sigma(A,\gamma) : \mathrm{OP}_D}$$

  $\sigma(A,\gamma)$ (where σ stands for the Σ-type) is a code for an inductive-recursive definition the constructor of which has one non-inductive argument $a : A$ and depending on it other arguments given by $\gamma(a)$. The result of T for an element introduced by a constructor with argument starting with $a : A$ is the result obtained for the remaining arguments with respect to $\gamma(a)$:

  - $\mathbb{F}^{\mathrm{U}}_{\sigma(A,\gamma)}(U,T) = (a : A) \times \mathbb{F}^{\mathrm{U}}_{\gamma(a)}(U,T)$,
  - $\mathbb{F}^{\mathrm{T}}_{\sigma(A,\gamma)}(U,T,\langle a,b\rangle) = \mathbb{F}^{\mathrm{T}}_{\gamma(a)}(U,T,b)$.

- Addition of an inductive argument:

$$\frac{A : \mathsf{stype} \qquad \gamma : (A \to D) \to \mathrm{OP}_D}{\delta(A,\gamma) : \mathrm{OP}_D}$$

  $\delta(A,\gamma)$ (where δ stands for **d**ependent Σ) is a code for an inductive-recursive definition the constructor of which has one inductive argument indexed over A and, if this argument is $f : A \to U$, the other arguments determined by $\gamma(T \circ f)$. The result of T for an element introduced by a constructor with such an argument is the result of it for the remaining arguments with respect to $\gamma(T \circ f)$:

- $\mathbb{F}^{\mathrm{U}}_{\delta(A,\gamma)}(U,T) = (f : A \to U) \times \mathbb{F}^{\mathrm{U}}_{\gamma(T\circ f)}(U,T)$,
- $\mathbb{F}^{\mathrm{T}}_{\delta(A,\gamma)}(U,T,\langle f,b\rangle) = \mathbb{F}^{\mathrm{T}}_{\gamma(T\circ f)}(U,T,b)$.

- Base case:

$$\frac{\psi : D}{\iota(\psi) : \mathrm{OP}_D}$$

$\iota(\psi)$ is an inductive-recursive definition with no arguments of the constructor and ψ as the result of T for an element introduced by it:

- $\mathbb{F}^{\mathrm{U}}_{\iota(\psi)}(U,T) = \mathbf{1}$,
- $\mathbb{F}^{\mathrm{T}}_{\iota(\psi)}(U,T,*) = \psi$.

Definition 3.3.2 *The formation and introduction rules for* OP *are the rules in this subsection.*

3.4 Formation and Introduction Rules for $\mathbf{U}, \mathbf{T}$

Definition 3.4.1 *Assume* $\gamma : \mathrm{OP}_D$*.*
The formation rules for U_γ*,* T_γ *are the following:*

$$\mathrm{U}_\gamma : \mathsf{set} \qquad\qquad \mathrm{T}_\gamma : \mathrm{U}_\gamma \to D$$

The introduction rules for U_γ *and equality rules for* T_γ *are*

$$\frac{a : \mathbb{F}^{\mathrm{U}}_\gamma(\mathrm{U}_\gamma,\mathrm{T}_\gamma)}{\mathrm{intro}_\gamma(a) : \mathrm{U}_\gamma} \qquad \frac{a : \mathbb{F}^{\mathrm{U}}_\gamma(\mathrm{U}_\gamma,\mathrm{T}_\gamma)}{\mathrm{T}_\gamma(\mathrm{intro}_\gamma(a)) = \mathbb{F}^{\mathrm{T}}_\gamma(\mathrm{U}_\gamma,\mathrm{T}_\gamma,a) : D}$$

The formation/introduction rules for U*,* T *are the rules above.*

3.5 Elimination and Equality Rules for $\mathbf{U}, \mathbf{T}$

We are going to define elimination rules for U_γ. Inductively defined sets like the set of natural numbers or the W-type are special cases of inductive-recursive definitions, so we obtain elimination rules for these sets as well as we do for universes (as introduced by Palmgren [24]).

We have to collect the induction hypotheses with respect to an argument of intro_γ, that is, with respect to an element u of $\mathbb{F}^{\mathrm{U}}_\gamma(\mathrm{U}_\gamma,\mathrm{T}_\gamma)$. The induction hypothesis consists of the value of the function to be defined for all references to U_γ by inductive arguments in u, and will be an element of type $\mathbb{F}^{\mathrm{IH}}_\gamma(\mathrm{U}_\gamma,\mathrm{T}_\gamma,E,u)$ with the following formation and equality rules:

$$\frac{\gamma : \mathrm{OP}_D \qquad U : \mathsf{set} \qquad T : U \to D \qquad u : \mathbb{F}^{\mathrm{U}}_\gamma(U,T) \qquad x : U \Rightarrow E[x] : \mathsf{type}}{\mathbb{F}^{\mathrm{IH}}_\gamma(U,T,E,u) : \mathsf{type}}$$

$$\begin{array}{rcl}
\mathbb{F}^{\mathrm{IH}}_{\iota(\psi)}(U,T,E,u) & = & \mathbf{1}\ , \\
\mathbb{F}^{\mathrm{IH}}_{\sigma(A,\gamma)}(U,T,E,\langle a,b\rangle) & = & \mathbb{F}^{\mathrm{IH}}_{\gamma(a)}(U,T,E,b)\ , \\
\mathbb{F}^{\mathrm{IH}}_{\delta(A,\gamma)}(U,T,E,\langle f,b\rangle) & = & ((x : A) \to E[f(x)]) \times \mathbb{F}^{\mathrm{IH}}_{\gamma(T\circ f)}(U,T,E,b)\ .
\end{array}$$

Note that we allow $E[x]$ to be an arbitrary type, that is, it does not need to be a set as in ordinary elimination rules. See [11] for a discussion of the need for such a large elimination schema in induction-recursion.

For the equality rules we need to define elements of $\mathbb{F}^{\mathrm{IH}}_\gamma(\mathrm{U}_\gamma, \mathrm{T}_\gamma, E, u)$ from the values of recursively defined functions on the inductive arguments of u. This is the purpose of the operation $\mathbb{F}^{\mathrm{map}}_\gamma$:

$$\frac{\gamma : \mathrm{OP}_D \qquad U : \mathsf{set} \qquad T : U \to \mathsf{set} \qquad x : U \Rightarrow E[x] : \mathsf{type} \qquad h : (x : U) \to E[x]}{\mathbb{F}^{\mathrm{map}}_\gamma(U, T, E, h) : (u : \mathbb{F}^{\mathrm{U}}_\gamma(U, T)) \to \mathbb{F}^{\mathrm{IH}}_\gamma(U, T, E, u)}$$

$$\begin{array}{rcl} \mathbb{F}^{\mathrm{map}}_{\iota(\psi)}(U, T, E, h, *) & = & * \ , \\ \mathbb{F}^{\mathrm{map}}_{\sigma(A,\gamma)}(U, T, E, h, \langle a, b\rangle) & = & \mathbb{F}^{\mathrm{map}}_{\gamma(a)}(U, T, E, h, b) \ , \\ \mathbb{F}^{\mathrm{map}}_{\delta(A,\gamma)}(U, T, E, h, \langle f, b\rangle) & = & \langle h \circ f, \mathbb{F}^{\mathrm{map}}_{\gamma(T \circ f)}(U, T, E, h, b)\rangle \ . \end{array}$$

Definition 3.5.1 *Assume* $\gamma : \mathrm{OP}_D$.
The elimination rule for U_γ, T_γ *is the following:*

$$\frac{x : \mathrm{U}_\gamma \ \Rightarrow E[x] : \mathsf{type} \qquad g : (u : \mathbb{F}^{\mathrm{U}}_\gamma(\mathrm{U}_\gamma, \mathrm{T}_\gamma), \mathbb{F}^{\mathrm{IH}}_\gamma(\mathrm{U}_\gamma, \mathrm{T}_\gamma, E, u)) \to E[\mathrm{intro}_\gamma(u)]}{\mathrm{R}_{\gamma,E}(g) : (u : \mathrm{U}_\gamma) \to E[u]}$$

The equality rule is

$$\frac{x : \mathrm{U}_\gamma \Rightarrow E[x] : \mathsf{type} \qquad g : (u : \mathbb{F}^{\mathrm{U}}_\gamma(\mathrm{U}_\gamma, \mathrm{T}_\gamma), \mathbb{F}^{\mathrm{IH}}_\gamma(\mathrm{U}_\gamma, \mathrm{T}_\gamma, E, u)) \to E[\mathrm{intro}_\gamma(u)] \qquad u : \mathbb{F}^{\mathrm{U}}_\gamma(\mathrm{U}_\gamma, \mathrm{T}_\gamma)}{\mathrm{R}_{\gamma,E}(g, \mathrm{intro}_\gamma(u)) = g(u, \mathbb{F}^{\mathrm{map}}_\gamma(\mathrm{U}_\gamma, \mathrm{T}_\gamma, E, \mathrm{R}_{\gamma,E}(g), u)) : E[\mathrm{intro}_\gamma(u)]}$$

Note that these rules presuppose the formation and introduction rules for OP *and for* U, T.

Definition 3.5.2 $\mathbf{IR_{elim}}$ *is the extension of the logical framework by the formation and introduction rules for* OP *and the formation, introduction, elimination, and equality rules for* U, T. $\mathbf{IR^{ext}_{elim}}$ *is the extension of* $\mathbf{IR_{elim}}$ *by the rules of extensionality.*

3.6 $\mathbf{OP_D}$-Codes for Some Standard Sets

Let us briefly review the examples of inductive and inductive-recursive definitions in Section 3.2 and assign codes in OP_D to them.

In the first examples, we just have inductive definitions - there is no recursively defined T_γ participating in the generation of U_γ. In this case we introduce a dummy function $\mathrm{T}_\gamma : \mathrm{U}_\gamma \to \mathbf{1}$ so that the code for the inductive definition is an element $\gamma : \mathrm{OP}_{\mathbf{1}}$. We will only define the corresponding $\gamma : \mathrm{OP}_{\mathbf{1}}$, the sets defined are then U_γ. Let $\iota_* := \iota(*) : \mathrm{OP}_{\mathbf{1}}$.

The empty stype $\mathbf{0}$ is part of our logical framework and we can code the set N_0 as

$$\gamma_{\mathrm{N}_0} := \sigma(\mathbf{0}, (x)\iota_*) : \mathrm{OP}_{\mathbf{1}} \ .$$

But it is possible to define N_0 without $\mathbf{0}$, since it can be defined as the set with only one constructor with type $\mathrm{N}_0 \to \mathrm{N}_0$. This definition has code

$$\gamma'_{\mathrm{N}_0} := \delta(\mathbf{1}, (f)\iota_*) : \mathrm{OP}_{\mathbf{1}} \ .$$

In Appendix B we prove that for the second definition of N_0 we can define ex falsum quodlibet.

The other finite sets have codes

$$\begin{aligned}\gamma_{\mathrm{N}_1} &:= \iota_* : \mathrm{OP}_{\mathbf{1}} \ , \\ \gamma_{\mathrm{N}_{n+2}} &:= \sigma(\mathbf{2}, (x)\mathrm{case}_2(x, \gamma_{\mathrm{N}_{n+1}}, \iota_*)) : \mathrm{OP}_{\mathbf{1}} \ .\end{aligned}$$

$A + B$ and $\Sigma(A, B)$ have codes

$$\begin{aligned}\gamma_{A+B} &:= \sigma(\mathbf{2}, (x)\mathrm{case}_2(x, \sigma(A, (x)\iota_*), \sigma(B, (x)\iota_*))) \ , \\ \gamma_{\Sigma(A,B)} &:= \sigma(A, (x)\sigma(B(x), (y)\iota_*)) \ .\end{aligned}$$

With this definition the constructor of Σ has two arguments. An alternative is to have one argument having as type the dependent product of the logical framework:

$$\gamma'_{\Sigma(A,B)} \ := \ \sigma((x : A) \times B, (y)\iota_*) \ .$$

N has code

$$\gamma_{\mathrm{N}} \ := \ \sigma(\mathbf{2}, (x)\mathrm{case}_2(x, \iota_*, \delta(\mathbf{1}, (y)\iota_*))) \ .$$

Zero is here $\mathrm{intro}_{\gamma_{\mathrm{N}}}(\langle *_0, * \rangle)$, and the successor of n is $\mathrm{intro}_{\gamma_{\mathrm{N}}}(\langle *_1, \langle n, * \rangle\rangle)$.

$\mathrm{W}(A, B)$ has code

$$\gamma_{\mathrm{W}(A,B)} := \sigma(A, (x)\delta(B(x), (y)\iota_*)) \ .$$

Finally, the first universe (consisting of $\mathrm{U}_0 : \mathsf{set}$ and $\mathrm{T}_0 : \mathrm{U}_0 \to \mathsf{set}$ and for simplicity closed under N and Σ only) has code

$$\gamma_{\mathrm{U}_0,\mathrm{T}_0} := \sigma(\mathbf{2}, (x)\mathrm{case}_2(x, \iota(\mathrm{N}), \delta(\mathbf{1}, (A)\delta(A(*), (B)\iota(\Sigma(A(*), B)))))) : \mathrm{OP}_{\mathsf{set}} \ .$$

4 Initial Algebras in Slice Categories

In this section we pursue the categorical point of view and introduce the theory $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{init}}$ which expresses closure under initial $\mathbb{F}_\gamma$-algebras. The **ext** in $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{init}}$ indicates that we here assume the rules of extensional equality.

We will also introduce the principle of OP-elimination and prove that this principle entails the equivalence of $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{init}}$ and $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{elim}}$. This can be viewed as yet another theorem showing the correspondence between syntactic theories and categorical models, such as the correspondence between the typed lambda calculus and Cartesian closed categories, between (impredicative) intuitionistic type theory in the sense of Lambek and Scott [15] and toposes, etc. Note however, that we here only treat the categorical semantics of induction-recursion and not of the logical framework. The reader is referred to the literature on categorical semantics of dependent types for the latter, see for example Cartmell [4], Seely [32], Dybjer [10] or Hofmann [14].

The categorical semantics of universes has previously been investigated by Mendler [21]. There he considers various universes which are all inductive-recursive definitions with $D = \mathsf{set}$. Our approach goes further since we consider inductive-recursive definitions with arbitrary D and characterize the collection of endofunctors which have initial algebras.

4.1 Strictly Positive Endofunctors on Type/D

First we shall show how to define an endofunctor $\mathbb{F}_\gamma$ in the category $\mathbf{Type}/D$. For arguments U, T with $U : \mathsf{set}$ the object part of this functor coincides with $\mathbb{F}^{\mathrm{U}}_\gamma$

and $\mathbb{F}^{\mathrm{T}}_{\gamma}$ in $\mathbf{IR_{elim}}$. In order to obtain a functor in $\mathbf{Type}/D$, we have to allow the argument U to be a type as well. So we have the new rules:

$$\frac{\gamma : \mathrm{OP}_D \qquad U : \mathsf{type} \qquad T : U \to D}{\mathbb{F}^{\mathrm{U}}_{\gamma}(U, D) : \mathsf{type}}$$

$$\frac{\gamma : \mathrm{OP}_D \qquad U : \mathsf{type} \qquad T : U \to D}{\mathbb{F}^{\mathrm{T}}_{\gamma}(U, D) : \mathbb{F}^{\mathrm{U}}_{\gamma}(U, T) \to D}$$

and the equality rules extended to $U : \mathsf{type}$.

We shall now define the arrow part $\mathbb{F}^{\to}_{\gamma}$ of the functor:

$$\begin{tikzcd}
U \arrow[rr, "f_0"] \arrow[dr, "T"'] \arrow[dr, phantom, shift left=6ex, "f_1"] & & U' \arrow[dl, "T'"] & \mapsto & \mathbb{F}^{\mathrm{U}}_{\gamma}(U,T) \arrow[rr, "\mathbb{F}^{\to}_{\gamma}(f_0, f_1)"] \arrow[dr, "\mathbb{F}^{\mathrm{T}}_{\gamma}(U,T)"'] \arrow[dr, phantom, shift left=6ex, "(x)\mathrm{ref}"] & & \mathbb{F}^{\mathrm{U}}_{\gamma}(U',T') \arrow[dl, "\mathbb{F}^{\mathrm{T}}_{\gamma}(U',T')"] \\
& D & & & & D &
\end{tikzcd}$$

Note that in this type-theoretic formalization $\mathbb{F}^{\to}_{\gamma}(f_0, f_1)$ has two main arguments: the arrow f_0 and the proof f_1 that the triangle commutes. The rules are:

$$\frac{\begin{array}{c} \gamma : \mathrm{OP}_D \\ \begin{array}{ll} U : \mathsf{set} & T : U \to D \\ U' : \mathsf{set} & T' : U' \to D \\ f_0 : U \to U' & f_1 : T =^{\mathrm{fun}}_{U \to D} T' \circ f_0 \end{array} \end{array}}{\mathbb{F}^{\to}_{\gamma}(f_0, f_1) : \mathbb{F}^{\mathrm{U}}_{\gamma}(U, T) \to \mathbb{F}^{\mathrm{U}}_{\gamma}(U', T')}$$

(We have suppressed the arguments U, T, U', T' of $\mathbb{F}^{\to}_{\gamma}$, which are implicitly contained in f_0, f_1.)

$$\begin{array}{rcl}
\mathbb{F}^{\to}_{\iota(\psi)}(f_0, f_1, *) & = & * \ , \\
\mathbb{F}^{\to}_{\sigma(A,\gamma)}(f_0, f_1, \langle a, b \rangle) & = & \langle a, \mathbb{F}^{\to}_{\gamma(a)}(f_0, f_1, b) \rangle \ , \\
\mathbb{F}^{\to}_{\delta(A,\gamma)}(f_0, f_1, \langle g, b \rangle) & = & \langle f_0 \circ g, \mathbb{F}^{\to}_{\gamma(T \circ g)}(f_0, f_1, b) \rangle \ .
\end{array}$$

Note that in the last equality we use that $T' \circ f_0 \circ g = T \circ g$ by f_1 and extensionality.

