Constructive Sheaf Models of Type Theory

Thierry Coquand, Fabian Ruch and Christian Sattler

Computer Science Department, University of Gothenburg

Introduction

Despite being relatively recent, the notion of (pre)sheaf model has a rich and intricate history which mixes different intuitions of topological, logical and algebraic nature. Eilenberg and Zilber [11] used a presheaf model (simplicial sets) to represent geometrical objects, and the intuition is of a *spatial* nature: we think of the objects of the base category $I, J, \ldots$ as basic "shapes" and a presheaf $A(I)$ gives a set of objects of shape I, objects that are related by the restriction maps. A little later, but independently, Beth [5] and Kripke [16] used respectively a sheaf and a presheaf model over trees to provide a formal semantics of intuitionistic logic. The motivations there were logical, and the objects of the base category/poset are thought of as stages of knowledge and the restriction maps have a *temporal* intuition. Scott [23] describes a presheaf model of higher-order logic and pointed out the potential interest for the semantics of λ-calculus. This was refined by Martin Hofmann [13] who provided a presheaf model of dependent type theory with universes. Such a semantics was used in an essential way in works on constructive semantics of type theory with univalence [7, 8, 18].

The generalisation of such presheaf models of dependent type theory, and especially of universes, to a *sheaf* model semantics is however non-trivial. The problem in generalising this semantics for universes comes essentially from the fact that the collection of sheaves don't 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. [12, Section 3.3]), and in general patching of mathematical structures. The generalisation of this to patching of higher structures was the content of the first part of Joyal's letter to Grothendieck [14]. 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 [31, 29]. This uses in a crucial way the fact that we have a *constructive* interpretation of univalence as in [7, 18], which can be relativised to any presheaf model. The main point is then that the notion of descent data induced by giving a collection of compatible objects defines a *left exact modality* (see [29, 19, 20]), which can then be used to build internally models of univalent type theory [19].

This work opens the possibility of generalising works of sheaf models of intuitionistic logic as in [28] to sheaf models of univalent type theory. It extends the previous work in [9] to a complete model of univalence, and has no restrictions for representing (higher) data types. We give only one application (independence of countable choice), but we expect for instance that results such as in [17] can be generalised as well, and that we can give a constructive account of works such as in [25, 33]. The present semantics (in a preliminary version) has already been used by Weaver and Licata [32] for building a constructive model of directed univalence.

1 Abstract notion of descent data

1.1 Lex operation

A lex operation is an internal functor which preserves unit type and sum types up to strict isomorphism. We work in a dependent type theory with product types $\Pi_A B : \mathcal{U}$ and sum types $\Sigma_A B : \mathcal{U}$ for $A : \mathcal{U}$

[1] Joyal's argument is non-constructive since it uses the classical model structure on simplicial sets and then Barr's theorem (see [4]). The present paper can be developed directly in the constructive framework of CZF with universes introduced by Aczel [1].

and $B : A \to \mathcal{U}$ and unit types $\mathbf{1}$ and universes[2]. Note that path/identity types are not needed for the definition of lex operation. We write $()$ the element of $\mathbf{1}$ and $(a, b) : \Sigma_A B$ for $a : A$ and $b : B\,a$ and π_1, π_2 the projection maps.

A *lex operation*[3] is given by a function $D : \mathcal{U} \to \mathcal{U}$ which is a strict functor on types: we have $Df : DA \to DB$ if $f : A \to B$ with $D(g \circ f) = Dg \circ Df$ and $D\mathsf{id}_A = \mathsf{id}_{DA}$ as strict equalities.

The operation D should also preserve the (strict) unit type $\mathbf{1}$. We have an element $\langle\rangle$ in $D\mathbf{1}$ and $x = \langle\rangle$ strictly if x is in $D\mathbf{1}$.

We assume next an extension of D to dependent types with $\widetilde{D}B : DA \to \mathcal{U}$ if $B : A \to \mathcal{U}$ with operations ensuring that $D(\Sigma_A B)$ is naturally (strictly) isomorphic to $\Sigma_{DA}\widetilde{D}B$.

We thus require $\widetilde{D}(B \circ f) = \widetilde{D}B \circ Df : DC \to \mathcal{U}$ if $f : C \to A$ together with an operation $\widetilde{D}s : \Pi_{DA}\widetilde{D}B$ if $s : \Pi_A B$ and a pairing operation $\langle u, v\rangle$ in $\Sigma_{DA}\widetilde{D}B$ if $u : DA$ and $v : (\widetilde{D}B)u$ with (strict) equations

$$D\pi_1\langle u, v\rangle = u \qquad \widetilde{D}\pi_2\langle u, v\rangle = v \qquad \langle (D\pi_1)w, (\widetilde{D}\pi_2)w\rangle = w$$

if $u : DA$ and $v : (\widetilde{D}B)u$ and $w : D(\Sigma_A B)$.

Proposition 1.1. *Any lex operation D defines a (strict) pointed endofunctor*[4].

