Intensional Type Theory with Guarded Recursive Types qua Fixed Points on Universes

Lars Birkedal
Dept. of Comp. Science, Aarhus University

Rasmus Ejlers Møgelberg
IT University of Copenhagen

***Abstract*—Guarded recursive functions and types are useful for giving semantics to advanced programming languages and for higher-order programming with infinite data types, such as streams, e.g., for modeling reactive systems.**

We propose an extension of intensional type theory with rules for forming fixed points of guarded recursive functions. Guarded recursive types can be formed simply by taking fixed points of guarded recursive functions on the universe of types.

Moreover, we present a general model construction for constructing models of the intensional type theory with guarded recursive functions and types. When applied to the groupoid model of intensional type theory with the universe of small discrete groupoids, the construction gives a model of guarded recursion for which there is a one-to-one correspondence between fixed points of functions on the universe of types and fixed points of (suitable) operators on types.

In particular, we find that the functor category $\mathbf{Grpd}^{\omega^{\mathrm{op}}}$ from the preordered set of natural numbers to the category of groupoids is a model of intensional type theory with guarded recursive types.

I. Introduction

Recursive functions and recursive types are used pervasively in computer science. In particular, guarded recursive types and guarded recursive functions have been used in recent years to give semantics to realistic programming languages with dynamically-allocated higher-order store, such as ML-like references. Guarded recursive types and functions have also proved useful for programming with streams and other "infinite" data types, e.g., in *higher-order* reactive programming [4, 10, 13, 14].

In earlier work [3, 4], we showed how the topos of trees, i.e., the category $\mathbf{Set}^{\omega^{\mathrm{op}}}$ of presheaves over the natural numbers ω, models an extension of dependent type theory with guarded mixed-variance recursive types and functions. Roughly, the extension consists of a new type constructor $\blacktriangleright$ and the possibility of defining recursive functions and types guarded by $\blacktriangleright$. Recursive functions can be defined by means of a fixed-point operator $\mathrm{fix}\colon (\blacktriangleright A \to A) \to A$. Note that guardedness here is expressed using types. This is in contrast to the syntactic checks of function definitions used in Coq and Agda to check for productivity of recursive functions on *coinductive* data types. It is because of the use of types to express guardedness that guarded recursion is useful for *higher-order* programming [4, 13, 14]. For higher-order functions, syntactic checks of function definitions are too restrictive, since one cannot analyze function parameters — this has led to research on how to "code around" these limitations, e.g., [5]. Thus guarded recursive types and functions may usefully complement the use of coinductive types, as also suggested in [16]. To investigate this hypothesis further, we are interested in extending implementations of (intensional) type theory with guarded recursive types and functions. This paper lays the model-theoretic foundations for such extensions.

In contrast to the situation for recursive functions, in [3, 4] the well-definedness of guarded recursive *types* follows standard practise and relies on syntactic checks [4, Proposition 4.10]. But, if we consider a type theory with a universe of types, then it really ought to be possible to express the existence of guarded recursive types simply as fixed points of guarded recursive functions on the universe of types! In this paper, we show that is indeed the case.

In our previous work, we only considered models of guarded recursion for *extensional* type theory. Indeed, the class of models we considered were sheaf toposes over well-founded complete Heyting algebras (the concrete example model $\mathbf{Set}^{\omega^{\mathrm{op}}}$ mentioned above is equivalent to a sheaf topos of this form). In this paper we consider more general models of type theory, encompassing *intensional* type theory. We show, in particular, how any model of intensional type theory with a universe can be used to construct a new model of intensional type theory with guarded recursive functions and guarded recursive types qua fixed points on the universe of types. The construction preserves several type-theoretic principles. For instance, when instantiated to a model of intensional type theory satisfying univalence [21], e.g., Hofmann and Streicher's groupoid model [9] (with the universe of small discrete groupoids), the resulting model also satisfies univalence. This shows the consistency of the combination of guarded recursion and univalence. Moreover, we show that for the groupoid model there is a one-to-one correspondence between fixed points of endomaps on the universe of types and fixed points of suitable operators on types. This result is part of the motivation for considering models of intensional type theory.

The remainder of the paper is organized as follows. In Section II we recall a syntactic formulation of guarded recursion from [4] and also explain our new rules for guarded recursive types qua fixed points on a universe of types. Then in Section III we recall the extensional topos of trees model

and show how to extend it to one with a universe. We recall a categorical notion of model of intensional type theory in Section IV and consider the construction of models for guarded recursion for intensional type theory in Section V. The hope is that the concrete treatment in Section III may serve as an introduction to the more abstract categorical treatment in Section V. Finally, we relate fixed points qua contractive maps on the universe to fixed points of operators on types in Section VI and conclude in Section VII.

II. Syntactic Motivation

We begin by calling to mind how to use types and guarded recursion to ensure that functions on streams (infinite lists of natural numbers) are well defined. In this informal introduction we write $x :: xs$ for the stream with head element x and tail xs. When streams are represented as a coinductive data type, the stream of zeros can be defined recursively as $zeros = 0 :: zeros$. This is a well-defined stream since the right-hand side is productive, i.e., $zeros$ appears directly under :: on the right-hand side. On the other hand, an equation such as $xs = xs$ is not well-defined, since the right-hand side is not productive. If $merge$ is the function that merges two streams, and $tail$ is the function that returns the tail of a stream, then it should also be possible to define the stream of zeros by $zeros = 0 :: merge\ zeros\ (tail\ zeros)$, but, alas, this definition is not productive (since $zeros$ does not appear directly under :: on the right-hand side).

Instead of relying on syntactic productivity checks we can use *types* to ensure well-definedness of functions on streams. The idea is to introduce a type constructor $\blacktriangleright$, pronounced "later", with the intuition being that elements of $\blacktriangleright A$ are elements of A that are only available later (after a time-step has passed). We then use a guarded recursive fixed point combinator $\mathrm{fix}\colon (\blacktriangleright A \to A) \to A$ to ensure well-definedness of recursive definitions of streams. For instance, writing Str for the type of streams and $\mathbb{N}$ for the type of natural numbers, we can give :: the type $N \times \blacktriangleright Str \to Str$ and then define $zeros$ as $\mathrm{fix}(\lambda z : \blacktriangleright Str.\, 0 :: z)$. By typing $merge$ and $tail$ appropriately, it is also possible to show well-definedness of the above alternative definition of $zeros$. See [4, 10, 13, 14] for more examples.

With this approach, using guarded recursion, the type of streams satisfies the isomorphism $Str \cong \mathbb{N} \times \blacktriangleright Str$, and thus the type of streams is an example of a *guarded recursive type*. Of course, streams is just one such example and it is clearly useful to have a general mechanism for defining guarded recursive types. In our earlier models of guarded recursion [4], one can form the guarded recursive type of streams as follows:

$$Str \stackrel{\mathrm{def}}{=} \mu X.\, \mathbb{N} \times \blacktriangleright X.$$

Here μ is a new type constructor for guarded recursive types. The above is a well-defined guarded recursive type, since the definition follows the syntactic rules for guarded recursive types, in particular since the recursion variable X occurs under a $\blacktriangleright$. Observe that well-definedness just relies on X appearing under a $\blacktriangleright$; in particular, the models in [4] allow for *mixed-variance* guarded recursive types (see *loc. cit.* for an application to semantics of programming languages with general references).

A. Fixed Points on Universes

Note that the well-definedness of recursive *types* in [4], as recalled above, was expressed using a syntactic check, akin to the productivity check for recursive functions on coinductive types. Here, instead, we will consider a type theory with a universe U of types and define Str by an ordinary fixed point of a guarded function on the universe.

The type constructor $\blacktriangleright$ is modeled by a strong product preserving functor and hence satisfies the rules for an applicative functor in the sense of McBride and Paterson [17]. Thus there is the type formation rule

$$\frac{\Gamma \vdash A\colon \mathrm{Type}}{\Gamma \vdash \blacktriangleright A\colon \mathrm{Type}}$$

and two associated terms:

$$\frac{\Gamma \vdash t\colon A}{\Gamma \vdash \mathrm{next}(t)\colon \blacktriangleright A} \qquad \frac{\Gamma \vdash t\colon \blacktriangleright(A \to B) \quad \Gamma \vdash u\colon \blacktriangleright A}{\Gamma \vdash t \circledast u\colon \blacktriangleright B}$$

One intuition is that types are indexed over discrete time and next captures a form of Kripke monotonicity: if we have a proof t of A now, then $\mathrm{next}(t)$ is a proof of $\blacktriangleright A$, i.e., of A later. Another, related, intuition is that next transforms elements we know now to elements we will only use later. (See the following section for a concrete model.) From this point of view, $t \circledast u$ expresses that if we know later that t is a function from A to B, and we later have an element of type A, then we can apply the function at that later time to get an element of B at that later time, i.e., an element of $\blacktriangleright B$.

The above terms satisfy the following four equations for applicative functors (definitional equalities in the type theory):

$$\begin{aligned} \mathrm{next}(\lambda x.x) \circledast u &= u \\ \mathrm{next}(\circ) \circledast s \circledast t \circledast u &= s \circledast (t \circledast u) \\ \mathrm{next}(t) \circledast \mathrm{next}(u) &= \mathrm{next}(t(u)) \\ u \circledast \mathrm{next}(t) &= \mathrm{next}(\lambda g.g(t)) \circledast u \end{aligned}$$

where $\circ$ is the composition function.

