Constructive sheaf models of type theory

(article starts on next page)

PAPER

Constructive sheaf models of type theory

Thierry Coquand* , Fabian Ruch and Christian Sattler

Computer Science Department, Chalmers University and University of Gothenburg, Gothenburg, Sweden
*Corresponding author. Email: Thierry.Coquand@cse.gu.se

Abstract

We provide a constructive version of the notion of sheaf models of univalent type theory. We start by relativizing existing constructive models of univalent type theory to presheaves over a base category. Any Grothendieck topology of the base category then gives rise to a family of left-exact modalities, and we recover a model of type theory by localizing the presheaf model with respect to this family of left-exact modalities. We provide then some examples.

1. Introduction

Despite being relatively recent, the notion of (pre-)sheaf model has a rich and intricate history which mixes different intuitions coming from topology, logic, and algebra. Eilenberg and Zilber (1950) used a presheaf model (simplicial sets) to represent geometrical objects, and the intuition is *geometrical*: we think of the objects $I, J, \ldots$ of the base category as basic "shapes"; a presheaf A is then given by a family of sets $A(I)$ of objects of each shape I, which are related by the restriction maps $A(I) \to A(J)$. A little later, but independently, Beth (1956) and Kripke (1965) used a sheaf and a presheaf model over trees, respectively, to provide a formal semantics for intuitionistic logic. Their motivations were logical, and the intuition is of a *temporal* nature instead: we think of the nodes of the tree as "stages of knowledge" and of the tree ordering as "increase in knowledge." Scott (1980) described a presheaf model of higher-order logic and pointed out the potential interest for the semantics of λ-calculus. This was refined by Hofmann (1997) who presented a presheaf model of dependent type theory with universes. Hofmann's presheaf model was subsequently used in an essential way in works on constructive semantics of type theory with *univalent* universes (see Cohen et al. 2015; Coquand et al. 2018; Orton and Pitts 2016).

The generalization of such presheaf models of dependent type theory, and especially of universes, to a *sheaf* model semantics is however non-trivial. The problem in generalizing this semantics for universes comes essentially from the fact that the collection of sheaves does not form a sheaf in any natural way: if we are given locally sheaves that are compatible, one can patch them together but not in a *unique* way, only unique *up to isomorphism*. This problem was the motivation for the introduction of stacks and a more subtle notion of patching of sheaves (cf. Grothendieck 1960, Section 3.3), and in general patching of mathematical structures. The generalization of this to patching of higher structures was the content of the first part of Joyal's letter to Grothendieck (see Joyal 1984). One contribution of the present paper is to provide a constructive version of this notion[1] by describing a sheaf model semantics of type theory with univalence (see, e.g., Voevodsky 2015 and Univalent Foundations Program 2013). This uses in a crucial way the fact that we have a

constructive interpretation of univalence as in Cohen et al. (2015), Orton and Pitts (2016), which can be relativized to any presheaf model. The main point is then that the operation sending an object to its object of descent data (a compatible collection of elements of its restrictions) defines a *left-exact modality* (see Quirin 2016; Rijke et al. 2020; Univalent Foundations Program 2013), which can then be used to build internally models of univalent type theory, as shown in Quirin (2016).

This work opens the possibility of generalizing the rich collection of results about sheaf models of intuitionistic logic as in Troelstra and van Dalen (1988) to sheaf models of univalent type theory. It extends the previous work in Coquand et al. (2017) to a complete model of univalence, and has no restrictions for representing (higher) data types. We give here only two applications (independence of countable choice, and a model of the algebraic closure of a field), but we expect also that this can be used to build constructive account of works such as in Shulman (2018) and Wellen (2017). The present semantics (in a preliminary version) has already been used by Weaver and Licata (2020) for building a constructive model of directed univalence.

This paper is organized as follows. We first introduce the notion of lex operation as an operation acting on types and families of types. A descent data operation is then a lex operation which defines a left-exact modality (cf. Quirin 2016; Rijke et al. 2020; Univalent Foundations Program 2013). These two notions are formulated purely syntactically in the framework of type theory. We show next how to instantiate these operations for cubical presheaves. In this setting, we can understand the notion of being modal for a descent data operation as a generalization of the sheaf condition, where the compatibility requirements are expressed up to path equality instead of being expressed as strict equalities. We then provide some examples and the applications to the unprovability of countable choice and to the model of the algebraic closure of a field. In an appendix, we explain how some of our results about descent data operations can be generalized to accessible left-exact modalities.

2. Abstract Notion of Descent Data

We use the following notations. We write $\mathbf{1}$ for the unit type and $() : \mathbf{1}$ for its unique element. If $a : A$, we write ε_a the associated function $\mathbf{1} \to A$. Given a type A and a family B of types over A, we write $\sum_A B$ for their sum type and $\prod_A B$ for their product type. The pairing operation is denoted by $(a, b) : \sum_A B$ for $a : A$ and $b : B\,a$. The projection maps are denoted by π_1 and π_2. We write K_B for the constant family B over a type A. As usual, function space $A \to B$ is defined to be $\prod_A K_B$ and product $A \times B$ is defined to be $\sum_A K_B$. We write id_A for the identity function on A and $g \circ f$ for the composition of $f : A \to B$ and $g : B \to C$. If B is a family of types over A and $f : A' \to A$, we also write $B \circ f$ for the family of types over A' obtained from B by reindexing along f, and we write $B\,a$ for the instantiation of the family B at the element a of type A.

In this section, equality (denoted by the symbol $=$) refers to the strict equality (as opposed to identity or path types), and isomorphisms refer to strictly invertible maps.

2.1 Lex operation

The concept of lex operations is defined for a dependent type theory with only unit type, dependent sums, dependent product, and universes. In particular, path types are not needed. Intuitively, a lex operation is an endofunctor on the category of types and functions (compatible with substitution) which preserves the unit type and dependent projections of sum types up to isomorphism. It is defined as an extension of type theory with some constants satisfying some (strict) equalities.

Definition 1. *A lex operation D^2 is given by the following constants:*

$$D(g \circ f) = Dg \circ Df \qquad D(\mathsf{id}_A) = \mathsf{id}_{DA} \qquad x = \langle\rangle \quad (x : D\mathbf{1})$$
$$\tilde{D}(B \circ f) = \tilde{D}B \circ Df \qquad \tilde{D}(s \circ f) = \tilde{D}s \circ Df$$
$$(D\pi_1)\langle u, v \rangle = u \qquad (\tilde{D}\pi_2)\langle u, v \rangle = v \qquad \langle (D\pi_1)\, w, (\tilde{D}\pi_2)\, w \rangle = w$$
$$Df \circ \eta_A = \eta_B \circ f \qquad \tilde{D}(K_B) = K_{DB} \qquad Ds = \tilde{D}s \quad (s : \textstyle\prod_A K_B)$$

Figure 1. Equations for lex operations.

(1) *an operation D on types and functions with a function $\eta_A : A \to DA$*

(2) *an operation $\tilde{D}$ on families of types: $\tilde{D}B$ is a family of types over DA if B is a family of types over A, with an operation $\tilde{D}s : \prod_{DA}\tilde{D}B$ on sections $s : \prod_A B$*

(3) *an element $\langle\rangle$ in $D\mathbf{1}$ and a pairing operation $\langle u, v \rangle : D(\sum_A B)$ for elements $u : DA$ and $v : (\tilde{D}B)\, u$*

satisfying the (strict) equations of Figure 1 and such that that universes reflect[3] these operations.

These constants and equations express that D forms a strict pointed endofunctor on types which preserves the unit type $\mathbf{1}$ up to isomorphism and dependent projections of sum types up to isomorphism.

The canonical example of a lex operation is exponentiation with a fixed type R (assumed to be in all universes). We define $DA = A^R$, $(\tilde{D}B)\, u = \prod_{x:R} B(u\, x)$, and $(\tilde{D}s)\, u = \lambda_{x:R}\, s(u\, x)$. The pairing is given by $\langle u, v \rangle = \lambda_{x:R}\, (u\, x, v\, x)$ and $\langle\rangle$ is the constant function $\lambda_{x:R}\, ()$ and η_A is $\lambda_{a:A}\, \lambda_{x:R}\, a$.

Remark 2. Let $\mathscr{U}$ be a universe. The action of the operation $\tilde{D}$ on $\mathscr{U}$-small families is uniquely determined by the universal case $L = \tilde{D}\,\mathsf{id}_{\mathscr{U}} : D\mathscr{U} \to \mathscr{U}$: we have (and can define) $\tilde{D}B = L \circ DB$ with $DB : DA \to D\mathscr{U}$ for $B : A \to \mathscr{U}$. This corresponds to the "escaping" function in Section 2.5 of Schreiber and Shulman (2012). We can describe the action of $\tilde{D}$ on $\mathscr{U}$-small families and associated operations by requiring that D applied to the "universal $\mathscr{U}$-small fibration" $\sum_{X:\mathscr{U}} X \to \mathscr{U}$ is isomorphic to a "$\mathscr{U}$-small fibration" (a projection of a type in $\mathscr{U}$), and that D preserves pullbacks of this map.

Remark 3. Any strict endofunctor D on types which preserves the unit type $\mathbf{1}$ up to isomorphism is uniquely pointed[4] since we should have $\eta_{\mathbf{1}}\, () = \langle\rangle$ and so $\eta_A\, a = \eta_A(\varepsilon_a\, ()) = (D\varepsilon_a)(\eta_{\mathbf{1}}\, ()) = (D\varepsilon_a)\,\langle\rangle$. It follows that the map η in Definition 1 (with associated laws) can be recovered from the other operations and laws.

It also follows that we have $\tilde{D}B\,(\eta_A\, a) = \tilde{D}B\,(D\varepsilon_a\,\langle\rangle) = \tilde{D}(K_{B\, a})\,\langle\rangle = K_{D(B\, a)}\,\langle\rangle = D(B\, a)$.

Remark 4. If we present type theory as a generalized algebraic theory, for instance as in Dybjer (1995), with sorts of contexts and substitutions $\Delta \to \Gamma$ and types $\mathsf{Type}(\Gamma)$ and elements, then a lex operation can be seen as an *extension* of this generalized algebraic theory, since it is defined by constants and equations. We then have DA in $\mathsf{Type}(\Gamma)$ for A in $\mathsf{Type}(\Gamma)$ and $\tilde{D}B$ in $\mathsf{Type}(\Gamma.DA)$ for B in $\mathsf{Type}(\Gamma.A)$.[5] As usual, we have to add that these constants commute with substitutions: for instance, $(DA)\sigma = D(A\sigma)$ and $(\tilde{D}B)\sigma^+ = \tilde{D}(B\sigma^+)$ if $\sigma : \Delta \to \Gamma$.

We get in this way a notion of model of type theory together with a lex operation.

Remark 5. A lex operation can be defined from a pseudomorphism of cwfs with universes Kaposi et al. (2019) from the cwf to itself that is pointed as an endofunctor.

2.2 D-modal types

The notion of lex operation is defined at the level of "pure" dependent type theory, without assuming any notion of path types. In the presence of path types with function extensionality, we automatically have the following preservation property.

Theorem. *Let D be a lex operation. Then D preserves equivalences.*

Proof. Note that if f_0 and f_1 are path equal then so are Df_0 and Df_1 by path induction. It follows that if f and g are inverses, then so are Df and Dg. □

Avigad et al. (2015) explain how to build a fibration category from a model of dependent type theory. Theorem 2.2 implies that any lex operation defines an endomorphism of the associated fibration category. A lex operation preserves all finite homotopy limits (e.g., contractible types, homotopy pullbacks, homotopy equalizers, homotopy fibers, etc.).

In the presence of path types, we can also define the following important notion of modal types.

