For the Metatheory of Type Theory, Internal Sconing Is Enough

Rafaël Bocquet
Eötvös Loránd University, Budapest, Hungary

Ambrus Kaposi
Eötvös Loránd University, Budapest, Hungary

Christian Sattler
Chalmers University of Technology, Gothenburg, Sweden

Abstract

Metatheorems about type theories are often proven by interpreting the syntax into models constructed using categorical gluing. We propose to use only sconing (gluing along a global section functor) instead of general gluing. The sconing is performed internally to a presheaf category, and we recover the original glued model by externalization.

Our method relies on constructions involving two notions of models: first-order models (with explicit contexts) and higher-order models (without explicit contexts). Sconing turns a displayed higher-order model into a displayed first-order model.

Using these, we derive specialized induction principles for the syntax of type theory. The input of such an induction principle is a boilerplate-free description of its motives and methods, not mentioning contexts. The output is a section with computation rules specified in the same internal language. We illustrate our framework by proofs of canonicity and normalization for type theory.

2012 ACM Subject Classification Theory of computation → Type theory

Keywords and phrases type theory, presheaves, canonicity, normalization, sconing, gluing

Digital Object Identifier 10.4230/LIPIcs.FSCD.2023.18

Related Version *Full Version including Appendices for Parametricity and Cubical Type Theory*: `https://arxiv.org/abs/2302.05190`

Funding *Ambrus Kaposi*: Supported by the Bolyai Fellowship of the Hungarian Academy of Sciences and by the "Application Domain Specific Highly Reliable IT Solutions" project of the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 funding scheme.

1 Introduction

The syntax of a type theory can be presented as the initial object in the category of models of a generalized algebraic theory (GAT), i.e. as a quotient inductive-inductive type (QIIT). Initiality provides an induction principle for the syntax, namely the dependent eliminator of the QIIT. Metatheoretic properties of the syntax, such as canonicity or normalization, can then be proven by carefully constructing models of the theory displayed over the syntax, or equivalently the motives and methods of the induction principle. However, the presentation of the syntax as a QIIT includes an explicit encoding of the substitution calculus of the theory; in particular every type or term former comes with a substitution rule. If all of the components of a complicated model are written explicitly, one has to prove that they all respect substitution. More importantly, when working exclusively at this level of generality, it is not easy to abstract the proof methods into reusable theorems.

An alternative to the (first-order) genereralized algebraic presentation is to present a type theory as a second-order or higher-order theory, for example by using a Logical Framework [20] or Uemura's representable map categories [36]. A higher-order presentation enables the use of higher-order abstract syntax (HOAS), in which binders are encoded by metatheoretic functions. In practice, this means that stability under substitution is implicit. Semantically, HOAS admits an interpretation in the internal language of presheaf categories [22]. Within the internal language of a presheaf category, all constructions are automatically stable under the morphisms of the base category.

In this work we propose an approach that combines the strengths of the first-order and higher-order presentations. We consider two notions of models of a type theory: first-order models correspond to the first-order presentation, while higher-order models correspond to the higher-order presentation. Typically, first-order models are categorical models of type theory (categories with families equipped with additional structure), while higher-order models are approximately universes closed under the type formers of the theory. Both of these notions make sense both externally and in the internal language of a presheaf category. We also have notions of displayed higher-order and first-order models, corresponding to the motives and methods of induction principles. We present a small number of constructions switching between external, internal, first-order and higher-order models. Any of these constructions is individually simple, but they can be composed to derive the induction principles we need to prove metatheoretic results. The constructions are listed below (FOM = first-order model, HOM = higher-order model). The restriction and externalization operations can also be applied to displayed first-order models and morphisms of first-order models.