Guarded recursive fixed points are then captured by the following rule, introducing the fixed point term, and the following equation, expressing that it yields a fixed point.

$$\frac{\Gamma, x\colon \blacktriangleright A \vdash t\colon A}{\Gamma \vdash \mathrm{fix}\, x.t\colon A} \tag{1}$$

$$t[\mathrm{next}(\mathrm{fix}\, x.t)/x] = \mathrm{fix}\, x.t \tag{2}$$

Fixed points are unique internally, as expressed by the following rule:

$$\frac{\Gamma, f\colon \blacktriangleright A \to A, y\colon A \vdash p\colon \mathrm{Id}_A(y, f(\mathrm{next}(y)))}{\Gamma, f\colon \blacktriangleright A \to A, y\colon A \vdash \mathrm{UFP}(p)\colon \mathrm{Id}_A(y, \mathrm{fix}\, x.f(x))} \quad (3)$$

The universe of types U is closed under the later operator, that is, there is also a later operator on codes for types, denoted $\triangleright$,

$$\frac{\Gamma \vdash A\colon \blacktriangleright U}{\Gamma \vdash \triangleright A\colon U} \quad (4)$$

The assumption of $A\colon \blacktriangleright U$ rather than $A\colon U$ is crucial for representing guarded recursive domain equations as maps $\blacktriangleright U \to U$.

The standard elements-of operator El taking elements of the universe to types is related to $\blacktriangleright$ and next as expressed by the following equation (in a context with $A : U$):

$$\mathrm{El}(\triangleright(\mathrm{next}(A))) = \blacktriangleright El(A) \quad (5)$$

We follow standard practise and often omit El when it can be inferred from context.

Using the above rules, we have the following derived rule

$$\frac{\Gamma, X\colon \blacktriangleright U \vdash A\colon U}{\Gamma \vdash \mu X.A\colon U} \quad (6)$$

where $\mu X.A$ is simply notation for $\mathrm{fix}\, X.A$. As an example, the name of the type of streams can now be defined like before (assuming $\mathbb{N}\colon U$)

$$Str = \mu X.\mathbb{N} \times \triangleright X : U$$

Note that

$$\begin{aligned}\mathrm{El}(Str) &= \mathrm{El}(\mathbb{N} \times \triangleright(\mathrm{next}(Str))) \\ &= \mathbb{N} \times \blacktriangleright \mathrm{El}(Str)\end{aligned}$$

which implies that the type $\mathrm{El}(Str)$ is a solution to the type equation corresponding to the given operation on the universe. Note that we obtain a solution up to identity, rather than isomorphism — this is due to the fact that solutions are found for endomaps on U, i.e., on maps from codes of types to codes of types. We relate fixed points for endomaps on U to fixed points for operators on types in Section VI.

Thus we can now define guarded recursive types simply as fixed points of guarded functions on the universe. For functions that are suitably functorial and locally contractive one can show that the fixed point is both an initial algebra and a final coalgebra. We just consider initiality: Say that $F : U \to U$ is guarded if there exists a $G : \blacktriangleright U \to U$ such that $F = G \circ \mathrm{next}$. Let μF be the fixed point of F. Assume that F has a contractive functorial action, i.e., a term $F : \prod X, Y.(X \to Y) \to (FX \to FY)$ such that, for any X, Y, there exists a $G_{X,Y}$ such that $F_{X,Y} = G_{X,Y} \circ \mathrm{next}$ and $G_{X,Y}$ satisfies the functor laws up to external equality. Then $F(\mu F) = \mu F$ is an initial F-algebra: if $a : FA \to A$ is another algebra, then there exists a unique map $fold : \mu F \to A$ such that $a \circ F_{\mu F,A}(fold) = fold$. Indeed, $fold$ is obtained by taking the fixed point of $H = \lambda h : \mu F \to A.\, a \circ F_{\mu F,A}(h)$, which exists since H is contractive: $H = \lambda h : \mu F \to A.\, a \circ G_{\mu F,A}(\mathrm{next}(h))$.

So far, perhaps the main achievement, compared to the situation where recursive types are a basic concept and well-definedness is checked syntactically, is simplicity – which is important! But using fixed points on universes to define recursive types allows us also to consider more general "higher-kinded" types and also provide a simple approach to guarded recursive dependent types, as we shall now show.

For a simple example of a "higher-kinded" type, consider

$$\lambda G\colon \blacktriangleright U \to U.\lambda A\colon U.\mathrm{fix}\, X.A \times G(X)$$

which is parameterized on any contractive type operator G and any type name A. Note that this is easily seen to be a well-defined type according to the rules given above. However, because of the use of the parameter G, standard syntactic checks of the body of the fix-expression would not allow us to conclude that the above is a well-defined guarded recursive type.

B. Example: A domain equation using dependent types

Not all recursively defined types used in type theory can be reduced to fixed points of endomaps on U. We now show how to solve guarded dependent domain equations, by considering the following example, from Altenkirch and Morris [1], of a domain equation defining a dependent type $\mathit{Lam}(n)$ of lambda terms with at most n free variables:

$$\mathit{Lam}(n) = \mathit{Fin}(n) + \mathit{Lam}(n) \times \mathit{Lam}(n) + \mathit{Lam}(n+1)$$

Here $\mathit{Fin}(n)\colon U$ is a type containing n elements representing variables. A guarded recursive version of this type is

$$\mathit{Lam}(n) = \mathit{Fin}(n) + (\blacktriangleright \mathit{Lam}(n))^2 + \blacktriangleright \mathit{Lam}(n+1)\,. \quad (7)$$

In the model of Section III global elements of $\mathit{Lam}(0)$ (i.e. maps $1 \to \mathit{Lam}(0)$) correspond to possibly infinite lambda terms in de Bruijn representation. This domain equation is not of the form (6) considered above; we cannot reduce it to taking a family of fixed points indexed by n (because we use $n+1$ on the right-hand side of (7)). Rather we must solve a single dependent domain equation. To that end, we can use the following definable generalized fixed point operator:

Proposition II.1. *Define*

$$\frac{\Gamma, x\colon A \to \blacktriangleright B \vdash t\colon A \to B}{\Gamma \vdash \mathrm{pfix}\, x.t\colon A \to B}$$

as $\mathrm{pfix}\, x.t = \mathrm{fix}\, y\colon \blacktriangleright(A \to B).t[\lambda a.y \circledast \mathrm{next}(a)/x]$. *Then*

$$\mathrm{pfix}\, x.t = t[\mathrm{next} \circ (\mathrm{pfix}\, x.t)/x].$$

Now define

$$n\colon \mathbb{N}, X\colon \mathbb{N} \to \blacktriangleright U \vdash F(X, n)\colon U$$

as $F(X,n) = \mathit{Fin}(n) + (\triangleright X(n))^2 + \triangleright X(n+1) + 1$. Then $\mathit{Lam}(n) = \mathrm{El}((\mathrm{pfix}\, X.\lambda n.F(X,n))(n))$ satisfies equation (7).

III. An Extensional Model

We now recall the model construction of [4] and describe how one can construct a universe inside this model from a given universe in $\mathbf{Set}$. We also explain how to give semantics to rule (4).

The model considered in [4] is the presheaf category $\mathbf{Set}^{\omega^{\mathrm{op}}}$. Thus a type A is modeled as a sequence of sets and functions

$$A_1 \xleftarrow{r_1^A} A_2 \xleftarrow{r_2^A} A_3 \xleftarrow{r_3^A} \dots$$

and a morphism $f\colon A \to B$ is a family of functions making the squares below commute

$$\begin{array}{ccccccc} A_1 & \xleftarrow{r_1^A} & A_2 & \xleftarrow{r_2^A} & A_3 & \xleftarrow{r_3^A} & \dots \\ \downarrow{\scriptstyle f_1} & & \downarrow{\scriptstyle f_2} & & \downarrow{\scriptstyle f_3} & & \\ B_1 & \xleftarrow{r_1^B} & B_2 & \xleftarrow{r_2^B} & B_3 & \xleftarrow{r_3^B} & \dots \end{array}$$

Our intuition is that the set A_n describes A up to the information computable in n computation steps, and the map r_n^A describes how the information obtained in $n+1$ computation steps relates to the information available after n steps. For example, the type of streams from the previous section is modeled by the object Str

$$\mathbb{N} \xleftarrow{\pi} \mathbb{N}^2 \xleftarrow{\pi} \mathbb{N}^3 \xleftarrow{\pi} \dots$$

Where the maps π are projections.

Since $\mathbf{Set}^{\omega^{\mathrm{op}}}$ is a topos it models dependent types, along with subset types for a higher order logic. Here we just describe how to model constructions specific to guarded recursion.

Given a type A, the type $\blacktriangleright A$ is defined as the sequence

$$1 \xleftarrow{!} A_1 \xleftarrow{r_1^A} A_2 \xleftarrow{r_2^A} \dots$$

The term $\mathrm{next}\colon A \to \blacktriangleright A$ is modeled as the natural transformation $\mathrm{next}_1 =!$, $\mathrm{next}_{n+1} = r_n^A$.

We say that a map $f\colon A \to A$ is *contractive* if there is a g such that $f = g \circ \mathrm{next}$. The next theorem states that contractive maps have unique fixed points.