Definition 6. *A type A is called D-modal if the unit map $\eta_A : A \to DA$ is an equivalence.*

Proposition 7. *If A is D-modal and B is a family of types over A, then B is a family of D-modal types over A if and only if $T = \sum_A B$ is D-modal.*

Proof. Let f be the map $T \to \sum_{DA} \tilde{D}B$, $(a, b) \mapsto (\eta_A\, a, \eta_{B\,a}\, b)$. Since η_A is an equivalence, each map $\eta_{B\,a}$ is an equivalence if and only if the map f is an equivalence (Univalent Foundations Program 2013). But f is an equivalence if and only if η_T is an equivalence. □

Lemma 8. *If B is D-modal and we have f and g maps $DA \to B$ such that $f \circ \eta_A$ and $g \circ \eta_A$ are path equal, and $D\eta_A$ is an equivalence then f and g are path equal.*

Proof. Since $D(f \circ \eta_A) = Df \circ D\eta_A$ and $D(g \circ \eta_A) = Dg \circ D\eta_A$ and $D\eta_A$ is an equivalence we get that Df and Dg are path equal. Hence, $Df \circ \eta_{DA} = \eta_B \circ f$ and $Dg \circ \eta_{DA} = \eta_B \circ g$ are path equal, which implies that f and g are path equal since η_B is an equivalence. □

2.3 Abstract notion of descent data

Theorem. *The following conditions are equivalent, for a lex operation D*

(1) *D defines a modality as axiomatized in Quirin (2016) and Rijke et al. (2020)*

(2) *the map $D\eta_A$ is an equivalence, and $D\eta_A$ and η_{DA} are path equal*

(3) *both maps $D\eta_A$ and η_{DA} are equivalences*

Proof. The first condition implies the second using the results in Quirin (2016) and Rijke et al. (2020) and the second condition clearly implies the third.

Conversely, assume that the map $D\eta_A$ is an equivalence, and $D\eta_A$ and η_{DA} are path equal. Then η_{DA} is an equivalence as well and each type DA is D-modal. Proposition 7 shows that D-modal types are closed under dependent sums. We thus only have to prove that the map

$$F : (DA \to B) \longrightarrow (A \to B) \qquad f \longmapsto f \circ \eta_A$$

is an equivalence if B is D-modal (Quirin 2016).

Let p_B be a map $DB \to B$ such that $p_B \circ \eta_B$ is path equal to id_B. We define a map

$$G : (A \to B) \longrightarrow (DA \to B) \qquad u \longmapsto p_B \circ Du$$

We then have $F(Gu) = p_B \circ Du \circ \eta_A = p_B \circ \eta_B \circ u$ which is path equal to u and $G(Ff) = p_B \circ D(f \circ \eta_A) = p_B \circ Df \circ D\eta_A$ which is path equal to $p_B \circ Df \circ \eta_{DA} = p_B \circ \eta_B \circ f$ which is path equal to f. Hence, G is an inverse to F and F is an equivalence.

Finally, we show that the third condition implies the second condition. We want to show that $D\eta_A$ and η_{DA} are path equal. By the previous lemma, it is enough to check that $D\eta_A \circ \eta_A$ and $\eta_{DA} \circ \eta_A$ are path equal, but they are even strictly equal since D is a lex operation. $\square$

Definition 9. *A descent data operation is a lex operation D satisfying the equivalent conditions of Theorem 2.3. It is then automatically a lex modality, as defined in Quirin (2016), Rijke et al. (2020) since it preserves homotopy pullbacks.*[6]

Note that the first and third conditions of Theorem 2.3 are (homotopy) propositions. The second condition is the one which will be convenient to verify for the main example.

We write $\mathsf{isMod}_D(A)$ for the type (proposition) expressing that A is D-modal.

In particular, it follows from the general properties of lex modalities that we have.

Lemma 10. *Let D be a descent data operation. For A to be D-modal, it is enough to have a patch function $p_A : DA \to A$ such that $p_A \circ \eta_A$ is path equal to the identity of A. More precisely, the type of such patch functions is a proposition equivalent to the proposition $\mathsf{isMod}_D(A)$.*

2.4 Example

If R is a *proposition*, then for the lex operation defined by $DA = A^R$ the two maps $D\eta_A$ and η_{DA} are path equal equivalences and hence exponentiation defines a descent data operation in that case.

In Section 4, we define a new kind of descent data operation for any presheaf model.

3. Lex Modalities and Universes

The following results were suggested by the proof that the universe of sheaves is a stack in Coquand et al. (2017). They can be stated and proved at the level of lex modalities and will be crucial for building the sheaf model of univalent type theory in the next sections. The appendix presents a more general version and connections with the notion of accessible modalities introduced in Rijke et al. (2020).

3.1 Universe of modal types

Let D be a lex modality, such that that $DA : \mathscr{U}$ if $A : \mathscr{U}$.

Proposition 11. *If D respects $\mathscr{U}$-small families, then $\mathscr{U}_D = \sum_{\mathscr{U}} \mathsf{isMod}_D$ is D-modal.*

Proof. The hypothesis implies Schreiber and Shulman (2012) that we have an "escaping" function $L : D\mathscr{U} \to \mathscr{U}$ such that $L \circ \eta_{\mathscr{U}}$ is path equal to $D : \mathscr{U} \to \mathscr{U}$ and L defines a family of D-modal types over $D\mathscr{U}$. We can then define $\alpha : D\mathscr{U} \to \mathscr{U}_D$ such that $\pi_1 \circ \alpha = L$, and then $\pi_1 \circ \alpha \circ \eta_{\mathscr{U}}$ is path equal to D. We claim that $\alpha \circ D\pi_1$ is a left inverse of $\eta_{\mathscr{U}_D}$. This implies that $\mathscr{U}_D$ is D-modal, since D is a lex modality.

By univalence(!), the map $D \circ \pi_1 : \mathscr{U}_D \to \mathscr{U}$ is path equal to $\pi_1 : \mathscr{U}_D \to \mathscr{U}$. It follows that $\pi_1 \circ \alpha \circ \eta_{\mathscr{U}} \circ \pi_1$ is path equal to π_1. Since $\pi_1 : \mathscr{U}_D \to \mathscr{U}$ is an embedding, $\mathsf{id}_{\mathscr{U}_D}$ is path equal to $\alpha \circ \eta_{\mathscr{U}} \circ \pi_1$. But $\eta_{\mathscr{U}} \circ \pi_1 : \mathscr{U}_D \to D\mathscr{U}$ is path equal to $D\pi_1 \circ \eta_{\mathscr{U}_D}$, since D is a modality, and so $\alpha \circ D\pi_1$ is a left inverse of $\eta_{\mathscr{U}_D}$. $\square$

3.2 Generalization to a family of descent data operations

More generally, if we have a *family* of lex modalities D_S indexed by a given type $S : \mathsf{C}$, with corresponding maps $\eta_A^S : A \to D_S A$, we can consider $\mathsf{isMod}_\mathsf{C}(A)$ to be the proposition $\prod_{S:\mathsf{C}} \mathsf{isMod}_{D_S}(A)$

and $\mathscr{U}_{\mathsf{C}}$ which is $\sum_{\mathscr{U}}$ isMod$_{\mathsf{C}}$. We use the slightly shorter notation $\mathscr{U}_S$ to denote the previously defined type $\mathscr{U}_{D_S} = \sum_{\mathscr{U}}$ isMod$_{D_S}$

We assume a preorder relation $\leq$ on C such that $S' \leq S$ implies that any $D_{S'}$-modal type is D_S-modal. We say that C is *filtered* if we have $\exists_{S:\mathsf{C}}. S \leq S_1 \wedge S \leq S_2$ for any $S_1, S_2 : \mathsf{C}$.[7]

We assume that $D_S A : \mathscr{U}$ if $A : \mathscr{U}$ and each D_S preserves $\mathscr{U}$-small maps.

Theorem. *If C is filtered, then $\mathscr{U}_{\mathsf{C}}$ satisfies* isMod$_{\mathsf{C}}$.

Proof. We write $L_S : D_S \mathscr{U} \to \mathscr{U}$ the "escape" map for D_S such that $L_S \circ \eta^S$ is path equal to D_S. For any $S' \leq S$ in C, $\mathscr{U}_{S'}$ is $D_{S'}$-modal by Proposition 11 and so D_S-modal, and hence $\eta^S : \mathscr{U}_{S'} \to D_S \mathscr{U}_{S'}$ is an equivalence. The following diagram (commuting up to path equality)

$$
\begin{array}{ccc}
\mathscr{U}_{S'} & \xrightarrow{\eta^S} & D_S \mathscr{U}_{S'} \\
{\scriptstyle \pi_1}\downarrow & & \downarrow{\scriptstyle D_S \pi_1} \\
\mathscr{U} & \xrightarrow[\;\eta^S\;]{} & D_S \mathscr{U} \xrightarrow{L_S} \mathscr{U} \\
& \underset{D_S}{\searrow} &
\end{array}
$$

and the fact that $\eta^S : \mathscr{U}_{S'} \to D_S \mathscr{U}_{S'}$ is an equivalence show that for any $A : D_S \mathscr{U}_{S'}$, the type $L_S(D_S \pi_1 A)$ is $D_{S'}$-modal. It follows that, for any $A : D_S \mathscr{U}_{\mathsf{C}}$, the type $L_S(D_S \pi_1 A)$ is $D_{S'}$-modal.

If C is filtered, this implies that the type $L_S(D_S \pi_1 A)$ is $D_{S'}$-modal for any S' in C. Hence, the map $L_S \circ D_S \pi_1 : D_S \mathscr{U}_{\mathsf{C}} \to \mathscr{U}$ factorizes through $\mathscr{U}_{\mathsf{C}} \to \mathscr{U}$ by a map $p : D_S \mathscr{U}_{\mathsf{C}} \to \mathscr{U}_{\mathsf{C}}$ and we have a (homotopy) commuting diagram

$$
\begin{array}{ccccc}
\mathscr{U}_{\mathsf{C}} & \xrightarrow{\eta^S} & D_S \mathscr{U}_{\mathsf{C}} & \xrightarrow{p} & \mathscr{U}_{\mathsf{C}} \\
{\scriptstyle \pi_1}\downarrow & & \downarrow{\scriptstyle D_S \pi_1} & & \downarrow{\scriptstyle \pi_1} \\
\mathscr{U} & \xrightarrow[\;\eta^S\;]{} & D_S \mathscr{U} & \xrightarrow{L_S} & \mathscr{U} \\
& & \underset{D_S}{\searrow} & &
\end{array}
$$

which, together with the fact that $\pi_1 : \mathscr{U}_{\mathsf{C}} \to \mathscr{U}$ is an embedding, and that $D_S \circ \pi_1$ is path equal to π_1 by univalence, shows that p is a left inverse of $\mathscr{U}_{\mathsf{C}} \to D_S \mathscr{U}_{\mathsf{C}}$. Hence, $\mathscr{U}_{\mathsf{C}}$ is D_S-modal for any S in C, since D_S is a lex modality. $\qquad\square$

4. Cubical Presheaf Models

4.1 *Cubical models*

Cubical models are presheaf models of univalent type theory specified by two parameters, an *interval object* $\mathbf{I}$ and a *cofibration classifier* Φ. Formally, we say that a *cubical model* is a presheaf category with the following structure, as in Orton and Pitts (2016).

- The interval object $\mathbf{I}$ is connected and has distinct points 0 and 1. Product with $\mathbf{I}$ preserves representable presheaves. We also assume that $\mathbf{I}$ has the structure of a bounded distributive lattice.[8]
- The universal cofibration $\top : 1 \to \Phi$ is a levelwise decidable inclusion. In the internal language of presheaves, we will work with Φ as a universe of certain propositions and leave the decoding function (given by equality with $\top : \Phi$) implicit. Isomorphic cofibrations are equal.[9] The interval endpoint inclusions $0, 1 : 1 \to \mathbf{I}$ are cofibrations. Cofibrations are closed

under finite union (finite disjunction), composition (dependent conjunction), and universal quantification over **I**.

It is then known, following the work in Cohen et al. (2015), Orton and Pitts (2016), Coquand et al. (2018), how to define a model of univalent type theory with higher inductive types.

4.2 Presheaves in cubical models

For the remainder of this subsection, we fix a cubical model given by presheaves over a small category $\mathscr{B}$. We refer to this as the *base model* (e.g., it can be cubical sets). We write $I, J, K, \ldots$ for the objects of $\mathscr{B}$. We have the interval object $\mathbf{I}_{\mathscr{B}}$ and the cofibration classifier $\Phi_{\mathscr{B}}$.

Let $\mathscr{C}$ be another small category. We write $X, Y, Z, \ldots$ for its objects. We describe possibilities for turning presheaves over $\mathscr{C} \times \mathscr{B}$ into a cubical model. For the interval object $\mathbf{I}$, we simply take $\mathbf{I}(X, I) = \mathbf{I}_{\mathscr{B}}(I)$. For the cofibration classifier, we have two reasonable options:

(1) The first example is simply to take $\Phi(X, I) = \Phi_{\mathscr{B}}(I)$.