The commutativity of the right triangle in the diagram above is expressed by the following rule:

$$\frac{\begin{array}{c} \gamma : \mathrm{OP}_D \\ \begin{array}{ll} U : \mathsf{set} & T : U \to D \\ U' : \mathsf{set} & T' : U' \to D \\ f_0 : U \to U' & f_1 : T =^{\mathrm{fun}}_{U \to D} T' \circ f_0 \end{array} \end{array}}{\mathbb{F}^{\mathrm{T}}_{\gamma}(U', T') \circ \mathbb{F}^{\to}_{\gamma}(f_0, f_1) = \mathbb{F}^{\mathrm{T}}_{\gamma}(U, T) : \mathbb{F}^{\mathrm{U}}_{\gamma}(U, T) \to D}$$

(In extensional type theory the proof object of an equality type is irrelevant, since it is equal to ref. Therefore, when stating rules which generate elements of equality types, we will not introduce a new constant which generates a proof object, but instead write the conclusion of such a rule in the form of a judgement $r = s : A$, as in the rule above).

Further we have rules expressing the functor laws:

$$\begin{array}{rcl}
\mathbb{F}^{\to}_{\gamma}(\mathrm{id}, (x)\mathrm{ref}) & = & \mathrm{id} : \mathbb{F}^{\mathrm{U}}_{\gamma}(U, T) \to \mathbb{F}^{\mathrm{U}}_{\gamma}(U, T) \quad , \\
\mathbb{F}^{\to}_{\gamma}(f_0 \circ f_1, (x)\mathrm{ref}) & = & \mathbb{F}^{\to}_{\gamma}(f_0, (x)\mathrm{ref}) \circ \mathbb{F}^{\to}_{\gamma}(f_1, (x)\mathrm{ref}) \ , \\
& & \quad : \mathbb{F}^{\mathrm{U}}_{\gamma}(U, T) \to \mathbb{F}^{\mathrm{U}}_{\gamma}(U'', T'') \ .
\end{array}$$

Definition 4.1.1 *The rules for* $\mathbb{F}^{\rightarrow}$ *are the rules above in this subsection. (They presuppose the formation/introduction rules for* OP *and for* U, T *and the rules of extensionality).*

Remark 4.1.2 *In the presence of elimination rules for* OP *(see below) the rules expressing the functor laws and the equality* $\mathbb{F}^{\mathrm{T}}_{\gamma}(U', T') \circ \mathbb{F}^{\rightarrow}_{\gamma}(f_0, f_1) = \mathbb{F}^{\mathrm{T}}_{\gamma}(U, T)$ *can be proved by induction on* γ *and therefore be omitted in the formal theory.*

The object part of the functors $\mathbb{F}_{\gamma}$ refers to the argument U only strictly positively, but to T applied to these arguments both positively and negatively. This motivates part (b) of the following definition:

Definition 4.1.3 *(a) Let for* $\gamma : \mathrm{OP}_D$ *be* $\mathbb{F}_{\gamma}$ *the endofunctor on* $\mathbf{Type}/D$ *(with respect to rules* R *which contain the rules introduced in this section and those presupposed by it) with*

- *object part* $\mathbb{F}_{\gamma}(\langle U, T\rangle) := \langle \mathbb{F}^{\mathrm{U}}_{\gamma}(U, T), \mathbb{F}^{\mathrm{T}}_{\gamma}(U, T)\rangle$ *and*
- *arrow part* $\mathbb{F}_{\gamma}(\langle h_0, h_1\rangle) := \langle \mathbb{F}^{\rightarrow}_{\gamma}(h_0, h_1), (x)\mathrm{ref}\rangle$.

(b) The strictly positive endofunctors on $\mathbf{Type}/D$ *are the functors* $\mathbb{F}_{\gamma}$ *for* $\gamma : \mathrm{OP}_D$.

We can give the following names to strictly positive endofunctors on $\mathbf{Type}/D$:

- $\mathbb{F}_{\iota(\psi)}$ is the "constant functor", the result of which does not depend on the arguments.
- $\mathbb{F}_{\sigma(A,\gamma)}$ is the "disjoint union of functors": the first component of the object part is a disjoint union of the first components of the object parts of $\mathbb{F}_{\gamma(a)}$ $(a : A)$, and the other parts are defined accordingly.
- $\mathbb{F}_{\delta(A,\gamma)}$ is the "dependent disjoint union of functors": the first component of the object part is the disjoint union of the first components of the object parts of $\mathbb{F}_{\gamma(T\circ f)}$ for $f : A \rightarrow U$, referring to the arguments of the functor, and the other parts are again defined accordingly.

The introduction rules for U_{γ} and equality rules for T_{γ} express that with

$$\mathrm{eq}_{\gamma} := (x)\mathrm{ref} : \mathrm{T}_{\gamma} \circ \mathrm{intro}_{\gamma} =^{\mathrm{fun}}_{\mathbb{F}^{\mathrm{U}}_{\gamma}(\mathrm{U}_{\gamma}, \mathrm{T}_{\gamma}) \rightarrow D} \mathbb{F}^{\mathrm{T}}_{\gamma}(\mathrm{U}_{\gamma}, \mathrm{T}_{\gamma})$$

$\langle \mathrm{U}_{\gamma}, \mathrm{T}_{\gamma}, \mathrm{intro}_{\gamma}, \mathrm{eq}_{\gamma}\rangle$ is an $\mathbb{F}_{\gamma}$-algebra:

Definition 4.1.4 *An* $\mathbb{F}_{\gamma}$*-algebra is a quadruple* $\langle U, T, f_0, f_1\rangle$*, s.t.*

$$\begin{array}{lllllll} U & : & \mathsf{type} \ , & & T & : & U \rightarrow D \ , \\ f_0 & : & \mathbb{F}^{\mathrm{U}}_{\gamma}(U, T) \rightarrow U \ , & & f_1 & : & T \circ f_0 =^{\mathrm{fun}}_{\mathbb{F}^{\mathrm{U}}_{\gamma}(U,T) \rightarrow D} \mathbb{F}^{\mathrm{T}}_{\gamma}(U, T) \ , \end{array}$$

as expressed by the diagram

$$\begin{tikzcd} \mathbb{F}^{\mathrm{U}}_{\gamma}(U,T) \arrow[rr, "f_0"] \arrow[dr, "\mathbb{F}^{\mathrm{T}}_{\gamma}(U,T)"'] & {} \arrow[d, phantom, "f_1"] & U \arrow[dl, "T"] \\ & D & \end{tikzcd}$$

In the following we will define the rules expressing that $\langle \mathrm{U}_{\gamma}, \mathrm{T}_{\gamma}, \mathrm{intro}_{\gamma}, \mathrm{eq}_{\gamma}\rangle$ is an initial algebra and show that these rules are extensionally equivalent to the standard elimination and equality rules for U_{γ} and T_{γ}.

4.2 Rules for Initial Algebras in Slice Categories

We presuppose in this subsection the rules of extensionality, formation/introduction rules for OP and for U, T (which express that $\langle \mathrm{U}_\gamma, \mathrm{T}_\gamma, \mathrm{intro}_\gamma, \mathrm{eq}_\gamma \rangle$ is an $\mathbb{F}_\gamma$-algebra), and the rules for $\mathbb{F}^{\rightarrow}$.

Initiality of $\langle \mathrm{U}_\gamma, \mathrm{T}_\gamma, \mathrm{intro}_\gamma, \mathrm{eq}_\gamma \rangle$ means that for any other $\mathbb{F}_\gamma$-algebra $\langle U', T', f_0, f_1 \rangle$ there is a unique mediating arrow $\langle h_0, h_1 \rangle$, such that the following diagram commutes:

$$\begin{tikzcd}
\mathbb{F}^{\mathrm{U}}_\gamma(\mathrm{U}_\gamma, \mathrm{T}_\gamma) \arrow[rr, "\mathrm{intro}_\gamma"] \arrow[dr, "\mathbb{F}^{\mathrm{T}}_\gamma(\mathrm{U}_\gamma, \mathrm{T}_\gamma)"] \arrow[dd, dashed, "\mathbb{F}^{\rightarrow}_\gamma(h_0, h_1)"'] & \mathrm{eq}_\gamma & \mathrm{U}_\gamma \arrow[dl, "\mathrm{T}_\gamma"'] \arrow[dd, dashed, "h_0"] \\
(x)\mathrm{ref} & D & h_1 \\
\mathbb{F}^{\mathrm{U}}_\gamma(U', T') \arrow[rr, "f_0"'] \arrow[ur, "\mathbb{F}^{\mathrm{T}}_\gamma(U', T')"] & f_1 & U' \arrow[ul, "T'"']
\end{tikzcd} \qquad (*)$$

The rules are (under additional assumption $\gamma : \mathrm{OP}_D$)

$$\frac{\begin{array}{ll} U' : \mathsf{type} & T' : U' \to D \\ f_0 : \mathbb{F}^{\mathrm{U}}_\gamma(U', T') \to U' & f_1 : T' \circ f_0 =^{\mathrm{fun}}_{\mathbb{F}^{\mathrm{U}}_\gamma(U',T') \to D} \mathbb{F}^{\mathrm{T}}_\gamma(U', T') \end{array}}{\mathrm{initmap}_\gamma(U', T', f_0, f_1) : \mathrm{U}_\gamma \to U'}$$

and, under the assumptions of the last rule,

$$T' \circ \mathrm{initmap}_\gamma(U', T', f_0, f_1) = \mathrm{T}_\gamma : \mathrm{U}_\gamma \to D$$

$$\begin{array}{c} \mathrm{initmap}_\gamma(U', T', f_0, f_1) \circ \mathrm{intro}_\gamma = f_0 \circ \mathbb{F}^{\rightarrow}_\gamma(\mathrm{initmap}_\gamma(U', T', f_0, f_1), (x)\mathrm{ref}) \\ : \mathbb{F}^{\mathrm{U}}_\gamma(\mathrm{U}_\gamma, \mathrm{T}_\gamma) \to U' \end{array}$$

$$\frac{\begin{array}{ll} h'_0 : \mathrm{U}_\gamma \to U' & h'_1 : T' \circ h'_0 =^{\mathrm{fun}}_{\mathrm{U}_\gamma \to D} \mathrm{T}_\gamma \\ \multicolumn{2}{c}{q : h'_0 \circ \mathrm{intro}_\gamma =^{\mathrm{fun}}_{\mathbb{F}^{\mathrm{U}}_\gamma(\mathrm{U}_\gamma, \mathrm{T}_\gamma) \to U'} f_0 \circ \mathbb{F}^{\rightarrow}_\gamma(h'_0, h'_1)} \end{array}}{\mathrm{initmap}_\gamma(U', T', f_0, f_1) = h'_0 : \mathrm{U}_\gamma \to U'}$$

Definition 4.2.1 *The theory* $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{init}}$ *is the extension of the logical framework by the formation/introduction rules for* OP *and for* U, T*, the rules of extensionality, the rules for* $\mathbb{F}^{\rightarrow}$ *and the rules mentioned in this subsection.*

4.3 Elimination Rules for OP

In Subsection 4.4 we will show that the elimination rules of U, T are equivalent to the rules for the same sets as an initial algebra. This will be shown by induction on $\gamma : \mathrm{OP}_D$ and we need therefore to add elimination and equality rules for OP.

Definition 4.3.1 *The elimination and equality rules for* OP *are the following:*

$$\frac{\begin{array}{l}\gamma_0 : \mathrm{OP}_D \\ \gamma : \mathrm{OP}_D \Rightarrow E[\gamma] : \mathsf{type} \\ a : (\psi : D) \to E[\iota(\psi)] \\ b : (A : \mathsf{stype}, \gamma : A \to \mathrm{OP}_D, f : (x : A) \to E[\gamma(x)]) \to E[\sigma(A, \gamma)] \\ c : (A : \mathsf{stype}, \gamma : (A \to D) \to \mathrm{OP}_D, f : (x : A \to D) \to E[\gamma(x)]) \\ \quad \to E[\delta(A, \gamma)]\end{array}}{\mathrm{R}^{\mathrm{OP}}_{D,E}(\gamma_0, a, b, c) : E[\gamma]}$$

$$\begin{array}{rcl}\mathrm{R}^{\mathrm{OP}}_{D,E}(\iota(\psi), a, b, c) & = & a(\psi) \ , \\ \mathrm{R}^{\mathrm{OP}}_{D,E}(\sigma(A, \gamma), a, b, c) & = & b(A, \gamma, (y)\mathrm{R}^{\mathrm{OP}}_{D,E}(\gamma(y), a, b, c)) \ , \\ \mathrm{R}^{\mathrm{OP}}_{D,E}(\delta(A, \gamma), a, b, c) & = & c(A, \gamma, (y)\mathrm{R}^{\mathrm{OP}}_{D,E}(\gamma(y), a, b, c)) \ .\end{array}$$

We call these rules $\mathbf{OP_{elim}}$. *They presuppose the formation/introduction rules for* OP.

4.4 Equivalence of the Elimination Principle and the Existence of Initial Algebras

We shall show that the two theories $\mathbf{IR^{ext}_{elim}}$ and $\mathbf{IR^{ext}_{init}}$ are equivalent under the assumption $\mathbf{OP_{elim}}$ by interpreting them in each other. See the diagram at the end of Sect. 5.3, p. 30 for a summary of the relationships.

Theorem 4.4.1 $\mathbf{IR^{ext}_{init}}$ *can be interpreted in* $\mathbf{IR^{ext}_{elim}} + \mathbf{OP_{elim}}$.

Remark 4.4.2 *More precisely, Theorem 4.4.1 means that we can translate each symbol in the language of* $\mathbf{IR^{ext}_{init}}$ *to a term in the language of* $\mathbf{IR^{ext}_{elim}} + \mathbf{OP_{elim}}$*, such that each translated rule in* $\mathbf{IR^{ext}_{init}}$ *is provable in* $\mathbf{IR^{ext}_{elim}} + \mathbf{OP_{elim}}$*, that is, if the translated premises of the rule are provable so is the translated conclusion.*

This translation can be extended by additional symbols and rules.

Proof: We work in $\mathbf{IR^{ext}_{elim}}$ extended with OP-elimination and construct the family of functors $\mathbb{F}_\gamma$ and initial algebras $\langle \mathrm{U}_\gamma, \mathrm{T}_\gamma, \mathrm{intro}_\gamma, \mathrm{eq}_\gamma \rangle$ for $\gamma : \mathrm{OP}_D$. First the extensions of $\mathbb{F}^{\mathrm{U}}_\gamma$ and $\mathbb{F}^{\mathrm{T}}_\gamma$ to $U : \mathsf{type}$ and $\mathbb{F}^{\to}_\gamma$ can be defined by straightforward induction on γ such that the rules for $\mathbb{F}^{\mathrm{U}}$, $\mathbb{F}^{\mathrm{T}}$ and $\mathbb{F}^{\to}$ hold.