Theorem III.1 ([4])**.** *Let A, B be objects of $\mathbf{Set}^{\omega^{\mathrm{op}}}$, and let $f\colon B \times \blacktriangleright A \to A$. There is a unique morphism $\mathrm{fix} f\colon B \to A$ such that*

$$\mathrm{fix} f = f \circ \langle \mathrm{id}_B, \mathrm{next}_A \circ \mathrm{fix} f \rangle$$

From Theorem III.1 one can derive a fixed-point operator $\mathrm{fix}\colon (\blacktriangleright A \to A) \to A$. To give some intuition for how fixed points are computed, we recall the construction for the case of $B = 1$. In that case the input is a function $f\colon \blacktriangleright A \to A$, i.e. a family of maps $f_1\colon 1 \to A_1$, $f_{n+1}\colon A_n \to A_{n+1}$. The fixed point is defined as

$$(\mathrm{fix} f)_n = f_n \circ f_{n-1} \circ \dots f_1 \colon 1 \to A_n.$$

The intuition for $\mathrm{fix} f$ is that to compute the n'th approximation of the fixed point, we apply f^n to an arbitrary argument.

In [4] we also proved the existence of solutions to a large class of recursive domain equations. Recall that an enrichment of a functor $F\colon \mathbf{Set}^{\omega^{\mathrm{op}}} \to \mathbf{Set}^{\omega^{\mathrm{op}}}$ is a family of maps

$$F_{X,Y}\colon (X \to Y) \to FX \to FY$$

internalizing the action of F on morphisms in a precise sense [12], and say that F is locally contractive if it has an enrichment such that each $F_{X,Y}$ is contractive. Then every locally contractive functor has a unique fixed point up to isomorphism, and moreover this fixed point is at the same time an initial algebra and final coalgebra. This result generalizes to mixed-variance functors and (for dependent types) functors on slices [4].

For example, the expression $F(X) = \mathbb{N} \times \blacktriangleright X$ defines a locally contractive functor. Since products and natural numbers are given pointwise in $\mathbf{Set}^{\omega^{\mathrm{op}}}$, the type Str satisfies $\mathit{Str} \cong \mathbb{N} \times \blacktriangleright \mathit{Str}$ and so is the unique solution to the equation given by F.

In fact, any type expression in simply typed lambda calculus, as well as many expressions in the dependent type theory of toposes define locally contractive functors as long as any occurrence of X is under a $\blacktriangleright$ operator. See [4] for a precise statement.

For A in $\mathbf{Set}^{\omega^{\mathrm{op}}}$, by the n'th approximation of A we mean the finite sequence

$$A_1 \xleftarrow{r_1^A} A_2 \xleftarrow{r_2^A} \dots \xleftarrow{r_{n-1}^A} A_n$$

The key property that locally contractive functors satisfy is that they are *productive* in the sense that the $(n+1)$'th approximation of FX only depends on the n'th approximation of X (up to isomorphism). Productivity implies the existence of unique fixed points: the n'th approximation of the fixed point can be computed by applying F^n to any object. Thus fixed points on the type level are computed essentially the same way as fixed points on the term level. We now show how productivity can be captured using universes.

A. Universes in $\mathbf{Set}^{\omega^{\mathrm{op}}}$

We now assume that we are given a universe U in $\mathbf{Set}$ and construct a universe V in $\mathbf{Set}^{\omega^{\mathrm{op}}}$. By a universe in $\mathbf{Set}$ we mean a set U of sets containing the set of natural numbers and being closed under subsets, as well as dependent products and sums indexed over sets in U, as is the case for example for a Grothendieck universe. A set A is *small* if $A \in U$. Following [8] we construct V:

$$V_1 \xleftarrow{r_1^V} V_2 \xleftarrow{r_2^V} V_3 \xleftarrow{r_3^V} \dots$$

by defining $V_1 = U$ and V_{n+1} as the set

$$\{(X_1, \ldots, X_{n+1}, f_1, \ldots, f_n) \mid \forall i. X_i \in U, f_i \colon X_{i+1} \to X_i\}$$

the restriction maps are defined by

$$r_n^V(X_1, \ldots, X_{n+1}, f_1, \ldots f_n) = (X_1, \ldots, X_n, f_1, \ldots f_{n-1}).$$

In Section IV we describe the universal properties that this universe satisfies, but for now we just note the intuition: if we say an object A of $\mathbf{Set}^{\omega^{\mathrm{op}}}$ is *small* if each A_i is small, then an element in V_n is a finite approximation of a small object. Note also that there is a bijective correspondence between small objects of $\mathbf{Set}^{\omega^{\mathrm{op}}}$ and global elements of V, i.e., morphisms $1 \to V$. Maps $F \colon V \to V$ thus induce constructions on small objects and the contractive maps correspond to the constructions that are productive in the sense of the previous subsection. The process of computing the unique fixed point of F thus corresponds exactly to the process described at the end of the previous subsection.

Define the map $\triangleright \colon \blacktriangleright V \to V$ as

$$\underset{1}{\triangleright}(\star) = 1$$
$$\underset{n+1}{\triangleright}(X_1 \ldots X_n, f_1 \ldots f_{n-1}) = (1, X_1 \ldots X_n, !, f_1 \ldots f_{n-1})$$

This map internalizes $\blacktriangleright$ in the sense that if the small object A corresponds to $\bar{A} \colon 1 \to V$ then $\blacktriangleright A$ corresponds to the composition

$$1 \xrightarrow{\bar{A}} V \xrightarrow{\mathrm{next}_V} \blacktriangleright V \xrightarrow{\triangleright} V$$

Since U is closed under products, we can similarly define a map $V \times V \to V$ capturing products of small sets. Using this, $\triangleright$ and $\mathbb{N} \in U$ we can define $F(X) = \mathbb{N} \times \blacktriangleright X$ as a contractive map $V \to V$. The unique fixed point of F is Str.

B. Dependent types

In the set theoretic model of dependent type theory a dependent type over A is modeled as a family of sets indexed over A. Such a family can equivalently be represented by a map $p \colon B \to A$. Varying the definition of the previous section slightly, we now say a map p is *small* if each fibre $B_a = \{b \colon B \mid pb = a\}$ is *isomorphic* to an element in U. Let $\mathrm{El} \colon E \to U$ where $E = \coprod_{X \in U} X$ be the dependent projection. Then p is small iff it fits in a pullback diagram

$$\begin{array}{ccc} B & \longrightarrow & E \\ {\scriptstyle p}\downarrow & \lrcorner & \downarrow{\scriptstyle \mathrm{El}} \\ A & \xrightarrow{\bar{B}} & U \end{array}$$

We call $\bar{B}$ a code of B. Codes are usually not unique.

Since $\mathbf{Set}^{\omega^{\mathrm{op}}}$ is locally cartesian closed we can interpret dependent type theory in it. A type dependent on A is modeled as a map $B \to A$. We can define the collection of elements in the universe as a dependent type

$$\begin{aligned} \mathrm{El}_V &\colon E_V \to V \\ (E_V)_1 &= E \\ (E_V)_{n+1} &= \textstyle\coprod_{(X_1, \ldots, X_{n+1}, f_1, \ldots, f_n) \in V_{n+1}} X_{n+1} \end{aligned}$$

with restriction maps

$$\begin{aligned} &r_n^V((X_1, \ldots, X_{n+1}, f_1, \ldots, f_n), x) \\ &\quad = ((X_1, \ldots, X_n, f_1, \ldots, f_{n-1}), f_n(x)) \,. \end{aligned}$$

Theorem III.2 (Hofmann and Streicher [8])**.** *Let $p \colon B \to A$ be a morphism in $\mathbf{Set}^{\omega^{\mathrm{op}}}$. Say p is small if each p_n is small. Then p is small iff it fits in a pullback diagram of the form*

$$\begin{array}{ccc} B & \longrightarrow & E_V \\ {\scriptstyle p}\downarrow & \lrcorner & \downarrow{\scriptstyle \mathrm{El}_V} \\ A & \longrightarrow & V \end{array}$$

For a category $\mathbb{C}$ and an object A in $\mathbb{C}$, we write $\mathbb{C}/A$ for the slice category. In [4] we generalized the definition of $\blacktriangleright$ to dependent types by defining a functor $\blacktriangleright_A \colon \mathbf{Set}^{\omega^{\mathrm{op}}}/A \to \mathbf{Set}^{\omega^{\mathrm{op}}}/A$ mapping $p_B \colon B \to A$ to the map $p_{\blacktriangleright B}$ fitting in the following pullback diagram.

$$\begin{array}{ccc} \blacktriangleright_A B & \longrightarrow & \blacktriangleright B \\ {\scriptstyle p_{\blacktriangleright B}}\downarrow & \lrcorner & \downarrow{\scriptstyle \blacktriangleright p_B} \\ A & \xrightarrow{\mathrm{next}} & \blacktriangleright A \end{array} \tag{8}$$

where $\blacktriangleright p_B$ is the functorial action of $\blacktriangleright$ applied to p_B. We usually omit the subscript and simply write $\blacktriangleright$ also for the operation on slices.