Construction	Input	Output
Internalization	FOM $\mathcal{M}$	HOM $\mathbb{C}$ in $\mathbf{Psh}(\mathcal{M})$
Set-contextualization	HOM $\mathbb{M}$	FOM $\mathbf{Set}_{\mathbb{M}}$
Telescopic contextualization	HOM $\mathbb{M}$	FOM $\mathbf{Tele}_{\mathbb{M}}$
Restriction	Functor $F : \mathcal{C} \to \mathcal{D}$, FOM $\mathcal{M}$ in $\mathbf{Psh}(\mathcal{D})$	FOM $F^*(\mathcal{M})$ in $\mathbf{Psh}(\mathcal{C})$
Externalization	FOM $\mathcal{M}$ in $\mathbf{Psh}(\mathcal{D})$	External FOM $1^*_{\mathcal{D}}(\mathcal{M})$
Scone-contextualization	FOM $\mathcal{M}$, Displayed HOM $\mathbb{M}^\bullet$ over $\mathcal{M}$	Displayed FOM $\mathbf{Scone}_{\mathbb{M}^\bullet}$ over $\mathcal{M}$

The most notable operations are the contextualizations[1], which turn higher-order models into first-order models. The **Set**-contextualization generalizes the construction of a first-order model from a universe; its underlying category is always the category of sets. The telescopic contextualization is the contextual core of the **Set**-contextualization; it restricts the underlying category to a category of telescopes. The **Scone**-contextualization is a generalization of the **Set**-contextualization to displayed higher-order and first-order models, which correspond to the motives and methods of induction principles; its underlying category is always the Sierpinski cone, also called scone, of some first-order model $\mathcal{M}$.

Our main observation is that sconing (that is **Scone**-contextualization) internally to a presheaf category corresponds when viewed externally to a more complicated gluing construction. For example, the normalization model from [12] can be recovered as the externalization of the **Scone**-contextualization of a displayed higher-order model constructed in presheaves over the category of renamings.

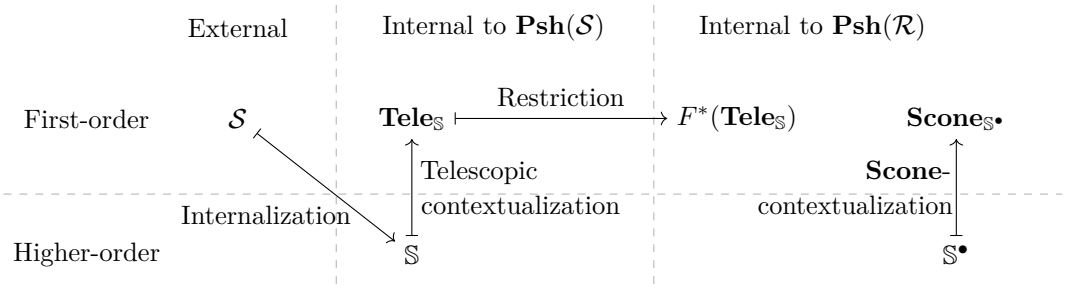
[1] The word contextualization reflects the fact that these operations make contexts explicit. It is not related to the notion of contextual model.

The main intended application of these constructions is the statement and proofs of relative induction principles. It is typically the case that the result of induction over the syntax of type theory only holds over some of the syntactic contexts, or is only stable under some of the syntactic substitutions. For example, canonicity only holds over the empty context. Normalization holds over every context, but is only stable under renamings. This situation is described by a functor into the syntax $F : \mathcal{R} \to \mathcal{S}$: the result of the induction should be stable under the morphisms of $\mathcal{R}$. These functors are called "figure shapes" by Sterling [32], they are also related to worlds in Twelf [29]. We prove the relative induction principles for the following functors. For cubical type theories, see Appendix C in the Full Version of the paper.