(2) The second example is to define an element ψ of $\Phi_{\mathsf{lw}}(X, I)$ to be a family ψ_f in $\Phi_{\mathscr{B}}(I)$ for $f : Y \to X$ such that $\psi_f \leq \psi_{fg}$ if furthermore $g : Z \to Y$. We then define the restriction operation $\psi(f, l)$ to be the family $\psi(f, l)_g = \psi_{fg} l$ for $f : Y \to X$ and $g : Z \to Y$ and $l : J \to I$.

The motivation for the second example is that if $\Phi_{\mathscr{B}}(I)$ is the collection of (decidable) sieves on I, then $\Phi_{\mathsf{lw}}(X, I)$ becomes the collection of (decidable) sieves on (X,I).[10]

The interval object $\mathbf{I}$ and any of the choices Φ and Φ_{lw} fit all the requirements listed in Subsection 4.1. This turns presheaves over $\mathscr{C} \times \mathscr{B}$ into a cubical model. In particular, we get a model of univalent type theory (and higher inductive types). We are going to analyze the model obtained using the choice Φ for the cofibration classifier and then indicate how to adapt these results for Φ_{lw}.

In this model, a context Γ is interpreted by a presheaf over $\mathscr{C} \times \mathscr{B}$ so a family of sets $\Gamma(X, I)$ with suitable restriction maps $\rho \mapsto \rho(f, l)$ with $f : Y \to X$ in $\mathscr{C}$ and $l : J \to I$ in $\mathscr{B}$.

A dependent type A over Γ is then given by a presheaf over the category of elements of Γ: for any ρ in $\Gamma(X, I)$ we have a set $A\rho$ with suitable restriction maps $A\rho \to A\rho(f, l)$ denoted by $u \mapsto u(f, l)$ together with a filling operation (see Cohen et al. 2015; Orton and Pitts 2016). We write $\mathsf{Type}(\Gamma)$ for the collection of all types with a composition operation over Γ. For A in $\mathsf{Type}(\Gamma)$ the extension $\Gamma.A$ is defined by taking $(\Gamma.A)(X, I)$ to be the set of pairs ρ, u with ρ in $\Gamma(X, I)$, u in $A\rho$, and $(\rho, u)(f, l) = \rho(f, l), u(f, l)$. The set $\mathsf{Elem}(\Gamma, A)$ is then the set of sections: a family $a\rho$ in $A\rho$ such that $(a\rho)(f, l) = a(\rho(f, l))$ for any ρ in $\Gamma(X, I)$ and map f,l of codomain X,I.

Given a constructive Grothendieck universe U (see Aczel 1998) containing $\mathscr{B}$ and $\mathscr{C}$, we write $\mathsf{Type}_U(\Gamma)$ for the set of U-small types. The presheaf Type_U is then represented by a fibrant type $\mathscr{U}$, which is univalent as shown in Cohen et al. (2015).

4.3 Internal language description

This was an external description of the presheaf model. It is also possible to describe this model using the internal logic of the presheaf topos over $\mathscr{C} \times \mathscr{B}$ as in Orton and Pitts (2016), Coquand et al. (2018) but also using the internal logic of the presheaf topos over $\mathscr{B}$. We will use both descriptions.

In the internal logic of the presheaf topos over $\mathscr{B}$, a context of the presheaf model over $\mathscr{C}$ is interpreted as a family of "spaces" $\Gamma(X)$ with restriction maps $\rho \mapsto \rho f$ for $f : Y \to X$. (Each space $\Gamma(X)$ is itself a presheaf over $\mathscr{B}$ with $\Gamma(X)(I) = \Gamma(X, I)$.) A dependent type A over Γ is given by a family of spaces $A\rho$ for ρ in $\Gamma(X)$ with restriction maps $u \mapsto uf$. The presheaf Φ of cofibrations is the constant presheaf $\Phi(X) = \Phi_{\mathscr{B}}$. The interval $\mathbf{I}$ is the constant interval $\mathbf{I}(X) = \mathbf{I}_{\mathscr{B}}$.

It will be convenient to introduce the following notation: if γ is an element of $\Gamma(X)^{\mathbf{I}}$ and $f : Y \to X$, we write γf^+ in $\Gamma(Y)^{\mathbf{I}}$ for $\lambda_i \, \gamma(i)f$. Similarly, if $u(i)$ is a section in $A\gamma(i)$, we write uf^+ for $\lambda_i \, u(i)f$.

A *filling operation* (see Coquand et al. 2018; Orton and Pitts 2016) for A is given by an operation c_A which takes as argument γ in $\Gamma(X)^{\mathbf{I}_{\mathscr{B}}}$ and ψ in $\Phi(X) = \Phi_{\mathscr{B}}$ and a family of elements $u(i)$ in $A\gamma(i)$ on the extent $\psi \vee i = 0$. (There is a dual operation with $i = 1$ instead.) It produces an element $c_A(X, \gamma, \psi, u)(i)$ in $A\gamma(i)$ such that

(1) $c_A(X, \gamma, \psi, u)(i) = u(i)$ on $\psi \vee i = 0$,
(2) $c_A(X, \gamma, \psi, u)(i)f = c_A(Y, \gamma f^+, \psi, uf^+)(i)$ for $f : Y \to X$.

Given such an operation, we also call *A fibrant* (note this is structure rather than property). If we only have the first condition, we say that A is *levelwise* fibrant, and c_A a *levelwise* filling operation.

If A is a type over Γ, we get a family of dependent types $A(X)$ over $\Gamma(X)$, each of them having a filling operation, but furthermore these filling operations commute with the restriction maps.

Similarly, an *extension operation* for A, witnessing that A is contractible (see Cohen et al. 2015), is given by an operation e_A which takes as argument ρ in $\Gamma(X)$ and a partial element u on the extent ψ and produces an element $e_A(X, \rho, \psi, u)$ in $A\rho$ such that

(1) $e_A(X, \rho, \psi, u) = u$ on ψ,
(2) $e_A(X, \rho, \psi, u)f = e_A(Y, \rho f, \psi, uf)$ for $f : Y \to X$.

If we only have the first condition, we say that A is *levelwise* contractible.

If A is contractible, each $A(X)$ is a contractible family of types over $\Gamma(X)$. But conversely, it may be that each $A(X)$ has an extension operation $e_A(X)$ which does not commute with restriction (see Subsection 4.4). Similarly, a map $\sigma : A \to B$ which is an equivalence defines a family of equivalences $\sigma_X : A(X) \to B(X)$ but it may be that each map σ_X is an equivalence, without σ being an equivalence.

Remark 12. We have a canonical map from Φ to Φ_{lw} sending $\psi : \Phi(X)$ to the constant family on ψ. This map commutes (up to isomorphism) with the decoding to propositions. It follows that there is a natural map from extension operations for Φ to extension operations for Φ_{lw}, and the same holds for filling operations. It follows that a (contractible) type for the cubical presheaf model for Φ is naturally also a (contractible) type for the cubical presheaf model for Φ_{lw}.

Remark 13. Let $\mathscr{C}$ be a groupoid. Then for $\psi : \Phi_{\mathsf{lw}}(X)$ and $f : Y \to X$, we have $\psi_{\mathsf{id}_X} \leq \psi_f \leq \psi_{ff^{-1}} = \psi_{\mathsf{id}_X}$. It follows that ψ is the constant family on ψ_{id_X}. Thus, the map $\Phi \to \Phi_{\mathsf{lw}}$ from Remark 12 is invertible. It follows that the cubical presheaf models for Φ and Φ_{lw} are the same.[11]

4.4 Examples

Let $\mathscr{B}$ be a concrete cube category, for instance the Cartesian, distributive lattice, or de Morgan one (as defined in Angiuli et al. 2017; Cohen et al. 2015; Orton and Pitts 2016). Then we have a nerve functor from groupoids to cubical sets in the sense of presheaves over $\mathscr{B}$. In this way, we can see any groupoid as a cubical set with a canonical filling operation.

For the first example, let $\mathscr{C}$ be the group $\mathbb{Z}/(2)$. Let τ be the non-trivial element of this group. A context is a space with an involutive action $\rho \mapsto \rho\tau$. A dependent type A over Γ has also an involutive action $A\rho \to A\rho\tau$ denoted by $u \mapsto u\tau$ with a filling operation which is equivariant, meaning $c_A(\gamma, \psi, u)(i)\tau = c_A(\gamma\tau^+, \psi, u\tau^+)(i)$. Let A be the groupoid with two isomorphic objects swapped by τ. Then A is levelwise contractible, but is not contractible in the presheaf model, since it has no global point. Another way to describe this example is that the unique map $A \to \mathbf{1}$ is a levelwise equivalence, but is not an equivalence.

For the second example, let $\mathscr{C}$ be the poset on objects $\bot$, 0, 1 with $\bot < 0$ and $\bot < 1$. We define a global type A as follows. We take $A(0)$ and $A(1)$ to consist of a single object a_0 and a_1, respectively. We take $A(\bot)$ to consist of an isomorphism between the restrictions of a_0 and a_1. Then A is levelwise contractible (i.e., $A(\bot)$, $A(0)$, $A(1)$ are contractible), but A is not contractible since it has no global point.

We note that the second example is fixed by working with the cofibration classifier Φ_{lw}. However, as explained by Remark 13, this does not apply to the first example.

5. Homotopy Descent Data in the Presheaf Model

5.1 A lex operation

In this subsection, we work in the internal language of the presheaf topos over $\mathscr{B}$. We first define a lex operation on presheaf types, and then show that this lex operation extends to types with a filling operation.

For any presheaf A over Γ, we define a presheaf EA over Γ. For ρ in $\Gamma(X)$, we define

$$(EA)\rho = \prod_{Y, f : Y \to X} A\rho f$$

so that an element u of $(EA)\rho$ by a family of elements $u(f)$ in $A\rho f$ for $f : Y \to X$. We define the restriction uf in $(EA)\rho f$ by $uf(g) = u(fg)$ if $f : Y \to X$ and $g : Z \to Y$.

If B is a presheaf over $\Gamma.A$, then we define a presheaf $\tilde{E}(B)$ over $\Gamma.EA$. If ρ is in $\Gamma(X)$ and u is in $(EA)\rho$, then we define $\tilde{E}(B)(\rho, u)$ as the space of families $v(f)$ in $B(\rho f, u(f))$.

We define a natural transformation $\alpha : A \to EA$ by $(\alpha a)(f) = af$.

Next, we extend the action of E to types with a filling operation. Actually, we define a filling operation $E(c_A)$ on EA assuming only that c_A is a *levelwise* filling operation on A.

Proposition 14. *Given a levelwise filling operation c_A on A over Γ, we have a filling operation $E(c_A)$ on EA. Given a levelwise filling operation c_B on B over $\Gamma.A$, we have a filling operation $\tilde{E}(c_B)$ on $\tilde{E}B$. These constructions commute with substitution in Γ.*

Proof. We assume that A has a levelwise filling operation $c_A(X)$. We define then, for $f : Y \to X$

$$E(c_A)(X, \gamma, \psi, u)(i)(f) = c_A(Y)(\gamma f^+, \psi, uf^+)(i)$$

We can then check for $f : Y \to X$ and $g : Z \to Y$

$$c_{EA}(X, \gamma, \psi, u)(i)f(g) = c_A(Z)(\gamma(fg)^+, \psi, u(fg)^+)(i) = c_{EA}(Y, \gamma f^+, \psi, uf^+)(i)(g)$$

and hence c_{EA} is natural in X.

The construction of $\tilde{E}(c_B)$ is similar. $\qquad\square$

Corollary 15. *E defines a lex operation.*

Proof. We define $\langle\rangle$ in $E\mathbf{1}$ by $\langle\rangle(f) = ()$ and $\langle u, v\rangle : E(\sum_A B)\rho$ by $\langle u, v\rangle(f) = (u(f), v(f))$ for u in $(EA)\rho$ and v in $(\tilde{E}B)(\rho, u)$. Any universe $\mathscr{U}$ reflects the operations E and $\tilde{E}$ since the Grothendieck universe U used to construct $\mathscr{U}$ was assumed to contain $\mathscr{C}$. We then check that all conditions for a lex operation are satisfied. $\qquad\square$

Proposition 16. *If A is levelwise contractible over Γ, then EA is contractible. If B is levelwise contractible on $\Gamma.A$, then $\tilde{E}B$ is contractible on $\Gamma.EA$. These constructions commute with substitution in Γ.*

Proof. We assume that A has a levelwise extension operation $e_A(X)$. We define then, for $f : Y \to X$

$$e_{EA}(X, \rho, \psi, u)(f) = e_A(Y)(\rho f, \psi, uf)$$

We can then check for $f : Y \to X$ and $g : Z \to Y$

$$e_{EA}(X, \rho, \psi, u)f(g) = e_A(Z)(\rho fg, \psi, ufg) = e_{EA}(Y, \rho f, \psi, uf)(g)$$

and hence e_{EA} is an extension operation for EA natural in X.