Proof. We define $\eta_A\, a = (D\epsilon_a)\,\langle\rangle$ with $\epsilon_a = \lambda_{x:\mathbf{1}}\, a$. We then have for $f : A \to B$

$$(Df)(\eta_A\, a) = (Df \circ D\epsilon_a)\,\langle\rangle = (D\epsilon_{f\,a})\,\langle\rangle = \eta_B(f\,a)$$

Note furthermore that this natural transformation η_A is uniquely determined, since we should have $\eta_{\mathbf{1}}\,() = \langle\rangle$ and so $\eta_A\, a = \eta_A(\epsilon_a\,()) = (D\epsilon_a)(\eta_{\mathbf{1}}\,()) = (D\epsilon_a)\,\langle\rangle$. □

We assume furthermore that we have $(\widetilde{D}B)(\eta_A\, a) = D(B\,a)$ strictly for any $A : \mathcal{U}$ and $B : A \to \mathcal{U}$ and $a : A$. This further condition is not needed (cf. [21]) but will hold for the examples of lex operations we will analyse and it simplifies some arguments.

The canonical example of a lex operation is exponentiation with respect to a fixed given type R. We define $D : \mathcal{U} \to \mathcal{U}$ to be $DX = X^R$ and $(\widetilde{D}P)\,u$ to be $\Pi_{x:R}\, P(u\,x)$ for u in DX. We can then define $\langle u, v\rangle = \lambda_{x:R}\,(u\,x, v\,x)$. We have $\eta_X\, a = \lambda_{x:R}\, a$ and $(\widetilde{D}P)(\eta_X\, a) = (P\,a)^R = D(P\,a)$.

1.2 D-modal types

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

Theorem 1.2. *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. [3] explain how to build a fibration category in term of dependent type theory. Theorem 1.2 states that any lex operation defines a endomap of the associated fibration category. A lex operation preserves any homotopy limits (homotopy pullbacks, equalisers, fibers, ...).

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

Definition 1.1. A type A is D-modal if and only if the unit map $\eta_A : A \to DA$ is an equivalence.

[2] We follow the typical ambiguous notation used in [29].

[3] The notion of lex operation appears implicitly in a natural way when describing the rules of inductive data types [10]. If we have a family D_a of lex operations indexed over $a : A$, we can consider the inductive type T with constructor $\mathsf{sup} : \Pi_{a:A} D_a T \to T$ and elimination rule $\mathsf{rec}\ f : \Pi_T P$ for $f : \Pi_{a:A, u:D_a T}(\widetilde{D}_a P)u \to P(\mathsf{sup}\ \ a\ \ u)$. We can then write the computation rule

$$\mathsf{rec}\ f\ (\mathsf{sup}\ \ a\ \ u) = f\ u\ (\widetilde{D}_a(\mathsf{rec}\ f)\ u)$$

For justifying the use of such inductive definitions, we need some "accessibility" assumption on the (strict) functor D, which will be satisfied in the examples. In the special case where $D_a X$ is $X^{B\,a}$ for $B : A \to \mathcal{U}$ this generalizes the W-type WAB.

[4] We owe this observation to Dan Licata.

Proposition 1.3. *If A is D-modal and $B : A \to \mathcal{U}$ then B is a family of D-modal types over A if, and only if, $\Sigma_A B$ is D-modal.*

Proof. Let f be the map $\Sigma_A B \to \Sigma_{DA} \widetilde{D}B$ defined by $(a, b) \mapsto (\eta_A\, a, \eta_{B\,a}\, b)$. Since η_A is an equivalence, the map f is an equivalence if, and only if, each map $\eta_{B\,a}$ is an equivalence [29]. But f is an equivalence if and only if $\eta_{\Sigma_A B}$ is an equivalence. □

1.3 Abstract notion of descent data

Theorem 1.4. *The following conditions are equivalent, for a lex operation D*

1. *D defines a modality as axiomatised in [19, 20]*
2. *the map $D\eta_A$ is an equivalence, and $D\eta_A$ and η_{DA} are path equal*

Proof. The first condition implies the second using the results in [19, 20].

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 1.3 shows that D-modal types are closed by sum types. We thus only have to prove that the map

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

is an equivalence if B is D-modal [19].

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

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

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. □

Definition 1.2. An *abstract notion of descent data* is a lex operation D satisfying the equivalent conditions of Theorem 1.4.

Note that the first condition of Theorem 1.4 is a (homotopy) proposition. The second condition is the one which will be convenient to verify for the main examples.

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

1.4 Closure properties

Let D be a descent data.