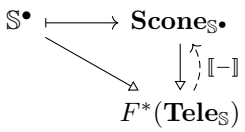
$1_{\mathbf{Cat}}$	$\to \mathcal{S}_{\mathcal{T}}$	Canonicity [12] (Theorem 14)
$\mathbf{Ren}_{\mathcal{S}}$	$\to \mathcal{S}_{\mathcal{T}}$	Normalization [4, 12] (Theorem 18)
$\square$	$\to \mathcal{S}_{\mathsf{CTT}}$	(Homotopy/strict) canonicity for cubical type theory [13]
$\mathcal{A}_{\square}$	$\to \mathcal{S}_{\mathsf{CTT}}$	Normalization for cubical type theory [34]

We also show how to prove canonicity and normalization by instantiating the relative induction principles for $1_{\mathbf{Cat}} \to \mathcal{S}_{\mathcal{T}}$ and $\mathbf{Ren}_{\mathcal{S}} \to \mathcal{S}_{\mathcal{T}}$. We don't prove canonicity nor normalization for cubical type theory, but we expect that the currently known proofs could be reformulated as instances of the relative induction principles for $\square \to \mathcal{S}_{\mathsf{CTT}}$ and $\mathcal{A}_{\square} \to \mathcal{S}_{\mathsf{CTT}}$.

Typically, the category $\mathcal{R}$ can be described as the initial object of some category of structured categories. A relative induction principle with respect to the functor F is an induction principle that combines the universal properties of $\mathcal{R}$ and $\mathcal{S}$. In our previous work [10], the relative induction principles were stated in terms of the rather ad-hoc notions of "displayed models without context extensions" and "relative sections". In the present work, the input of a relative induction principle is a displayed higher-order model $\mathbb{S}^\bullet$, and the result is just a (first-order) section $[\![-]\!]$ of $\mathbf{Scone}_{\mathbb{S}^\bullet}$. The previous notion of "displayed model without context extension" is recovered in the special case of displayed higher-order models over $F^*(\mathbf{Tele}_{\mathbb{S}})$. The following diagram illustrates the constructions involved in the statement and proof of a relative induction principle ($F : \mathcal{R} \to \mathcal{S}$).



Both $\mathbb{S}^\bullet$ and $\mathbf{Scone}_{\mathbb{S}^\bullet}$ are displayed over the internal first-order model $F^*(\mathbf{Tele}_{\mathbb{S}})$.



An important feature of our work is that the section $[\![-]\!]$ admits good computational behaviour, although we do not formally analyze this behaviour. In fact, part of our understanding comes from looking at the computational behaviour of the congruence operation ap in higher observational type theory [31, 6], which is also a morphism of (first-order) models internally to presheaves over the syntax of H.O.T.T.

Related work

Logical relations and categorical gluing. The initial motivation for this work was the understanding of algebraic and reduction-free normalization proofs for dependent type theories. Variants of categorical gluing were used to prove canonicity or normalization for simple types [2, 15, 35], System F [3], and dependent types [4, 12, 14]. These can be contrasted with reduction based normalization proofs such as [1, 30].

Logical relations were used to prove syntactic parametricity for type theory [9, 5] and definability for simply typed lambda calculus [25]. It was shown that categorical gluing generalizes both syntactic parametricity and canonicity proofs [27].

Logical frameworks and higher-order abstract syntax. Higher-order abstract syntax (HOAS) is the use of metatheoretic functions to specify syntactic binders. Hofmann [22] has explained how HOAS can be interpreted in the internal language of presheaf categories.

Uemura [36] has given a general definition of type theory based on these ideas, which we will call second-order generalized algebraic theories (SOGATs). It generalizes notions of second-order algebraic theories that have been studied by Fiore and Mahmoud [16]. Harper presents an equational variant of logical framework [19] for defining theories with bindings corresponding to SOGATs. We believe that the constructions in our paper generalize to any SOGAT. Gratzer and Sterling [18] propose using LCCCs to define higher order theories (without representability conditions) which correspond to our higher order models, but we also consider first order models.

Synthetic Tait computability. Synthetic Tait computability (STC, [32, 33]) is an approach that relies on the internal language of the Artin gluing of toposes, typically constructed by gluing syntactic and semantic toposes. A pair of open/closed modalities can be used to distinguish the syntactic and semantic parts in the internal language of the Artin gluing. STC has been applied to proofs of normalization for cubical type theory [34], multimodal type theory [17] and simplicial type theory [37].