Theorem III.3. *If $\bar{B} \colon A \to V$ is a code for $p_B \colon B \to A$ then $\triangleright \circ \mathrm{next} \circ \bar{B} \colon A \to V$ is a code for $p_{\blacktriangleright B}$.*

Theorem III.3 states that $\triangleright \colon \blacktriangleright V \to V$ also internalizes $\blacktriangleright$ on dependent types. Moreover, it shows the soundness of rule (5) (up to coherence issues to be addressed in Section IV) as we now describe. In the model, a term $\Gamma \vdash A \colon U$ is interpreted as a map $[\![A]\!] \colon \Gamma \to V$, and $\Gamma \vdash \mathrm{El}(A) \colon \mathrm{Type}$ is interpreted as $[\![A]\!]^* \mathrm{El}_V$ where $[\![A]\!]^*$ denotes pullback along $[\![A]\!]$. We can interpret (4) as $[\![\triangleright A]\!] = \triangleright \circ [\![A]\!]$. Soundness of (5) up to isomorphism can be proved as

$$\begin{aligned} [\![\mathrm{El}(\triangleright(\mathrm{next}(A)))]\!] &= (\triangleright \circ \mathrm{next} \circ [\![A]\!])^* \mathrm{El} \\ &\cong \blacktriangleright_\Gamma [\![\mathrm{El}(A)]\!] \qquad \text{Theorem III.3} \end{aligned}$$

IV. Intensional Models

Before we generalize the model of the previous section to a model construction on intensional models we fix a notion of model of type theory. For this, we follow recent literature on homotopy type theory, e.g. [2, 6, 11, 19], A type dependent on A is still modeled as a map $B \to A$, but unlike locally cartesian closed categories, not all maps are used to denote dependent types. Following the recent literature on homotopy type theory, we call the class of maps denoting dependent types *fibrations*, but other authors use the words *display maps* or *dependent projections*. For example, in the groupoid model of type theory [9] a type dependent on A is modelled as an isofibration $p\colon B \to A$, i.e., a homomorphism such that for any $b \in B$, $f\colon a \to pb$ there exists a map $g\colon b' \to b$ such that $pg = f$.

Definition IV.1 ([19])**.** *A* type-theoretic fibration category *is a category $\mathbb{C}$ with a terminal object 1 together with a collection of morphisms called fibrations closed under composition and containing all isomorphisms. Say an object A is fibrant if the unique map $!\colon A \to 1$ is a fibration. The following conditions must be satisfied*

- *All pullbacks of fibrations between fibrant objects exist and are fibrations*
- *For any fibration $f\colon A \to B$ between fibrant objects the reindexing functor*

$$\mathbb{C}/B \to \mathbb{C}/A$$

has a partial right adjoint defined on fibrations and whose values are fibrations.

Call a map i an acyclic cofibration *if it has the left lifting property wrt all fibrations p, i.e. if $pf = gi$ then there exists an h such that $hi = f$, $ph = g$.*

- *If $p\colon A \to B$ is a fibration between fibrant objects then its diagonal $A \to A \times_B A$ factors as an acyclic cofibration followed by a fibration:*

$$A \xrightarrow{\ r\ } P_B A \longrightarrow A \times_B A$$

Moreover, this factorization should be preserved by reindexing along all maps into B.

The object $A \times_B A$ is the one obtained by pullback of p along p. This is the product in the category $\mathbb{C}/B$. The fibration $P_B A \to A \times_B A$ models the dependent type $b\colon B, x\colon A(b), y\colon A(b) \vdash \mathrm{Id}_{A(b)}(x, y)$. In the groupoid model, if $B = 1$ the groupoid $P_B A$ is the discrete groupoid of morphisms of A. As a degenerate example, note that any locally cartesian closed category can be seen as a type-theoretic fibration category where all morphisms are fibrations. Examples also include the category of simplicial sets [11] and a type-theoretic fibration category constructed from the syntax of dependent type theory [19].

Note that although type-theoretic fibration categories are not locally cartesian closed, the products $p \times q$ and exponentials p^q do exist in slices over fibrant objects for fibrations p and q.

Definition IV.2. *A* universe *in a type-theoretic fibration category $\mathbb{C}$ is a fibration $\mathrm{El}\colon E \to U$ in $\mathbb{C}$ such that U is fibrant and such that $\mathbb{C}$ with the family of fibrations arising as pullbacks of* El *is a type-theoretic fibration category.*

A fibration in $\mathbb{C}$ is small (with respect to El) if it can be seen as a pullback of El, and an object A is small if $A \to 1$ is small.

Example IV.3. *In* $\mathbf{Set}$ *given a universe U, defining* El *as in Section III-B gives a universe in the sense of Definition IV.2. Say a groupoid X is small if the sets of objects and morphisms are both in U, and consider the groupoid $Grpd$ of all small groupoids and isomorphisms between them. This defines a universe in the category of groupoids: the groupoid E_{Grpd} is obtained by applying the Grothendieck construction to the inclusion from $Grpd$ to the (large) category* $\mathbf{Grpd}$ *of all groupoids. Explicitly, the objects are pairs (X, x) such that $x \in X \in Grpd$ and morphisms (X, x) to (Y, y) are pairs (f, g) such that $f\colon X \to Y$ and $g\colon x \to f^{-1}(y)$ in X. One can also construct the universe $Grpd_\Delta$ of small discrete groupoids similarly.*

In the example of the groupoid model, all small fibrations are split.

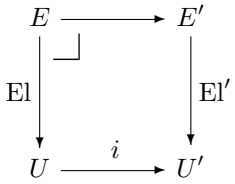
Universes are used for two purposes: to model universes in type theory and to solve coherence issues [7], i.e., the problem that pullback, which models substitution, usually does not commute with constructions such as dependent products and is not associative. We follow a recent approach for dealing with the coherence issue developed independently by Streicher [20] and Voevodsky [11]. Essentially the idea is to model types depending on A as morphisms $A \to U$. Operations on types are then modeled as chosen maps on the universe. For example to model binary products, one considers the generic fibered product $\pi_1^*\mathrm{El} \times_{U\times U} \pi_2^*\mathrm{El} \to U \times U$. This can be seen as the interpretation of the dependent type $X, Y\colon U \vdash X \times Y$. Choose a code $\bar{\times}\colon U \times U \to U$ for $\pi_1^*\mathrm{El} \times_{U\times U} \pi_2^*\mathrm{El}$, and define $[\![\Gamma \vdash A \times B]\!] = \bar{\times} \circ \langle [\![A]\!], [\![B]\!] \rangle\colon [\![\Gamma]\!] \to U$. For dependent products and sums one considers the object $U^{(1)}$ defined as the exponential

$$(U \times U \to U)^{\mathrm{El}}$$

in the category $\mathbb{C}/U$. The object $U^{(1)}$ is the interpretation of the generic dependent type context $X\colon U, Y\colon X \to U$. Dependent products are modeled by choosing a code $U^{(1)} \to U$ for the generic dependent type $X\colon U, Y\colon X \to U \vdash \prod x\colon X.Y(x)$. We refer to [11, 19] for details.

A *universe structure* is a universe together with a choice of codes implementing dependent products, sums, unit and

identity types. A *universe embedding* is a monomorphism $U \to U'$ between universes which fits in a pullback



(i.e. i is a code for El) and which commutes with the universe structure.

Definition IV.4 ([19])**.** *A* model of MLTT_U *(Martin-Löf type theory with one universe) is a type-theoretic fibration category with a universe embedding $i\colon U \to U'$ such that U is small with respect to U'.*

The universe U' is used to solve the coherence issue and U is used to model the universe of the type theory. More precisely, a type in context Γ is modeled as a map $\Gamma \to U'$, and a judgement $\Gamma \vdash A\colon U$ as a map $[\![A]\!]\colon \Gamma \to U$. The elements of a term in the universe is modeled as $[\![\mathrm{El}(A)]\!] = i \circ [\![A]\!]$. Terms $\Gamma \vdash t\colon B$ are modeled as sections of $[\![B]\!]^*\mathrm{El}'$.

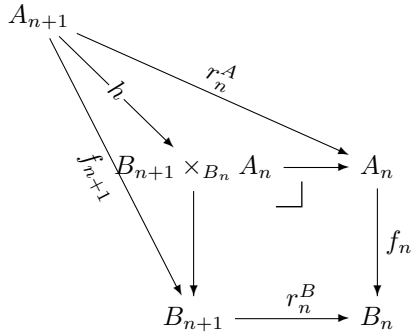
The above approach can of course be extended to type theories with more than one universe. Generally, in these cases we need to assume one more universe in the semantic setup than in the type theory. The exception is for type theories with an increasing family of universes (such as extended calculus of constructions [15]), where one does not need a universe containing all universes in the family [19].

Example IV.5. *Given two Grothendieck universes U_0, U_1 such that $U_0 \in U_1$ and $U_0 \subset U_1$ the inclusion is a universe embedding, and also induces universe embeddings between the corresponding universes of groupoids.*

V. A GENERAL MODEL CONSTRUCTION

In this section we assume given a model $\mathbb{C}$ of MLTT_U as in Definition IV.4. We generalize the construction of [3] by equipping $\mathbb{C}^{\omega^{\mathrm{op}}}$ with the structure of a model of type theory. Following [19] we use the fibration category structure on $\mathbb{C}^{\omega^{\mathrm{op}}}$ given by Reedy fibrations.

Definition V.1. *A morphism $f\colon A \to B$ in $\mathbb{C}^{\omega^{\mathrm{op}}}$ is a* Reedy fibration *if $f_1\colon A_1 \to B_1$ is a fibration and for each n, the induced map h into the pullback*



is a fibration.

One can prove that if f is a Reedy fibration, then each f_n is a fibration. Recall the following from [19].