The proof for $\tilde{E}B$ is similar. $\qquad \square$

In general, E may not be a descent data operation, since EA does not need to be E-modal. The next subsection will use the lex operation E to define a descent data operation.

5.2 Homotopy descent data

In this subsection, unless explicitly stated, we work in the internal language of the presheaf model over $\mathscr{C} \times \mathscr{B}$. Starting from the lex operation E, we define a new lex operation D. As before, we first define D on presheaves, and then show that it extends to a lex operation on presheaves with a filling operation. On presheaves with a filling operation, D will be a descent data operation.

We let P_n be the subpresheaf of $\mathbf{I}^{n+1}$ of elements $(i_0, i_1, \ldots, i_n)$ satisfying $i_0 = 1 \vee \cdots \vee i_n = 1$.

Let $s_k : \mathbf{I}^{n+1} \to \mathbf{I}^n$ be the map which omits the kth component, for $k = 0, \ldots, n$. Note that $s_k \vec{i}$ is in P_{n-1} if $\vec{i}$ is in P_n and $i_k = 0$.

Definition 17. *An element of DA is given by a family $u(\vec{i})$ in $E^{n+1}A$ defined on P_n and satisfying the compatibility conditions*[12] $u(\vec{i}) = E^k(\alpha)u(s_k\vec{i})$ *on $i_k = 0$.*

For instance, we have

$$u(0, i_1, i_2) = \alpha u(i_1, i_2) \qquad u(i_0, 0, i_2) = E(\alpha)u(i_0, i_2) \qquad u(i_0, i_1, 0) = E^2(\alpha)u(i_0, i_1)$$

We have an element $u(\vec{1})$ in each $E^{n+1}A$. We have a path $u(i,1)$ between $\alpha\, u(1)$ and $u(1,1)$ and a path $u(1,i)$ between $E(\alpha)\, u(1)$ and $u(1,1)$ in E^2A. But, in general, we need further higher coherence conditions.

We define $\eta_A : A \to DA$ by $(\eta_A\, a)(i_0, i_1, \ldots, i_n) = \alpha^{n+1}a$.

If B is a family of types over A, we define $\tilde{D}B\, u$ as the type of families $v(\vec{i})$ in $\tilde{E}^{n+1}B\, u(\vec{i})$ such that $u(\vec{i}), v(\vec{i}) = E^k(\alpha)(u(s_k\vec{i}), v(s_k\vec{i}))$ on $i_k = 0$.

If A is a family of types over Γ, we define DA family of types over Γ by $(DA)\rho = D(A\rho)$.

Proposition 18. *If A is a family of types over Γ with a levelwise filling operation, then DA has a filling operation. If B is a family of types over $\Gamma.A$ with a levelwise filling operation, then $\tilde{D}B$ has a filling operation. These constructions commute with substitution in Γ.*

Proof. We use that each $E^{n+1}A$ has a (uniform) filling operation by Proposition 14 hence is a family of types in the model over $\mathscr{C} \times \mathscr{B}$. We assume given γ in $\Gamma^{\mathbf{I}}$ and ψ in Φ and a partial element u_j in $(DA)\gamma(j)$ defined over $\psi \vee j = 0$. We explain how to define a total extension v_j in $(DA)\gamma(j)$. For this we define $v_j(\vec{i})$ in $E^{n+1}A$ by induction on n. Since $E^{n+1}A$ has a filling operation, we apply this filling operation to the partial element equal to $u_j(\vec{i})$ on $\psi \vee j = 0$ and equal to $E^k(\alpha)\, v_j(s_k(\vec{i}))$ if $i_k = 0$.

The construction for $\tilde{D}B$ is similar. $\qquad \square$

Corollary 19. *D defines a lex operation.*

One way to understand the definition of D from E is the following. Being a pointed endofunctor, E defines a cosemisimplicial diagram starting from EA, and DA is a strict way to realize the homotopy limit of this diagram using a P-weighted limit. We can think of P as a cofibrant resolution of the constant diagram on 1. A remark is that E, and hence each E^l, preserves the P-weighted

limit defining D. In particular, an element of $E^l(DA)$ is determined by a family $u(\vec{i})$ in $E^{l+n+1}A$ satisfying $u(\vec{i}) = E^{l+k}(\alpha)\, u(s_k\vec{i})$ on $i_k = 0$.

Proposition 20. *Let A be a fibrant family of types over Γ. We can build a path between η_{DA} and $D\eta_A$.*

Proof. An element of $(D^2A)\rho$ is given by a family $v(\vec{i})(\vec{j})$ in $E^{n+m+2}A$ satisfying the conditions

(1) $v(\vec{i})(\vec{j}) = E^k(\alpha)\, v(s_k\vec{i})(\vec{j})$ on $i_k = 0$
(2) $v(\vec{i})(\vec{j}) = E^{n+1+l}(\alpha)\, v(\vec{i})(s_l\vec{j})$ on $j_l = 0$

Given u in $(DA)\rho$, we define an element $\tilde{u}$ in $(D^2A)\rho$ by $\tilde{u}(\vec{i})(\vec{j}) = u(\vec{i}, \vec{j})$.
We compute, for u in $(DA)\rho$

$$(\eta_{DA}\, u)(\vec{i})(\vec{j}) = \alpha^{n+1}\, u(\vec{j}) = u(\vec{0}, \vec{j})$$

and we have a homotopy[13] connecting this map to $\tilde{u}$ by defining

$$v_k(\vec{i})(\vec{j}) = u(\vec{i} \wedge k, \vec{j}).$$

We also have

$$(D\eta_A\, u)(\vec{i})(\vec{j}) = E^{n+1}(\alpha^{m+1})\, u(\vec{i}) = u(\vec{i}, \vec{0})$$

and we have a homotopy connecting this map to $\tilde{u}$ by defining

$$w_k(\vec{i})(\vec{j}) = u(\vec{i}, k \wedge \vec{j})$$

By composition, we have a path between $D\eta_A$ and η_{DA}. □

Proposition 21. *If A is a levelwise contractible family of types over Γ, then DA is contractible. If B is a levelwise contractible family of types over a family of types A over Γ, then $\tilde{D}B$ is contractible over DA.* □

Proof. Let u be a partial element of $(DA)\rho$ of extent ψ. We use that each $E^{n+1}A$ has an extension operation by Proposition 16. We define $v(\vec{i})$ extending $u(\vec{i})$ by induction on n. We apply the extension operation of $(E^{n+1}A)\rho$ to find an element equal to $u(\vec{i})$ on ψ and equal to $E^k(\alpha)v(\vec{i})$ on $i_k = 0$.
    The proof of the second part is similar. □

Corollary 22. *Let $\sigma : A \to B$ be map between fibrant families of types over Γ. If σ is levelwise an equivalence, then $D\sigma$ is an equivalence.*

Proof. The fiber $\mathrm{fib}(\sigma)$ defines a levelwise contractible family of types over B. Hence, $\tilde{D}\mathrm{fib}(\sigma)$ is contractible over DB. Since D is a lex operation, $\mathrm{fib}(D\sigma)$ is contractible over DB and $D\sigma$ is an equivalence. □

Proposition 23. *Let A be a fibrant family of types over Γ. Then η_A is levelwise an equivalence and $D\eta_A$ is an equivalence.*

Proof. For this proposition, we work in the presheaf model over $\mathscr{B}$. If $\vec{f}$ is a composable chain of arrows, we write $\langle \vec{f} \rangle$ for its composition.
    Let A be a type over Γ. For ρ in $\Gamma(X)$, an element u of $(DA)\rho$ is a family of elements $u(\vec{i})(\vec{f})$ in $A\rho\langle\vec{f}\rangle$ satisfying the compatibility conditions. For a in $A\rho$, the element $\eta_A\, a$ is the family of elements

$$(\eta_A\, a)(\vec{i})(\vec{f}) = a\langle\vec{f}\rangle$$

We define an inverse $G : (DA)(X) \to A(X)$ of $\eta_A(X)$ by taking Gu to be the element $u(1)(\mathsf{id}_X)$. We then have $G(\eta_A\,a) = a$. The element $\eta_A\,(G\,u)$ satisfies

$$(\eta_A\,(G\,u))(\vec{i})(\vec{f}) = (G\,u)\langle\vec{f}\rangle = u(1)(\mathsf{id})\langle\vec{f}\rangle = u(1,\vec{0})(\mathsf{id},\vec{f})$$

Define the element $\tilde{u}$ in $(DA)\rho$ by $\tilde{u}(\vec{i})(\vec{f}) = u(1,\vec{i})(\mathsf{id},\vec{f})$. We can define a homotopy

$$u_k(\vec{i})(\vec{f}) = u(1, k \wedge \vec{i})(\mathsf{id},\vec{f})$$

between $\eta_A\,(G\,u)$ and $\tilde{u}$ and we can define a homotopy

$$v_k(\vec{i})(\vec{f}) = u(k,\vec{i})(\mathsf{id},\vec{f})$$

between u and $\tilde{u}$. By composition, there is a path between u and $\eta_A\,(G\,u)$ and G is an inverse of $\eta_A(X)$.

This shows that η_A is levelwise an equivalence. Then $D\eta_A$ is an equivalence by Corollary 22. □

Corollary 24. *D defines a descent data operation.*

Proof. By Propositions 20 and 23. □

Note that a direct consequence of Corollary 22 is the following strictification result.

Theorem. *Let A and B be fibrant families of types over Γ that are D-modal. Then any levelwise equivalence $\sigma : A \to B$ is an equivalence.* □

5.3 Coherently constant functions

The way from which we get D from E can also be applied to the lex operation $EA = A^R$, where R is an arbitrary type. This amounts to give a map which is *coherently constant* as defined by Rezk (2010), Kraus (2015). It is there shown that a coherently constant function $R \to A$ can be factorized through a map $\|R\| \to A$ from the propositional truncation of R to A.

Our development actually provides a way to recover this result in the cubical setting.

Lemma 25. *There is a path between $D\eta_A$ and η_{DA}. Hence, η_A is an equivalence as soon as it has a left inverse.*

Proof. The first part is proved exactly as in Proposition 20. If $p \circ \eta_A$ is path equal to id, we also have $Dp \circ D\eta_A$ path equal to $D\mathsf{id} = \mathsf{id}$ and hence $Dp \circ \eta_{DA}$ is path equal to id. But we have $Dp \circ \eta_{DA} = \eta_A \circ p$ and hence p is also a right inverse of η_A, and η_A is an equivalence. □

Theorem. *If $u : DA$, then the coherently constant function $u(1) : R \to A$ can be factorized through a map $\|R\| \to A$.*

Proof. An element of DA is a sequence of elements $u(\vec{i})(\vec{x})$ in A for $\vec{i}$ in P_n and $\vec{x}$ in R^{n+1} with $u(\vec{i})(\vec{x}) = u(s_k\vec{i})(s_k\vec{x})$ on $i_k = 0$. Given an element x in R, we can build a left inverse p of $\eta_A : A \to DA$ by taking $pu = u(1)(x)$. By the previous lemma, we thus get an element of $R \to \mathsf{isEquiv}(\eta_A)$, and so of $\|R\| \to \mathsf{isEquiv}(\eta_A)$ which provides a factorization of a coherently constant map $R \to A$ through $R \to \|R\|$. □

5.4 Case of a monoid

We consider the special case where the base category is a monoid $\mathcal{M}$. If $\vec{x}$ is a sequence $(x_0, \ldots, x_n)$, we write $t_k\vec{x}$ for the sequence where we omit x_k and replace x_{k+1} by $x_k x_{k+1}$ for $k < n$ and $t_n\vec{x}$ is the sequence where we omit x_n. A type in the presheaf model is a type A with an $\mathcal{M}$-action, and an element of DA is then a family of elements $u(\vec{i})(\vec{x})$ in A with $\vec{i}$ in P_n and $\vec{x}$ in $\mathcal{M}^{n+1}$ satisfying the compatibility conditions

(1) $u(\vec{i})(\vec{x}) = u(s_k\vec{i})(t_k\vec{x})$ on $i_k = 0$ for $k < n$ and

(2) $u(\vec{i})(\vec{x}) = u(s_n\vec{i})(t_n\vec{x})x_n$ on $i_n = 0$

We define the $\mathscr{M}$-action on DA by $ux(\vec{i})(x_0, \ldots, x_n) = u(\vec{i})(xx_0, \ldots, x_n)$.