Lemma 1.5. *For the map $\eta_A : A \to DA$ to be an equivalence, 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.*

Proof. If p_A is such a patch function, we have $\mathsf{id}_{DA} = D\mathsf{id}_A = D(p_A \circ \eta_A) = Dp_A \circ D\eta_A$ which is path equal to $Dp_A \circ \eta_{DA} = \eta_A \circ p_A$. Hence p_A is an inverse of η_A and η_A is an equivalence. □

Lemma 1.6. *For $B : A \to \mathcal{U}$ and any $u : DA$ the type $(\widetilde{D}B)u$ is D-modal.*

Proof. Since $D(\Sigma_A B)$ is D-modal, so is the isomorphic type $\Sigma_{DA} \widetilde{D}B$. Using Proposition 1.3, we have that $(\widetilde{D}B)u$ is a D-modal type for any $u : DA$. □

Proposition 1.7. *The type $\mathcal{U}_D = \Sigma_{\mathcal{U}} \mathsf{isMod}_D$ is a D-modal.*

Proof. Consider the following diagram

$$\begin{tikzcd} D\mathcal{U}_D \arrow[dr, "\widetilde{D}\pi_1"] & \\ \mathcal{U}_D \arrow[u, "\eta"] \arrow[r, "\pi_1"'] & \mathcal{U} \end{tikzcd}$$

It is homotopy commuting since $(\widetilde{D}\pi_1)(\eta\, X) = D(\pi_1\, X)$ is path equal to $\pi_1\, X$ for any $X : \mathcal{U}_D$ by univalence(!). Note also that π_1 is an embedding since isMod_D is a property.

Since $(\widetilde{D}\pi_1)\, A$ is D-modal by Lemma 1.6 for any $A : D\mathcal{U}_D$ the map $\widetilde{D}\pi_1 : D\mathcal{U}_D \to \mathcal{U}$ factorises through $\pi_1 : \mathcal{U}_D \to \mathcal{U}$ and the corresponding map $D\mathcal{U}_D \to \mathcal{U}_D$ is a left inverse of $\eta : \mathcal{U}_D \to D\mathcal{U}_D$ since π_1 is an embedding. Hence $\mathcal{U}_D$ is D-modal by Lemma 1.5. □

1.5 Model associated to a family of descent data

We can now define an internal translation which provides a new model of univalent type theory for any descent data D, following the work in [19]. 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.

In order to interpret the type of natural numbers with the desired computation rules (not covered in [19]), we need to use the following higher inductive type[5]

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

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.

The same idea applies to the interpretation of other inductive types such as the W-type. What is noteworthy is that this description of inductive types is actually in some sense simpler in this higher setting than in the setting of ordinary 1-sheaf (cf. the direct description of sheaves of W-types in [30]).

It also works for higher inductive type. For instance the suspension of a type A will be defined as

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

Note that having D defined as a *strict* functor is essential for such definitions.

1.6 Generalisation to a family of descent data

More generally, if we have a *family* of descent data D_S indexed by a given type $S : \mathsf{C}$, with corresponding maps $\eta^S_A : A \to D_S A$, we can consider $\mathsf{isMod}_{\mathsf{C}}(A)$ to be the proposition $\Pi_{S:\mathsf{C}}\, \mathsf{isMod}_{D_S}(A)$ and $\mathcal{U}_{\mathsf{C}}$ which is $\Sigma_{\mathcal{U}} \mathsf{isMod}_{\mathsf{C}}$. We let $\mathcal{U}_S$ to be $\Sigma_{\mathcal{U}} \mathsf{isMod}_{D_S}$

We let the preorder $D_1 \leqslant D_2$ on descent data mean that any D_1-modal type is D_2-modal. We say that C is *filtered* if we have $\exists_{S:\mathsf{C}}\, D_S \leqslant D_{S_1} \wedge D_S \leqslant D_{S_2}$ for any $S_1, S_2 : \mathsf{C}$[6].

Theorem 1.8. *If* C *is filtered then* $\mathcal{U}_{\mathsf{C}}$ *satisfies* $\mathsf{isMod}_{\mathsf{C}}$*.*

Proof. We have the following homotopy commuting diagram for any $S \leqslant S_1$ in C

$$\begin{tikzcd} D_{S_1}\mathcal{U}_{\mathsf{C}} \arrow[rr] \arrow[ddr, "\widetilde{D}_{S_1}\pi_1"] & & D_{S_1}\mathcal{U}_S \arrow[ddl, "\widetilde{D}_{S_1}\pi_1"'] \\ \mathcal{U}_{\mathsf{C}} \arrow[u, "\eta^{S_1}"] \arrow[rr] \arrow[dr, "\pi_1"'] & & \mathcal{U}_S \arrow[u, "\eta^{S_1}"'] \arrow[dl, "\pi_1"] \\ & \mathcal{U} & \end{tikzcd}$$

[5]To justify the use of such inductive definitions, we need some accessibility assumption on the strict functor D that will be satisfied in the examples.

[6]Existence is defined as the propositional truncation of sum type [29].