Theorem V.2 (Shulman)**.** *Let $\mathbb{C}$ be a model of MLTT_U. Then $\mathbb{C}^{\omega^{\mathrm{op}}}$ equipped with Reedy fibrations is also a model of MLTT_U. If, moreover $\mathbb{C}$ is a model of univalence, so is $\mathbb{C}^{\omega^{\mathrm{op}}}$.*

The interpretation of type theory in $\mathbb{C}^{\omega^{\mathrm{op}}}$ models contexts as fibrant objects. Note that the fibrant objects are the sequences such that B_1 is fibrant and each $r_n^B\colon B_{n+1} \to B_n$ is a fibration. Since fibrations in $\mathbb{C}$ model dependent types we can think of types in the model $\mathbb{C}^{\omega^{\mathrm{op}}}$ as infinite contexts $a_1\colon A_1, a_2\colon A_2(a_1), a_3\colon A_3(a_1, a_2), \ldots$. A type depending on A is a Reedy fibration on A, and the condition of Definition V.1 can be understood as requiring that A_{n+1} is a type dependent on B_{n+1} and A_n:

$$b_n\colon B_n, a_n\colon A_n(b_n), b_{n+1}\colon B_{n+1}(b_n) \vdash A_{n+1}(b_n, a_n, b_{n+1})\colon \mathrm{Type}$$

We recall Shulman's construction of the universe because it is needed in the next section. For the construction, we first assume given a universe $\mathrm{El}\colon E \to U$ in $\mathbb{C}$ and construct a Reedy fibration $\mathrm{El}^V\colon E^V \to V$ in $\mathbb{C}^{\omega^{\mathrm{op}}}$ inductively by

$$V_1 = U \qquad E_1^V = E \qquad \mathrm{El}_1^V = \mathrm{El}$$

$$V_{n+1} = (V_n \times U \xrightarrow{\pi_1} V_n)^{\mathrm{El}_n^V}$$

where in the latter definition, the exponent is defined in the slice $\mathbb{C}/V_n$. For the Reedy fibration we need a fibration $\mathrm{El}_{n+1}^V\colon E_{n+1}^V \to V_{n+1} \times_{V_n} E_n^V$, and this is defined as $\mathrm{El}_{n+1}^V = (\pi_2 \mathrm{ev})^*\mathrm{El}$ where $\mathrm{ev}\colon V_{n+1} \times_{V_n} \mathrm{El}_n^V \to V_n \times U$ is the evaluation map.

Proposition V.3 (Shulman [19])**.** *Let $\mathrm{El}\colon E \to U$ be a given universe in $\mathbb{C}$ and say that a map $p\colon A \to B$ in $\mathbb{C}^{\omega^{\mathrm{op}}}$ is a* Reedy small-fibration *if p_1 as well as each induced map $A_{n+1} \to A_n \times_{B_n} B_{n+1}$ is small with respect to* El*. Then $p\colon A \to B$ is a Reedy small-fibration if and only if it arises as a pullback of El^V along some map $\bar{p}\colon B \to V$.*

Shulman also proves that any universe embedding $i\colon U \to U'$ in $\mathbb{C}$ induces a universe embedding $j\colon V \to V'$ between universes in $\mathbb{C}^{\omega^{\mathrm{op}}}$. Of course $j_1 = i$. To describe j_{n+1}, note first that a map $X \to V_{n+1}$ corresponds to a pair of maps $a\colon X \to V_n$, $b\colon a^*E_n^V \to U$. Thus we can define j_{n+1} using the Yoneda lemma by defining a map

$$\mathrm{Hom}_{\mathbb{C}}(X, V_{n+1}) \to \mathrm{Hom}_{\mathbb{C}}(X, V'_{n+1})\,.$$

which maps (a, b) to the pair consisting of $j_n \circ a$ and

$$(j_n \circ a)^* E_n^{V'} \cong a^* E_n^V \xrightarrow{b} U \xrightarrow{i} U'$$

(using the property $j_n^* E_n^{V'} \cong E_n^V$ which can be proved by induction.)

A. Modeling guarded recursion

The definition of $\blacktriangleright$ generalizes immediately from $\mathbf{Set}^{\omega^{\mathrm{op}}}$ to $\mathbb{C}^{\omega^{\mathrm{op}}}$: we first define $\blacktriangleright$ as a functor on $\mathbb{C}^{\omega^{\mathrm{op}}}$ mapping A to

$$1 \xleftarrow{!} A_1 \xleftarrow{r_1^A} A_2 \xleftarrow{r_2^A} \dots$$

and then generalize to slices over $\mathbb{C}^{\omega^{\mathrm{op}}}$ using (8).

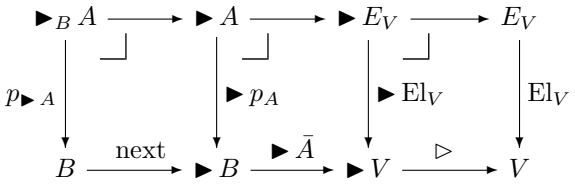
Proposition V.4. *If B is Reedy fibrant, so is $\blacktriangleright B$. If $p_B \colon B \to A$ is a Reedy fibration and A is Reedy fibrant, then $p_{\blacktriangleright B}$ is a Reedy fibration.*

Proof: Note first that $\blacktriangleright p_B$ is a Reedy fibration if p_B is. This implies the first statement because $\blacktriangleright 1 = 1$. For the second part, note that from Theorem V.2 it follows that Reedy fibrations are closed under pullbacks along all maps between fibrant objects, so $p_{\blacktriangleright B} = \mathrm{next}^*(\blacktriangleright p_B)$ is a Reedy fibration. ∎

To model (4) we need a map $\triangleright \colon \blacktriangleright V \to V$ internalizing $\blacktriangleright$. One way to obtain this is to observe that since $\blacktriangleright \mathrm{El}_V \colon \blacktriangleright E_V \to \blacktriangleright V$ is a Reedy small-fibration, by Proposition V.3 it has a code $\triangleright \colon \blacktriangleright V \to V$. The following is then a generalization of Theorem III.3.

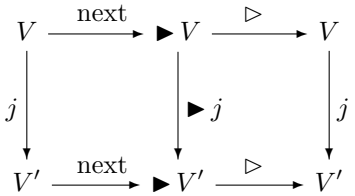
Proposition V.5. *If $\bar{A} \colon B \to V$ is a code for $p_A \colon A \to B$, then $\triangleright \circ \mathrm{next} \circ \bar{A}$ is a code for $\blacktriangleright_B p_A$.*

Proof: Using naturality of next it suffices to observe that the following is a pullback (using that $\blacktriangleright$ preserves pullbacks).



∎

We now show that $\triangleright$ can be chosen in such a way that it commutes with the universe embedding. To see why this is needed recall that judgements $\Gamma \vdash A \colon U$ are interpreted as maps $[\![A]\!] \colon [\![\Gamma]\!] \to V$, judgements $\Gamma \vdash A \colon \mathrm{Type}$ as maps $[\![A]\!] \colon [\![\Gamma]\!] \to V'$, and moreover that $[\![\mathrm{El}(A)]\!] = j \circ [\![A]\!]$ where $j \colon V \to V'$ is the universe embedding. Thus for soundness of (5) we need commutativity of the outer square below



(9)

Note that the square on the left commutes by naturality of next.

Define $\triangleright_1 \colon 1 \to V_1 = U$ as the code for the unit (henceforth denoted $\bar{1}$), which is part of the universe structure. For the inductive definition of $\triangleright_{n+1} \colon V_n \to V_{n+1}$, again we define a map

$$\mathrm{Hom}_{\mathbb{C}}(X, V_n) \to \mathrm{Hom}_{\mathbb{C}}(X, V_{n+1})$$

natural in X and appeal to the Yoneda lemma. First define $\triangleright_2 \colon V_1 = U \to V_2$ by mapping $a \colon X \to U$ to the pair consisting of $\bar{1} \circ ! \colon X \to U$ and

$$(\bar{1} \circ !)^* E \cong X \xrightarrow{\ a\ } U$$

Define $\triangleright_{n+2} \colon V_{n+1} \to V_{n+2}$ by mapping a pair $(a \colon X \to V_n, b \colon a^* E_n^V \to U)$ to the pair consisting of $\triangleright_{n+1} \circ a$ and

$$(\underset{n+1}{\triangleright} \circ a)^* E_{n+1}^V \cong a^* E_n^V \xrightarrow{\ b\ } U$$

using the isomorphism $\triangleright_{n+1}^* E_{n+1}^V \cong E_n^V$ which can be constructed by induction.

Theorem V.6. *The family $(\triangleright_n)_n$ as defined above defines a map $\blacktriangleright V \to V$ which is a code of $\blacktriangleright \mathrm{El}^V$. Moreover, if $j \colon V \to V'$ is a universe embedding in $\mathbb{C}^{\omega^{\mathrm{op}}}$ induced by a universe embedding $i \colon U \to U'$ then the diagram (9) commutes. As a consequence, Rules* (4) *and* (5) *can be modeled soundly in $\mathbb{C}^{\omega^{\mathrm{op}}}$.*

B. Term level fixed points

Terms of the form $\Gamma, x \colon \blacktriangleright A \vdash t \colon A$ are interpreted as morphisms $\blacktriangleright A \to A$ in the slice category $\mathbb{C}^{\omega^{\mathrm{op}}}/\Gamma$. The next proposition states that guarded recursion can be modeled soundly in $\mathbb{C}^{\omega^{\mathrm{op}}}$.