As a special case, let $\mathscr{M}$ be the walking idempotent. Let $e^2 = e$ be the non-trivial idempotent element of $\mathscr{M}$. Here is an example of a non-modal type which is levelwise contractible, but not contractible. Let Γ be the set with elements ρ_1, ρ_2, and ρ with $\rho_1 e = \rho_2 e = \rho$. We let A be the following type. We let $A\rho_1$ be the point a_1 and $A\rho_2$ be the point a_2 and $A\rho$ be the groupoid with two isomorphic objects u_1, u_2 with $a_i e = u_i$ for $i = 1, 2$. The type A is then levelwise contractible, but it has no global point.[14]

5.5 Generalization to a Grothendieck topology

A Grothendieck topology J on the category $\mathscr{C}$ defines a set $\mathsf{C}(X, I) = \mathsf{J}(X)$ and we have a family E_S indexed by $S : \mathsf{C}$ defined as follows.

Let ρ be in $\Gamma(X)$ and S in $\Gamma \to \mathsf{C}$ so that $S\rho$ is in $\mathsf{C}(X)$, which is the set $\mathsf{J}(X)$ of sieves on X seen as a discrete space. We now define

$$(E_S A)\rho = \prod_{f \in S\rho} A\rho f$$

In this way, we get a family of lex operations E_S and an associated family of descent data operations D_S indexed by $S : \mathsf{C}$.

All previous results about E and D adapt to this generalization. For instance, Proposition 16 becomes the following.

Proposition 26. *Let S be in $\Gamma \to \mathsf{C}$. If A in $\mathsf{Type}(\Gamma)$ is such that we have an extension operation $e(\rho, f)$ for each ρ in $\Gamma(X)$ and f in $S\rho$ which extends any partial element of $A\rho f$ to a total element, then $E_S A$ and $D_S A$ are contractible.*

Proof. We show that $E_S A$ is contractible by defining a uniform extension operation $e_{E_S A}(X, \rho, \psi, u)(f) = e(\rho, f)(\psi, uf)$ like in Proposition 16. It follows that each $E_S^n A$ is contractible. Like in Proposition 21, we can then show that $D_S A$ is contractible. $\qquad\square$

Note that if $S_1\rho$ is a subset of $S_2\rho$ for all ρ, then we have a canonical projection map $D_{S_2}A \to D_{S_1}A$ that coheres with the pointings. If A is D_{S_1}-modal, then a left inverse of $\eta_A^{S_1}$ composed with this projection map is a left inverse of $\eta_A^{S_2}$. Hence, a D_{S_1}-modal type is also D_{S_2}-modal and we have $D_{S_1} \leq D_{S_2}$ for the preorder defined in Subsection 3.2. Since J is a Grothendieck topology, the family D_S over $S : \mathsf{C}$ is filtered. Thus, we can apply Theorem 3.2.

We also note the following.

Proposition 27. *For any (ordinary) sheaf A for the topology J, the associated presheaf $C_A(X, I) = A(X)$ is D_S-modal for all S in J.*

Proof. We reason in the presheaf model over $\mathscr{B}$. The map $A(X) \to A(X)^{\mathbf{I}}$ is an isomorphism, since $\mathbf{I}$ is tiny. If S is a sieve on X, then $D_S C_A(X)$ is *isomorphic* to the set of families u_f in $A(Y)$ for $f : Y \to X$ in S such that $u_f g = u_{fg}$ if $g : Z \to Y$ and $\eta_{C_A}(a)$ for a in $A(X)$ corresponds to the family $u_f = af$ through this isomorphism. Since A is a sheaf, η_{C_A} is a (strict) isomorphism and an equivalence. $\qquad\square$

Each $S : \mathsf{C}$ defines a strict proposition, and hence a lex modality by exponentiation. An element of v of $(A^S)\rho$ is a family $v(f)$ in $A\rho f$, for f in the sieve $S\rho$, and such that $v(fg) = v(f)g$. We also define vh in $(A^S)\rho h$ by $vh(g) = v(hg)$.

Proposition 28. *There is a strict isomorphism between $D_S A$ and $(DA)^S$. Through this isomorphism, the unit map $A \to D_S A$ can be seen as the composition of the canonical map $A \to A^S$ and $(\eta_A)^S : A^S \to (DA)^S$. It follows that if A is D-modal, then A is modal for D_S if and only if it is modal for the lex modality defined by exponentiation with S.*

Proof. An element of $(DA)\rho$ for ρ in $\Gamma(X)$ is given by a sequence $u(\vec{i})(\vec{f})$ in $A\rho\langle\vec{f}\rangle$ with $\vec{i}$ in P_n, and $\vec{f}$ a sequence of composable maps ending in X and satisfying some compatibility conditions. We have defined uf by $uf(\vec{i})(f_0, \vec{f}) = u(\vec{i})(ff_0, \vec{f})$.

For v in $(DA)^S\rho$, we define $\varphi v : D_S A$ by taking $\varphi v(\vec{i})(f_0, \vec{f}) = v(f_0)(\vec{i})(\mathrm{id}, \vec{f})$.

For u in $(D_S A)\rho$, we define $\psi u : (DA)^S\rho$ by taking $(\psi u)(f)(\vec{i})(f_0, \vec{f}) = u(\vec{i})(ff_0, \vec{f})$.

We can then check that φ and ψ are strict inverses. $\qquad\square$

6. Sheaf Model

We can now define an internal translation which provides a new model of univalent type theory for the descent data operation D, following the work in Quirin (2016). A type A, p of the new model is a type A *together with* a proof p that this type is D-modal, while an element of a pair A, p is an element of A. We go further than the interpretation in Quirin (2016) by interpreting (higher) inductive types as well in this model.

For this, it is convenient to introduce the type of *patch structures* on a type A as the type of (homotopy) left inverses of η_A, that is, the type of pairs p_A, l_A with $p_A : DA \to A$ and l_A a proof that p_A is a (homotopy) left inverse of η_A. Since D is a lex modality, this type is a *proposition* equivalent to the type $\mathsf{isMod}_D(A)$. If p_A, l_A is a patch structure on A, and B a family of types over A, we let the type of *dependent patch structures* on B over p_A, l_A be the type of operations $p_B : \prod_{u:DA} (\tilde{D}B)u \to B(p_A u)$ together with $l_B : \prod_{a : A, b:B a} p_B (\eta_A a)(\eta_B b) =_{l_A a} b$, where $u =_{l_A a} v$ denotes the type of paths over the path $l_A a$ connecting $u : B(p_A(\eta_A a))$ and $v : B a$.

Lemma 29. *Given a patch structure p_A, l_A on a type A and B a family of types over A, the type of dependent patch structures on B over p_A, l_A is a proposition equivalent to the product over $a : A$ of the type of patch structures over $B a$.*

Proof. Since η_A is an equivalence, this type of dependent patch structures is equivalent to the type of operations $f : \prod_{a:A} D(B a) \to B(p_A(\eta_A a))$ together with an element of $\prod_{a : A, b:B a} f a (\eta_B b) =_{l_A a} b$. By the type-theoretic axiom of choice, this type is equivalent to

$$\prod_{a : A} \sum_{g : D(B a) \to B(p_A(\eta_A a))} \prod_{b : B a} g (\eta_B b) =_{l_A a} b$$

which is equivalent to

$$\prod_{a : A} \sum_{g : D(B a) \to B a} \prod_{b : B a} g\, b =_{B a} b$$

which is exactly the product over $a : A$ of the type of patch structures over $B a$. $\qquad\square$

In order to interpret the type of natural numbers with the desired computation rules, we use the following higher inductive type,[15] which "forces" the type Nat to have a patch structure:

$$
\begin{array}{lcl}
\mathsf{zero} & : & \mathsf{Nat} \\
\mathsf{succ} & : & \mathsf{Nat} \to \mathsf{Nat} \\
\mathsf{patch} & : & D\,\mathsf{Nat} \to \mathsf{Nat} \\
\mathsf{linv} & : & \prod_{x:\mathsf{Nat}} \mathsf{patch}(\eta_{\mathsf{Nat}}\, x) =_{\mathsf{Nat}} x
\end{array}
$$

In order to build this type in the model, we can follow the method explained in Coquand et al. (2018), which is presented in Appendix B. This type Nat satisfies the following induction principle. Given a family of types P over Nat with operations $a : P$ zero and $f : \prod_{n:\text{Nat}} P\, n \to P\,(\text{succ}\, n)$ and a dependent patch structure p, l for P over patch, linv, we have $\text{rec} : \prod_{\text{Nat}} P$ such that

$$\text{rec zero} = a \quad \text{rec}\,(\text{succ}\, n) = f\, n\,(\text{rec}\, n) \quad \text{rec}\,(\text{patch}\, u) = p\, u\,(\tilde{D}\text{rec}\, u) \quad \text{rec}\,(\text{linv}\, n\, r) = l\, n\,(\text{rec}\, n)\, r$$

Lemma 29 shows then that Nat interprets the type of natural numbers in the model of D-modal types.

The same idea applies to the interpretation of other inductive types such as W-types, and also works for higher inductive types. For instance, the suspension of a type A will be defined as

$$
\begin{array}{lcl}
\text{north, south} & : & T \\
\text{merid} & : & A \to \text{north} =_T \text{south} \\
\text{patch} & : & DT \to T \\
\text{linv} & : & \prod_{z:T} \text{patch}(\eta_T\, z) =_T z
\end{array}
$$

This interpretation of type theory with univalence and higher inductive types generalizes to the case where we have a Grothendieck topology J on the base category defining a *filtered family* of descent data operations D_S, $S : \mathsf{C}$ (see Subsection 5.5). Types that are modal for all D_S form then a model of univalent type theory, and we can interpret inductive types as well. For instance, for the type of natural numbers, we consider

$$
\begin{array}{lcl}
\text{zero} & : & \text{Nat} \\
\text{succ} & : & \text{Nat} \to \text{Nat} \\
\text{patch} & : & \prod_{S:\mathsf{C}} D_S\, \text{Nat} \to \text{Nat} \\
\text{linv} & : & \prod_{S:\mathsf{C}, x:\text{Nat}} \text{patch}_S(\eta_{\text{Nat}}^S\, x) =_{\text{Nat}} x
\end{array}
$$

We get in this way a model of higher sheaves over J.

6.1 A model with the negation of countable choice

As a first example of an application of sheaf models, we build a model of univalent type theory with higher inductive types with a countable family of sets E_n such that each of the homotopy propositional truncation $\|E_n\|$ is inhabited, but $\left\|\prod_{n:N} E_n\right\|$ is not globally inhabited.

We consider the following space, corresponding to the lattice generated by formal elements X_n and L_n with the relations $X_0 = 1$, $X_n = L_n \vee X_{n+1}$ and $L_{n+1} = L_n \wedge X_{n+1}$.
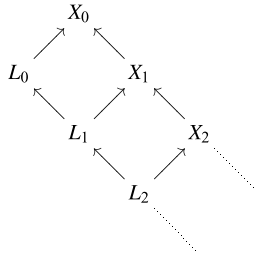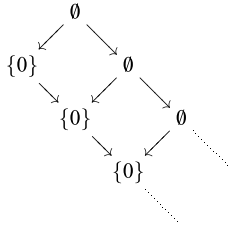

We can then consider the following family of *strict* sheaves $E_0, E_1, E_2, \dots$
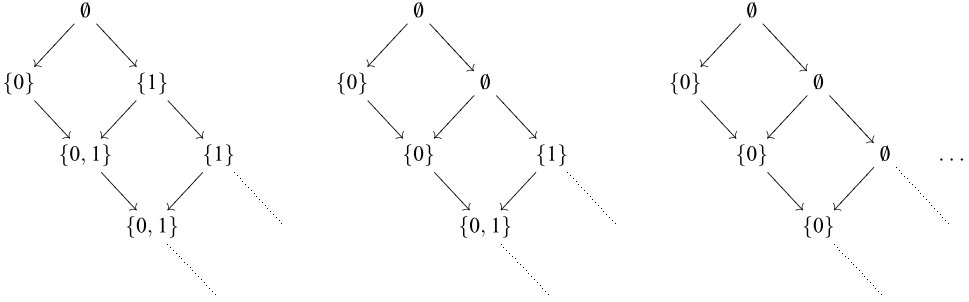
Using Proposition 26, one can show the following result.