By Proposition 1.7, $\mathcal{U}_S$ is D_S-modal and so D_{S_1}-modal, and hence $\eta^{S_1} : \mathcal{U}_S \to D_{S_1}\mathcal{U}_S$ has an inverse. It follows that the map $\widetilde{D}_{S_1}\pi_1 : D_{S_1}\mathcal{U}_{\mathsf{C}} \to \mathcal{U}$ factorises through $\mathcal{U}_S \to \mathcal{U}$ and hence that for any $A : D_{S_1}\mathcal{U}_{\mathsf{C}}$ the type $(\widetilde{D}_{S_1}\pi_1)A$ is D_S-modal.

If C is filtered, this implies that the type $(\widetilde{D}_{S_1}\pi_1)A$ is D_{S_2}-modal for any S_2 in C. Hence the map $\widetilde{D}_{S_1}\pi_1 : D_{S_1}\mathcal{U}_{\mathsf{C}} \to \mathcal{U}$ factorises through $\mathcal{U}_{\mathsf{C}} \to \mathcal{U}$ and the corresponding map $\widetilde{D}_{S_1}\mathcal{U}_{\mathsf{C}} \to \mathcal{U}_{\mathsf{C}}$ is a left inverse of $\mathcal{U}_{\mathsf{C}} \to D_{S_1}\mathcal{U}_{\mathsf{C}}$. Hence $\mathcal{U}_{\mathsf{C}}$ is D_{S_1}-modal for any S_1 in C by Lemma 1.5. □

1.7 Example

If R is a *proposition*, and DX is X^R the two maps $D\eta_A$ and η_{DA} are path equals, and are equivalence, and D defines a notion of descent data.

The next section will define a new kind of descent data for any presheaf model.

2 Presheaf models

2.1 Cubical sets

We now assume given a small Cartesian category $\mathcal{B}$ with an object $\mathbf{I}$ which has two distinct global points 0 and 1. We also assume that $\mathbf{I}$ has a structure of bounded distributive lattice[7].

We write $I, J, K, \ldots$ the objects of $\mathcal{B}$. We assume given a presheaf $\Phi_{\mathcal{B}}$ which, together with $\mathbb{I}_{\mathcal{B}}$ satisfy the axioms of Orton and Pitts [18] (or Angiuli et al. [2] if we work in the Cartesian cubical set model). It is then known, following the work in [7, 18, 8], how to define a model of univalent type theory with higher inductive types, using as the interval the presheaf $\mathbb{I}_{\mathcal{B}}$ and as cofibrations the maps classified by $\Phi_{\mathcal{B}}$.

2.2 Presheaf model over cubical sets

Let $\mathcal{C}$ be another small 1-category. We write $X, Y, Z, \ldots$ the objects of $\mathcal{C}$. We write (X, I) the objects of $\mathcal{C} \times \mathcal{B}$ and (f, g) the morphisms of this category where f is a map of $\mathcal{C}$ and g a map of $\mathcal{B}$.

We can define an interval $\mathbb{I}$ on $\mathcal{C} \times \mathcal{B}$ by $\mathbb{I}(X, I) = \mathbb{I}_{\mathcal{B}}(I)$. We will consider two main examples for defining the new notion of cofibration:

1. the first example is simply to take $\Phi_0(X, I) = \Phi_{\mathcal{B}}(I)$.
2. the second example is to define an element ψ of $\Phi(X, I)$ to be a family ψ_f in $\Phi_{\mathcal{B}}(I)$ for $f : Y \to X$ such that $\psi_f \leqslant \psi_{ff_1}$ if furthermore $f_1 : Z \to Y$. We then define the restriction operation $\psi(f, g)$ to be the family $\psi(f, g)_{f_1} = \psi_{ff_1} g$ for $f : Y \to X$ and $f_1 : Z \to Y$.

The motivation for the second example is that if $\Phi_{\mathcal{B}}(I)$ is the collection of (decidable) sieves on I, then $\Phi(X, I)$ becomes the collection of (decidable) sieves on (X, I)[8].

Both define a notion of cofibration which still satisfies all required conditions of [18, 2]. We still get a model of univalent type theory (and higher inductive types) using the interval $\mathbb{I}$ and cofibrations Φ_0 or Φ. We are going to analyse the model obtained using the notion of cofibrations defined by Φ_0 and then indicate how to adapt these results for the other notion of cofibration.

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

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 $A\rho \to A\rho(f, g)$ denoted by $u \mapsto u(f, g)$ together with a composition operation (see [7, 18]). We write $\mathsf{Type}(\Gamma)$ the collection of all types with a composition operation over Γ. The set $\mathsf{Elem}(\Gamma, A)$ is then the set of sections: a family $a\rho$ in $A\rho$ such that $(a\rho)(f, g) = a(\rho(f, g))$ for any ρ in $\Gamma(X, I)$ and f, g map of codomain X, I.

Given a constructive Grothendieck universe U (see [1]), we write $\mathsf{Type}_U(\Gamma)$ the set of U-types, such that each set $A\rho$ is in U. The presheaf Type_U is then represented by a fibrant type $\mathcal{U}$ which is univalent [7].