Theorem V.7. *Let $f \colon \blacktriangleright A \to A$ be a morphism in $\mathbb{C}^{\omega^{\mathrm{op}}}/\Gamma$. Then there is a unique map $\mathrm{fix} f \colon 1_\Gamma \to A$ such that $\mathrm{fix} f = f \circ \mathrm{next} \circ \mathrm{fix} f$. As a consequence, Rules* (1) *and* (2) *can be modeled soundly in $\mathbb{C}^{\omega^{\mathrm{op}}}$.*

We can also prove that fixed points are unique internally in the model.

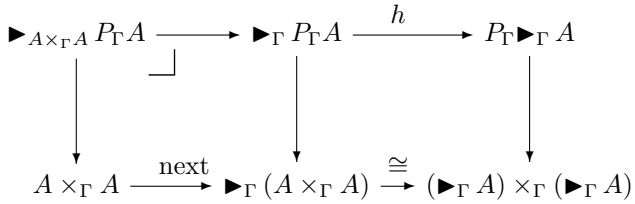
Theorem V.8. *Rule* (3) *can be modeled soundly in $\mathbb{C}^{\omega^{\mathrm{op}}}$.*

For the proof we need the following contractiveness principle for next.

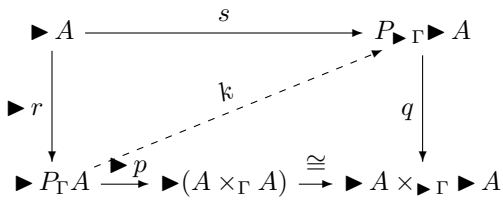
Lemma V.9. *The following principle is sound in $\mathbb{C}^{\omega^{\mathrm{op}}}$*

$$\frac{\Gamma, x, y \colon A \vdash p \colon \blacktriangleright \mathrm{Id}_A(x, y)}{\Gamma, x, y \colon A \vdash \mathrm{NC}(p) \colon \mathrm{Id}_{\blacktriangleright A}(\mathrm{next}(x), \mathrm{next}(y))}$$

Proof (outline). The term NC is modeled as the top horizontal composite in the following diagram. This has the right type because the bottom composite is next × next.



The left most square is a generalisation of (8). To construct h first use the left lifting property (Definition IV.1) to obtain k in the following diagram



here (r, p) is the factorization of the diagonal as in Definition IV.1 and likewise (s, q). Since q is a fibration, to apply the left lifting property, we need $\blacktriangleright r$ an acyclic cofibration. This it is indeed, since acyclic cofibrations in the Reedy model structure are given pointwise. Now, h can be defined as the result of pulling back k along $\mathrm{next}\colon \Gamma \to \blacktriangleright\Gamma$ using that identity types commute with reindexing. ■

Proof of Theorem V.8. Using Lemma V.9 we can construct $\mathrm{UFP}(p)$ in the type theory as a fixed point of a map $\blacktriangleright \mathrm{Id}_A(y, \mathrm{fix}\, x.f) \to \mathrm{Id}_A(y, \mathrm{fix}\, x.f)$ constructed as follows: if $q\colon \blacktriangleright \mathrm{Id}_A(y, \mathrm{fix}\, x.f(x))$ then $\mathrm{NC}(q)\colon \mathrm{Id}_{\blacktriangleright A}(\mathrm{next}(y), \mathrm{next}(\mathrm{fix}\, x.f(x)))$. Since $f(\mathrm{next}(\mathrm{fix}\, x.f(x)))$ is externally equal to $\mathrm{fix}\, x.f(x)$ we thus get an element of $\mathrm{Id}_A(f(\mathrm{next}(y)), \mathrm{fix}\, x.f(x))$, and we can apply transitivity to this and p to get an element of type $\mathrm{Id}_A(y, \mathrm{fix}\, x.f(x))$ as desired. ■

VI. Type constructors vs. universe maps

We now turn to the question of the relationship between type constructions in the sense of constructions defined on objects of a category, and maps on the universe.

Definition VI.1. *Let $\mathbb{C}$ be a type-theoretic fibration category and let $\mathrm{El}\colon E \to U$ be a universe in $\mathbb{C}$. A* small object operation *is a family of operations indexed over fibrant objects B mapping small fibrations $p_A\colon A \to B$ to small fibrations $F(p_A)\colon FA \to B$ such that*

- *F commutes with reindexing along all maps*
- *F maps isomorphic objects to isomorphic objects*

We say that $\bar{F}\colon U \to U$ is a code for F if $\bar{F} \circ \bar{A}$ is a code for FA whenever $\bar{A}\colon B \to U$ is a code for A.

Proposition VI.2. *Every small object operation F has a code $\bar{F}$. A fixed point for $\bar{F}$ is the code of a fixed point (up to isomorphism) for F.*

Proof: Define $\bar{F}$ to be a code for $F(\mathrm{El})$. Suppose $\bar{A}\colon 1 \to U$ is a fixed point for $\bar{F}\colon U \to U$. Then

$$F(\bar{A}^*\mathrm{El}) \cong \bar{A}^* F(\mathrm{El}) \cong (\bar{F} \circ \bar{A})^*\mathrm{El} = \bar{A}^*\mathrm{El}$$

■

For example, assuming that $\mathbb{C}$ has a natural numbers object which is fibrant, we can define a small object operation F by $F(X) = \Delta(\mathbb{N}) \times \blacktriangleright X$, where $\Delta(\mathbb{N})$ is the object whose components are all $\mathbb{N}$ and whose restriction maps are all identities. If $\mathbb{N}$ is small, by Proposition V.3 so is $\Delta(\mathbb{N})$ and we write $\overline{\Delta\mathbb{N}}$ for the code of $\Delta(\mathbb{N})$. We can define a contractive code for F as follows

$$\bar{F}(X) = \overline{\Delta\mathbb{N}} \bar{\times} \triangleright(\mathrm{next}(X))\,.$$

By Proposition VI.2, the unique fixed point of $\bar{F}$ is the code of a fixed point for F.

Remark VI.3. *Some readers may find it helpful to think of the above by analogy to Scott's approach to domains using information systems [18, 22]: Recall that Scott domains can be represented as information systems and that there is a (large) cpo of information systems, such that a solution to a recursive domain equation can be obtained by taking a fixed point of an endomap on the cpo of information systems. This yields a fixed point up to equality; when combined with the representation of cpos as information systems one gets solutions up to isomorphism of cpos.*

A. The groupoid model

Proposition VI.2 gives the existence of fixed points for small object operations with contractive codes. Even though the codes of such operations have unique fixed points, the small object operations themselves need not have unique fixed points, since the code of a fixed point for F need not be a fixed point for $\bar{F}$. A related question is that of giving an inverse of Proposition VI.2: is any map $\bar{F}\colon U \to U$ a code for a small object operation? In general the answer is 'no': the mapping of $p_A\colon A \to B$ to $(\bar{F} \circ \bar{A})^*\mathrm{El}$, where $\bar{A}$ is a code for A is generally not well-defined. In particular, for the set theoretic model of Section III different choices of $\bar{A}$ may give objects that are not even isomorphic.

The universe of discrete groupoids in the groupoid model is much more well-behaved as we now explain. We denote by $\mathrm{El}_{V_\Delta}\colon E_{V_\Delta} \to V_\Delta$ the universe in $\mathbf{Grpd}^{\omega^{\mathrm{op}}}$ obtained by applying the construction of Theorem V.2 to the universe of small discrete groupoids $Grpd_\Delta$ of Example IV.3.

Lemma VI.4. *Let A be an object of $\mathbf{Grpd}^{\omega^{\mathrm{op}}}$ and let $\bar{B}, \bar{C}\colon A \to V_\Delta$ be morphisms. The following two statements are equivalent*

- *$\bar{B}^*\mathrm{El}_{V_\Delta}$ is isomorphic to $\bar{C}^*\mathrm{El}_{V_\Delta}$ as objects of the slice category over A*
- *The type $\prod x\colon A.\mathrm{Id}_{V_\Delta}(\bar{B}(x), \bar{C}(x))$ is inhabited, i.e., $\bar{B}$ and $\bar{C}$ are pointwise internally equal.*

Theorem VI.5. *Any morphism $\bar{F}\colon V_\Delta \to V_\Delta$ is the code of a small object operation. If A is a small fibrant object and $\bar{A}\colon 1 \to V_\Delta$ is the code for A then A is a fixed point for F up to isomorphism, i.e., $F(A) \cong A$ if and only if $\bar{A}$ is a fixed point $\bar{F}$ up to internal equality.*

Proof: The second statement follows directly from Lemma VI.4. To prove the first statement we must show that

if $\bar{B}^*\mathrm{El} \cong \bar{C}^*\mathrm{El}$ also $(\bar{F} \circ \bar{B})^*\mathrm{El} \cong (\bar{F} \circ \bar{C})^*\mathrm{El}$. But this follows because if $\prod x\colon A.\mathrm{Id}_{V_\Delta}(\bar{B}(x), \bar{C}(x))$ is inhabited so is $\prod x\colon A.\mathrm{Id}_{V_\Delta}(\bar{F} \circ \bar{B}(x), \bar{F} \circ \bar{C}(x))$. ∎

Corollary VI.6. *Let F be a small object operation on $\mathbf{Grpd}^{\omega^{\mathrm{op}}}$ and suppose F has a contractive code $\bar{F}\colon V_\Delta \to V_\Delta$. Then F has a fixed point which is unique up to isomorphism.*

Proof: Existence of the fixed point is the second statement of Proposition VI.2. If A is a fixed point of F and $\bar{A}$ is a code for A, then $\bar{A}$ is a fixed point of $\bar{F}$ up to internal equality. By Rule (3) (which holds in the model by Theorem V.8), this implies that $\bar{A}$ is internally equal to $\mathrm{fix}(\bar{F})$, which by Lemma VI.4 implies that A is isomorphic to $\mathrm{fix}(\bar{F})^*\mathrm{El}$. ∎

If F in addition to the assumptions of Corollary VI.6 has a functorial action (possibly of mixed variance), and is locally contractive in the sense of [3] then universal properties can be derived for the unique solution using the techniques of [4, Section 7]. But not all small object operations can be equipped with functorial actions.