We are going to show that $\langle \mathrm{U}_\gamma, \mathrm{T}_\gamma, \mathrm{intro}_\gamma, \mathrm{eq}_\gamma \rangle$ is an initial $\mathbb{F}_\gamma$-algebra for $\gamma : \mathrm{OP}_D$. So let $\gamma : \mathrm{OP}_D$, $\langle U', T', f_0, f_1 \rangle$ be another $\mathbb{F}_\gamma$-algebra, and construct a unique mediating arrow $h = \langle h_0, h_1 \rangle$, such that diagram (∗) on page 17 commutes. To this end we shall use the elimination rule for U_γ with

$$\mathrm{E}[u] := (u' : U') \times (T'(u') =_D \mathrm{T}_\gamma(u))$$

for $u : \mathrm{U}_\gamma$. To this end define locally by induction on $\gamma' : \mathrm{OP}_D$:

$$\begin{array}{rcl}\multicolumn{3}{l}{\mathrm{k}_{\gamma'} : ((u : \mathbb{F}^{\mathrm{U}}_{\gamma'}(\mathrm{U}_\gamma, \mathrm{T}_\gamma)) \times \mathbb{F}^{\mathrm{IH}}_{\gamma'}(\mathrm{U}_\gamma, \mathrm{T}_\gamma, \mathrm{E}, u)) \to \mathbb{F}^{\mathrm{U}}_{\gamma'}(U', T') \ ,} \\ \mathrm{k}_{\iota(\psi)}(\langle *, * \rangle) & = & * \ , \\ \mathrm{k}_{\sigma(A,\gamma')}(\langle \langle a, b \rangle, c \rangle) & = & \langle a, \mathrm{k}_{\gamma'(a)}(\langle b, c \rangle) \rangle \ , \\ \mathrm{k}_{\delta(A,\gamma')}(\langle \langle f', b \rangle, \langle g', c \rangle \rangle) & = & \langle \pi_0 \circ g', \mathrm{k}_{\gamma'(\mathrm{T}_\gamma \circ f')}(\langle b, c \rangle) \rangle \ .\end{array}$$

In the last equality we use that $T' \circ \pi_0 \circ g' = \mathrm{T}_\gamma \circ f'$ by extensionality and the equality proof $\pi_1 \circ g'$. Moreover, we can show, by induction on γ', that $\mathrm{k}_{\gamma'}$ has the property that both triangles in the following diagram commute ($h'_0 : \mathrm{U}_\gamma \to U'$, $h'_1 : T' \circ h'_0 =^{\mathrm{fun}}_{\mathrm{U}_\gamma \to U'} \mathrm{T}_\gamma$):

$$\begin{tikzcd}
 & \mathbb{F}^{\mathrm{U}}_{\gamma'}(\mathrm{U}_\gamma, \mathrm{T}_\gamma) \arrow[dl, "\langle \mathrm{id}, \mathbb{F}^{\mathrm{map}}_{\gamma'}(\mathrm{U}_\gamma, \mathrm{T}_\gamma, \mathrm{E}, \langle h_0', h_1'\rangle^{\mathrm{fun}})\rangle^{\mathrm{fun}}"'] \arrow[d, "\mathbb{F}^{\rightarrow}_{\gamma'}(h_0', h_1')"] \\
(u : \mathbb{F}^{\mathrm{U}}_{\gamma'}(\mathrm{U}_\gamma, \mathrm{T}_\gamma)) \times \mathbb{F}^{\mathrm{IH}}_{\gamma'}(\mathrm{U}_\gamma, \mathrm{T}_\gamma, \mathrm{E}, u) \arrow[r, "\mathrm{k}_{\gamma'}"'] \arrow[dr, "\mathbb{F}^{\mathrm{T}}_{\gamma'}(U_\gamma, T_\gamma) \circ \pi_0"'] & \mathbb{F}^{\mathrm{U}}_{\gamma'}(U', T') \arrow[d, "\mathbb{F}^{\mathrm{T}}_{\gamma'}(U', T')"] \\
 & D
\end{tikzcd}$$

Now let

$$\begin{array}{rcl} g & = & (u, v)\langle f_0(\mathrm{k}_\gamma(\langle u, v\rangle)), \mathrm{ref}\rangle \\ & : & (u : \mathbb{F}^{\mathrm{U}}_{\gamma}(\mathrm{U}_\gamma, \mathrm{T}_\gamma), \mathbb{F}^{\mathrm{IH}}_{\gamma}(\mathrm{U}_\gamma, \mathrm{T}_\gamma, \mathrm{E}, u)) \to \mathrm{E}[\mathrm{intro}_\gamma(u)] \ . \end{array}$$

ref has correct type since

$$\begin{array}{rcl} \mathrm{T}_\gamma(\mathrm{intro}_\gamma(u)) = \mathbb{F}^{\mathrm{T}}_{\gamma}(\mathrm{U}_\gamma, \mathrm{T}_\gamma, u) & = & \mathbb{F}^{\mathrm{T}}_{\gamma}(U', T', \mathrm{k}_\gamma(\langle u, v\rangle)) \\ & \overset{\text{diagram } (*)}{=} & T'(f_0(\mathrm{k}_\gamma(\langle u, v\rangle))) \ . \end{array}$$

Now we can define the two components of the mediating arrow by

$$\begin{array}{rcl} h_0 & := & \pi_0 \circ \mathrm{R}_{\gamma,\mathrm{E}}(g) : \mathrm{U}_\gamma \to U' \ , \\ h_1 & := & \pi_1 \circ \mathrm{R}_{\gamma,\mathrm{E}}(g) : T' \circ h_0 =^{\mathrm{fun}}_{\mathrm{U}_\gamma \to D} \mathrm{T}_\gamma \ . \end{array}$$

To show the commutativity of the outer square in the initial algebra diagram we use the equality rule for R_γ:

$$\begin{array}{rcl} h_0(\mathrm{intro}_\gamma(u)) & = & \pi_0(\mathrm{R}_{\gamma,\mathrm{E}}(g, \mathrm{intro}_\gamma(u))) \\ & = & \pi_0(g(u, \mathbb{F}^{\mathrm{map}}_{\gamma}(\mathrm{U}_\gamma, \mathrm{T}_\gamma, \mathrm{E}, \mathrm{R}_{\gamma,\mathrm{E}}(g), u))) \\ & = & f_0(\mathrm{k}_\gamma(\langle u, \mathbb{F}^{\mathrm{map}}_{\gamma}(\mathrm{U}_\gamma, \mathrm{T}_\gamma, \mathrm{E}, \langle h_0, h_1\rangle^{\mathrm{fun}}, u)\rangle)) \\ & = & f_0(\mathbb{F}^{\rightarrow}_{\gamma}(h_0, h_1, u)) : U' \ . \end{array}$$

Finally, to prove uniqueness of h we assume that we have another mediating arrow $\langle h_0', h_1'\rangle$, that is we have $h_0' : \mathrm{U}_\gamma \to U'$, $h_1' : T' \circ h_0' =^{\mathrm{fun}}_{\mathrm{U}_\gamma \to D} \mathrm{T}_\gamma$, such that $h_0' \circ \mathrm{intro}_\gamma = f_0 \circ \mathbb{F}^{\rightarrow}_{\gamma}(h_0', h_1')$.

Let for $u : \mathrm{U}_\gamma$ $\mathrm{E}'[u] := (h_0(u) =_{U'} h_0'(u))$. By induction γ' we can prove for all $u : \mathbb{F}^{\mathrm{U}}_{\gamma'}(\mathrm{U}_\gamma, \mathrm{T}_\gamma)$ and $v : \mathbb{F}^{\mathrm{IH}}_{\gamma'}(\mathrm{U}_\gamma, \mathrm{T}_\gamma, \mathrm{E}', u)$

$$\mathbb{F}^{\rightarrow}_{\gamma'}(h_0, h_1, u) =_{\mathbb{F}^{\mathrm{U}}_{\gamma'}(U', T')} \mathbb{F}^{\rightarrow}_{\gamma'}(h_0', h_1', u) \ .$$

So if $u : \mathbb{F}^{\mathrm{U}}_{\gamma}(\mathrm{U}_\gamma, \mathrm{T}_\gamma)$ and $\mathbb{F}^{\mathrm{IH}}_{\gamma}(\mathrm{U}_\gamma, \mathrm{T}_\gamma, \mathrm{E}', u)$ it follows that

$$h_0(\mathrm{intro}_\gamma(u)) = f_0(\mathbb{F}^{\rightarrow}_{\gamma}(h_0, h_1, u)) = f_0(\mathbb{F}^{\rightarrow}_{\gamma}(h_0', h_1', u)) = h_0'(\mathrm{intro}_\gamma(u)) : U' \ .$$

Hence, by the induction principle it follows that $h_0(u) =_{U'} h_0'(u)$ for all $u : \mathrm{U}_\gamma$, and thus by extensionality $h_0 = h_0' : \mathrm{U}_\gamma \to U'$. Hence, $\langle h_0, h_1\rangle = \langle h_0', h_1'\rangle$ as arrows in the slice category.

Theorem 4.4.3 $\mathbf{IR_{elim}}$ *can be interpreted in* $\mathbf{IR^{ext}_{init}} + \mathbf{OP_{elim}}$.

Proof: We shall work in $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{init}}$ extended with OP-elimination and show how to define the constants $\mathbb{F}^{\,\mathrm{IH}}, \mathbb{F}^{\,\mathrm{map}}$, and R, which are specific to $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{elim}}$, so that their equality rules can be verified.

First, we can define in a straightforward way $\mathbb{F}^{\,\mathrm{IH}}_{\gamma}$ and $\mathbb{F}^{\,\mathrm{map}}_{\gamma}$ by OP-elimination and verify their equality rules.

We define now R_γ (the recursion operator for the inductive-recursive definition of U_γ and T_γ) and verify the corresponding equality rule. So let $E[u]$ be a type for $u : \mathrm{U}_\gamma$ and assume

$$g : (u : \mathbb{F}^{\,\mathrm{U}}_{\gamma}(\mathrm{U}_\gamma, \mathrm{T}_\gamma), \mathbb{F}^{\,\mathrm{IH}}_{\gamma}(\mathrm{U}_\gamma, \mathrm{T}_\gamma, E, u)) \to E[\mathrm{intro}_\gamma(u)] \ \ .$$

We will define

$$\mathrm{R}_{\gamma,E}(g) := \pi_1 \circ h_0 : (u : \mathrm{U}_\gamma) \to E[\pi_0(h_0(u))] \ \ ,$$

and verify that $\pi_0(h_0(u)) = u$, where $\langle h_0, h_1\rangle$ is the unique mediating morphism in the following initial $\mathbb{F}_{\gamma}$-algebra diagram:

$$\begin{tikzcd}
\mathbb{F}^{\,\mathrm{U}}_{\gamma}(\mathrm{U}_\gamma, \mathrm{T}_\gamma) \arrow[rr, "\mathrm{intro}_\gamma"] \arrow[dr, "\mathbb{F}^{\,\mathrm{T}}_{\gamma}(\mathrm{U}_\gamma, \mathrm{T}_\gamma)"] \arrow[dd, dashed, "\mathbb{F}^{\,\to}_{\gamma}(h_0, h_1)"'] & \arrow[d, phantom, "\mathrm{eq}_\gamma"] & \mathrm{U}_\gamma \arrow[dl, "\mathrm{T}_\gamma"'] \arrow[dd, dashed, "h_0"] \\
\arrow[r, phantom, "(x)\mathrm{ref}"] & D & \arrow[r, phantom, "h_1"] & {} \\
\mathbb{F}^{\,\mathrm{U}}_{\gamma}(\mathrm{U}^E_\gamma, \mathrm{T}^E_\gamma) \arrow[ur, "\mathbb{F}^{\,\mathrm{T}}_{\gamma}(\mathrm{U}^E_\gamma, \mathrm{T}^E_\gamma)"] \arrow[r, "\cong", "\langle \mathbb{F}^{\,\to}_{\gamma}(\pi_0, (x)\mathrm{ref}), \mathrm{j}_\gamma\rangle^{\mathrm{fun}}"'] & (u : \mathbb{F}^{\,\mathrm{U}}_{\gamma}(\mathrm{U}_\gamma, \mathrm{T}_\gamma)) \times \mathbb{F}^{\,\mathrm{IH}}_{\gamma}(\mathrm{U}_\gamma, \mathrm{T}_\gamma, E, u) \arrow[u, "\mathbb{F}^{\,\mathrm{T}}_{\gamma}(\mathrm{U}_\gamma, \mathrm{T}_\gamma) \circ \pi_0"] \arrow[r, "\langle \mathrm{intro}_\gamma \circ \pi_0, g'\rangle^{\mathrm{fun}}"'] & \mathrm{U}^E_\gamma \arrow[ul, "T^E_\gamma"']
\end{tikzcd}$$

Here $g' : ((u : \mathbb{F}^{\,\mathrm{U}}_{\gamma}(\mathrm{U}_\gamma, \mathrm{T}_\gamma)) \times \mathbb{F}^{\,\mathrm{IH}}_{\gamma}(\mathrm{U}_\gamma, \mathrm{T}_\gamma, E, u)) \to E[\mathrm{intro}_\gamma(u)]$ is the uncurried version of g, $\mathrm{U}^E_\gamma := (x : \mathrm{U}_\gamma) \times E[x]$ and $\mathrm{T}^E_\gamma := \mathrm{T}_\gamma \circ \pi_0 : \mathrm{U}^E_\gamma \to D$. Furthermore $\mathrm{j}_{\gamma'}$ is defined by induction on $\gamma' : \mathrm{OP}_D$ as follows (this is a local definition):

$$\begin{array}{rcl}
\mathrm{j}_{\gamma'} & : & (u : \mathbb{F}^{\,\mathrm{U}}_{\gamma'}(\mathrm{U}^E_\gamma, \mathrm{T}^E_\gamma)) \to \mathbb{F}^{\,\mathrm{IH}}_{\gamma'}(\mathrm{U}_\gamma, \mathrm{T}_\gamma, E, \mathbb{F}^{\,\to}_{\gamma'}(\pi_0, (x)\mathrm{ref}, u)) \ \ , \\
\mathrm{j}_{\iota(\psi)}(*) & = & * \ \ , \\
\mathrm{j}_{\sigma(A,\gamma')}(\langle a, b\rangle) & = & \mathrm{j}_{\gamma'(a)}(b) \ \ , \\
\mathrm{j}_{\delta(A,\gamma')}(\langle f, b\rangle) & = & \langle \pi_1 \circ f, \mathrm{j}_{\gamma'(\mathrm{T}_\gamma \circ (\pi_0 \circ f))}(b)\rangle \ \ .
\end{array}$$

(One can show by induction on γ' that $\langle \mathbb{F}^{\,\to}_{\gamma'}(\pi_0, (x)\mathrm{ref}), \mathrm{j}_{\gamma'}\rangle^{\mathrm{fun}}$ is an isomorphism $\mathbb{F}^{\,\mathrm{U}}_{\gamma'}(\mathrm{U}^E_\gamma, \mathrm{T}^E_\gamma) \xrightarrow{\cong} (u : \mathbb{F}^{\,\mathrm{U}}_{\gamma'}(\mathrm{U}_\gamma, \mathrm{T}_\gamma)) \times \mathbb{F}^{\,\mathrm{IH}}_{\gamma'}(\mathrm{U}_\gamma, \mathrm{T}_\gamma, E, u)$, but this is not needed in the current proof.) The lower left triangle in the initial algebra diagram commutes, since $\mathbb{F}^{\,\to}_{\gamma}(\pi_0, (x)\mathrm{ref})$ is an arrow in the slice category and the lower right triangle commutes by $\mathrm{T}^E_\gamma = \mathrm{T}_\gamma \circ \pi_0$. Therefore it follows that the two triangles together form an $\mathbb{F}_{\gamma}$-algebra, and hence we can construct the unique mediating morphism $\langle h_0, h_1\rangle$.

We show $\pi_0 \circ h_0 =^{\mathrm{fun}}_{\mathrm{U}_\gamma \to \mathrm{U}_\gamma} \mathrm{id}$ and therefore $\mathrm{R}_{\gamma,E}(g) := \pi_1 \circ h_0 : (u : \mathrm{U}_\gamma) \to E[u]$. The following diagram commutes

$$\begin{tikzcd}
\mathbb{F}^{\mathrm{U}}_\gamma(\mathrm{U}_\gamma, \mathrm{T}_\gamma) \arrow[r, "\mathrm{intro}_\gamma"] \arrow[d, "\mathbb{F}^{\to}_\gamma(h_0, h_1)"] & \mathrm{U}_\gamma \arrow[d, "h_0"'] \\
\mathbb{F}^{\mathrm{U}}_\gamma(\mathrm{U}^E_\gamma, \mathrm{T}^E_\gamma) \arrow[r, "\langle \mathrm{intro}_\gamma \circ \pi_0, g' \rangle^{\mathrm{fun}} \circ \langle \mathbb{F}^{\to}_\gamma(\pi_0, (x)\mathrm{ref}), \mathrm{j}_\gamma \rangle^{\mathrm{fun}}"'] \arrow[d, "\mathbb{F}^{\to}_\gamma(\pi_0, (x)\mathrm{ref})"] & \mathrm{U}^E_\gamma \arrow[d, "\pi_0"'] \\
\mathbb{F}^{\mathrm{U}}_\gamma(\mathrm{U}_\gamma, \mathrm{T}_\gamma) \arrow[r, "\mathrm{intro}_\gamma"] & \mathrm{U}_\gamma
\end{tikzcd}$$

and all arrows are arrows in the slice category. So $\langle \pi_0 \circ h_0, (x)\mathrm{ref} \rangle^{\mathrm{fun}}$ and $\langle \mathrm{id}, (x)\mathrm{ref} \rangle^{\mathrm{fun}}$ are two arrows $\langle h_0', h_1' \rangle$ from $\langle \mathrm{U}_\gamma, \mathrm{T}_\gamma \rangle$ to itself in the slice category such that $h_0' \circ \mathrm{intro}_\gamma = \mathrm{intro}_\gamma \circ \mathbb{F}^{\to}_\gamma(h_0', h_1')$. Uniqueness of the arrows from an initial algebra implies now $\pi_0 \circ h_0 = \mathrm{id}$, the assertion.