[7]This assumption simplifies some arguments but our results still hold without this hypothesis and apply also to the Cartesian cubical sets (see [2]).

[8]Classically, this corresponds to having all monomorphisms as cofibrations.

2.3 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 $\mathcal{C} \times \mathcal{B}$ as in [18, 8] but also using the internal logic of the presheaf topos over $\mathcal{B}$. We will use both descriptions.

In the internal logic of the presheaf topos over $\mathcal{B}$, a context of the presheaf model over $\mathcal{C}$ 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 $\mathcal{B}$ 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_0 of cofibration is the constant presheaf $\Phi_0(X) = \Phi_{\mathcal{B}}$. The interval $\mathbb{I}$ is the constant interval $\mathbb{I}(X) = \mathbb{I}_{\mathcal{B}}$.

It will be convenient to introduce the following notation: if γ is an element of $\Gamma(X)^{\mathbb{I}}$ and $f : Y \to X$ we write γf^+ in $\Gamma(Y)^{\mathbb{I}}$ 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 [18, 8]) for A is given by an operation c_A which takes as argument γ in $\Gamma(X)^{\mathbb{I}_{\mathcal{B}}}$ and ψ in $\Phi_0(X) = \Phi_{\mathcal{B}}$ and a family of elements $u(i)$ in $A\gamma(i)f$ 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 $f : Y \to X$

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 [7]), 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 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 Examples below). 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.

2.4 Examples

We consider first the case where the base category is the group $\mathbb{Z}/2\mathbb{Z}$. We write τ the non-trivial element of this group. A context can be seen as a space with an action $\rho \mapsto \rho\tau$. A dependent type A over Γ has also an action $A\rho \to A\rho\tau$ denoted by $u \mapsto u\tau$ with a filling operation which is equivariant: we have $c_A(\gamma, \psi, u)(i)\tau = c_A(\gamma\tau^+, \psi, u\tau^+)(i)$.

Let A be the groupoid with two isomorphic objects swapped by τ. Then A is pointwise 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 pointwise equivalence but is not an equivalence.

The second example is when $\mathcal{C}$ is the poset $0 \leqslant 1$. In this case, a type A is given by two spaces with a map $A(1) \to A(0)$ which commutes with the filling operation.

Let us take $\Gamma(0)$ to be a point ρ_0 and $\Gamma(1)$ to be two points ρ_1, ρ_1'. We define $A\rho_0$ to be the groupoid with two isomorphic objects a_0, a_0' and $A\rho_1$ to be a point a_1 and $A\rho_1'$ to be a point a_1'. Then $A(0)$ and $A(1)$ are contractible, but A is not contractible since it has no global point[9].

3 Homotopy descent data

3.1 A lex operation

In this section, we work in the internal language of the presheaf topos over $\mathcal{B}$.

[9]Indeed, a global point a should satisfy the conditions $a\rho$ in $A\rho$ and $(a\rho)f = a(\rho f)$. It should thus satisfy $a\rho_1 = a_1$ and $a\rho_1' = a_1'$. If f is the unique map $0 \leqslant 1$ we have $\rho_1 f = \rho_1' f = \rho_0$ and $a_1 f = a_0$ and $a_1' f = a_0'$. There is then no possibility for the choice of $a\rho_0$.

For any A in $\mathsf{Type}(\Gamma)$ we define EA in $\mathsf{Type}(\Gamma)$. An element u of $(EA)\rho$, for ρ in $\Gamma(X)$ is given 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 in $\mathsf{Type}(\Gamma.A)$, we define $\widetilde{E}(B)$ in $\mathsf{Type}(\Gamma.EA)$. If ρ is in $\Gamma(X)$ and u is in $(EA)\rho$, then $\widetilde{E}(B)(\rho, u)$ is the space of families $v(f)$ in $B(\rho f, u(f))$.

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

Proposition 3.1. *If A has a pointwise filling operation then EA has a (uniform) filling operation and E defines a lex operation.*

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

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

We can then check for $f : Y \to X$ and $f_1 : Z \to Y$

$$c_{EA}(X, \gamma, \psi, u)(i)f(f_1) = c_A(Z)(\gamma(ff_1)^+, \psi, u(ff_1)^+)(i) = c_{EA}(Y, \gamma f^+, \psi, uf^+)(i)(f_1)$$

and hence c_{EA} is natural in X.

We can also define $\langle\rangle$ in $E\mathbf{1}$ by $\langle\rangle(f) = ()$ and $\langle u, v\rangle : E(\Sigma AB)\rho$ by $\langle u, v\rangle(f) = (u(f), v(f))$ for u in $EA\rho$ and v in $(\widetilde{E}B)(\rho, u)$, and check that all conditions for a lex operations are satisfied. $\square$

Proposition 3.2. *If A is pointwise contractible then EA is contractible.*

Proof. We assume that A has a pointwise 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 $f_1 : Z \to Y$

$$e_{EA}(X, \rho, \psi, u)f(f_1) = e_A(Z)(\rho ff_1, \psi, uff_1) = e_{EA}(Y, \rho f, \psi, uf)(f_1)$$

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

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

3.2 Homotopy descent data

In this section, unless explicitly stated, we work in the internal language of the presheaf model over $\mathcal{C} \times \mathcal{B}$. Starting from the lex operation E, we define a new lex operation D which is now a notion of descent data. We let P_n be the subpresheaf of $\mathbb{I}^{n+1}$ of elements $(i_0, i_1, \ldots, i_n)$ satisfying $i_0 = 1 \vee \cdots \vee i_n = 1$.