VII. Conclusion and Future Work

We have shown that, for any model $\mathbb{C}$ of type theory with a universe, $\mathbb{C}^{\omega^{\mathrm{op}}}$ is a model of type theory with guarded recursion in which guarded recursive types can be obtained simply as fixed points of contractive functions on the universe of types. Moreover, we have proved that for the groupoid model $\mathbf{Grpd}$, there is a one-to-one correspondence between fixed points of functions on the universe of discrete types in $\mathbf{Grpd}^{\omega^{\mathrm{op}}}$ and fixed points of small object operations on types.

In future work we plan to investigate the relationship between guarded recursive types and coinductive types. For some models $\mathbb{C}$ it is the case that elements of a coinductive type in $\mathbb{C}$ correspond to global elements of a corresponding guarded recursive type in $\mathbb{C}^{\omega^{\mathrm{op}}}$, suggesting that guarded recursive types can be used to as a crutch for higher-order programming with coinductive types.

Future work also includes extending implementations of type theory with guarded recursive types so as to allow for experimentation with larger case studies of applications of guarded recursion.

Acknowledgments: We thank Andreas Abel, Bob Atkey, Benno van den Berg, Nils Anders Danielsson, Daniel Gustafsson, Nicolas Pouillard, Michael Shulman, and Thomas Streicher for stimulating discussions regarding the work presented in this paper.

References

[1] Thorsten Altenkirch and Peter Morris. Indexed containers. In *LICS*, pages 277–285. IEEE Computer Society, 2009.

[2] S. Awodey and M.A. Warren. Homotopy theoretic models of identity types. *Math. Proc. Camb. Phil. Soc.*, 146(45), 2009.

[3] L. Birkedal, R. Møgelberg, J. Schwinghammer, and K. Støvring. First steps in synthetic guarded domain theory: step-indexing in the topos of trees. In *Proceedings of LICS*, 2011.

[4] L. Birkedal, R. Møgelberg, J. Schwinghammer, and K. Støvring. First steps in synthetic guarded domain theory: step-indexing in the topos of trees. *Logical Methods in Computer Science*, 8(4), October 2012.

[5] N.A. Danielsson. Beating the productivity checker using embedded languages. In *Proceedings of PAR*, Electronic Proceedings of Theoretical Computer Science, 2010.

[6] N. Gambino and R. Garner. The identity type weak factorization system. *Theoretical Computer Science*, 409, 2008.

[7] M. Hofmann. On the interpretation of type theory in locally cartesian closed categories. In *Proc. of CSL*, 1994.

[8] Martin Hofmann and Thomas Streicher. Lifting grothendieck universes. Available online at `http://www.mathematik.tu-darmstadt.de/~streicher/`

[9] Martin Hofmann and Thomas Streicher. The groupoid interpretation of type theory. In *Twenty-five years of constructive type theory. Proceedings of a congress*, pages 83–111. Oxford: Clarendon Press. Oxf. Logic Guides, 1998.

[10] G. Hutton and M. Jaskelioff. Representing contractive functions on streams. 2012. Available online at `http://www.cs.nott.ac.uk/~gmh/contractive.pdf`

[11] C. Kapulkin, P.L. Lumsdaine, and V. Voevodsky. The simplicial model of univalent foundations. *arXiv*, 1211.2851, 2012.

[12] G. M. Kelly. *Basic Concepts of Enriched Category Theory*. Cambridge University Press, 1982.

[13] N. Krishnaswami. Simple and efficient higher-order reactive programming. 2012. Available online at `http://www.mpi-sws.org/~neelk/simple-frp.pdf`

[14] N.R. Krishnaswami and N. Benton. Ultrametric semantics of reactive programs. In *Proceedings LICS*, 2011.

[15] Z. Luo. *Computation and Reasoning. A Type Theory for Computer Science*. Number 11 in International Series of Monographs on Computer Science. Oxford University Press, 1994.

[16] C. McBride. Time flies like an applicative functor. Available online at `http://www.e-pig.org/epilogue/?p=186`, 2009.

[17] C. McBride and R. Paterson. Applicative programming with effects. *Journal of Functional Programming*, 18(1), 2008.

[18] D.S. Scott. Domains for denotational semantics. In *Proceedings of ICALP*, volume 140 of *LNCS*, 1982.

[19] M. Shulman. The univalence axiom for inverse diagrams. *arxiv*, 1203.3253, 2012.

[20] Thomas Streicher. A model of type theory in simplicial sets. Unpublished, 2011.

[21] V. Voevodsky. Univalent foundations. Available at `http://www.math.ias.edu/~vladimir/`, 2012.

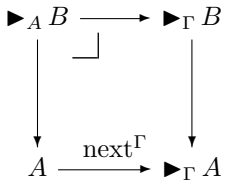
[22] G. Winskel and K.G. Larsen. Using information systems to solve recursive domain equations effectively. In *Proceedings of Conference on Abstract Data Types*, volume 173 of *LNCS*, 1984.

APPENDIX

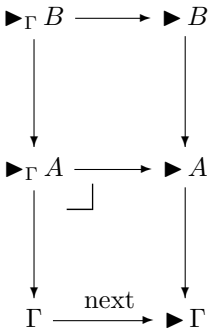
A. Proofs for Section V

The following lemma is the generalisation of (8) needed in the proof of Lemma V.9.

Lemma A.1. *Let $B \to A \to \Gamma$ be fibrations and Γ fibrant. There is a pullback diagram*

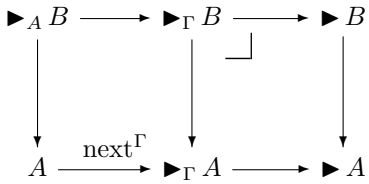


Proof: Consider first the commutative diagram



The lower diagram and the outer diagram are both versions of the pullback (8). The map $\blacktriangleright_\Gamma B \to \blacktriangleright_\Gamma A$ is defined using the universal property of the lower pullback square. By the pullback lemma also the upper square is a pullback.

Now consider the diagram



The right square is the one we have just proved is a pullback. The outer square is the pullback (8) and the map $\blacktriangleright_A B \to \blacktriangleright_\Gamma B$ is defined using the universal property of the right pullback diagram. By the pullback lemma the left diagram is a pullback. ∎

B. Proofs for Section VI

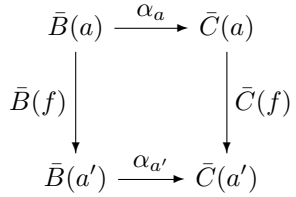
As a warm-up before proving Lemma VI.4 we prove a similar statement in the groupoid model

Lemma A.2. *Let A be a groupoid and let $\bar{B}, \bar{C} \colon A \to \mathit{Grpd}_\Delta$ be homomorphisms. The following two statements are equivalent*

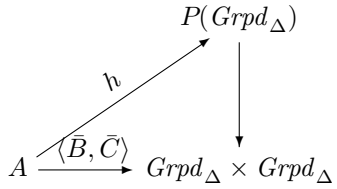
- *$\bar{B}^*\mathrm{El}$ is isomorphic to $\bar{C}^*\mathrm{El}$ as objects of the slice category over A*
- *The type $\prod x \colon A.\mathrm{Id}_{\mathit{Grpd}_\Delta}(\bar{B}(x), \bar{C}(x))$ is inhabited, i.e., $\bar{B}$ and $\bar{C}$ are pointwise internally equal.*

Proof: We first spell out the first statement of the lemma. The homomorphism $\bar{B} \colon A \to \mathit{Grpd}_\Delta$ maps each a in A to a set $\bar{B}(a)$ and each morphism $f \colon a \to a'$ in A to a bijection $\bar{B}(f) \colon \bar{B}(a) \to \bar{B}(a')$. The pullback of El along $\bar{B}$ is the groupoid $\int \bar{B}$ obtained by the Grothendieck construction. The objects are pairs (a, x) such that $a \in A$, $x \in \bar{B}(a)$ and a morphism from (a, x) to (a', x') is a map $f \colon a \to a'$ such that $\bar{B}(f)(x) = x'$.

An isomorphism α as in the first statement is a family of bijections $\alpha_a \colon \bar{B}(a) \to \bar{C}(a)$ such that if $f \colon (a, x) \to (a', x')$ in $\bar{B}^*\mathrm{El}$ also $f \colon (a, \alpha_a(x)) \to (a', \alpha_a(x'))$ in $\bar{C}^*\mathrm{El}$. The latter is equivalent to requiring that the following diagram commutes for all $f \colon a \to a'$ in A:



The second statement is equivalent to the existence of a homomorphism h making the following diagram commute.