Finally, we show that the equality rules hold with the above interpretation:

$$\begin{array}{ll}
\multicolumn{2}{l}{\mathrm{R}_{\gamma,E}(g, \mathrm{intro}_\gamma(u))} \\
= & \pi_1(h_0(\mathrm{intro}_\gamma(u))) \\
= & g'(\langle (\mathbb{F}^{\to}_\gamma(\pi_0, (x)\mathrm{ref}) \circ \mathbb{F}^{\to}_\gamma(h_0, h_1))(u), (\mathrm{j}_\gamma \circ \mathbb{F}^{\to}_\gamma(h_0, h_1))(u) \rangle) \\
= & g'(\langle \mathbb{F}^{\to}_\gamma(\underbrace{\pi_0 \circ h_0}_{=\mathrm{id}}, (x)\mathrm{ref}, u), \mathbb{F}^{\mathrm{map}}_\gamma(\mathrm{U}_\gamma, \mathrm{T}_\gamma, E, \pi_1 \circ h_0, u) \rangle) \\
= & g'(\langle u, \mathbb{F}^{\mathrm{map}}_\gamma(\mathrm{U}_\gamma, \mathrm{T}_\gamma, E, \mathrm{R}_{\gamma,E}(g), u) \rangle) \\
= & g(u, \mathbb{F}^{\mathrm{map}}_\gamma(\mathrm{U}_\gamma, \mathrm{T}_\gamma, E, \mathrm{R}_{\gamma,E}(g), u)) \ ,
\end{array}$$

where the third equality uses the commutativity of the following diagram (which can be proved by induction on $\gamma' : \mathrm{OP}_D$):

$$\begin{tikzcd}
u : \mathbb{F}^{\mathrm{U}}_{\gamma'}(\mathrm{U}_\gamma, \mathrm{T}_\gamma) \arrow[d, "\mathbb{F}^{\to}_{\gamma'}(h_0, h_1)"'] \arrow[dr, "\mathbb{F}^{\mathrm{map}}_{\gamma'}(\mathrm{U}_\gamma, \mathrm{T}_\gamma, E, \pi_1 \circ h_0)"] & \\
\mathbb{F}^{\mathrm{U}}_{\gamma'}(\mathrm{U}^E_\gamma, \mathrm{T}^E_\gamma) \arrow[r, "\mathrm{j}_{\gamma'}"'] & \mathbb{F}^{\mathrm{IH}}_{\gamma'}(\mathrm{U}_\gamma, \mathrm{T}_\gamma, E, u)
\end{tikzcd}$$

4.5 Conclusion

We have seen that in extensional type theory together with the elimination rules for OP, the principle of $\langle \mathrm{U}_\gamma, \mathrm{T}_\gamma, \mathrm{intro}_\gamma, \mathrm{eq}_\gamma \rangle$ being an initial algebra is equivalent to the elimination/equality rules for U_γ. In intensional type theory we cannot even express the principle of being an initial algebra, since the arrow part of the functors cannot be defined. However, because of the above mentioned equivalence, the principle of universe elimination can be described as a principle which can be formulated in intensional type theory and in the presence of extensionality expresses the initiality property.

5 Induction-recursion as a Reflection Principle

5.1 Background

When working in the slice category both U_γ and T_γ become part of the initial algebra and we break the pattern

inductively defined set	initial algebra
recursively defined function	initial arrow

The slice algebra approach is an abstraction of the set theoretic semantics of inductive-recursive definitions in terms of inductive definitions. It suggests the view that T_γ (and not only U_γ) is inductively generated.

From a type-theoretic point of view, however, it is unnatural to view inductive-recursive definitions as special cases of inductive ones. We shall therefore recall an alternative formalization of induction-recursion which maintains the distinctions in the table above. This formalization was previously presented in Dybjer and Setzer [12]. It expresses induction-recursion as a reflection principle: for any type D and any D-operation d of "arity" ϕ, there is a set $\mathrm{U}'_{\phi,d}$ which is closed under d and has decoding function $\mathrm{T}'_{\phi,d} : \mathrm{U}'_{\phi,d} \to D$ (we add an accent to U, T, intro, R in the current theory in order to distinguish them from the corresponding constants in $\mathbf{IR_{elim}}$).

Consider again the case of the constructor $\hat{\Sigma}$ for the first universe. We can express the fact that $\hat{\Sigma}$ reflects (inside U'_0) the set-operation Σ by the following diagram:

$$\begin{tikzcd}
(a : \mathrm{U}'_0) \times (\mathrm{T}'_0(a) \to \mathrm{U}'_0) \arrow[r, "\hat{\Sigma}"] \arrow[d, "\langle a, b\rangle \mapsto \langle \mathrm{T}'_0(a), \mathrm{T}'_0 \circ b\rangle"'] & \mathrm{U}'_0 \arrow[d, "\mathrm{T}'_0"] \\
(A : \mathsf{set}) \times (A \to \mathsf{set}) \arrow[r, "\Sigma"'] & \mathsf{set}
\end{tikzcd}$$

We have simply observed that the diagonal arrow in the diagram in Section 3.1 factors through Σ.

The general reflection principle is captured by the following commuting diagram:

$$\begin{tikzcd}
\mathrm{arg}_\phi(\mathrm{U}'_{\phi,d}, \mathrm{T}'_{\phi,d}) \arrow[r, "\mathrm{intro}'_{\phi,d}"] \arrow[d, "\mathrm{map}_\phi(\mathrm{U}'_{\phi,d}, \mathrm{T}'_{\phi,d})"'] & \mathrm{U}'_{\phi,d} \arrow[d, "\mathrm{T}'_{\phi,d}"] \\
\mathrm{Arg}_{D,\phi} \arrow[r, "d"'] & D
\end{tikzcd}$$

Here ϕ is an element of the type SP_D of D-arities. It encodes both the domain $\mathrm{Arg}_{D,\phi}$ of d, the domain $\mathrm{arg}_\phi(\mathrm{U}'_{\phi,d}, \mathrm{T}'_{\phi,d})$ of the constructor $\mathrm{intro}'_{\phi,d}$ which reflects d, and also the function $\mathrm{map}_\phi(\mathrm{U}'_{\phi,d}, \mathrm{T}'_{\phi,d})$ which decodes the arguments of $\mathrm{intro}'_{\phi,d}$.

Note that this relationship with initial algebras is different from the initial algebra diagram in the slice category $\mathbf{Type}/D$ discussed in the previous section. There U_γ, T_γ arose as the carrier of an initial algebra and universe elimination arose as the initial arrow.

In this section we shall show the equivalence between the two formulations. To this end we briefly summarize the formalization in Dybjer and Setzer [12] and refer the reader to that paper for more details and discussion.

5.2 An Alternative Formalization

The first step is to introduce a new type SP_D

$$\mathrm{SP}_D : \mathsf{type} \ ,$$

containing codes for arities of D-operations. (The elements of SP_D can also be viewed as codes for *strictly positive* "functors", hence the name.) SP_D has five associated operations (Again we write ϕ instead of D, ϕ in argument position. One exception is $\mathrm{Arg}_{D,\phi}$, where the equality rules refer to D.)

$$\frac{\phi : \mathrm{SP}_D}{\mathrm{Arg}_{D,\phi} : \mathsf{type}}$$

$$\frac{\phi : \mathrm{SP}_D \qquad U : \mathsf{set} \qquad T : U \to D}{\mathrm{arg}_\phi(U,T) : \mathsf{stype}}$$

$$\frac{\phi : \mathrm{SP}_D \qquad U : \mathsf{set} \qquad T : U \to D}{\mathrm{map}_\phi(U,T) : (\mathrm{arg}_\phi(U,T)) \to \mathrm{Arg}_{D,\phi}}$$

$$\frac{\begin{array}{ccc}\phi : \mathrm{SP}_D & U : \mathsf{set} & T : (x : U) \to D \\ x : U \Rightarrow E[x] : \mathsf{type} & & b : \mathrm{arg}_\phi(U,T)\end{array}}{\mathrm{IH}_{\phi,U,T,E}(b) : \mathsf{type}}$$

$$\frac{\begin{array}{ccc}\phi : \mathrm{SP}_D & U : \mathsf{set} & T : (x : U) \to D \\ x : U \Rightarrow E[x] : \mathsf{type} & & h : (x : U) \to E[x]\end{array}}{\mathrm{mapIH}_{\phi,U,T,E}(h) : (x : \mathrm{arg}_\phi(U,T)) \to \mathrm{IH}_{\phi,U,T,E}(x)}$$

We have the following introduction rules for SP:

$$\mathrm{nil} : \mathrm{SP}_D$$

$$\frac{A \ \mathsf{stype} \qquad \phi : A \to \mathrm{SP}_D}{\mathrm{nonind}(A,\phi) : \mathrm{SP}_D}$$

$$\frac{A \ \mathsf{stype} \qquad \phi : (A \to D) \to \mathrm{SP}_D}{\mathrm{ind}(A,\phi) : \mathrm{SP}_D}$$

$$\begin{array}{rcl}
\mathrm{Arg}_{D,\mathrm{nil}} & = & \mathbf{1} \ , \\
\mathrm{Arg}_{D,\mathrm{nonind}(A,\phi)} & = & (x : A) \times \mathrm{Arg}_{D,\phi(x)} \ , \\
\mathrm{Arg}_{D,\mathrm{ind}(A,\phi)} & = & (f : A \to D) \times \mathrm{Arg}_{D,\phi(f)} \ .
\end{array}$$

$$\begin{array}{rcl}
\mathrm{arg}_{\mathrm{nil}}(U,T) & = & \mathbf{1} \ , \\
\mathrm{arg}_{\mathrm{nonind}(A,\phi)}(U,T) & = & (x : A) \times (\mathrm{arg}_{\phi(x)}(U,T)) \ , \\
\mathrm{arg}_{\mathrm{ind}(A,\phi)}(U,T) & = & (f : A \to U) \times (\mathrm{arg}_{\phi(T \circ f)}(U,T)) \ .
\end{array}$$

$$\begin{array}{rcl}
\mathrm{map}_{\mathrm{nil}}(U,T,*) & = & * \ , \\
\mathrm{map}_{\mathrm{nonind}(A,\phi)}(U,T,\langle a,b\rangle) & = & \langle a, \mathrm{map}_{\phi(a)}(U,T,b)\rangle \ , \\
\mathrm{map}_{\mathrm{ind}(A,\phi)}(U,T,\langle f,b\rangle) & = & \langle T \circ f, \mathrm{map}_{\phi(T\circ f)}(U,T,b)\rangle \ .
\end{array}$$

$$\begin{array}{rcl}
\mathrm{IH}_{\mathrm{nil},U,T,E}(*) & = & \mathbf{1} \ , \\
\mathrm{IH}_{\sigma(A,\phi),U,T,E}(\langle a,b\rangle) & = & \mathrm{IH}_{\phi(a),U,T,E}(b) \ ,
\end{array}$$

$$\mathrm{IH}_{\delta(A,\phi),U,T,E}(\langle f,b\rangle) \quad = \quad ((y:A)\to E[f(y)])\times(\mathrm{IH}_{\phi(T\circ f),U,T,E}(b)) \ .$$

$$\begin{array}{rcl}\mathrm{mapIH}_{\mathrm{nil},U,T,E}(h,*) & = & * \ , \\ \mathrm{mapIH}_{\sigma(A,\phi),U,T,E}(h,\langle a,b\rangle) & = & \mathrm{mapIH}_{\phi(a),U,T,E}(h,b) \ , \\ \mathrm{mapIH}_{\delta(A,\phi),U,T,E}(h,\langle f,b\rangle) & = & \langle h\circ f, \mathrm{mapIH}_{\phi(T\circ f),U,T,E}(h,b)\rangle \ .\end{array}$$

We are now ready to give the formal rules for U' and T'. These rules have the common additional premises $\phi : \mathrm{SP}_D$ and $d : \mathrm{Arg}_{D,\phi} \to D$:

Formation rules:

$$\mathrm{U}'_{\phi,d} : \mathsf{set}$$

$$\mathrm{T}'_{\phi,d} : \mathrm{U}'_{\phi,d} \to D$$

Introduction rule:

$$\frac{a : \arg_\phi(\mathrm{U}'_{\phi,d}, \mathrm{T}'_{\phi,d})}{\mathrm{intro}'_{\phi,d}(a) : \mathrm{U}'_{\phi,d}}$$

Equality rule for T':

$$\frac{a : \arg_\phi(\mathrm{U}'_{\phi,d}, \mathrm{T}'_{\phi,d})}{\mathrm{T}'_{\phi,d}(\mathrm{intro}'_{\phi,d}(a)) = d(\mathrm{map}_\phi(\mathrm{U}'_{\phi,d}, \mathrm{T}'_{\phi,d}, a))}$$

Elimination rule:

$$\frac{e : (x : \arg_\phi(\mathrm{U}'_{\phi,d}, \mathrm{T}'_{\phi,d}), \mathrm{IH}_{\phi,\mathrm{U}'_{\phi,d},\mathrm{T}'_{\phi,d},E}(x)) \to (E[\mathrm{intro}'_{\phi,d}(x)])}{\mathrm{R}'_{\phi,d,E}(e) : (a : \mathrm{U}'_{\phi,d}) \to E[a]}$$

Equality rule:

$$\mathrm{R}'_{\phi,d,E}(e, \mathrm{intro}'_{\phi,d}(b)) = e(b, \mathrm{mapIH}_{\phi,\mathrm{U}'_{\phi,d},\mathrm{T}'_{\phi,d},E}(\mathrm{R}'_{\phi,d,E}(e), b)) \ .$$

Definition 5.2.1 *(a) The theory* $\mathbf{IR_{refl}}$ *consists of the rules above in this subsection.*

(b) $\mathbf{IR^{ext}_{refl}}$ *is the extension of* $\mathbf{IR_{refl}}$ *by the rules of extensionality.*

(c) The following are the elimination and equality rules for SP*, called* $\mathbf{SP_{elim}}$ *(they presuppose the formation and introduction rules for* SP*):*

$$\frac{\begin{array}{l}\phi_0 : \mathrm{SP}_D \\ \phi : \mathrm{SP}_D \Rightarrow E[\phi] : \mathsf{type} \\ a : E[\mathrm{nil}] \\ b : (A : \mathsf{stype}, \phi : A \to \mathrm{SP}_D, f : (x:A) \to E[\phi(x)]) \\ \quad \to E[\mathrm{nonind}(A,\phi)] \\ c : (A : \mathsf{stype}, \phi : (A \to D) \to \mathrm{SP}_D, f : (x : A \to D) \to E[\phi(x)]) \\ \quad \to E[\mathrm{ind}(A,\phi)]\end{array}}{\mathrm{R}^{\mathsf{SP}}_{D,E}(\phi_0, a, b, c) : E[\phi]}$$

$$\begin{array}{rcl}\mathrm{R}^{\mathsf{SP}}_{D,E}(\mathrm{nil},a,b,c) & = & a \ , \\ \mathrm{R}^{\mathsf{SP}}_{D,E}(\mathrm{nonind}(A,\phi),a,b,c) & = & b(A,\phi,(y)\mathrm{R}^{\mathsf{SP}}_{D,E}(\phi(y),a,b,c)) \ , \\ \mathrm{R}^{\mathsf{SP}}_{D,E}(\mathrm{ind}(A,\phi),a,b,c) & = & c(A,\phi,(y)\mathrm{R}^{\mathsf{SP}}_{D,E}(\phi(y),a,b,c)) \ .\end{array}$$

5.3 The Correspondence between $\mathbf{IR_{elim}}$ and $\mathbf{IR_{refl}}$

We are going to analyze the correspondence between $\mathbf{IR_{elim}}$ and $\mathbf{IR_{refl}}$. See the diagram at the end of this section, p. 30 for a summary of the relationships.

First we show that, in the type theory containing rules of both theories, $\mathbf{OP_{elim}}$, $\mathbf{SP_{elim}}$ and extensionality laws, there is a 1-1 correspondence between objects $\gamma : \mathrm{OP}_D$ and pairs $(\phi : \mathrm{SP}_D, d : \mathrm{Arg}_{D,\phi} \to D)$ and that we obtain translations between the associated operations. Then we interpret the theory $\mathbf{IR_{elim}}$ in $\mathbf{IR_{refl}}$ and $\mathbf{IR_{elim}} + \mathbf{OP_{elim}}$ in $\mathbf{IR_{refl}} + \mathbf{SP_{elim}}$ (Theorem 5.3.3), and in a last step interpret, using one additional rule, $\mathbf{IR^{ext}_{refl}} + \mathbf{SP_{elim}}$ in $\mathbf{IR^{ext}_{elim}} + \mathbf{OP_{elim}}$ (Theorem 5.3.9). Using the results of [12] the consistency of all theories considered in this article follow (Corollary 5.3.4, Remark 5.3.6).