Let $s_k : \mathbb{I}^{n+1} \to \mathbb{I}^n$ 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 3.1. 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*[10] $u(\vec{i}) = E^k(\alpha)u(s_k\vec{i})$ 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 $u(1, i)$ between $\alpha\, u(1)$ and $u(1, 1)$ and a path $u(i, 1)$ 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 A is a family of types over Γ we define DA family of types over Γ by $(DA)\rho = D(A\rho)$.

[10]It is suggestive to think of the elements of DA as *choice sequences* [28] extended in a *spatial* rather than *temporal* dimension.

Proposition 3.3. *If A is a family of types with a pointwise filling operation, then DA has a filling operation.*

Proof. We use that each $E^{n+1}A$ has a (uniform) filling operation by Proposition 3.1 hence is a family of types in the model over $\mathcal{C} \times \mathcal{B}$. We assume given γ in $\Gamma^{\mathbb{I}}$ 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 $i_k = 0$. □

Corollary 3.4. *D defines a lex operation.*

A similar argument as the one for Proposition 3.3 using Proposition 3.2 instead proves the following.

Proposition 3.5. *If A is a family of types which is pointwise contractible, then DA is contractible. If B is a family of types over A which is pointwise contractible, then $\widetilde{D}B$ is contractible over DA.*

Corollary 3.6. *If $\sigma : A \to B$ is pointwise an equivalence then $D\sigma$ is an equivalence.*

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

Proposition 3.7. *η_A is pointwise an equivalence, and hence $D\eta_A$ is an equivalence by Corollary 3.6.*

Proof. For this proposition, we work in the presheaf model over $\mathcal{B}$.

If $\vec{f}$ is a composable chain of arrows we write $\langle \vec{f} \rangle$ 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 element

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

We define a (pointwise) inverse G of η_A by taking Gu to be the element $u(1)(\mathsf{id}_X)$. We then have $G(\eta_A\, a) = a$ strictly. 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 η_A[11]. □

One way to understand the definition of D from E is the following. Being a pointed endofunctor, E defines a semisimplical diagram starting from EA, and DA is a strict way to realise the homotopy limit of this diagram. A remark is that E, and hence each E^l, commutes strictly with such limit. 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 3.8. *We can build a path between the two maps 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 $j_l = 0$

[11] At this point that we use that the object $\mathbf{I}$ in $\mathcal{B}$ 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.

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 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}. □

Corollary 3.9. *The operation D defines a notion of descent data.*

Proof. By Propositions 3.7 and 3.8. □

Note that a direct consequence of Corollary 3.6 is the following strictification result.

Theorem 3.10. *If A and B are D-modal types and $\sigma : A \to B$ is pointwise an equivalence then σ is an equivalence.*

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 Kraus [15] and so a map $\|R\| \to A$ from the propositional truncation of R to A [15].

Our development actually provides a way to recover this result. Indeed, 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_A of $\eta_A : A \to DA$ by taking $p_A u = u(1)(x)$. Hence $R \to \mathsf{isEquiv}(\eta_A)$, and so $\|R\| \to \mathsf{isEquiv}(\eta_A)$ which provides a factorisation of a coherently constant map $R \to A$ through $R \to \|R\|$.

3.3 Case of a monoid

We consider the special case where the base category is a monoid M.n If $\vec{x}$ is a sequence $(x_0, \dots, x_n)$ we write $t_k\vec{x}$ the sequence where we omit x_k and replace x_{k+1} by x_kx_{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 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 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 M-action on DA by $ux(\vec{i})(x_0, \dots, x_n) = u(\vec{i})(xx_0, \dots, x_n)$.

The special case where M is the walking idempotent is particularly relevant since the corresponding model represents the model of family of pointed types (where the point is *strictly* preserved by any map). It will be interesting to find a natural statement in this model which requires the restriction to modal types.

Here is an example of a non-modal type which is pointwise contractible but not contractible. Let $e^2 = e$ be the non trivial idempotent element of M. 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 $a_ie = u_i$ for $i = 1, 2$. The type A is then pointwise contractible but it has no global point[12].

[12] 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.