Proposition 30. *The types* $\|E_n\|$ *are all contractible, while* $\prod_{n:N} E_n$ *is a strict proposition which does not hold globally.* □

Proof. $\prod_{n:N} E_n$ is isomorphic to the following proposition:

and the fact that $\|E_n\|$ is contractible follows from Proposition 26. □

Corollary 31. *There exists a model of univalent type theory with higher inductive types where countable choice does not hold.* □

As stressed in Swan and Uemura (2019), it is yet unknown how to build a model of univalent type theory and higher inductive types satisfying countable choice in a constructive metatheory. (Countable choice holds in a classical metatheory in the simplicial set model.)

6.2 A model of the algebraic closure of a field

Let F be a field that we suppose of characteristic 0 to simplify the discussion. In Mannaa and Coquand (2013), we present a sheaf model where we can build the algebraic closure of F in a constructive way. The objects of the base category are triangular F-algebras, and the two basic coverings are

(1) the two projections $A \to A_1$ and $A \to A_2$ if $A = A_1 \times A_2$
(2) the map $A \to A[X]/(P)$ if P is a monic separable polynomial

We can then show, as in Mannaa and Coquand (2013), that the presheaf $L(A) = \hom (F[X], A)$ is actually a sheaf and it can be seen as the algebraic closure of F. Similarly, the presheaf $G_m(A) = \hom (F[X, 1/X], A)$ is a sheaf. As explained in Mannaa and Coquand (2013), there is *no* operation in the model which associates a root as a function of the coefficients of the polynomial.

In the type-theoretic sheaf model over this site, we can consider the presheaf $C_L(A, I) = L(A)$, which is a type-theoretic sheaf by Proposition 27. We can then show using Proposition 26.

Proposition 32. *We have* $\prod_{a_1,\dots,a_n : C_L} \left\| \sum_{x : C_L} x^n + a_1 x^{n-1} + \cdots + a_n =_{C_L} 0 \right\|.$

It would be interesting to explore further this model, for instance computing the cohomology groups of G_m, but we leave this for future work.

7. Variation with Another Notion of Cofibration

We explain how to modify the definition of filling operation if we work with the other notion of cofibration classified by Φ_{lw}. Recall that an element of $\Phi_{\mathsf{lw}}(X)$ is no longer constant, but is given by a family of elements ψ_f in $\Phi_{\mathscr{B}}$ for $f : Y \to X$ and satisfying $\psi_f \leq \psi_{fg}$ if $g : Z \to Y$.

All the main results above still hold for this new notion of cofibration, suitably modified. The notion of *filling operation* for A is given by an operation c_A which takes as argument γ in $\Gamma(X)^{\mathbf{I}_{\mathscr{B}}}$ and ψ in $\Phi_{\mathsf{lw}}(X)$ and a family of elements $u_f(i)$ in $A\gamma(i)f$ for $f : Y \to X$ on the extent $\psi_f \vee i = 0$ such that $u_f(i)g = u_{fg}(i)$ for $g : Z \to Y$ on the extent $\psi_f \vee i = 0$. (There is a dual operation with $i = 1$ instead.) It produces an element $c_A(X, \gamma, \psi, u)(i)$ in $A\gamma(i)$ such that

(1) $c_A(X, \gamma, \psi, u)(i)f = u_f(i)$ on $\psi_f \vee i = 0$,
(2) $c_A(X, \gamma, \psi, u)(i)f = c_A(Y, \gamma', \psi f, u')(i)$ with $\gamma'(i) = \gamma(i)f$ and $u'_g(i) = u_{fg}(i)$ on the extent $\psi_{fg} \vee i = 0$ for $g : Z \to Y$.

For instance, Proposition 14 becomes the following result.

Lemma 33. *If A has a levelwise filling operation $c_A(X)$, then EA has a filling operation.*

Proof. We take γ in $\Gamma(X)^{\mathbf{I}_{\mathscr{B}}}$ and $u_f(i)$ in $(EA)\gamma(i)f$ on the extent $\psi_f \vee i = 0$ and we define $v(i) = c_{EA}(X, \gamma, \psi, u)(i)$ in $(EA)\gamma(i)$. For $f : Y \to X$, we take (filling at level Y)

$$v(i)(f) = c_A(Y)(\gamma', \psi', u')$$

where $\gamma'(i) = \gamma(i)f$ and $\psi' = \psi_f$ and $u'(i) = u_f(i)(\mathrm{id}_Y)$ in $A\gamma(i)f$ on the extent $\psi_f \vee i = 0$. ☐

Let us give some examples.

The first example is when $\mathscr{C}$ is the poset $0 \leq 1$. In this case, a global type A is given by two spaces with a map $A(1) \to A(0)$. An element of $\Phi_{\mathsf{lw}}(0)$ is an element of $\Phi_{\mathscr{B}}$ while an element of $\Phi_{\mathsf{lw}}(1)$ is a pair ψ_1, ψ_0 of elements of $\Phi_{\mathscr{B}}$ with $\psi_1 \leq \psi_0$. One can check that A is fibrant exactly if $A(0)$ is fibrant and $A(1) \to A(0)$ is a fibration, and a similar characterization holds in the relative situation (for a type A over Γ) and for trivial fibrations.

Let us note the following consequence of Proposition 23.

Proposition 34. *The following conditions are equivalent:*

(1) *all fibrant families of types are D-modal,*
(2) *all levelwise equivalences between fibrant families of types are equivalences,*
(3) *all fibrant families of types that are levelwise contractible are contractible.*

Proof. The fact that the first condition implies the second one is Theorem 5.2. In the reverse direction, given a fibrant family of types A, recall that η_A is a levelwise equivalence by Proposition 23. Then η_A is an equivalence and hence D-modal. The third condition is a special case of the second. The reverse direction holds since a (levelwise) equivalence can be described as a map with (levelwise) contractible fibers. ☐

Using this proposition, one sees that every type in the model is D-modal. The model coincides with the Reedy presheaf model described in Shulman (2015b) over the direct category $\mathscr{C}$ in the model of univalent type theory given by the base model. More generally, this will be the case for an arbitrary direct category $\mathscr{C}$ for which the inclusion of objects into morphisms given by identities is decidable.

The second example is the walking retract $\mathscr{C}$ generated by maps $f: 0 \to 1$ and $g: 1 \to 0$ satisfying $gf = \mathrm{id}_0$. Note that $\mathscr{C}$ is the idempotent splitting of the walking idempotent monoid $\mathscr{M}$ considered in Subsection 5.4. This makes the cubical presheaf models (for both Φ and Φ_{lw}) over $\mathscr{C}$ and $\mathscr{M}$ equivalent. Level 0 in the model over $\mathscr{C}$ corresponds to the fixpoints of the action of e in the model over $\mathscr{M}$. Taking Φ_{lw} as the cofibration classifier, the model of modal types gives a model for pointed families in a cubical model. It is homotopically correct in the sense that the equivalences are levelwise.

One might ask if types in the above model are already D-modal, similar to what happens for the poset $0 \leq 1$. More generally, one might attempt to generalize from a direct category $\mathscr{C}$ to a Reedy category $\mathscr{C}$ that is elegant (cf. Bergner and Rezk 2013); the walking retract is an example of an elegant Reedy category, with coface map f and codegeneracy map g. Taking Φ_{lw} as the cofibration classifier, one might ask if the (trivial) fibrations are given by the (trivial) Reedy fibrations; as before, this would imply that every type in the model is D-modal. An equivalent condition is that the levelwise cofibrations (classified by Φ_{lw}) are also the Reedy cofibrations. This holds true in classical situations where cofibrations and monomorphisms coincide and gives rise to the classical model (Shulman 2015a) over an elegant Reedy category. This does not hold in our constructive setting. Ultimately, this is because the inclusions $A(X) \to A(Y)$ are not generally cofibrations for a global type A and a codegeneracy map $Y \to X$ in $\mathscr{C}$. For the case of the walking retract, this is the inclusion $A(0) \to A(1)$. In terms of a global type A in the model over the walking idempotent monoid $\mathscr{M}$, it is the inclusion of fixpoints of the action of e on A. For a counterexample, let S be a discrete space with undecidable equality in one of the concrete cubical models listed in subsection 4.4. Take $A = S \times S$ with the action of e given by swapping. Because of this, we conjecture that there should be fibrant types in this model that are not D-modal.

8. Related and Future Work

Shulman (2019) shows that all $(\infty, 1)$-toposes have strict univalent universes, using a classical metatheory. This work does not cover however (yet) higher inductive types and cumulativity of universes. There are close connections between Shulman's work and ours, which we plan to explore in future work. His work inspired some results about levelwise weak equivalences in Section 5.2, in particular Corollary 22.

Once we have a presheaf model of univalence with homotopical features such as ours, it is now understood (see, e.g., Boulier 2018; Sattler 2017) how to define a Quillen model structure whose (trivial) fibrations coincide with the (contractible) types. For the model of D-modal types, we expect that, similar to Shulman (2019), the weak equivalences are the levelwise weak equivalences and the fibrations are a variation[16] of the injective fibrations. If we use Φ_{lw}, and we start with a model of type theory which has a corresponding Quillen model structure equivalent classically to the one of spaces, we expect to get exactly the injective Quillen model structure of sheaves. Thus, we should get an elementary[17] description of the classical ∞-sheaf toposes. We leave this to future work.

Instead of parameterizing our construction over an external category $\mathscr{C}$, we could start from a internal category $\mathscr{C}$ in presheaves over $\mathscr{B}$. Note that the category of presheaves over an internal category in presheaves is still a presheaf category. Compared to the construction of Shulman (2019) (which instantiates at this level of generality), we seem to need less fibrancy assumptions on this internal category. We leave this generalization to future work.

Notes

1 Joyal's argument was using non-constructive reasoning in simplicial sets and then Barr's theorem (see Barr 1974). The present paper can be developed directly in the constructive framework of CZF with universes introduced by Aczel (1998).

2 The notion of lex operation appears implicitly in a natural way when describing the rules of inductive data types (Coquand and Paulin 1988). If we have a family D_a of lex operations indexed over $a : A$, we can consider the inductive type T with constructor $\mathsf{sup} : \prod_{a:A} (D_a T \to T)$ and elimination rule $\mathsf{rec}\, f : \prod_T P$ for $f : \prod_{a:A} \prod_{u:D_a T} (\tilde{D}_a P\, u \to P(\mathsf{sup}_a u))$. We can then write the computation rule

$$\mathsf{rec}\, f\, (\mathsf{sup}_a u) = f_a\, u\, (\tilde{D}_a (\mathsf{rec}\, f)\, u)$$

For justifying the use of such inductive definitions, we need some "accessibility" assumption on the functors D_a, which will be satisfied in the examples. In the special case where $D_a X$ is $X^{B\,a}$ for B a family of types over A we recover the W-type $W_A B$.

3 This means we have $DA : \mathscr{U}$ if $A : \mathscr{U}$, and $\tilde{D}B : DA \to \mathscr{U}$ if $B : A \to \mathscr{U}$ and A a type (crucially, A need not be in $\mathscr{U}$ here).

4 We owe this observation to Dan Licata.

5 Here $\Gamma.A$ denotes the extension of the context Γ by the type A and we have an extension map $\sigma^+ : \Delta.A\sigma \to \Gamma.A$ if $\sigma : \Delta \to \Gamma$.

6 This is one characterization of lex modalities in Rijke et al. (2020), Theorem 3.1.

7 Existence is defined as the propositional truncation of the dependent sum type.

8 This assumption simplifies two of our arguments, namely Propositions 20 and 23. However, our results should apply to the Cartesian variation of cubical models of Angiuli et al. (2017). There, one removes this hypothesis and instead adds that the diagonal $\mathbf{I} \to \mathbf{I} \times \mathbf{I}$ is a cofibration.

9 This assumption is not strictly speaking necessary, but simplifies the theory.

10 Classically, this corresponds to having all monomorphisms as cofibrations.

11 We thank Emily Riehl for this observation.

12 It is suggestive to think of the elements of DA as *choice sequences* extended in a *spatial* rather than *temporal* dimension.

13 At this point that we use that the object $\mathbf{I}$ in $\mathscr{B}$ has lattice operations but one could however instead define a homotopy in a more complex way by induction on the dimension for Cartesian cubes. The same remark applies for the proof of the next Proposition 23.

14 If a is such a point, we should have $a\rho_i = a_i$ and then $(a\rho_i)e = u_i$ and $a(\rho_1 e) = a(\rho_2 e) = a\rho$ which is not possible since u_1, u_2 are distinct.