We start with translations between OP_D and $(\phi : \mathrm{SP}_D) \times (\mathrm{Arg}_\phi \to D)$:

Definition 5.3.1 *We define in a type theory containing formation/introduction/elimination/equality rules for* OP *and* SP

$$\begin{array}{rcl} \mathrm{sp}_D & : & \mathrm{OP}_D \to \mathrm{SP}_D \ , \\ \mathrm{d}_D & : & (\gamma : \mathrm{OP}_D, \mathrm{Arg}_{\mathrm{sp}(\gamma)}) \to D \ , \\ \mathrm{op}_D & : & (\phi : \mathrm{SP}_D, d : \mathrm{Arg}_\phi \to D) \to \mathrm{OP}_D \ , \end{array}$$

by (we omit the index D in sp, d, op*)*

$$\begin{array}{rcl} \mathrm{sp}(\iota(\psi)) & = & \mathrm{nil} \ , \\ \mathrm{sp}(\sigma(A,\gamma)) & = & \mathrm{nonind}(A, \mathrm{sp} \circ \gamma) \ , \\ \mathrm{sp}(\delta(A,\gamma)) & = & \mathrm{ind}(A, \mathrm{sp} \circ \gamma) \ , \end{array}$$

$$\begin{array}{rcl} \mathrm{d}(\iota(\psi), *) & = & \psi \ , \\ \mathrm{d}(\sigma(A,\gamma), \langle a, b\rangle) & = & \mathrm{d}(\gamma(a), b) \ , \\ \mathrm{d}(\delta(A,\gamma), \langle f, b\rangle) & = & \mathrm{d}(\gamma(f), b) \ . \end{array}$$

$$\begin{array}{rcl} \mathrm{op}(\mathrm{nil}, d) & = & \iota(d(*)) \ , \\ \mathrm{op}(\mathrm{nonind}(A,\gamma), d) & = & \sigma(A, (a)\mathrm{op}(\gamma(a), (b)d(\langle a, b\rangle))) \ , \\ \mathrm{op}(\mathrm{ind}(A,\gamma), d) & = & \delta(A, (f)\mathrm{op}(\gamma(f), (b)d(\langle f, b\rangle))) \ . \end{array}$$

Theorem 5.3.2 *Assume a type theory including formation/introduction/elimination/equality rules for* OP *and* SP*, rules for* $\mathbb{F}^{\mathrm{U}}$, $\mathbb{F}^{\mathrm{T}}$, $\mathbb{F}^{\mathrm{IH}}$, $\mathbb{F}^{\mathrm{mapIH}}$, Arg*,* arg*,* map*,* IH*,* mapIH *and extensional equality. Then, with variables chosen of appropriate type, the following holds (we omit in* op, sp, d *the parameter D):*

$$\begin{array}{rcl} \mathbb{F}^{\mathrm{U}}_\gamma(U,T) & = & \mathrm{arg}_{\mathrm{sp}(\gamma)}(U,T) \ , \\ \mathbb{F}^{\mathrm{T}}_\gamma(U,T,a) & = & \mathrm{d}(\gamma, \mathrm{map}_{\mathrm{sp}(\gamma)}(U,T,a)) \ , \\ \mathbb{F}^{\mathrm{IH}}_\gamma(U,T,E,a) & = & \mathrm{IH}_{\mathrm{sp}(\gamma),U,T,E}(a) \ , \\ \mathbb{F}^{\mathrm{map}}_\gamma(U,T,E,h,a) & = & \mathrm{mapIH}_{\mathrm{sp}(\gamma),U,T,E}(h,a) \ ; \end{array}$$

$$\begin{array}{rcl} \mathrm{arg}_\phi(U,T) & = & \mathbb{F}^{\mathrm{U}}_{\mathrm{op}(\phi,d)}(U,T) \ , \\ d(\mathrm{map}_\phi(U,T,a)) & = & \mathbb{F}^{\mathrm{T}}_{\mathrm{op}(\phi,d)}(U,T,a) \ , \\ \mathrm{IH}_{\phi,U,T,E}(u) & = & \mathbb{F}^{\mathrm{IH}}_{\mathrm{op}(\phi,d)}(U,T,E,u) \ , \\ \mathrm{mapIH}_{\phi,U,T,E}(h,a) & = & \mathbb{F}^{\mathrm{map}}_{\mathrm{op}(\phi,d)}(U,T,E,h,a) \ , \\ \mathrm{Arg}_{D,\phi} & = & \mathbb{F}^{\mathrm{U}}_{\mathrm{op}(\phi,d)}(D, \mathrm{id}) \ , \\ \mathrm{map}_\phi & = & \mathbb{F}^{\to}_{\mathrm{op}(\phi,d)}(\mathrm{T}'_{\phi,d}, (x)\mathrm{ref}) \ ; \end{array}$$

$$\begin{array}{rcl} \mathrm{op}(\mathrm{sp}(\gamma), \mathrm{d}(\gamma)) & = & \gamma \ , \\ \mathrm{sp}(\mathrm{op}(\phi,d)) & = & \phi \ , \\ \mathrm{d}(\mathrm{op}(\phi,d)) & = & d \ . \end{array}$$

The following diagram summarizes the correspondence ($\gamma = \mathrm{op}(\phi, d)$):

$$\begin{tikzcd}
\arg_\phi(\mathrm{U}'_{\phi,d}, \mathrm{T}'_{\phi,d}) = \mathbb{F}^{\mathrm{U}}_\gamma(\mathrm{U}'_{\phi,d}, \mathrm{T}'_{\phi,d}) \arrow[r, "\mathrm{intro}'_{\phi,d}"] \arrow[d, "\mathrm{map}_\phi(\mathrm{U}'_{\phi,d}, \mathrm{T}'_{\phi,d}) = \mathbb{F}^{\rightarrow}_\gamma(\mathrm{T}'_{\phi,d}, (x)\mathrm{ref})"'] \arrow[dr, "\mathbb{F}^{T}_\gamma(U'_{\phi,d}, T'_{\phi,d})"] & \mathrm{U}'_{\phi,d} \arrow[d, "\mathrm{T}'_{\phi,d}"] \\
\mathrm{Arg}_{D,\phi} = \mathbb{F}^{\mathrm{U}}_\gamma(D, \mathrm{id}) \arrow[r, "d"'] & D
\end{tikzcd}$$

Proof of Theorem 5.3.2: Straightforward induction on OP and SP.

Theorem 5.3.3 *(a)* $\mathbf{IR_{elim}}$ *can be interpreted in* $\mathbf{IR_{refl}}$.

(b) $\mathbf{IR_{elim}} + \mathbf{OP_{elim}}$ *can be interpreted in* $\mathbf{IR_{refl}} + \mathbf{SP_{elim}}$.

Note that in Theorem 5.3.3 extensionality is not needed: the constants of $\mathbf{IR_{elim}}$ can all be defined in $\mathbf{IR_{refl}}$ in such a way that all the equality rules of $\mathbf{IR_{elim}}$ are translated into definitional equalities in $\mathbf{IR_{refl}}$.

Corollary 5.3.4 *(a)* $\mathbf{IR^{ext}_{refl}} + \mathbf{SP_{elim}}$, $\mathbf{IR^{ext}_{elim}} + \mathbf{OP_{elim}}$ *and* $\mathbf{IR^{ext}_{init}} + \mathbf{OP_{elim}}$ *are consistent.*

(b) The same holds with subtheories $\mathbf{IR_{refl}}$, $\mathbf{IR^{ext}_{refl}}$, $\mathbf{IR_{refl}} + \mathbf{SP_{elim}}$, $\mathbf{IR_{elim}}$, $\mathbf{IR^{ext}_{elim}}$, $\mathbf{IR_{elim}} + \mathbf{OP_{elim}}$, $\mathbf{IR^{ext}_{init}}$.

Proof of the corollary: In [12] we gave a model for $\mathbf{IR_{refl}}$. This model fulfills the extensionality rules (with $\mathrm{ref}^* := 0$, $r =_A s := \{0 \mid r^* = s^* \wedge s^* \in A^*\}$), and we can easily interpret R^{SP} and verify $\mathbf{SP_{elim}}$. Therefore $\mathbf{IR^{ext}_{refl}} + \mathbf{SP_{elim}}$ is consistent. By Theorems 4.4.1, 5.3.3 and the fact that the above interpretations hold if one extends the theories by additional rules and constants, the consistency of $\mathbf{IR^{ext}_{elim}} + \mathbf{OP_{elim}}$ and $\mathbf{IR^{ext}_{init}} + \mathbf{OP_{elim}}$ follows.

Proof of Theorem 5.3.3:
The following list gives an interpretation of all terms in the language of $\mathbf{IR_{elim}}$, which are not in the language of $\mathbf{IR_{refl}}$ (this interpretation has to be applied inductively to subterms as well)

$$\begin{array}{rcl}
\mathrm{OP}_D & \mapsto & \mathrm{OP}^*_D := (\phi : \mathrm{SP}_D) \times (\mathrm{Arg}_\phi \to D) \ , \\
\text{and with} & & \\
\mathrm{sp}' & := & (\gamma)\pi_0(\gamma) : \mathrm{OP}^*_D \to \mathrm{SP}_D \ , \\
\mathrm{d}' & := & (\gamma)\pi_1(\gamma) : (\gamma : \mathrm{OP}^*_D, \mathrm{Arg}_{\mathrm{sp}'(\gamma)}) \to D \ , \\
\iota(\psi) & \mapsto & \langle \mathrm{nil}, (x)\psi \rangle \ , \\
\sigma(A, \gamma) & \mapsto & \langle \mathrm{nonind}(A, \mathrm{sp}' \circ \gamma), \widetilde{d} \rangle \text{ with } \widetilde{d}(\langle x, y \rangle) = \mathrm{d}'(\gamma(x), y) \ , \\
\delta(A, \gamma) & \mapsto & \langle \mathrm{ind}(A, \mathrm{sp}' \circ \gamma), \widetilde{d} \rangle \text{ with } \widetilde{d}(\langle x, y \rangle) = \mathrm{d}'(\gamma(x), y) \ , \\
\mathbb{F}^{\mathrm{U}}_\gamma(U, T) & \mapsto & \arg_{\mathrm{sp}'(\gamma)}(U, T) \ , \\
\mathbb{F}^{\mathrm{T}}_\gamma(U, T, a) & \mapsto & \mathrm{d}'(\gamma, \mathrm{map}_{\mathrm{sp}'(\gamma)}(U, T, a) \ , \\
\mathbb{F}^{\mathrm{IH}}_\gamma(U, T, E, a) & \mapsto & \mathrm{IH}_{\mathrm{sp}'(\gamma), U, T, E}(a) \ , \\
\mathbb{F}^{\mathrm{map}}_\gamma(U, T, E, h, a) & \mapsto & \mathrm{mapIH}_{\mathrm{sp}'(\gamma), U, T, E}(h, a) \ , \\
\mathrm{U}_\gamma & \mapsto & \mathrm{U}'_{\mathrm{sp}'(\gamma), \mathrm{d}'(\gamma)} \ , \\
\mathrm{T}_\gamma(a) & \mapsto & \mathrm{T}'_{\mathrm{sp}'(\gamma), \mathrm{d}'(\gamma)}(a) \ , \\
\mathrm{intro}_\gamma(a) & \mapsto & \mathrm{intro}'_{\mathrm{sp}'(\gamma), \mathrm{d}'(\gamma)}(a) \ , \\
\mathrm{R}_{\gamma, E}(e) & \mapsto & \mathrm{R}'_{\mathrm{sp}'(\gamma), \mathrm{d}'(\gamma), E}(e) \ .
\end{array}$$

Further, in part (b) we have, with $E'[\phi] := (d : \mathrm{Arg}_\phi \to D) \to E[\langle\phi, d\rangle]$ the translation

$$\begin{array}{lll}
\mathrm{R}^{\mathrm{OP}}_{D,E}(\gamma, a, b, c) & \mapsto & \mathrm{R}^{\mathrm{SP}}_{D,E'}(\mathrm{sp}'(\gamma), \\
& & \quad (d) \qquad a(d(*)), \\
& & \quad (A, \phi, f, d) b(A, \\
& & \qquad\qquad (x)\langle\phi(x), (y)d(\langle x, y\rangle)\rangle \\
& & \qquad\qquad (x)f(x, (y)d(\langle x, y\rangle))), \\
& & \quad (A, \phi, f, d) c(A, \\
& & \qquad\qquad (x)\langle\phi(x), (y)d(\langle x, y\rangle)\rangle \\
& & \qquad\qquad (x)f(x, (y)d(\langle x, y\rangle)))) \\
& & (\mathrm{d}'(\gamma))
\end{array}$$

One easily verifies that with this interpretation the rules of $\mathbf{IR_{elim}}$ and $\mathbf{IR_{elim}} + \mathbf{OP_{elim}}$ hold in $\mathbf{IR_{refl}}$, $\mathbf{IR_{refl}} + \mathbf{SP_{elim}}$ respectively.

We are now going to study the interpretation of $\mathbf{IR_{refl}}$ in $\mathbf{IR_{elim}}$. We will need additionally $\mathbf{OP_{elim}}$, extensionality and the following rules:

Definition 5.3.5 *(a) Let* $\mathbf{Case_2^{type}}$ *be the following rules, expressing case distinction for* $\mathbf{2}$ *into* type*:*

$$\frac{a : \mathbf{2} \qquad A : \mathsf{type} \qquad B : \mathsf{type}}{\mathrm{case}_2^{\mathsf{type}}(a, A, B) : \mathsf{type}}$$

$$\mathrm{case}_2^{\mathsf{type}}(*_0, A, B) = A$$

$$\mathrm{case}_2^{\mathsf{type}}(*_1, A, B) = B$$

(b) Using $\mathbf{Case_2^{type}}$ *we define for* $A : \mathsf{type}$*,* $B : \mathsf{type}$*:*

- $A + B := (x : \mathbf{2}) \times \mathrm{case}_2^{\mathsf{type}}(x, A, B) : \mathsf{type}$.
- *For* $a : A$*,* $\mathrm{inl}(a) := \langle *_0, a\rangle : A + B$.
- *For* $b : A$*,* $\mathrm{inr}(b) := \langle *_1, b\rangle : A + B$.
- *Using additionally an equality on* $\mathbf{2}$*, let for* $a : A + B$ $\mathrm{isl}(a) := (\pi_0(a) =_{\mathbf{2}} *_0) : \mathsf{stype}$.

Remark 5.3.6 $\mathbf{Case_2^{type}}$ *can be interpreted in the model of [12] in a straightforward way, therefore* $\mathbf{IR^{ext}_{refl}} + \mathbf{SP_{elim}} + \mathbf{Case_2^{type}}$*,* $\mathbf{IR^{ext}_{elim}} + \mathbf{OP_{elim}} + \mathbf{Case_2^{type}}$*,* $\mathbf{IR^{ext}_{init}} + \mathbf{OP_{elim}} + \mathbf{Case_2^{type}}$ *are consistent as well.*

Further we need the subtree relation on OP_D:

Lemma 5.3.7 *In* $\mathbf{IR^{ext}_{elim}} + \mathbf{OP_{elim}}$ *the following holds under assumption* $D : \mathsf{type}$*:*

(a) We can define for $\gamma : \mathrm{OP}_D$

$$\begin{array}{lll}
\mathrm{case}^{\mathrm{OP}}_D(\gamma) & : & D + (((A : \mathsf{stype}) \times (A \to \mathrm{OP}_D)) \\
& & \quad + ((A : \mathsf{stype}) \times ((A \to D) \to \mathrm{OP}_D))) \ ,
\end{array}$$

such that we can prove

- $\mathrm{case}^{\mathrm{OP}}_D(\iota(\psi)) = \mathrm{inl}(\psi)$,
- $\mathrm{case}^{\mathrm{OP}}_D(\sigma(A, \gamma)) = \mathrm{inr}(\mathrm{inl}(\langle A, \gamma\rangle))$,
- $\mathrm{case}^{\mathrm{OP}}_D(\delta(A, \gamma)) = \mathrm{inr}(\mathrm{inr}(\langle A, \gamma\rangle))$.

(b) For $\gamma, \gamma' : \mathrm{OP}_D$ *we can define* $\gamma' \preceq_D \gamma : \mathsf{type}$ *(expressing "*γ' *is a subtree of* γ *or equal to* γ*"), such that we can prove*

- $\gamma' \preceq_D \iota(\psi)$ *iff* $\gamma' = \iota(\psi)$,
- $\gamma' \preceq_D \sigma(A, \gamma)$ *iff* $\gamma' = \sigma(A, \gamma)$ *or* $\gamma' \preceq_D \gamma(a)$ *for some* $a : A$,
- $\gamma' \preceq_D \delta(A, \gamma)$ *iff* $\gamma' = \delta(A, \gamma)$ *or* $\gamma' \preceq_D \gamma(f)$ *for some* $f : A \to D$,
- $\preceq_D$ *is transitive and reflexive.*

Definition 5.3.8 *In the situation of the last lemma we write* $\forall \gamma' \preceq_D \gamma.E[\gamma']$ *for*

$$(\gamma' : \mathrm{OP}_D) \to \gamma' \preceq_D \gamma \to E[\gamma'] \ .$$

Proof of Lemma 5.3.7: (a) $\mathrm{case}^{\mathrm{OP}}_D(\gamma)$ can be defined by induction on γ. In (b) we cannot simply use elimination rules on γ since for this we need a type to collect $\gamma' \preceq_D \gamma$ for all $\gamma : \mathrm{OP}_D$.
We define $\gamma' \preceq_D \gamma$ iff there exists $n : \mathrm{N}$, $f : \mathrm{N}_{n+1} \to \mathrm{OP}_D$, such that

- $f(0_{n+1}) = \gamma$,
- $f(n_{n+1}) = \gamma'$,
- if $k + 1 < n + 1$, then $f(k_{n+1}) \neq \iota(\psi)$ and
  - if $f(k_{n+1}) = \sigma(A, \gamma'')$, then $f((k+1)_{n+1}) = \gamma''(a)$ for some $a : A$, and
  - if $f(k_{n+1}) = \delta(A, \gamma'')$, then $f((k+1)_{n+1}) = \gamma''(g)$ for some $g : A \to D$.