3.4 Generalisation to a Grothendieck topology

A Grothendieck topology J on the category $\mathcal{C}$ defines a (strict) 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 is in $\Gamma \to \mathsf{C}$. Note that if $S\rho$ is in $\mathsf{C}(X) = \mathsf{J}(X)$, which is a set of sieves on X. An element of $(E_S A)\rho$ is now a family $u(f)$ in $A\rho f$ *with* f *in* $S\rho$. We define in this way an associated family of descent data D_S indexed by $S : \mathsf{C}$. This family is filtered and we can apply Theorem 1.8. Note that if $S_1\rho$ is a subset of $S_2\rho$ for all ρ then we have a canonical projection map $D_{S_2}(A) \to D_{S_1}A$. If A is D_{S_1}-modal 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 $D_{S_1} \leqslant D_{S_2}$ for the preorder defined in subsection 1.6.

Proposition 3.11. *If A in $\mathsf{Type}(\Gamma)$ and S in $\Gamma \to \mathsf{C}$ and ρ in $\Gamma(X)$ and $A\rho f$ is (pointwise) contractible for each f in $S\rho$ then we can find a uniform extension operation $e_{A\rho}(f,\psi,u)$ in $A\rho f$ for all $f : Y \to X$ and u partial element in $A\rho f$ of extent ψ.*

3.5 A model with the negation of countable choice

Using in an essential way the notion of *homotopy* descent data, we build a model with a countable family of (strict) sets P_n such that each $\|P_n\|$ is the true proposition, while $\Pi_{n:N}\, P_n$ is a strict proposition 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 $L_{n+1} = L_n \wedge X_{n+1}$.

Using Proposition 3.11 one can show the following result.

Proposition 3.12. *We have $\|L_0 + X_n\| = 1$ for all n while $\Pi_{n:N}\,(L_0 + X_n)$ is L_0.*

Corollary 3.13. *There exists a model of univalent type theory with higher inductive types where countable choice does not hold.*

As stressed in [27], 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.)

4 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, where an element of $\Phi(X)$ is no longer constant, but is given by a family of elements ψ_f in $\Phi_{\mathcal{B}}$ for $f : Y \to X$ and satisfying $\psi_f \leqslant \psi_{ff_1}$ if $f_1 : Z \to Y$.

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 $\Gamma(X)^{\mathbb{I}_{\mathcal{B}}}$ and ψ in $\Phi(X)$ and a family of elements $u(i)$ in $A\gamma(i)f$ on the extent $\psi \vee i = 0$ such that $u_f(i)g = u_{fg}(i)$ if $g : Z \to Y$ 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 3.1 becomes the following result.

Lemma 4.1. *If A has a pointwise filling operation $c_A(X)$ then EA has a filling operation.*

Proof. We take γ in $\Gamma(X)^{\mathbb{I}_{\mathcal{B}}}$ 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 $u'(i) = u_f(i)(\mathsf{id}_Y)$ in $A\gamma(i)f$ on the extent $\psi_f \vee i = 0$. □

4.1 Examples

We consider first the case where the base category is the group $\mathbb{Z}/2\mathbb{Z}$. We write τ the non-trivial element of this group. A context can be seen as a space with an action $\rho \mapsto \rho\tau$. An element of Φ is given by a pair ψ_1, ψ_τ of elements of $\Phi_\mathcal{B}$ with $\psi_1 \leqslant \psi_\tau \leqslant \psi_1$ so $\psi_1 = \psi_\tau$ and we can identify Φ with $\Phi_\mathcal{B}$ with the constant action. So in this case, the two notions of cofibration Φ_0 and Φ coincide.

The second example is when $\mathcal{C}$ is the poset $0 \leqslant 1$. In this case, a type A is given by two spaces with a map $A(1) \to A(0)$. An element of $\Phi(0)$ is an element of $\Phi_\mathcal{B}$ while an element of $\Phi(1)$ is a pair ψ_1, ψ_0 of elements of $\Phi_\mathcal{B}$ with $\psi_1 \leqslant \psi_0$. One can then check that a composition for A implies that the map $A(1) \to A(0)$ is a fibration. It is thus natural to conjecture that the model we obtain should be equivalent to the Reedy model described in [24], but we leave this point for further research.

5 Related and future works

Shulman [26] shows that all $(\infty, 1)$-toposes have strict univalent universes, using a classical metatheory. The work does not cover however (yet) higher inductive types and cumulativity of universes.

Once we have a presheaf model of univalence with a fibrant universe, it is now understood (see e.g. [22, 6]) how to define a notion of weak equivalences on all presheaves (not necessarily fibrant) with a corresponding Quillen model structure which satisfies the Frobenius condition and the fibration extension property. We expect that this should be possible as well for the sheaf models we describe, and that in the presheaf case, we get the injective model structure (since equivalences are pointwise equivalences).

Acknowledgement

Many thanks to Mathieu Anel, Steve Awodey, Martín Escardó, Dan Licata, Emily Riehl, Bas Spitters and Matthew Weaver for many discussions and remarks.

References

[1] Peter Aczel. On relating type theories and set theories. In Thorsten Altenkirch, Wolfgang Naraschewski, and Bernhard Reus, editors, *Types for Proofs and Programs, International Workshop TYPES '98, Kloster Irsee, Germany, March 27-31, 1998, Selected Papers*, volume 1657 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 1998.