Both STC and our approach provide a synthetic setting for proofs of metatheorems. Our approach is perhaps simpler in some aspects, e.g. we don't use any modalities and don't need to use realignment; we acknowledge that this is partly a matter of preference. The main advantage of our approach over STC is that we have an internal specification of the result of an induction principle.

In his thesis [32], Sterling briefly discusses the notion of Henkin model of a higher-order theory. A Henkin model is a higher-order model with a non-standard interpretation of the dependent products. We note that Henkin models are closely related to first-order models: the Henkin models of a second-order theory are equivalent to the democratic first-order models (this is [36, Theorem 7.30]).

Contributions

The main takeaway of this paper is that when using HOAS, we should not discard the first order presentation, even in an internal setting. In particular, the right notion of displayed higher-order model (input of an induction principle) lies over a first-order model. The output of an induction principle is also a section of first-order models, still internal.

The technical contributions are:

- the relative induction principles which combine the initiality of the underlying category and initiality of the syntax; we derive four such induction principles;
- the internalization, contextualization, externalization constructions which let us formulate the relative induction principles;
- the derivation of internal sections by analyzing the category of sections.

The main takeaway is supported by applications of the induction principles: boilerplate-free proofs of canonicity and normalization. The fact that the notion of section is specified internally has the feature that we can reuse it directly in subsequent inductions. We exploit this in the proof of uniqueness of normal forms which is proven in a separate step after normalization, relying on how the section computes normal forms.

Structure of the paper

In Section 2 we define first-order- and higher-order models of an example type theory, and constructions relating them. In Section 3, we define displayed higher order models which collect the motives and methods of an induction principle. We also define sconing which turns a displayed higher order model into its first order variant, also providing a notion of section. Then we move on to applications: we prove canonicity in Section 4 using an induction principle relative to the empty context (Theorem 14) which is a trivial consequence of our previous definitions. In Section 5, we prove normalization using an induction principle relative to renamings (Theorem 18). This induction principle is proved using the methods described in Section 6. Another application (syntactic parametricity) of Theorem 14 is described in the Full Version of the paper, Appendix B. In Appendix C of the Full Version, cubical variants of the above induction principles are also proven.

Background

We assume some familiarity with the categorical semantics of type theory [11, 5] and with the use of extensional type theory as the internal language of presheaf categories [21].

We use the notion of locally representable dependent presheaf to encode context extensions (see Definition 21). This definition is a more indexed formulation of the notion of representable natural transformation, which was used by Awodey to give an alternative definition of CwFs, known as natural models [8].

2 First-order and higher-order models

Our running example is a minimal dependent type theory $\mathcal{T}$ with only Π-types, but our constructions directly generalize to larger type theories. Some other type theories are considered in the Full Version of this paper, including dependent type theories with universes and cubical type theories. We leave generalization to arbitrary second-order generalized algebraic theories to future work.

We define notions of higher-order and first-order models for $\mathcal{T}$. A higher-order model is essentially a universe closed under dependent products, while a first-order model is a category with families (CwF) equipped with Π-types. The higher-order models are the models of some higher-order theory $\mathcal{T}^{\mathsf{ho}}$ (a theory whose operations can have a higher-order sort; classified by some locally cartesian closed category), whereas the first-order models are the models of some (first-order) essentially[2] algebraic theory $\mathcal{T}^{\mathsf{fo}}$ (a theory whose operations have a first-order sort; classified by some finitely complete category).

[2] The distinction between generalized algebraic theories and essentially algebraic theories is not relevant in this paper.