15 This is equivalent to the type DN where N is the usual inductive type with constructors zero and succ, but the type DN does not satisfy the required computation rules.

16 We define a family of types to be injectively fibrant if it lifts against cofibrations that are levelwise trivial cofibrations.

17 Our development can be done in a weak predicative metatheory such as the one described in Aczel (1998).

References

Aczel, P. (1998). On relating type theories and set theories. In: Altenkirch, T., Naraschewski, W. and Reus, B. (eds.) *Types for Proofs and Programs, International Workshop TYPES'98, Kloster Irsee, Germany, March 27–31, 1998, Selected Papers*, Lecture Notes in Computer Science, vol. 1657, Springer, 1–18.

Angiuli, C., Brunerie, G., Coquand, T., Hou (Favonia), K.-B., Harper, R. and Licata, D. R. (2017). Cartesian cubical type theory. Draft.

Avigad, J., Kapulkin, K. and Lumsdaine, P. L. (2015). Homotopy limits in type theory. *Mathematical Structures in Computer Science* **25** (5) 1040–1070.

Barr, M. (1974). Toposes without points. *Journal of Pure and Applied Algebra* **5** 265–280.

Bergner, J. E. and Rezk, C. (2013). Reedy categories and the Θ-construction. *Mathematische Zeitschrift* **274** (1–2) 499–514.

Beth, E. W. (1956). *Semantic Construction of Intuitionistic Logic*. Medededlingen der koninklijke Nederlandse Akademie van Wetenschappen, afd. Letterkunde. Nieuwe Reeks, Deel 19, No. 11. N. V. Noord-Hollandsche Uitgevers Maatschappij, Amsterdam.

Boulier, S. P. (2018). *Extending Type Theory with Syntactic Models. (Etendre la théorie des types à l'aide de modèles syntaxiques)*. Phd thesis, Ecole nationale supérieure Mines-Télécom Atlantique Bretagne Pays de la Loire, France.

Cohen, C., Coquand, T., Huber, S. and Mörtberg, A. (2015). Cubical type theory: A constructive interpretation of the univalence axiom. In: Uustalu, T. (ed.) *21st International Conference on Types for Proofs and Programs, TYPES 2015, May 18–21, 2015, Tallinn, Estonia*, LIPIcs, vol. 69, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 5:1–5:34.

Coquand, T., Huber, S. and Mörtberg, A. (2018). On higher inductive types in cubical type theory. In: Dawar, A. and Grädel, E. (eds.) *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09–12, 2018*, ACM, 255–264.

Coquand, T., Mannaa, B. and Ruch, F. (2017). Stack semantics of type theory. In: *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20–23, 2017*, IEEE Computer Society, 1–11.

Coquand, T. and Paulin, C. (1988). Inductively defined types. In: Martin-Löf, P. and Mints, G. (eds.) *COLOG-88, International Conference on Computer Logic, Tallinn, USSR, December 1988, Proceedings*, Lecture Notes in Computer Science, vol. 417, Springer, 50–66.

Dybjer, P. (1995). Internal type theory. In: Berardi, S. and Coppo, M. (eds.) *Types for Proofs and Programs, International Workshop TYPES'95, Torino, Italy, June 5–8, 1995, Selected Papers*, Lecture Notes in Computer Science, vol. 1158, Springer, 120–134.

Eilenberg, S. and Zilber, J. A. (1950). Semi-simplicial complexes and singular homology. *Annals of Mathematics (2)* 51 499–513.

Grothendieck, A. (1960). Éléments de géométrie algébrique. I. Le langage des schémas. *Inst. Hautes Études Sci. Publ. Math.* 4 228.

Hofmann, M. (1997). Syntax and semantics of dependent types. In: *Semantics and Logics of Computation (Cambridge, 1995)*, Publications of the Newton Institute, vol. 14, Cambridge, Cambridge University Press, 79–130.

Joyal, A. (1984). Lettre à Grothendieck.

Kaposi, A., Huber, S. and Sattler, C. (2019). Gluing for type theory. In: Geuvers, H. (ed.) *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24–30, 2019, Dortmund, Germany*, LIPIcs, vol. 131, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 25:1–25:19.

Kraus, N. (2015). *Truncation Levels in Homotopy Type Theory*. Phd thesis, University of Nottingham, UK.

Kripke, S. A. (1965). Semantical analysis of intuitionistic logic. I. In: *Formal Systems and Recursive Functions (Proc. Eighth Logic Colloq., Oxford, 1963)*, North-Holland, Amsterdam, 92–130.

Mannaa, B. and Coquand, T. (2013). Dynamic Newton-Puiseux theorem. *Journal of Logic & Analysis* **5** 1–22.

Orton, I. and Pitts, A. M. (2016). Axioms for modelling cubical type theory in a topos. In: Talbot, J.-M. and Regnier, L. (eds.) *25th EACSL Annual Conference on Computer Science Logic, CSL 2016, August 29 - September 1, 2016, Marseille, France*, LIPIcs, vol. 62, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 24:1–24:19.

Quirin, K. (2016). *Lawvere-Tierney Sheafification in Homotopy Type Theory. (Faisceautisation de Lawvere-Tierney en théorie des types homotopiques)*. Phd thesis, école des mines de Nantes, France.

Rezk, C. (2010). Toposes and homotopy toposes. Unpublished manuscript.

Rijke, E., Shulman, M. and Spitters, B. (2020). Modalities in homotopy type theory. *Logical Methods in Computer Science* **16** (1).

Sattler, C. (2017). The equivalence extension property and model structures. CoRR, abs/1704.06911.

Schreiber, U. and Shulman, M. (2012). Quantum gauge field theory in cohesive homotopy type theory. In: Duncan, R. and Panangaden, P. (eds.) *Proceedings 9th Workshop on Quantum Physics and Logic, QPL 2012, Brussels, Belgium, 10–12 October 2012*, EPTCS, vol. 158, 109–126.

Scott, D. S. (1980). Relating theories of the λ-calculus. In: *To HB. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*, London-New York, Academic Press, 403–450.

Shulman, M. (2015a). The univalence axiom for elegant Reedy presheaves. *Homology, Homotopy and Applications* **17** (2) 81–106.

Shulman, M. (2015b). Univalence for inverse diagrams and homotopy canonicity. *Mathematical Structures in Computer Science* **25** (5) 1203–1277.

Shulman, M. (2018). Brouwer's fixed-point theorem in real-cohesive homotopy type theory. *Mathematical Structures in Computer Science* **28** (6) 856–941.

Shulman, M. (2019). All $(\infty, 1)$-toposes have strict univalent universes. CoRR, abs/1904.07004.

Swan, A. and Uemura, T. (2019). On Church's thesis in cubical assemblies. CoRR, abs/1905.03014.

Troelstra, A. S. and van Dalen, D. (1988). *Constructivism in mathematics. Vol. II*, *Studies in Logic and the Foundations of Mathematics*, vol. **123**, Amsterdam, North-Holland Publishing Co. An introduction.

Univalent Foundations Program, T. (2013). *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study.

Voevodsky, V. (2015). An experimental library of formalized mathematics based on the univalent foundations. *Mathematical Structures in Computer Science* **25** (5) 1278–1294.

Weaver, M. Z. and Licata, D. R. (2020). A constructive model of directed univalence in bicubical sets. In: Hermanns, H., Zhang, L., Kobayashi, N. and Miller, D. (eds.) *LICS'20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8–11, 2020*, ACM, 915–928.

Wellen, F. (2017). *Formalizing Cartan Geometry in Modal Homotopy Type Theory*. Phd thesis, Karlsruher Institut für Technologie, Germany.

A. General Results for Lex Modalities

Some of our results hold for modalities in the sense of Rijke et al. (2020) that are not necessarily presented in a strict manner by a lex operation. The main example is the case of accessible modalities, which are implemented using higher inductive types that rarely give rise to a lex operation.

The purpose of this section is to prove these more general statements. We work in the homotopy type theory setting of Rijke et al. (2020). Universes are assumed univalent and closed under dependent sums, dependent products, and identity types. For statements involving accessible modalities, we also assume closure under higher inductive types.

In this section, we take terminology with potentially both strict and homotopical meaning to have the homotopical meaning by default. For example, equality refers to the identity type, and pullbacks refer to homotopy pullbacks (expressed using the identity type).

We write $\mathsf{Modality}(\mathscr{U})$ for the type of modalities on a universe $\mathscr{U}$. Recall from Rijke et al. (2020) that $M : \mathsf{Modality}(\mathscr{U})$ has an underlying subuniverse[18] of $\mathscr{U}$, the M-modal types $\mathscr{U}_M$. Subuniverses of $\mathscr{U}$ carry an evident poset structure. Following Rijke et al. (2020, Subsection 3.2), we obtain a poset structure also on $\mathsf{Modality}(\mathscr{U})$.

Definition 35. *Let $\mathscr{U}$ be a universe contained in a universe $\mathscr{U}'$. A modality M' on $\mathscr{U}'$ is an extension of a modality M on $\mathscr{U}$ if every M-modal type in $\mathscr{U}$ is M'-modal in $\mathscr{U}'$ and for $X : \mathscr{U}$, the canonical map $M'X \to MX$ is invertible.*

The above conditions mean that a $\mathscr{U}$-small type is M-modal exactly if it is M'-modal and M-connected exactly if it is M'-connected. In terms of the stable factorization systems $(\mathscr{L}, \mathscr{R})$ and $(\mathscr{L}', \mathscr{R}')$ corresponding to M and M', this means that $\mathscr{L}$ and $\mathscr{R}$ are the restrictions of $\mathscr{L}'$ and $\mathscr{R}'$ to maps between $\mathscr{U}$-small types. For this, recall Rijke et al. (2020, Subsection 1.2), that the left and right classes of the stable factorization system corresponding to a modality are the connected and modal maps, which are defined by having connected and modal fibers, respectively.

We write $\mathsf{Modality}(\mathscr{U} < \mathscr{U}')$ for the type of pairs (M, M') with M a modality on $\mathscr{U}$ and M' an extension of M to $\mathscr{U}'$. The poset structures on $\mathsf{Modality}(\mathscr{U})$ and $\mathsf{Modality}(\mathscr{U}')$ extend to a poset structure on $\mathsf{Modality}(\mathscr{U} < \mathscr{U}')$.

The following statement makes precise that up to (essential) size issues, a modality is lex exactly if the universe of modal types is modal. In particular, a "size-polymorphic" modality (acting compatibly on all universes) whose action on maps preserves smallness of fibers is lex exactly if universes of modal types are modal. This generalizes Proposition 11 to modalities; the smallness condition on fibers mirrors the dependent action $\tilde{D}$ on $\mathscr{U}$-small types we require for a lex operation D. For $M : \mathsf{Modality}(\mathscr{U})$, we denote by $\mathscr{U}_M$ the subuniverse of $\mathscr{U}$ of M-modal types.

Proposition 36. *For $(M, M') : \mathsf{Modality}(\mathscr{U} < \mathscr{U}')$:*

(1) *if M' is lex and preserves maps with $\mathscr{U}$-small fibers, then $\mathscr{U}_M$ is M'-modal;*

(2) *if $\mathscr{U}_M$ is M'-modal, then M is lex and M' preserves maps with $\mathscr{U}$-small fibers.*

Proof. For the first part of the statement, let M' be lex and preserve maps with $\mathscr{U}$-small fibers. To show that $\mathscr{U}_M$ is M'-modal, it suffices to construct a left inverse to $\eta^{M'}_{\mathscr{U}_M}$ (Rijke et al. 2020, Lemma 1.20). By univalence of $\mathscr{U}_M$, this means to find an extension

$$
\begin{array}{ccc}
\sum_{X:\mathscr{U}_M} X & \dashrightarrow & Z \\
\downarrow & & \vdots \;\; {\scriptstyle \mathscr{U}\text{-small and}} \\
& & {\scriptstyle M\text{-modal fibers}} \\
\mathscr{U}_M & \xrightarrow{\;\eta^{M'}_{\mathscr{U}_M}\;} & M'\mathscr{U}_M.
\end{array}
$$

We use the naturality square of $\eta^{M'}$ at the left map. The square is a pullback because M' is lex. The right map has $\mathscr{U}$-small fibers by assumption and has M-modal fibers because it is M'-modal as it goes between M'-modal types.