The verification of the properties of $\preceq_D$ is now easy.

Theorem 5.3.9 $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{refl}} + \mathbf{SP}_{\mathbf{elim}}$ *can be interpreted in* $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{elim}} + \mathbf{OP}_{\mathbf{elim}} + \mathbf{Case}^{\mathbf{type}}_{\mathbf{2}}$.

Proof: The main problem is the interpretation of SP. Once this is done and the formation/introduction/elimination/equality rules for SP are verified, we can in a straightforward way define Arg, arg, map, IH, mapIH, op, sp, d and therefore the equations in Theorem 5.3.2 hold. Now interpret $\mathrm{U}'_{\phi,d}$, $\mathrm{T}'_{\phi,d}(a)$, $\mathrm{intro}'_{\phi,d}(a)$, $\mathrm{R}'_{\phi,d,E}(f)$ as $\mathrm{U}_{\mathrm{op}(\phi,d)}$, $\mathrm{T}_{\mathrm{op}(\phi,d)}(a)$, $\mathrm{intro}_{\mathrm{op}(\phi,d)}(a)$, $\mathrm{R}_{\mathrm{op}(\phi,d),E}(f)$. All rules are then trivially fulfilled and we are done.

We will now interpret SP and verify the rules for it and work in the following in $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{elim}} + \mathbf{OP}_{\mathbf{elim}} + \mathbf{Case}^{\mathbf{type}}_{\mathbf{2}}$.

We cannot interpret SP_D as OP_D. For instance, if D is empty, OP_D is empty but SP_D is not empty. Instead we will interpret elements of SP_D as $\gamma : \mathrm{OP}_{D+\mathbf{1}}$, such that

- for all subtrees of γ of the form $\iota(\psi)$ we have $\psi = \mathrm{inl}(*)$, which corresponds to the fact that in SP_D leaves do not refer to D,
- all subtrees of γ of the form $\delta(A, \gamma')$ are such that

  $$\gamma'(f) = \sigma((x : A) \to \mathrm{isl}(f(x)), \gamma''(f)) \ ,$$

  which means that (since all proofs of $(x : A) \to \mathrm{isl}(f(x))$ are equal and force f to be equal to $\mathrm{inl} \circ f'$ for some $f' : A \to D$) $\gamma''(f, g) = \gamma'''(f')$ where $f' : A \to D$ such that $f = \mathrm{inl} \circ f'$.

However, there will be no direct relationship between the functors on slice categories coded by the corresponding elements in SP_D and $\mathrm{OP}_{D+\mathbf{1}}$, we use $\mathrm{OP}_{D+\mathbf{1}}$ only as a type of trees.

So SP_D is interpreted as

$$\mathrm{SP}^*_D := (\gamma : \mathrm{OP}_{D+\mathbf{1}}) \times \mathrm{Cor}_D(\gamma) \ ,$$

and we define

$$\begin{array}{rcl}\widetilde{\mathrm{op}} := (\phi)\pi_0(\phi) & : & \mathrm{SP}^*_D \to \mathrm{OP}_{D+\mathbf{1}} \ , \\ \mathrm{cor} := (\phi)\pi_1(\phi) & : & (\phi : \mathrm{SP}^*_D) \to \mathrm{Cor}_D(\widetilde{\mathrm{op}}(\phi)) \ ,\end{array}$$

where for $\gamma : \mathrm{OP}_{D+\mathbf{1}}$, $\mathrm{Cor}_D(\gamma)$ iff for all $\gamma' \preceq_{D+\mathbf{1}} \gamma$:

- if $\gamma' = \iota(\psi)$, then $\psi = \mathrm{inl}(*)$;
- if $\gamma' = \delta(A, \gamma'')$, then $\gamma''(f) = \sigma((x : A) \to \mathrm{isl}(f(x)), \gamma''')$ for some γ'''.

ψ, A, γ'', γ''' can be expressed as terms in γ' by using $\mathrm{case}^{\mathrm{OP}}_D$, and therefore $\mathrm{Cor}_D(\gamma)$ is built from universal quantifications, implication and conjunction only with the right side of all implications being an equality. By uniqueness of equality proofs follows therefore

$$\text{for all } p, p' : \mathrm{Cor}_D(\gamma) \ p = p' \ .$$

Further by transitivity of $\preceq_{D+\mathbf{1}}$ it follows

$$\mathrm{Cor}_D(\gamma) \to \forall \gamma' \preceq_{D+\mathbf{1}} \gamma . \mathrm{Cor}_D(\gamma') \ .$$

We now interpret

(i) nil as $\langle \iota(\mathrm{inr}(*)), p \rangle$,

(ii) $\mathrm{nonind}(A, \phi)$ as $\langle \sigma(A, \widetilde{\mathrm{op}} \circ \phi), q \rangle$, and

(iii) $\mathrm{ind}(A, \phi)$ as $\langle \delta(A, (f)\sigma((x : A) \to \mathrm{isl}(f(x)), (g)\widetilde{\mathrm{op}}(\phi(f')))), r \rangle$.

Here p, q, r are suitable proofs of $\mathrm{Cor}_D(\gamma')$ for the corresponding $\gamma' : \mathrm{OP}_{D+\mathbf{1}}$ we obtain using in (ii), (iii) $\mathrm{cor} \circ \phi$. Further in (iii) $f' : A \to D$ is obtained from f and g such that $\mathrm{inl} \circ f' = f$.

By the uniqueness of elements of $\mathrm{Cor}_D(\gamma)$ and the uniqueness of proofs of $(x : A) \to \mathrm{isl}(f(x))$ it follows:

- If $\gamma : \mathrm{OP}_D$ such that $\mathrm{Cor}_D(\gamma)$, then
  - $\gamma = \widetilde{\mathrm{op}}(\mathrm{nil})$ or
  - $\gamma = \widetilde{\mathrm{op}}(\mathrm{nonind}(A, \phi))$ for some unique A, ϕ or
  - $\gamma = \widetilde{\mathrm{op}}(\mathrm{ind}(A, \phi))$ for some unique A, ϕ.

We can now interpret R^{SP} and verify its rules in the following way.

Assume γ_0, E, a, b, c as in the premise of the elimination rule for SP_D. We show for $\gamma : \mathrm{OP}_{D+\mathbf{1}}$,

$$g(\gamma) : (p : \mathrm{Cor}_D(\gamma), \gamma' : \mathrm{OP}_D, q : \gamma' \preceq_{D+\mathbf{1}} \gamma).E[\langle \gamma', p' \rangle] \ ,$$

where $g(\gamma)$ is defined using $\mathbf{OP_{elim}}$ and p' is a proof of $\mathrm{Cor}_D(\gamma')$ obtained from p and q.

Assume the assertion for immediate subtrees of γ, $\gamma' \preceq_{D+\mathbf{1}} \gamma$. If γ' is a proper subtree of γ, the assertion follows from the IH. Otherwise $\gamma' = \gamma$. Then $\gamma' = \widetilde{\mathrm{op}}(\mathrm{nil})$ or $\gamma' = \widetilde{\mathrm{op}}(\mathrm{nonind}(A, \phi))$ or $\gamma' = \widetilde{\mathrm{op}}(\mathrm{ind}(A, \phi))$ for some ϕ. In the last two cases we obtain, since for $x : A$ respective $x : A \to (D + \mathbf{1})$, $\widetilde{\mathrm{op}}(\phi(x))$ is a proper subtree of γ by IH $E[\phi(x)]$, and therefore in all three cases by the steps a, b, c $E[\langle \gamma', p \rangle]$.

We now define the interpretation of R^{SP}

$$\mathrm{R}^{\mathrm{SP},*}_{D,E}(\phi, a, b, c) := g(\widetilde{\mathrm{op}}(\phi), \mathrm{cor}(\phi), \widetilde{\mathrm{op}}(\phi), \mathrm{r}(\phi)) : E[\phi] \ ,$$

where $\mathrm{r}(\phi) : \widetilde{\mathrm{op}}(\phi) \preceq_{D+\mathbf{1}} \widetilde{\mathrm{op}}(\phi)$. Note that from the uniqueness of proofs $p : \mathrm{Cor}_D(\gamma)$ it follows that for all $p, p' : \mathrm{Cor}_D(\gamma)$

$$\mathrm{R}^{\mathrm{SP},*}_{D,E}(\langle \gamma, p \rangle, a, b, c) = \mathrm{R}^{\mathrm{SP},*}_{D,E}(\langle \gamma, p' \rangle, a, b, c)$$

.

It remains to verify that the equality rules for SP hold:

- $\mathrm{R}^{\mathrm{SP},*}_{D,E}(\mathrm{nil}, a, b, c) = a$ is immediate.
- $\mathrm{R}^{\mathrm{SP},*}_{D,E}(\mathrm{nonind}(A, \phi), a, b, c) = b(A, \phi, h)$.
- $\mathrm{R}^{\mathrm{SP},*}_{D,E}(\mathrm{ind}(A, \phi), a, b, c) = c(A, \phi, h)$.

In the last two equations $h(x)$ is a proof of $E[\phi(x)]$. It follows that

$$\mathrm{R}^{\mathrm{SP},*}_{D,E}(\phi(x), a, b, c) = h(x).$$

Summary. The following diagram summarizes the relationships between the theories considered above (5.3.9 requires the addition of $\mathbf{Case}^{\mathbf{type}}_{\mathbf{2}}$ to the logical framework):

$$\begin{tikzcd}
\mathbf{IR}^{\mathbf{ext}}_{\mathbf{refl}} + \mathbf{SP}_{\mathbf{elim}} \arrow[d, shift right, "{5.3.9,\ \text{with } \mathbf{Case}^{\mathbf{type}}_{\mathbf{2}}}"'] & \\
\mathbf{IR}^{\mathbf{ext}}_{\mathbf{elim}} + \mathbf{OP}_{\mathbf{elim}} \arrow[u, shift right, "5.3.3(b)"'] \arrow[r, shift left, "4.4.3"] & \mathbf{IR}^{\mathbf{ext}}_{\mathbf{init}} + \mathbf{OP}_{\mathbf{elim}} \arrow[l, shift left, "4.4.1"]
\end{tikzcd}$$

6 The Mahlo Universe

6.1 The Internal Mahlo Universe

In this final section we recall Setzer's definition of a Mahlo universe [36, 34, 35] in Martin-Löf type theory. In fact, we consider two versions of it, the original "internal" version, and another, to our knowledge yet unpublished "external" one, both of which have been the subject of much discussion during the last few years.

There are several interesting connections between Mahlo notions and induction-recursion. First, the external Mahlo universe is a powerful example of what can be defined by induction-recursion in the current theory $\mathbf{IR}_{\mathbf{elim}}$. Secondly, the internal Mahlo universe is a canonical example of a definition which goes beyond induction-recursion as formalized by $\mathbf{IR}_{\mathbf{elim}}$. Whereas the external Mahlo universe, as all inductive-recursive definitions in $\mathbf{IR}_{\mathbf{elim}}$, has constructors which are strictly positive in the set defined, this is not the case for the internal Mahlo universe.

The second author has determined a lower bound of the proof-theoretic strength of the internal Mahlo universe [36] and shown that its strength is substantially greater than the strength of the type theory known before (with W-type and finitely iterated universes). He will show in Subsection 6.4 how to modify this result to obtain a lower bound of the proof-theoretic strength of the external Mahlo universe, which is only slightly below the strength of the internal Mahlo universe. As a consequence we therefore get a lower bound of the proof-theoretic strength of the theory of inductive-recursive definitions $\mathbf{IR}_{\mathbf{elim}}$ and its variants $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{init}}$ and $\mathbf{IR}_{\mathbf{refl}}$.

The goal of the definition of the Mahlo universe is to find a constructive analogue of a Mahlo cardinal and its recursive analogue, a recursively Mahlo ordinal. We briefly repeat the definitions. A cardinal κ is Mahlo (or more precisely weakly Mahlo) if it is regular and every normal function $f : \kappa \to \kappa$ has a regular fixed point.

The recursive analogue of a regular cardinal is an admissible, where an ordinal κ is admissible if it is > 0 and for every (set theoretic) Π_2-formula φ with parameters in L_κ which holds in L_κ there exists an $\alpha < \kappa$ s.t. L_α contains the parameters and φ holds in L_α. An ordinal is recursively Mahlo if it is > 0 and for every Π_2-formula φ (as before with parameters) which holds in L_κ there exists an admissible $\pi < \kappa$ s.t. φ holds in L_π.

Related to the notion of a recursively Mahlo ordinal is the notion of recursively inaccessible: an ordinal is recursively inaccessible if it is admissible and the limit of admissibles. It can easily be seen that a recursively Mahlo ordinal is recursively inaccessible, and that the π mentioned above can always be chosen to be inaccessible. So an ordinal κ is recursively Mahlo, if it is recursively inaccessible and for every Π_2-formula φ with parameters which holds in L_κ there exists an inaccessible $\pi < \kappa$ such that φ holds in L_π, and we will take this characterization as a basis for the type-theoretic formulation.

In term models, W-sets correspond to inductive definitions which can be modeled by iterations of a certain operator up to an admissible κ such that the interpretations of the underlying sets are in L_κ. A universe is inductively defined and closed under the W-formation and therefore modeled by iterating an operation up to a recursively inaccessible ordinal. Roughly speaking recursively inaccessible ordinals correspond to universes.

A universe $V : \mathsf{set}$ with decoding function $S : V \to \mathsf{set}$ can be seen as the type theoretic analogue of a recursively inaccessible κ, and the type theoretic analogue of a Π_2-formula, which holds in L_κ, is a function $f : \mathrm{Fam}(V) \to \mathrm{Fam}(V)$. Here $\mathrm{Fam}(V) := (a : V) \times (S(a) \to V)$ are V-indexed families of sets in V. The analogy of the fact that for any Π_2-formula there exists a recursively inaccessible closed under it is now that for every function f as above there exists a universe U_f closed under f.

So a formulation of the Mahlo principle in type theory is as follows: There exists a universe V which is a set with decoding function S such that for every function $f : \mathrm{Fam}(\mathrm{V}) \to \mathrm{Fam}(\mathrm{V})$ there exists a subuniverse U_f of V, closed under f and represented in V.

We can simplify this by currying f and splitting it into two functions f, g: Instead of

$$f : ((a : \mathrm{V}) \times (\mathrm{S}(a) \to \mathrm{V})) \to ((a : \mathrm{V}) \times (\mathrm{S}(a) \to \mathrm{V}))$$

we take

$$\begin{aligned} f &: (a : \mathrm{V}, b : \mathrm{S}(a) \to \mathrm{V}) \to \mathrm{V} \ , \\ g &: (a : \mathrm{V}, b : \mathrm{S}(a) \to \mathrm{V}, \mathrm{S}(f(a,b))) \to \mathrm{V} \ . \end{aligned}$$

The precise formalization of the Mahlo principle in type theory is now as follows:

First of all

$$\mathrm{V} : \mathsf{set} \ , \qquad \mathrm{S} : \mathrm{V} \to \mathsf{set} \ ,$$

and V, S is closed under the standard universe constructions.