2.1 Definitions

▶ **Definition 1.** *A **higher-order model** of $\mathcal{T}$ consists of the following families and operations:*

$$\begin{aligned}
&\mathsf{Ty} &&: \mathrm{Set},\\
&\mathsf{Tm} &&: \mathsf{Ty} \to \mathrm{Set},\\
&\Pi &&: \forall(A : \mathsf{Ty})(B : \mathsf{Tm}(A) \to \mathsf{Ty}) \to \mathsf{Ty},\\
&\mathsf{app} &&: \forall A\ B\ (f : \mathsf{Tm}(\Pi(A, B)))\ (a : \mathsf{Tm}(A)) \to \mathsf{Tm}(B(a)),\\
&\mathsf{lam} &&: \forall A\ B\ (b : (a : \mathsf{Tm}(A)) \to \mathsf{Tm}(B(a))) \to \mathsf{Tm}(\Pi(A, B)),
\end{aligned}$$

subject to equations corresponding to the β- and η-rules:

$$\mathsf{lam}(\lambda a \mapsto \mathsf{app}(f, a)) = f, \qquad \mathsf{app}(\mathsf{lam}(b), a) = b(a). \qquad \lrcorner$$

▶ **Definition 2.** *A **first-order model** of $\mathcal{T}$ is a CwF $\mathcal{C}$ equipped with Π-types.* $\lrcorner$

The notion of first-order model can be presented by a first-order generalized algebraic theory $\mathcal{T}^{\mathsf{fo}}$. Equivalently, a first-order model is a category $\mathcal{C}$ with a terminal object $1_{\mathcal{C}}$, along with a (global) higher-order model $\mathbb{C}$ of $\mathcal{T}$ in $\mathbf{Psh}(\mathcal{C})$ such that the dependent presheaf $\mathbb{C}.\mathsf{Tm}$ is locally representable. The higher-order model $\mathbb{C}$ is called the internalization of $\mathcal{C}$.

We use blackboard bold letters $\mathbb{M}$, $\mathbb{N}$, $\mathbb{C}$, $\mathbb{S}$, $\mathbb{R}$, etc. to refer to higher-order models, and calligraphic letters $\mathcal{M}$, $\mathcal{N}$, $\mathcal{C}$, $\mathcal{S}$, $\mathcal{R}$, etc. to refer to first-order models. We try to use the corresponding letter for the underlying internal higher-order model of a first-order model, e.g. if $\mathcal{C}$ is an external first-order model, we use $\mathbb{C}$ for its underlying internal higher-order model in $\mathbf{Psh}(\mathcal{C})$. We denote the components of a model by $\mathbb{C}.\mathsf{Ty}$, $\mathcal{C}.\mathsf{Tm}$, $\mathcal{C}.\Pi$, etc.

We write $\mathbf{Mod}_{\mathcal{T}}$ for the 1-category of first-order models of $\mathcal{T}$, and $\mathcal{S}_{\mathcal{T}}$ or just $\mathcal{S}$ for its initial object (the letter "S" standing for both syntax and substitutions).

2.2 Contextualization

In general, we almost never want to construct all of the components of a first-order model explicitly, because checking functoriality and naturality conditions without relying on the internal language of a presheaf model is tedious. For some models however, the functoriality and naturality conditions hold trivially. This is the case for the "standard model" of type theory over the category of sets: when defining this standard model in intensional type theory, all naturality and functoriality conditions hold definitionally.

The construction of the standard model generalizes to the construction of a first-order model from a higher-order model, which we now describe.

▶ **Construction 3** (**Set**-Contextualization)**.** *Let $\mathbb{M}$ be a higher-order model of $\mathcal{T}$. We construct a first-order model $\mathbf{Set}_{\mathbb{M}}$, called the **Set-contextualization** of $\mathbb{M}$. Its underlying category is the category* **Set** *of sets.*

The types and terms of $\mathbf{Set}_{\mathbb{M}}$ are indexed families of types and terms of $\mathbb{M}$:

- *A type over $\Gamma \in \mathbf{Set}$ is a function $\Gamma \to \mathbb{M}.\mathsf{Ty}$.*
- *Type substitution along a function $f : \Delta \to \Gamma$ is precomposition with f.*
- *A term of type $A : \Gamma \to \mathbb{M}.\mathsf{Ty}$ is a dependent function $(\gamma : \Gamma) \to \mathbb{M}.\mathsf{Tm}(A(\gamma))$.*
- *Term substitution along a function $f : \Delta \to \Gamma$ is precomposition with f.*
- *The functoriality of substitution is associativity of function composition.*

The context extensions are given by dependent sums in **Set***:*

$$(\Gamma.A) \triangleq (\gamma : \Gamma) \times \mathbb{M}.\mathsf{Tm}(A(\gamma)).$$

The type-theoretic operations are all defined pointwise:

$$\begin{aligned}
\mathbf{Set}_{\mathbb{M}}.\Pi(\Gamma, A, B) &\triangleq \lambda\gamma \mapsto \mathbb{M}.\Pi(A(\gamma), \lambda a \mapsto B(\gamma, a)),\\
\mathbf{Set}_{\mathbb{M}}.\mathsf{app}(\Gamma, f, a) &\triangleq \lambda\gamma \mapsto \mathbb{M}.\mathsf{app}(f(\gamma), a(\gamma)),\\
\mathbf{Set}_{\mathbb{M}}.\mathsf{lam}(\Gamma, b) &\triangleq \lambda\gamma \mapsto \mathbb{M}.\mathsf{lam}(\lambda a \mapsto b(\gamma, a)).
\end{aligned}$$

The β- and η-rules for $\mathbf{Set}_{\mathbb{M}}$ *hold as a consequence of the corresponding rules for* $\mathbb{M}$*.*

The naturality conditions are all trivial. For example, in the case of the Π *type former, we have to check* $\Pi(\Gamma, A, B)[f] = \Pi(\Delta, A[f], B[f^+])$ *for any* $f : \Delta \to \Gamma$*, where* $f^+(\gamma, a) = (f(\gamma), a)$*. This amounts to checking the equality*

$$(\lambda\gamma \mapsto \mathbb{M}.\Pi(A(\gamma), \lambda a \mapsto B(\gamma, a))) \circ f = (\lambda\gamma \mapsto \mathbb{M}.\Pi((A \circ f)(\gamma), \lambda a \mapsto (B \circ f^+)(\gamma, a))).$$ ⌟

▶ Remark 4. Note that the underlying category **Set** of $\mathbf{Set}_{\mathbb{M}}$ now has two CwF structures:

- An *inner* CwF structure, as defined in Construction 3.
- An *outer* CwF structure, corresponding to the usual CwF structure on the category of sets, modeling extensional type theory.

Together, they form a model of two-level type theory [7].

2.3 Telescopic contextualization

A first-order order model is said to be contextual when every object can be uniquely written as an iterated context extension starting from the empty context. The contextual first-order models form a coreflective subcategory $\mathbf{Mod}_{\mathcal{T}}^{\mathsf{cxl}}$ of $\mathbf{Mod}_{\mathcal{T}}$: the inclusion $\mathbf{Mod}_{\mathcal{T}}^{\mathsf{cxl}} \to \mathbf{Mod}_{\mathcal{T}}$ has a right adjoint cxl: the *contextual core* $\mathsf{cxl}(\mathcal{C})$ has as objects the iterated context extensions of $\mathcal{C}$, also known as telescopes, over the empty context.

▶ **Definition 5.** *Let* $\mathbb{M}$ *be a higher-order model of* $\mathcal{T}$*. The* ***telescopic contextualization*** $\mathbf{Tele}_{\mathbb{M}}$ *is the contextual core of the* **Set***-contextualization* $\mathbf{Set}_{\mathbb{M}}$*.* ⌟

By general properties of the contextual core, there is a model morphism $\lfloor - \rfloor : \mathbf{Tele}_{\mathbb{M}} \to \mathbf{Set}_{\mathbb{M}}$ that is bijective on types and terms. (There is a cofibrantly generated factorization system on first-order models with such morphisms as its right class. The contextual models are precisely those in the left class.)