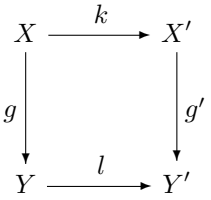


(10)

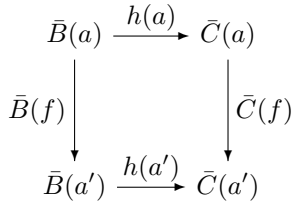
The object $P(\mathit{Grpd}_\Delta)$ is the groupoid whose set of objects is

$$\{(X, Y, g) \mid X, Y \in \mathit{Grpd}_\Delta, g \colon X \to Y \text{ bijection}\}$$

and whose morphisms from (X, Y, g) to (X', Y', g') are pairs of bijections $k \colon X \to X'$, $l \colon Y \to Y'$ such that the following diagram commutes



A map h as in (10) associates to each $a \in A$ a bijection $h(a) \colon \bar{B}(a) \to \bar{C}(a)$ such that for each $f \colon a \to a'$ the following diagram commutes.



It should now be clear that the α's and the h's are in bijective correspondence, proving the equivalence of the two statements of the lemma. ∎

The next step towards proving Lemma VI.4 is to spell out the universe $\mathrm{El}_{V_\Delta}\colon E_{V_\Delta} \to V_\Delta$ and the path space $P(V_\Delta)$.

The universe V_Δ is essentially the one of Section III equipped with a groupoid structure: the elements of $(V_\Delta)_n$ are families of small sets and maps

$$A_1 \xleftarrow{r_1^A} A_2 \xleftarrow{r_2^A} \dots \xleftarrow{r_{n-1}^A} A_n$$

and a morphism is a family of bijections $f_1, \dots, f_n$ making the following diagram commute.

$$\begin{array}{ccccccc} A_1 & \longleftarrow & A_2 & \longleftarrow & \dots & \longleftarrow & A_n \\ \downarrow{\scriptstyle f_1} & & \downarrow{\scriptstyle f_2} & & & & \downarrow{\scriptstyle f_n} \\ A_1' & \longleftarrow & A_2' & \longleftarrow & \dots & \longleftarrow & A_n' \end{array}$$

In the following, we shall refer to the condition above as $(f_1, \dots, f_n)$ being an isomorphism of finite sequences.

Shulman defines the path space (in the case of $B = 1$) over a fibrant object A as $(PA)_1 = P(A_1)$ and $(PA)_2$ is best defined using the internal language as

$$a_1, a_1'\colon A_1, a_2\colon A_2(a_1), a_2'\colon A_2(a_1'), p_1\colon \mathrm{Id}_{A_1}(a_1, a_1') \vdash \mathrm{Id}_{A_2(a_1)}(a_2, p_1^* a_2')$$

where p^* is the *transport* operation (which in [19] goes the other way). In general $(PA)_{n+1}$ is defined as

$$a_n, a_n'\colon A_n, a_{n+1}\colon A_{n+1}(a_n), a_{n+1}'\colon A_{n+1}(a_n'), p_n\colon \mathrm{Id}_{A_n}(a_n, a_n') \vdash \mathrm{Id}_{A_{n+1}(a_n)}(a_{n+1}, p_n^* a_{n+1}')$$

An element of $P(V_\Delta)_2$ over a pair

$$(A_1 \leftarrow A_2, A_1' \leftarrow A_2') \in (V_\Delta)_2 \times (V_\Delta)_2$$

is a pair of a bijection $f_1\colon A_1 \to A_1'$ and an isomorphism $f_2\colon A_2 \to f_1^* A_2'$ in the slice over A_1. This is the same as a pair of bijections f_1, f_2 making the square

$$\begin{array}{ccc} A_1 & \longleftarrow & A_2 \\ \downarrow{\scriptstyle f_1} & & \downarrow{\scriptstyle f_2} \\ A_1' & \longleftarrow & A_2' \end{array}$$

In general, an object of $P(V_\Delta)_n$ over a pair

$$(A_1 \leftarrow \dots \leftarrow A_n, A_1' \leftarrow \dots \leftarrow A_n') \in (V_\Delta)_n \times (V_\Delta)_n$$

is an isomorphism $(f_1, \dots, f_n)$ of finite sequences.

Proof of Lemma VI.4. Similarly to the proof of Lemma A.2 we start by spelling out the two statements to be proved equivalent.

This time the morphism $\bar{B}$ is a family of maps $\bar{B}_n\colon A_n \to (V_\Delta)_n$ mapping an element $a \in A_n$ to a finite sequence of sets and morphisms

$$(\bar{B}_n(a))_1 \leftarrow (\bar{B}_n(a))_2 \longleftarrow \dots \longleftarrow (\bar{B}_n(a))_n$$

such that the $n-1$ first components of this sequence are the same as

$$(\bar{B}_{n-1}(a_{|n-1}))_1 \longleftarrow \dots \longleftarrow (\bar{B}_{n-1}(a_{|n-1}))_{n-1}$$

and a morphism $f\colon a \to a'$ in A_n is mapped to an isomorphism of finite sequences.

The pullback $\bar{B}^*\mathrm{El}$ is

$$\begin{array}{ccccccccc} \int \bar{B}_1 & \longleftarrow & \int \pi_2 \bar{B}_2 & \longleftarrow & \dots & \longleftarrow & \int \pi_n \bar{B}_n & \longleftarrow & \dots \\ \downarrow & & \downarrow & & & & \downarrow & & \\ A_1 & \longleftarrow & A_2 & \longleftarrow & \dots & \longleftarrow & A_n & \longleftarrow & \dots \end{array}$$

where the π_i is the homomorphism $V_\Delta \to \mathit{Grpd}_\Delta$ mapping a sequence to its i'th element, and the integral sign denotes the Grothendieck construction as described in the proof of Lemma A.2.

An isomorphism $\bar{B}^*\mathrm{El} \cong \bar{C}^*\mathrm{El}$ is thus a family of bijections $\alpha_a^n\colon (\bar{B}_n(a))_n \to (\bar{C}_n(a))_n$ such that for all $f\colon a \to a'$ in A_n the following diagram commutes

$$\begin{array}{ccc} (\bar{B}_n(a))_n & \xrightarrow{\alpha_a^n} & (\bar{C}_n(a))_n \\ \downarrow{\scriptstyle (\bar{B}_n(f))_n} & & \downarrow{\scriptstyle (\bar{C}_n(f))_n} \\ (\bar{B}_n(a'))_n & \xrightarrow{\alpha_{a'}^n} & (\bar{C}_n(a'))_n \end{array} \tag{11}$$

and, moreover, such that the following commutes

$$\begin{array}{ccccc} (\bar{B}_{n-1}(a_{|n-1}))_{n-1} & = & (\bar{B}_n(a))_{n-1} & \longleftarrow & (\bar{B}_n(a))_n \\ \downarrow{\scriptstyle \alpha^{n-1}_{a_{|n-1}}} & & & & \downarrow{\scriptstyle \alpha_a^n} \\ (\bar{C}_{n-1}(a_{|n-1}))_{n-1} & = & (\bar{C}_n(a))_{n-1} & \longleftarrow & (\bar{C}_n(a))_n \end{array} \tag{12}$$

The second condition is equivalent to the existence of a map h making the following diagram commute

$$\begin{array}{ccc} & & P(V_\Delta) \\ & \overset{h}{\nearrow} & \downarrow \\ A & \xrightarrow{\langle \bar{B}, \bar{C} \rangle} & V_\Delta \times V_\Delta \end{array}$$

Such a map h associates to each $a \in A_n$ an isomorphism of sequences $h_n\colon \bar{B}(a) \to \bar{C}(a)$ with the requirement that

$$(h_n(a))_i = (h_{n-1}(a_{|n-1}))_i \tag{13}$$

for all $i < n$. The requirement that h is a homomorphism means that for every $f\colon a \to a'$ in A_n and every $i \leq n$ the following diagram must commute

$$\begin{array}{ccc} (\bar{B}_n(a))_i & \xrightarrow{(h_n(a))_i} & (\bar{C}_n(a'))_i \\ \Big\downarrow{\scriptstyle (\bar{B}_n(f))_i} & & \Big\downarrow{\scriptstyle (\bar{C}_n(f))_i} \\ (\bar{B}_n(a))_i & \xrightarrow{(h_n(a'))_i} & (\bar{C}_n(a'))_i \end{array} \tag{14}$$

To prove that the first statement implies the other, suppose we are given α. Construct h as $h(a) = (\alpha^1_{a|_1}, \ldots, \alpha^n_a)$. Condition (12) ensures that $h(a)$ is an isomorphism of sequences and condition (13) is almost trivial. Since $(\bar{B}_n(a))_i = (\bar{B}_i(a|_i))_i$ and $(h_n(a))_i = (h_i(a|_i))_i$ requirement (14) reduces to (11).

On the other hand, if we are given h we can construct an α as $\alpha^n_a = (h_n(a))_n$. Then (11) is a special case of (14) and (12) follows from $h_n(a)$ being an isomorphism of finite sequences, since $(h_{n-1}(a|_{n-1}))_{n-1} = (h_n(a))_{n-1}$. ■