For the second part, assume that $\mathscr{U}_M$ is M'-modal. Then $\mathscr{U}_M$ is right orthogonal against M'-connected types, in particular M-connected types. This verifies condition (xiii) of Rijke et al. (2020, Theorem 3.1), making M lex. It remains to show that M' preserves maps with

$\mathscr{U}$-small fibers. Given such a map, we factor it using M as an M'-connected map followed by a map with fibers in $\mathscr{U}_M$. Since M' sends M'-connected maps to equivalences, it remains to show, given $Y : X \to \mathscr{U}_M$, that $M'(\sum_X Y) \to M'X$ has $\mathscr{U}$-small fibers. Since $\mathscr{U}_M$ is M'-modal, it is right orthogonal against $X \to M'X$. Thus, $Y : X \to \mathscr{U}_M$ extends uniquely to a map $Y' : M'X \to \mathscr{U}_M$. Looking at the classified maps, we obtain the following commuting diagram:

$$
\begin{array}{ccc}
\sum_X Y & \longrightarrow & \sum_{z:M'X} Y'(z) \\
\downarrow & & \downarrow \\
X & \xrightarrow{\ \eta_X^{M'}\ } & M'X.
\end{array}
$$

Since the right map has M-modal (hence also M'-modal) fibers, it is M'-modal. The top map is a pullback of $\eta_X^{M'}$, hence M'-connected. Since

$$
\sum_X Y \longrightarrow \sum_{z:M'X} Y'(z) \longrightarrow M'X
$$

and

$$
\sum_X Y \longrightarrow M'(\sum_X Y) \longrightarrow M'X
$$

are $(M'$-connected, M'-modal)-factorizations of the same map, they coincide. This shows that the map $M'(\sum_X Y) \to M'X$ is equal to $\sum_{z:M'X} Y'(z) \to M'X$, hence has $\mathscr{U}$-small fibers. $\qquad\square$

Recall from Rijke et al. (2020, Subsection 2.3), that accessible modalities admit canonical extensions to larger universes. If the accessible modality is lex, we observe that it satisfies the technical condition on smallness of fibers of Proposition 36. This means that the first part of that statement can also be regarded as a generalization of the direction from condition (i) to condition (iii) in Rijke et al. (2020, Theorem 3.11).

Corollary 37. *Let M be an accessible lex modality on a universe $\mathscr{U}$. Let M' be its extension to a universe $\mathscr{U}'$ containing $\mathscr{U}$. Then M' preserves maps with $\mathscr{U}$-small fibers.*

Proof. This follows from the second part of Proposition 36 since $\mathscr{U}_M$ is M'-modal by Rijke et al. (2020, Theorem 3.11). $\qquad\square$

Let $M : I \to \mathsf{Modality}(\mathscr{U})$ be a family of modalities. We write

$$
\mathscr{U}(M) = \sum_{X:\mathscr{U}} \prod_{i:I} X \text{ is } M_i\text{-modal}. \tag{1}
$$

for the meet of the subuniverses of modal types of M_i over $i : I$. We call a given meet $\bigwedge M$ of M *structural* if it is preserved under the forgetful functor to the poset of subuniverses. This means that its subuniverse of modal types is $\mathscr{U}(M)$. By Rijke et al. (2020, Theorem 3.11), part (i), M has a structural meet exactly if $\mathscr{U}_M$ admits a reflection in $\mathscr{U}$. In that case, $\bigwedge M$ is given by the reflection operation.

Given a family $(M, M') : I \to \mathsf{Modality}(\mathscr{U} < \mathscr{U}')$, we say that a given meet of (M, M') is structural if it is sent to structural meets of M and M' by the forgetful functors. Note that (M, M') has a structural meet exactly if M and M' have structural meets $\bigwedge M$ and $\bigwedge M'$, respectively, and $\bigwedge M'$ is an extension of $\bigwedge M$ to $\mathscr{U}'$. This unfolds to the following conditions:

- the subuniverse $\mathscr{U}_M$ of $\mathscr{U}$ admits a reflection L,
- the subuniverse $\mathscr{U}'_{M'}$ of $\mathscr{U}'$ admits a reflection L',
- for $X : \mathscr{U}$, the canonical map $L'X \to LX$ is invertible.

When considering diagrams in a poset, we will restrict our attention to shapes that are themselves posets. Note that in any poset, the limit of a (poset-indexed) diagram coincides with the

meet over the object components of the diagram. Nonetheless, it is useful to speak about limits of diagrams because this allows us to constrain the relations between the inputs objects.

A poset I is *filtered* if it is merely inhabited and for any two elements $x_0, x_1 : I$, there merely exists $y : I$ with $x_0, x_1 \leq y$. It is *cofiltered* if I^{op} is filtered. The following statement generalizes Theorem 3.2 to modalities.

Proposition 38. *Let $(M, M') : I \to \mathsf{Modality}(\mathcal{U} < \mathcal{U}')$ be a $\mathcal{U}$-small cofiltered diagram. If $\mathcal{U}_{M_i}$ is M_i'-modal for all $i : I$, then $\mathcal{U}(M) : \mathcal{U}'$ belongs to $\mathcal{U}'(M)$.*

Proof. Given $i : I$, we have to show that $\mathcal{U}(M)$ is M_i'-modal. Because I is cofiltered, we have

$$\mathcal{U}(M) = \mathcal{U}((M_j)_{j \leq i}),$$

so it suffices to show that $\mathcal{U}((M_j)_{j \leq i})$ is M_i'-modal. By assumption, $\mathcal{U}_{M_j}$ is M_j'-modal, hence M_i'-modal for $j \leq i$. We now use that a type X over $\mathcal{U}_{M_i}$ (M_i'-modal) is M_i'-modal exactly if the map $X \to \mathcal{U}_{M_i}$ is M_i'-modal. Given that $\mathcal{U}_{M_j} \to \mathcal{U}_{M_i}$ is M_i'-modal for $j \leq i$, it suffices to show that $\mathcal{U}((M_j)_{j \leq i}) \to \mathcal{U}_{M_i}$ is M_i'-modal. Observe that the fibers of the latter embedding are products of the fibers of the former embeddings. So the claim holds since modal types are closed under product (Rijke et al. 2020, Lemma 1.26). □

Corollary 39. *Let $(M, M') : I \to \mathsf{Modality}(\mathcal{U} < \mathcal{U}')$ be a $\mathcal{U}$-small cofiltered diagram with a structural meet ($\bigwedge M, \bigwedge M'$). If $\mathcal{U}_{M_i}$ is M_i'-modal for all $i : I$, then $\mathcal{U}_{\bigwedge M}$ is $\bigwedge M'$-modal.*

Proof. This is a direct consequence of Proposition 38 and the definition of structural meet. □

The following statement says that, up to the same size issues of Proposition 36, lex modalities are closed under structural cofiltered limits of modalities. In particular, structural cofiltered limits of "size-polymorphic" modalities whose actions on maps preserve smallness of fibers preserve left exactness.

Corollary 40. *In the situation of Corollary 39, if M_i' is lex for $i : I$ and preserves maps with $\mathcal{U}$-small fibers, then $\bigwedge M$ is lex and $\bigwedge M'$ preserves maps with $\mathcal{U}$-small fibers.*

Proof. This is the combination of Proposition 36 and Corollary 39. □

Finally, we specialize to the important case of accessible modalities.

Corollary 41. *Let $M : I \to \mathsf{Modality}(\mathcal{U})$ be a $\mathcal{U}$-small cofiltered diagram. If M_i is lex and accessible for all $i : I$, then the meet $\bigwedge M$ exists and also has these properties.*

Proof. Let $\mathcal{U}'$ be a universe containing $\mathcal{U}$. Let $(M, M') : I \to \mathsf{Modality}(\mathcal{U} < \mathcal{U}')$ be the extension of M given by Rijke et al. (2020, Theorem 3.36). By Rijke et al. (2020, Theorem 3.29), the meet of (M, M') exists, is structural, and $\bigwedge M$ is again accessible. By Rijke et al. (2020, Theorem 3.11), $\mathcal{U}_{M_i}$ is M_i'-modal for $i : I$. Applying Corollary 39, $\mathcal{U}_{\bigwedge M}$ is $\bigwedge M'$-modal. By the second part of Proposition 36, this makes $\bigwedge M$ lex. □

B Justification of the Type Nat

As in Coquand et al. (2018), to simplify the presentation, we assume that $\mathbf{I}$ also has a reverse operation.

We first build an "upper approximation" of the type Nat by defining a family of sets $A(X, I)$ together with maps $A(X, I) \to A(Y, J)$ for $f : Y \to X$ and $l : J \to I$. An element of $A(X, I)$ is of the form

- zero or
- succ a with a in $A(X, I)$ or

- patch u where u is a family $u(l, \vec{r})(\vec{f})$ in $A(X_{n+1}, J)$ with $l : J \to I$ and $\vec{r}$ in $\mathsf{P}_n(J)$ and $\vec{f}$ a sequence $f_0, \ldots, f_n$ of composable maps $f_0 : X_1 \to X, \ldots, f_n : X_{n+1} \to X_n$
- linv $a\, r$ for a in $A(X, I)$ and r in $\mathbf{I}(I)$ with $r \neq 0, 1$
- hcomp $[\psi \mapsto a]\, a_0$ with $\psi \neq 1$ in $\Phi(I)$ and a_0 in $A(X, I)$ and a a family $a_{f,l,r}$ in $A(Y, J)$ for $l : J \to I$ and $f : Y \to X$ such that $\psi l = 1$ and r in $\mathbf{I}(J)$

In this way, an element of $A(X, I)$ can be seen as a well-founded tree. As in Coquand et al. (2018), we can then define $A(X, I) \to A(Y, J)$, $a \mapsto a(f, l)$ for $f : Y \to X$ and $l : J \to I$ by well-founded induction on a

- $\mathsf{zero}(f, l) = \mathsf{zero}$
- $(\mathsf{succ}\, a)(f, l) = \mathsf{succ}\, (a(f, l))$
- $(\mathsf{patch}\, u)(f, l) = \mathsf{patch}\, (u(f, l))$ where $u(f, l)$ is a family $u(f, l)(m, \vec{r})(\vec{f}) = u(ml, \vec{r}, (f, \vec{f}))$ for $l : J \to I$ and $m : K \to J$
- $(\mathsf{linv}\, a\, r)(f, l) = \mathsf{linv}\, a(f, l)\, rl$ if $rl \neq 0, 1$ and $(\mathsf{linv}\, a\, r)(f, l) = a(f, l)$ if $rl = 1$ and $(\mathsf{linv}\, a\, r)(f, l) = \mathsf{patch}\, u$ if $rl = 0$ with[19] $u(\vec{r}, \vec{f}) = a\langle f, \vec{f} \rangle$
- $(\mathsf{hcomp}\, [\psi \mapsto a]\, a_0)(f, l) = a_{f,l,1}$ if $\psi l = 1$ and $(\mathsf{hcomp}\, [\psi \mapsto a]\, a_0)(f, l) = (\mathsf{hcomp}\, [\psi \mapsto a(f, l^+)]\, a_0(f, l)$ if $\psi l \neq 1$ where $a(f, l^+)$ is the family $a(f, l^+)_{g,m,r} = a_{fg,lm,r}$

Note that, as in Coquand et al. (2018), $A(X, I)$ together with these maps may not define a presheaf. However, we can now define a subset of "good" elements inductively

- zero is good
- $\mathsf{succ}\, a$ is good if a is good
- $\mathsf{linv}\, a\, r$ is good if a is good
- $\mathsf{patch}\, u$ is good if $u(l, \vec{r}, \vec{f})$ is a family of good elements satisfying the equations[20] $u(l, \vec{r}, \vec{f}) = u(l, s_k \vec{r}, t_k \vec{f})$ on $r_k = 0$ for $k < n$ and $u(l, \vec{r}, \vec{f}) = u(l, s_n \vec{r}, t_n \vec{f})(f_n, \mathsf{id})$ on $r_n = 0$, and $u(l, \vec{r}, \vec{f})(\mathsf{id}, m) = u(lm, \vec{r}m, \vec{f})$ and finally
- $\mathsf{hcomp}\, [\psi \mapsto a]\, a_0$ is good of a_0 is good and $a_{f,l,r}$ is a family of good elements such that $a_0(f, l) = a_{f,l,0}$ and $a_{f,l,r}(g, m) = a_{fg,lm,rm}$ for $f : Y \to X$, $g : Z \to Y$ and $l : J \to I$, $m : K \to J$

If we now define $\mathsf{Nat}(X, I)$ to be the subset of good elements of $A(X, I)$, we obtain a presheaf, which is also a sheaf satisfying the induction principle presented in Section 6.