When working internally to some presheaf category $\mathbf{Psh}(\mathcal{C})$, another related construction involves the internal subcategory spanned by $ﾄ : \mathsf{Ob}_{\mathcal{C}} \to \mathrm{Set}$, where $\mathsf{Ob}_{\mathcal{C}}$ is the discrete presheaf on the set of objects of $\mathcal{C}$, and $ﾄ$ internalizes the Yoneda embedding. This "Yoneda universe" has been used by Hu et al. [23] to give semantics to contextual types.

2.4 Internal first-order models

Since the notion of first-order model is described by an essentially algebraic theory, it can be interpreted in any finitely complete category. In particular, there is a notion of internal first-order model in any category $\widehat{\mathcal{C}}$ of small presheaves, obtained by letting Set stand for the Hofmann-Streicher universe of the presheaf topos $\mathbf{Psh}(\mathcal{C})$ in the definition of model.

▶ **Proposition 6.** *The following three notions are equivalent:*

1. *First-order models of* $\mathcal{T}$ *in* $\mathbf{Psh}(\mathcal{C})$*;*
2. *Finite limit preserving functors* $\mathcal{T}^{\mathsf{fo}} \to \widehat{\mathcal{C}}$*, where* $\mathcal{T}^{\mathsf{fo}}$ *is the finitely complete category classifying the first-order models of* $\mathcal{T}$*;*
3. *Functors* $\mathcal{C} \to \mathbf{Mod}^{\mathsf{op}}_{\mathcal{T}}$*.*

Proof. This is well-known [24, D1.2.14]. The equivalence between (1) and (2) is the fact that $\mathcal{T}^{\mathsf{fo}}$ classifies the first-order models of $\mathcal{T}$. The equivalence between (2) and (3) follows from the fact that finite limits in $\widehat{\mathcal{C}}$ are computed pointwise. ◀

2.5 Restriction and externalization

Another important operation on first-order models is the restriction of a first-order model $\mathcal{M}$ internal to $\mathbf{Psh}(\mathcal{D})$ along a functor $F : \mathcal{C} \to \mathcal{D}$. This restricted model $F^*(\mathcal{M})$ is a first-order model internal to $\mathbf{Psh}(\mathcal{C})$. If $\mathcal{M}$ is seen as a functor $\mathcal{D} \to \mathbf{Mod}^{\mathsf{op}}_{\mathcal{T}}$, then the restriction $F^*(\mathcal{M}) : \mathcal{C} \to \mathbf{Mod}^{\mathsf{op}}_{\mathcal{T}}$ is simply the precomposition $(\mathcal{M} \circ F)$. If $\mathcal{M}$ is seen instead as a finite-limits preserving functor $\mathcal{T}^{\mathsf{fo}} \to \mathbf{Psh}(\mathcal{D})$, then $F^*(\mathcal{M})$ is postcomposition with the inverse image functor $F^* : \mathbf{Psh}(\mathcal{D}) \to \mathbf{Psh}(\mathcal{C})$. These two definitions coincide up to the equivalence of Proposition 6; which is thus natural in the base category.

▶ Remark 7. A more explicit computation of the restriction can be given in the internal language of $\mathbf{Psh}(\mathcal{C})$ using the dependent right adjoint associated to the adjunction $(F_! \dashv F^*)$. When $\mathcal{M}$ is the **Set**- or telescopic contextualization of a higher-order model, then the dependent right adjoint allows for the use of HOAS when working with $F^*(\mathcal{M})$. ⌟

A special case of the restriction is the *externalization* of an internal first-order model.