Assume now f, g as before. Then the Mahlo principle claims that we have a subuniverse $\mathrm{U}_{fg}, \widehat{\mathrm{T}}_{fg}$ of V, S closed under f and g and represented in V. So we have

$$\mathrm{U}_{fg} : \mathsf{set} \ , \qquad \widehat{\mathrm{T}}_{fg} : \mathrm{U}_{fg} \to \mathrm{V} \ ,$$

and define for $a : \mathrm{U}_{fg}$

$$\mathrm{T}_{fg}(a) := \mathrm{S}(\widehat{\mathrm{T}}_{fg}(a)) : \mathsf{set} \ .$$

For the standard constructors of V, like

$$\begin{aligned}
\widehat{\mathrm{N}} &: \mathrm{V} \ , \\
\mathrm{S}(\widehat{\mathrm{N}}) &= \mathrm{N} \ , \\
\widehat{\Pi} &: (a:\mathrm{V}, b:\mathrm{S}(a) \to \mathrm{V}) \to \mathrm{V} \ , \\
\mathrm{S}(\widehat{\Pi}(a,b)) &= \Pi(\mathrm{S}(a), \mathrm{S} \circ b) \ ,
\end{aligned}$$

we claim the existence of codes in U_{fg}, reflecting them, i.e.

$$\begin{aligned}
\widehat{\mathrm{N}}_{fg} &: \mathrm{U}_{fg} \ , \\
\widehat{\mathrm{T}}_{fg}(\widehat{\mathrm{N}}_{fg}) &= \widehat{\mathrm{N}} \ , \\
\widehat{\Pi}_{fg} &: (a:\mathrm{U}_{fg}, b:\mathrm{T}_{fg}(a) \to \mathrm{U}_{fg}) \to \mathrm{U}_{fg} \ , \\
\widehat{\mathrm{T}}_{fg}(\widehat{\Pi}_{fg}(a,b)) &= \widehat{\Pi}(\widehat{\mathrm{T}}_{fg}(a), \widehat{\mathrm{T}}_{fg} \circ b) \ .
\end{aligned}$$

Further $\widehat{\mathrm{U}}_{fg}$ is closed under f and g, i.e. we have constructors

$$\begin{aligned}
\widehat{\mathrm{f}}_{fg} &: (a:\mathrm{U}_{fg}, b:\mathrm{T}_{fg}(a) \to \mathrm{U}_{fg}) \to \mathrm{U}_{fg} \ , \\
\widehat{\mathrm{T}}_{fg}(\widehat{\mathrm{f}}_{fg}(a,b)) &= f(\widehat{\mathrm{T}}_{fg}(a), \widehat{\mathrm{T}}_{fg} \circ b) \ ; \\
\widehat{\mathrm{g}}_{fg} &: (a:\mathrm{U}_{fg}, b:\mathrm{T}_{fg}(a) \to \mathrm{U}_{fg}, c:\mathrm{S}(f(\widehat{\mathrm{T}}_{fg}(a), \widehat{\mathrm{T}}_{fg} \circ b))) \\
&\quad \to \mathrm{U}_{fg} \ , \\
\widehat{\mathrm{T}}_{fg}(\widehat{\mathrm{g}}_{fg}(a,b,c)) &= g(\widehat{\mathrm{T}}_{fg}(a), \widehat{\mathrm{T}}_{fg} \circ b, c) \ ;
\end{aligned}$$

and U_{fg} is represented in V, i.e.

$$\widehat{\mathrm{U}}_{fg} : \mathrm{V} \ , \qquad \mathrm{S}(\widehat{\mathrm{U}}_{fg}) = \mathrm{U}_{fg} \ .$$

U_{fg}, $\widehat{\mathrm{T}}_{fg}$ are inductive-recursively defined: They can be defined as

$$\mathrm{U}_{fg} = \mathrm{U}'(\mathrm{V}, \mathrm{S}, f, g, \widehat{\mathrm{N}}, \widehat{\Pi}, \ldots), \qquad \widehat{\mathrm{T}}_{fg} = \mathrm{T}'(\mathrm{V}, \mathrm{S}, f, g, \widehat{\mathrm{N}}, \widehat{\Pi}, \ldots) \ ,$$

("…" stands for other universe constructions) where for $V : \mathsf{set}$, $S : V \to \mathsf{set}$, $f : (x : V, y : S(x) \to V) \to V$, $g : (x : V, y : S(x) \to V, S(f(x,y))) \to V$, $a : V$, $b : (x : V, y : S(x) \to V) \to V$ etc.

$$\mathrm{U}'(V, S, f, g, a, b, \ldots) : \mathsf{set}, \qquad \mathrm{T}'(V, S, f, g, a, b, \ldots) : \mathrm{U}'(V, S, f, g, a, b, \ldots) \to V$$

can be defined by an inductive-recursive definition. However, V itself has apart from the standard constructors, which are strictly positive in V, also one constructor, which is not at all positive in it, namely

$$\begin{aligned}
\widehat{\mathrm{U}} \ : \ & (f : (a:\mathrm{V}, b:\mathrm{S}(a) \to \mathrm{V}) \to \mathrm{V}, \\
& \ g : (a:\mathrm{V}, b:\mathrm{S}(a) \to \mathrm{V}, \mathrm{S}(f(a,b))) \to \mathrm{V}) \to \mathrm{V} \ .
\end{aligned}$$

Therefore V definitely goes beyond induction-recursion as discussed in this article.

We call the above construction *internal universe*, since V is an element of set, in contrast to the construction in Section 6.3, where the Mahlo-universe is not an element of set, but set itself.

6.2 Simplification of U_{fg}

In the above definition we demanded U_{fg} to be closed under all standard universe construction. However these can be coded into suitable functions f, g typed as above.

Assume f, g as above. Define functions f', g' of the same type by

$$\begin{array}{rcl} f'(a,b) & = & \mathrm{N}_1 + \mathrm{N} + (\mathrm{S}(a) \times \mathrm{S}(a)) + (\mathrm{S}(a) \times \mathrm{S}(a)) \\ & & +\mathrm{N}_3 + \mathrm{N}_1 + \mathrm{S}(f(a,b)) \ . \end{array}$$

and, if we call the i-th injection into this disjoint union i_i, then

$$\begin{array}{rclrcl} g'(a,b,\mathrm{i}_0(0_1)) & = & \widehat{\mathrm{N}} \ , & g'(a,b,\mathrm{i}_1(k)) & = & \widehat{\mathrm{N}}_k \ , \\ g'(a,b,\mathrm{i}_2(\langle c,d\rangle)) & = & \widehat{\mathrm{I}}(a,c,d) \ , & g'(a,b,\mathrm{i}_3(\langle c,d\rangle)) & = & b(c)\widehat{+}b(d) \ , \\ g'(a,b,\mathrm{i}_4(0_3)) & = & \widehat{\Pi}(a,b) \ , & g'(a,b,\mathrm{i}_4(1_3)) & = & \hat{\Sigma}(a,b) \ , \\ g'(a,b,\mathrm{i}_4(2_3)) & = & \widehat{\mathrm{W}}(a,b) \ , & g'(a,b,\mathrm{i}_5(0_1)) & = & f(a,b) \ , \\ g'(a,b,\mathrm{i}_6(c)) & = & g(a,b,c) \ . & & & \end{array}$$

A non-empty sub-collection of sets of V, i.e. $U : \mathsf{set}$, $T : U \to \mathrm{V}$, which is closed under f', g', but not necessarily under the standard universe constructions, has representatives for all universe constructions and for f, g relativized to it, so it is essentially a subuniverse closed under f, g. So we can omit the closure of U_{fg} under universe constructions, except of one constant in order to obtain U_{fg} nonempty, and still have a universe which is sufficiently closed. The canonical choice for the constant would be $\widehat{\mathrm{N}}$. Alternatively one could add additional parameters $a : \mathrm{V}$, $b : \mathrm{S}(a) \to \mathrm{V}$ to f, g and demand that U_{abfg} contains additionally codes for a and $b(x)$ $(x : \mathrm{S}(a))$. Then closure under $\widehat{\mathrm{N}}$ is not needed. Note however that V has in any case to be closed under the standard universe constructions.

6.3 The External Mahlo Universe

The unproblematic part of the above definition was the definition of U_{fg}. Now, instead of making this definition relative to V, we can make it relative to set as well:

Assume

$$\begin{array}{rcl} f & : & (A : \mathsf{set}, B : A \to \mathsf{set}) \to \mathsf{set} \ , \\ g & : & (A : \mathsf{set}, B : A \to \mathsf{set}, f(A,B)) \to \mathsf{set} \ . \end{array}$$

Then we can define inductive-recursively a universe U_{fg}, T_{fg} closed under the standard universe constructions and under f, g. Again we can restrict the standard universe constructions to one, e.g. N, and have the following constructors of U_{fg}:

$$\begin{array}{rcl} \widehat{\mathrm{N}} & : & \mathrm{U}_{fg} \ , \\ \mathrm{T}_{fg}(\widehat{\mathrm{N}}) & = & \mathrm{N} \ , \\ \widehat{\mathrm{f}} & : & (a : \mathrm{U}_{fg}, b : \mathrm{T}_{fg}(a) \to \mathrm{U}_{fg}) \to \mathrm{U}_{fg} \ , \\ \mathrm{T}_{fg}(\widehat{\mathrm{f}}_{fg}(a,b)) & = & f(\mathrm{T}_{fg}(a), \mathrm{T}_{fg} \circ b) \ , \\ \widehat{\mathrm{g}} & : & (a : \mathrm{U}_{fg}, b : \mathrm{T}_{fg}(a) \to \mathrm{U}_{fg}, c : f(\mathrm{T}_{fg}(a), \mathrm{T}_{fg} \circ b)) \to \mathrm{U}_{fg} \ , \\ \mathrm{T}_{fg}(\widehat{\mathrm{g}}_{fg}(a,b,c)) & = & g(\mathrm{T}_{fg}(a), \mathrm{T}_{fg} \circ b, c) \ . \end{array}$$

We obtain the following code for U_{fg} in $\mathrm{OP}_{\mathsf{set}}$:

$$\begin{array}{rcl} \gamma_{\mathrm{U}_{fg}} & = & \sigma(\mathbf{2}, (x)\mathrm{case}_2(x, \iota(\mathrm{N}) \ , \\ & & \quad \sigma(\mathbf{2}, (x)\mathrm{case}_2(x, \delta(\mathbf{1}, (A)\delta(A(*), (B)\iota(f(A(\mathbf{1}), B))) \ , \\ & & \quad\quad \delta(\mathbf{1}, (A)\delta(A(*), (B)\sigma(f(A(\mathbf{1}), B)), (C)g(A(\mathbf{1}), B, C)))))))) \ . \end{array}$$

This is a nice example, which demonstrates how easy it is to verify that something is an inductive-recursive definition: one just has to find a code for it in OP_D. Note that we got inductive-recursive definitions relative to parameters for free: assuming f, g as above we can derive elements of OP_D and the corresponding sets and decoding functions, like U_{fg} and T_{fg} above, will depend on these parameters.

We call the above construction, in which set plays the role of a Mahlo-universe (although set can be closed under other constructions as well), and which is subsumed by inductive-recursive definitions, the external Mahlo universe construction.

6.4 The Strength of the External Mahlo Universe

In [36] the second author showed that the strength of the internal Mahlo universe is at least as strong as the extension of Rathjen's Kripke-Platek set theory for recursively Mahloness, KPM [28] by ω admissibles above a recursively Mahlo ordinal, KPM^+. [34] shows that this bound is sharp. The following theorem provides a lower bound for the strength of the external Mahlo universe. It is due to the second author.

Theorem 6.4.1 (Setzer). *Let T be the type theory having standard type constructions including the* W*-type, all with elimination rules into all types, and rules for the universes U_{fg}, T_{fg} as above for every f, g of the above mentioned type (but no elimination rules for U_{fg} or other universes). The strength of T is at least that of* KPM*.*

Roughly speaking, T as in the Theorem above can be called the type theory with the external Mahlo universe and full elimination rules

Corollary 6.4.2 $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{elim}}$, $\mathbf{IR}^{\mathbf{ext}}_{\mathbf{init}}$ *have at least the strength of* KPM*.*

Proof of Corollary 6.4.2. The external Mahlo universe is an instance of inductive-recursive definitions.

Proof of Theorem 6.4.1. We will show how to adapt the well-ordering proofs [36] for the internal Mahlo universe to its external variant. In a future article, the second author will give an alternative proof. There he will extend ordinal systems to recursively Mahlo ordinals, and obtain simpler and more perspicuous well-ordering proofs.

Most definitions, lemmata, theorems and proofs in [36] can be carried over directly to the external Mahlo universe, if we replace V everywhere by set. Especially $\mathcal{P}(\mathrm{N})$ becomes the type $\mathrm{N} \to \mathsf{set}$, and we can for $A : \mathcal{P}(\mathrm{N})$ define $\mathrm{M}(A), \mathrm{W}(A) : \mathcal{P}(\mathrm{N})$, $\mathrm{Ag}(A) : \mathsf{type}$. $\mathcal{W}$ can be defined as a class, i.e. we can define a predicate $\mathcal{W}(a)$ s.t.

$$a : \mathrm{N} \Rightarrow \mathcal{W}(a) : \mathsf{type}$$

by

$$\mathcal{W}(a) := (A : \mathcal{P}(A)) \times \mathrm{Ag}(A) \times A(a) \enspace .$$

The only exception, where we can no longer carry over proofs from the internal Mahlo universe, is from the last part of Lemma 5.11 (b) onwards, because there we used $\mathrm{W}(\mathcal{W})$, which cannot be defined, since we are not allowed to define $\mathrm{W}x : A.B$ for types A, B.

Instead we argue as follows. First we have transfinite induction over $\mathcal{W}$, for if we have

$$\forall x \in \mathcal{W}.(\forall y \prec x.y \in \mathcal{W} \to \varphi(y)) \to \varphi(x) \enspace ,$$

then, for every distinguished set A we have by $A \sqsubseteq \mathcal{W}$

$$\forall x \in A.(\forall y \prec x.y \in A \to \varphi(y)) \to \varphi(x) \enspace ,$$

and by transfinite induction over A therefore $\forall x \in A.\varphi(x)$. Since every element of $\mathcal{W}$ is in some distinguished set, it follows $\forall x \in \mathcal{W}.\varphi(x)$.

Next we define $\mathcal{W}'_i$ by:

$$\begin{aligned}\mathcal{W}'_0 &:= (\mathcal{W} \cap \mathrm{M}) \cup \{\mathrm{M}\} \ , \\ \mathcal{W}'_{i+1} &:= \{\omega^{\alpha_1} + \cdots + \omega^{\alpha_n} \mid \alpha_i \in \mathcal{W}'_i\} \ ,\end{aligned}$$

which can be defined as classes. We have transfinite induction over $\mathcal{W}'_0$ and then by Gentzen's trick (transfinite induction over ordinals built by Cantor normal form reduces to transfinite induction over the underlying ordinals) and Meta-induction on i we can show transfinite induction over $\mathcal{W}'_i$.

Since $\mathcal{W}$ is closed under Cantor normal form, it follows $\mathcal{W}'_i \cap \mathrm{M} \cong \mathcal{W} \cap \mathrm{M}$. Next we can show for (Meta-) all $i \in \mathrm{N}$:

$$\forall y \in \mathcal{W}'_i.\forall \kappa \in \mathcal{W}'_0 \cap \mathrm{R}.\{y, \kappa\} \subseteq \mathrm{C}_\kappa(y) \to \psi_\kappa(y) \in \mathcal{W} \ . \qquad (+)$$

This is done by induction on y. Assume y and the IH. We show

$$\mathrm{C}^{\psi_\kappa(y)}(\mathcal{W}) \cap \mathrm{C}_\kappa(y) \cap \omega_i(\mathrm{M}+1) \subseteq \mathcal{W}'_i \ ,$$

where $\omega_0(\alpha) := \alpha$, $\omega_{n+1}(\alpha) := \omega^{\omega_n(\alpha)}$.

This can be shown as in the proof of Lemma 5.2 (c), assertion $(*)$ with A replaced by $\mathcal{W}$, $\mathrm{W}(A)$ replaced by $\mathcal{W}'_i$, τ^+ replaced by $\omega_i(\mathrm{M}+1)$ throughout in the proof.

Now it follows

$$\mathrm{C}^{\psi_\kappa y}(\mathcal{W}) \cap \psi_\kappa y \subseteq \mathcal{W}'_i \cap \mathrm{M} \subseteq \mathcal{W} \ .$$

If $y \prec \widetilde{\psi_\kappa y}$, then

$$y \in \mathcal{W}'_i \cap \widetilde{\psi_\kappa y} \cong \mathcal{W} \cap \widetilde{\psi_\kappa y} \subseteq \mathrm{C}^{\psi_\kappa y}(\mathcal{W}) \ ,$$

Otherwise $y \in \mathrm{M}(\mathcal{W}'_i) \cong \mathrm{M}(\mathcal{W})$, $y \in \mathrm{C}^y(\mathcal{W}) \subseteq \mathrm{C}^{\psi_\kappa y}(\mathcal{W})$. Further $\kappa \in \mathcal{W}'_0$, $\kappa \in \mathrm{M}(\mathcal{W})$, $\kappa \in \mathrm{C}^\kappa(\mathcal{W}) \subseteq \mathrm{C}^{\psi_\kappa y}(\mathcal{W})$. It follows $\psi_\kappa y \in \mathrm{C}^{\psi_\kappa y}(\mathcal{W})$. Now we have $\psi_\kappa y \in \mathrm{M}(\mathcal{W})$, $\tau^{\mathcal{W}}(\psi_\kappa y) \cong \mathrm{C}^{\psi_\kappa y}(\mathcal{W}) \cap \psi_\kappa y \subseteq \mathcal{W}'_i \cap \mathrm{M} \cong \mathcal{W}$, $\psi_\kappa y \in \mathcal{A}^{\mathcal{W}}(\mathcal{W}) \cap \mathrm{M} \subseteq \mathcal{W}$, and $(+)$ is shown.