[2] Carlo Angiuli, Guillaume Brunerie, Thierry Coquand, Kuen-Bang Hou (Favonia), Robert Harper, and Daniel R. Licata. Cartesian cubical type theory. 12 2017.

[3] Jeremy Avigad, Krzysztof Kapulkin, and Peter LeFanu Lumsdaine. Homotopy limits in type theory. *Math. Struct. Comput. Sci.*, 25(5):1040–1070, 2015.

[4] Michael Barr. Toposes without points. *J. Pure Appl. Algebra*, 5:265–280, 1974.

[5] Evert Willem Beth. *Semantic Construction of Intuitionistic Logic.* Noord-Hollandsche Uitg. Mij, 1956.

[6] Simon Pierre Boulier. *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, 2018.

[7] Cyril Cohen, Thierry Coquand, Simon Huber, and Anders Mörtberg. Cubical type theory: A constructive interpretation of the univalence axiom. In Tarmo Uustalu, editor, *21st International Conference on Types for Proofs and Programs, TYPES 2015, May 18-21, 2015, Tallinn, Estonia*, volume 69 of *LIPIcs*, pages 5:1–5:34. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[8] Thierry Coquand, Simon Huber, and Anders Mörtberg. On higher inductive types in cubical type theory. In Anuj Dawar and Erich Grädel, editors, *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*, pages 255–264. ACM, 2018.

[9] Thierry Coquand, Bassel Mannaa, and Fabian Ruch. Stack semantics of type theory. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–11. IEEE Computer Society, 2017.

[10] Thierry Coquand and Christine Paulin. Inductively defined types. In Per Martin-Löf and Grigori Mints, editors, *COLOG-88, International Conference on Computer Logic, Tallinn, USSR, December 1988, Proceedings*, volume 417 of *Lecture Notes in Computer Science*, pages 50–66. Springer, 1988.

[11] Samuel Eilenberg and J. A. Zilber. Semi-simplicial complexes and singular homology. *Ann. of Math. (2)*, 51:499–513, 1950.

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

[13] Martin Hofmann. Syntax and semantics of dependent types. In *Semantics and logics of computation (Cambridge, 1995)*, volume 14 of *Publ. Newton Inst.*, pages 79–130. Cambridge Univ. Press, Cambridge, 1997.

[14] André Joyal. Lettre à grothendieck, 1984.

[15] Nicolai Kraus. *Truncation levels in homotopy type theory.* PhD thesis, University of Nottingham, UK, 2015.

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

[17] Bassel Mannaa and Thierry Coquand. Dynamic Newton-Puiseux theorem. *J. Logic & Analysis*, 5, 2013.

[18] Ian Orton and Andrew M. Pitts. Axioms for modelling cubical type theory in a topos. In Jean-Marc Talbot and Laurent Regnier, editors, *25th EACSL Annual Conference on Computer Science Logic, CSL 2016, August 29 - September 1, 2016, Marseille, France*, volume 62 of *LIPIcs*, pages 24:1–24:19. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[19] Kevin Quirin. *Lawvere-Tierney sheafification in Homotopy Type Theory. (Faisceautisation de Lawvere-Tierney en théorie des types homotopiques).* PhD thesis, École des mines de Nantes, France, 2016.

[20] Egbert Rijke, Michael Shulman, and Bas Spitters. Modalities in homotopy type theory. *CoRR*, abs/1706.07526, 2017.

[21] Fabian Ruch. *Sheaf models of type theory.* PhD thesis, Göteborgs Universitet, Sweden, to appear.

[22] Christian Sattler. The equivalence extension property and model structures. *CoRR*, abs/1704.06911, 2017.

[23] Dana S. Scott. Relating theories of the λ-calculus. In *To H. B. Curry: essays on combinatory logic, lambda calculus and formalism*, pages 403–450. Academic Press, London-New York, 1980.

[24] Michael Shulman. Univalence for inverse diagrams and homotopy canonicity. *Mathematical Structures in Computer Science*, 25(5):1203–1277, 2015.

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

[26] Michael Shulman. All $(\infty, 1)$-toposes have strict univalent universes. *CoRR*, abs/1904.07004, 2019.

[27] Andrew Swan and Taichi Uemura. On Church's thesis in cubical assemblies. *CoRR*, abs/1905.03014, 2019.

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

[29] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013.

[30] Benno van den Berg and Ieke Moerdijk. W-types in sheaves. *CoRR*, abs/0810.2398, 2008.

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

[32] Matthew Z. Weaver and Daniel R. Licata. A constructive model of directed univalence in bicubical sets. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 915–928. ACM, 2020.

[33] Felix Wellen. *Formalizing Cartan Geometry in Modal Homotopy Type Theory*. PhD thesis, Karlsruher Institut für Technologie, Germany, 2017.