Now $\omega_n(\mathrm{M}+1) \in \mathcal{W}'_{n+1}$, $\Omega_1 \in \mathcal{W}$, $\Omega_1, \omega_n(\mathrm{M}+1) \in \mathrm{C}_{\Omega_1}(\omega_n(\mathrm{M}+1))$, therefore by $(+)$

$$\psi_{\Omega_1}(\omega_n(\mathrm{M}+1)) \in \mathcal{W} \cap \Omega_1 \sqsubseteq \mathrm{OT} \ ,$$

and from transfinite induction over $\mathcal{W}$ follows transfinite induction up to $\psi_{\Omega_1}(\omega_n(\mathrm{M}+1))$ for $n \in \omega$, which in the limit reaches $\psi_{\Omega_1}(\epsilon_{\mathrm{M}+1})$. Rathjen determined the strength of KPM in [27, 28, 29]. The ordinal notation systems we used is based on [3], where it is shown that the strength of KPM is at most $\psi_{\Omega_1}(\epsilon_{\mathrm{M}+1})$ (which can be seen to be sharp as in [29] or by taking the above proof and adapting it to KPM). Therefore the assertion of the theorem follows.

A Complete Rules of the Logical Framework

In this article we omit in general additional contexts in rules. So for $n \geq 1$ a rule

$$\frac{\Delta_1 \Rightarrow \theta_1 \qquad \cdots \qquad \Delta_n \Rightarrow \theta_n}{\Delta \Rightarrow \theta}$$

stands for

$$\frac{\Gamma, \Delta_1 \Rightarrow \theta_1 \qquad \cdots \qquad \Gamma, \Delta_n \Rightarrow \theta_n}{\Gamma, \Delta \Rightarrow \theta}$$

and a rule without premises $\Delta \Rightarrow \theta$ stands for $\dfrac{\Gamma\ \mathsf{context}}{\Gamma, \Delta \Rightarrow \theta}$

The only exception are the context and assumption rules.

Context- and Assumption-rules

$$\emptyset\ \mathsf{context} \qquad \frac{\Gamma\ \mathsf{context} \qquad \Gamma \Rightarrow A : \mathsf{type}}{\Gamma, x : A\ \mathsf{context}}$$

$$\frac{\Gamma\ \mathsf{context} \qquad \Gamma \Rightarrow A : \mathsf{type}}{\Gamma, x : A \Rightarrow x : A} \qquad \frac{\Gamma \Rightarrow x : A \qquad \Gamma \Rightarrow B : \mathsf{type}}{\Gamma, y : B \Rightarrow x : A}\ (\text{if } x \neq y,\ y \notin \mathrm{FV}(A))$$

Equality Rules

$$\frac{a : A}{a = a : A} \qquad \frac{A : \mathsf{type}}{A = A : \mathsf{type}}$$

$$\frac{a = b : A}{b = a : A} \qquad \frac{A = B : \mathsf{type}}{B = A : \mathsf{type}}$$

$$\frac{a = b : A \qquad b = c : A}{a = c : A} \qquad \frac{A = B : \mathsf{type} \qquad B = C : \mathsf{type}}{A = C : \mathsf{type}}$$

$$\frac{a : A \qquad A = B : \mathsf{type}}{a : B} \qquad \frac{a = b : A \qquad A = B : \mathsf{type}}{a = b : B}$$

Rules for $\rightarrow$

$$\frac{A : \mathsf{stype} \qquad x : A \Rightarrow B : \mathsf{stype}}{(x : A) \rightarrow B : \mathsf{stype}} \qquad \frac{x : A \Rightarrow B : \mathsf{type}}{(x : A) \rightarrow B : \mathsf{type}}$$

$$\frac{A = A' : \mathsf{stype} \qquad x : A \Rightarrow B = B' : \mathsf{stype}}{(x : A) \rightarrow B = (x : A') \rightarrow B' : \mathsf{stype}}$$

$$\frac{A = A' : \mathsf{type} \qquad x : A \Rightarrow B = B' : \mathsf{type}}{(x : A) \rightarrow B = (x : A') \rightarrow B' : \mathsf{type}}$$

$$\frac{x : A \Rightarrow t : B}{(x : A)t : (x : A) \rightarrow B}$$

$$\frac{x : A \Rightarrow t = t' : B}{(x : A)t = (x : A)t' : (x : A) \rightarrow B}$$

$$\frac{x : A \Rightarrow B : \mathsf{type} \qquad t : (x : A) \rightarrow B \qquad s : A}{t(s) : B[x := s]}$$

$$\frac{x : A \Rightarrow B : \mathsf{type} \qquad t = t' : (x : A) \rightarrow B \qquad s = s' : A}{t(s) = t'(s') : B[x := s]}$$

$$\frac{x : A \Rightarrow r : B \qquad s : A}{((x : A)r)(s) = r[x := s] : B[x := s]}$$

$$\frac{x : A \Rightarrow B : \mathsf{type} \qquad s : (x : A) \rightarrow B}{s = (x : A)s(x) : (x : A) \rightarrow B}$$

Rules for $\times$

$$\frac{A : \mathsf{stype} \qquad x : A \Rightarrow B : \mathsf{stype}}{(x : A) \times B : \mathsf{stype}} \qquad \frac{x : A \Rightarrow B : \mathsf{type}}{(x : A) \times B : \mathsf{type}}$$

$$\frac{A = A' : \mathsf{stype} \qquad x : A \Rightarrow B = B' : \mathsf{stype}}{(x : A) \times B = (x : A') \times B' : \mathsf{stype}}$$

$$\frac{A = A' : \mathsf{type} \qquad x : A \Rightarrow B = B' : \mathsf{type}}{(x : A) \times B = (x : A') \times B' : \mathsf{type}}$$

$$\frac{r : A \qquad s : B[x := r] \qquad x : A \Rightarrow B : \mathsf{type}}{\langle r, s\rangle : (x : A) \times B}$$

$$\frac{r = r' : A \qquad s = s' : B[x := r] \qquad x : A \Rightarrow B : \mathsf{type}}{\langle r, s\rangle = \langle r', s'\rangle : (x : A) \times B}$$

$$\frac{x : A \Rightarrow B : \mathsf{type} \qquad r : (x : A) \times B}{\pi_0(r) : A}$$

$$\frac{x : A \Rightarrow B : \mathsf{type} \qquad r = r' : (x : A) \times B}{\pi_0(r) = \pi_0(r') : A}$$

$$\frac{x : A \Rightarrow B : \mathsf{type} \qquad r : (x : A) \times B}{\pi_1(r) : B[x := \pi_0(r)]}$$

$$\frac{x : A \Rightarrow B : \mathsf{type} \qquad r = r' : (x : A) \times B}{\pi_1(r) = \pi_1(r') : B[x := \pi_0(r)]}$$

$$\frac{r : A \qquad s : B[x := r] \qquad x : A \Rightarrow B : \mathsf{type}}{\pi_0(\langle r, s\rangle) = r : A}$$

$$\frac{r : A \qquad s : B[x := r] \qquad x : A \Rightarrow B : \mathsf{type}}{\pi_1(\langle r, s\rangle) = s : B[x := r]}$$

$$\frac{x : A \Rightarrow B : \mathsf{type} \qquad r : (x : A) \times B}{r = \langle \pi_0(r), \pi_1(r)\rangle : (x : A) \times B}$$

In the paper we have the following general assumption about equality versions of rules, omitting types in equality judgements and about bracket notations like $E[t]$:

General assumption A.0.3 *(a) In the following all rules are understood to be supplemented by additional equality rules. For instance the rule*

$$\frac{(x : A) \Rightarrow B : \mathsf{type}}{(x : A) \to B : \mathsf{type}}$$

should be supplemented by

$$\frac{A = A' : \mathsf{type} \qquad (x : A) \Rightarrow B = B' : \mathsf{type}}{(x : A) \to B = (x : A') \to B' : \mathsf{type}}$$

and the rule

$$\frac{(x:A) \Rightarrow b : B}{(x:A)b : (x:A) \to B}$$

should be supplemented by

$$\frac{(x:A) \Rightarrow b = b' : B}{(x:A)b = (x:A)b' : (x:A) \to B}$$

(b) *We will usually omit the type in an equality judgement and assumptions about the types of the variables in it, it they can easily be filled in by the reader.*

(c) *We follow a common convention and write $E[x]$ for an expression which may depend on a free variable x. After the first occurrence of it, $E[t]$ denotes the result of substituting the term t for the variable x in $E[x]$. Further, after such an occurrence, E not followed by a square bracket stands for $(x)E[x]$. The latter will be used to denote parameters only.*

B Derivation of Ex Falsum Quodlibet for N_0'

We verify that we can define ex falsum quodlibet for N_0' defined by $\gamma := \sigma(\mathbf{1}, (f)\iota_*)$:

Let $\mathrm{N}_0' := \mathrm{U}_\gamma$, $\mathrm{T}' := \mathrm{T}_\gamma$. Assume $x : \mathrm{N}_0' \Rightarrow E[x] : \mathsf{type}$. We show that there exists $f : (x : \mathrm{N}_0') \to E[x]$.

Define $E' := (x : \mathrm{N}_0') \to E[x]$, $E''[y] := E'$. Definitionally we have

$$\begin{aligned}
\mathbb{F}^{\mathrm{IH}}_{\gamma}(\mathrm{N}_0', \mathrm{T}', E'', u) &= \mathbb{F}^{\mathrm{IH}}_{\gamma}(\mathrm{N}_0', \mathrm{T}', E'', \langle \pi_0(u), \pi_1(u)\rangle) \\
&= ((x : \mathbf{1}) \to E''[\pi_0(u)]) \times \mathbb{F}^{\mathrm{IH}}_{\iota_*}(\mathrm{N}_0', \mathrm{T}', E'', \pi_1(u)) \\
&= (\mathbf{1} \to E') \times \mathbf{1} \ .
\end{aligned}$$

The argument of $\mathrm{R}_{\gamma,E''}$ has type

$$\begin{aligned}
& (u : \mathbb{F}^{\mathrm{U}}_{\gamma}(\mathrm{N}_0', \mathrm{T}'), \mathbb{F}^{\mathrm{IH}}_{\gamma}(\mathrm{N}_0', \mathrm{T}', E'', u)) \to E''[\mathrm{intro}_\gamma(u)] \\
= \ & \mathbb{F}^{\mathrm{U}}_{\gamma}(\mathrm{N}_0', \mathrm{T}') \to ((\mathbf{1} \to E') \times \mathbf{1}) \to E' \ .
\end{aligned}$$

$g := (u, v)\pi_0(v)(*)$ has this type. Therefore $g' := \mathrm{R}_{\gamma,E''}(g) : \mathrm{N}_0' \to (x : \mathrm{N}_0') \to E'[x]$. Define $f := (x)g'(x, x)$.

References

[1] P. Aczel. Frege structures and the notions of proposition, truth, and set. In J. Barwise, H. J. Keisler, and K. Kunen, editors, *The Kleene Symposium*, pages 31–59. North-Holland, 1980.

[2] S. Allen. *A Non-Type-Theoretic Semantics for Type-Theoretic Language*. PhD thesis, Department of Computer Science, Cornell University, 1987.

[3] W. Buchholz. A note on the ordinal analysis of KPM. In J. Oikkonen and J. Väänänen, editors, *Logic Colloquium '90, ASL Summer Meeting Helsinki*, volume 2 of *Springer Lecture Notes in Logic*, pages 1 – 9, 1993.

[4] J. Cartmell. Generalized algebraic theories and contextual categories. *Annals of Pure and Applied Logic*, 32:209–243, 1986.

[5] C. Coquand. *The Agda homepage*, February 2000. `http://www.cs.chalmers.se/~catarina/agda/`.

[6] T. Coquand and C. Paulin. Inductively defined types, preliminary version. In *LNCS 417, COLOG '88, International Conference on Computer Logic*. Springer-Verlag, 1990.

[7] P. Dybjer. Inductive sets and families in Martin-Löf's type theory and their set-theoretic semantics. In G. Huet and G. Plotkin, editors, *Logical Frameworks*, pages 280–306. Cambridge University Press, 1991.

[8] P. Dybjer. Universes and a general notion of simultaneous inductive-recursive definition in type theory. In B. Nordström, K. Petersson, and G. Plotkin, editors, *Proceedings of the 1992 Workshop on Types for Proofs and Programs*, 1992.

[9] P. Dybjer. Inductive families. *Formal Aspects of Computing*, 6:440–465, 1994.

[10] P. Dybjer. Internal type theory. In *TYPES '95, Types for Proofs and Programs*, volume 1158 of *Lecture Notes in Computer Science*, pages 120–134. Springer, 1996.

[11] P. Dybjer. A general formulation of simultaneous inductive-recursive definitions in type theory. *Journal of Symbolic Logic*, 65(2):525–549, 2000.

[12] P. Dybjer and A. Setzer. A finite axiomatization of inductive-recursive definitions. In J.-Y. Girard, editor, *Typed Lambda Calculi and Applications*, volume 1581 of *Lecture Notes in Computer Science*, pages 129–146. Springer, April 1999.

[13] P. Dybjer and A. Setzer. Indexed induction-recursion. In R. Kahle, P. Schroeder-Heister, and R. Stärk, editors, *Proof Theory in Computer Science*, pages 93 – 113. LNCS 2183, 2001.

[14] M. Hofmann. Syntax and semantics of dependent types. In A. Pitts and P. Dybjer, editors, *Semantics and Logics of Computation*, pages 79–130. Cambridge University Press, 1997.

[15] J. Lambek and P. J. Scott. *Introduction to higher order categorical logic*, volume 7 of *Cambridge studies in advanced mathematics*. Cambridge University Press, 1986.

[16] C. Löfwall and G. Sjödin. Strong normalizability in Martin-Löf's type theory. Technical Report R91-09, Swedish Institute of Computer Science, 1991.

[17] P. Martin-Löf. An intuitionistic theory of types: Predicative part. In H. E. Rose and J. C. Shepherdson, editors, *Logic Colloquium '73*, pages 73–118. North-Holland, 1975.

[18] P. Martin-Löf. Constructive mathematics and computer programming. In *Logic, Methodology and Philosophy of Science, VI, 1979*, pages 153–175. North-Holland, 1982.

[19] P. Martin-Löf. *Intuitionistic Type Theory*. Bibliopolis, 1984.

[20] P. Martin-Löf. An intuitionistic theoy of types. In G. Sambin and J. Smith, editors, *Twenty-Five Years of Constructive Type Theory*, pages 127–172. Oxford University Press, 1998. Reprinted version of an unpublished report from 1972.

[21] P. F. Mendler. Predicative type universes and primitive recursion. In *Proceedings Sixth Annual Synposium on Logic in Computer Science*. IEEE Computer Society Press, 1991.

[22] B. Nordström, K. Petersson, and J. Smith. *Programming in Martin-Löf's Type Theory: an Introduction.* Oxford University Press, 1990.

[23] E. Palmgren. *On Fixed Point Operators, Inductive Definitions and Universes in Martin-Löf's Type Theory.* PhD thesis, Uppsala University, 1991.

[24] E. Palmgren. Type-theoretic interpretation of iterated, strictly positive inductive definitions. *Arch. Math. Logic*, 32:75–99, 1992.

[25] E. Palmgren. On universes in type theory. In G. Sambin and J. Smith, editors, *Twenty five years of constructive type theory*, pages 191 – 204, Oxford, 1998. Oxford University Press.

[26] C. Paulin-Mohring. Inductive definitions in the system Coq - rules and properties. In *Typed lambda calculi and applications*, volume 664 of *Lecture Notes in Computer Science*, pages 328–245. Springer-Verlag, 1993.

[27] M. Rathjen. Ordinal notations based on a weakly Mahlo cardinal. *Arch. Math. Logic*, 29:249 – 263, 1990.

[28] M. Rathjen. Proof-theoretical analysis of KPM. *Arch. Math. Logic*, 30:377 – 403, 1991.

[29] M. Rathjen. Collapsing functions based on recursively large cardinals: A well-ordering proof for KPM. *Arch. Math. Logic*, 33:35–55, 1994.

[30] M. Rathjen, E. R. Griffor, and E. Palmgren. Inaccessibility in constructive set theory and type theory. *Annals of Pure and Applied Logic*, 94:181 – 200, 1998.

[31] D. S. Scott. Constructive validity. In *Symposium on Automatic Demonstration*, pages 237–275. Springer Lecture Notes in Mathematics 125, 1970.

[32] R. A. G. Seely. Locally cartesian closed categories and type theory. *Proceedings of the Cambridge Philosophical Society*, 95:33–48, 1984.

[33] A. Setzer. *Proof theoretical strength of Martin-Löf Type Theory with W-type and one universe.* PhD thesis, Fakultät für Mathematik der Ludwig-Maximilians-Universität München, 1993.

[34] A. Setzer. A model for a type theory with Mahlo universe. Draft, available from http://www-compsci.swan.ac.uk/∼csetzer/, 1996.

[35] A. Setzer. A type theory for Mahlo universes. Abstract for Logic Colloquium 95. *Bulletin of Symbolic Logic*, 3:128 – 129, 1997.

[36] A. Setzer. Extending Martin-Löf type theory by one Mahlo-universe. *Arch. Math. Logic*, 39:155 – 181, 2000.