▶ **Definition 8.** *Let* $\mathcal{C}$ *be any category with a terminal object* $1_{\mathcal{C}}$*, and consider the functor* $1_{\mathcal{C}} : 1_{\mathbf{Cat}} \to \mathcal{C}$ *that selects this terminal object. For any internal first-order model* $\mathcal{M}$ *in* $\mathbf{Psh}(\mathcal{C})$*, we have an external first-order model* $1^*_{\mathcal{C}}(\mathcal{M})$*, called the* ***externalization*** *of* $\mathcal{M}$*.* ⌟

Given any higher-order model $\mathbb{M}$ in $\mathbf{Psh}(\mathcal{C})$, we can construct the externalization $1^*_{\mathcal{C}}(\mathbf{Tele}_{\mathbb{M}})$ of its telescopic first-order model. Up to isomorphism, all external contextual first-order models arise as the externalization of a telescopic contextualization.

▶ **Lemma 9.** *Let* $\mathcal{C}$ *be an external first-order model, with* $\mathbb{C}$ *its underlying internal higher-order model. Then* $1^*_{\mathcal{C}}(\mathsf{Tele}_{\mathbb{C}})$ *is the contextual core of* $\mathcal{C}$*.*

Proof. See Corollary 24. ◀

In particular, since the initial model $\mathcal{S}$ is contextual, the externalization $1^*_{\mathcal{S}}(\mathsf{Tele}_{\mathbb{S}})$ of its telescopic contextualization is isomorphic to $\mathcal{S}$.

We can also construct the externalization $1^*_{\mathcal{C}}(\mathbf{Set}_{\mathbb{C}})$ of the internal **Set**-contextualization of an higher-order model $\mathbb{C}$. The underlying category of $1^*_{\mathcal{C}}(\mathbf{Set}_{\mathbb{C}})$ is the category of presheaves over $\mathcal{C}$ (restricted to some universe level); and $1^*_{\mathcal{C}}(\mathbf{Set}_{\mathbb{C}})$ is an external model of two-level type theory (its underlying category is the restriction of $\mathbf{Psh}(\mathcal{C})$ to some universe level). Recall that there is, internally to $\mathbf{Psh}(\mathcal{C})$, a morphism $\lfloor - \rfloor : \mathbf{Tele}_{\mathbb{C}} \to \mathbf{Set}_{\mathbb{C}}$ of first-order models. This morphism can also be externalized, giving an external morphism $1^*_{\mathcal{C}}(\lfloor - \rfloor) : 1^*_{\mathcal{C}}(\mathsf{Tele}_{\mathbb{C}}) \to 1^*_{\mathcal{C}}(\mathbf{Set}_{\mathbb{C}})$ of first-order models. When $1^*_{\mathcal{C}}(\mathsf{Tele}_{\mathbb{C}}) \cong \mathcal{C}$, this is a simple construction of the embedding of $\mathcal{C}$ into the presheaf model of two-level type theory.

3 Displayed higher-order models

3.1 Motives and methods

We now define the notion of displayed higher-order model, which collects the motives and methods of induction principles. One could expect that a displayed higher-order model would be displayed over a base higher-order model. We instead define the notion of displayed higher-order model over a base first-order model; it is always possible to turn higher-order models into first-order models using a contextualization, but not every first-order model arises in this way.

▶ **Definition 10.** *Let $\mathcal{M}$ be a first-order model of $\mathcal{T}$. A* ***displayed higher-order model*** *$\mathbb{M}^\bullet$ over $\mathcal{M}$ consists of the following data:*

$$
\begin{aligned}
&\mathsf{Ty}^\bullet \;: \mathcal{M}.\mathsf{Ty}(1_\mathcal{M}) \to \mathrm{Set},\\
&\mathsf{Tm}^\bullet : \forall(A : \mathcal{M}.\mathsf{Ty}(1_\mathcal{M}))\ (A^\bullet : \mathsf{Ty}^\bullet(A)) \to \mathcal{M}.\mathsf{Tm}(1_\mathcal{M}, A) \to \mathrm{Set},\\
&\Pi^\bullet \;: \forall(A : \mathcal{M}.\mathsf{Ty}(1_\mathcal{M}))\ (A^\bullet : \mathsf{Ty}^\bullet(A))\\
&\qquad (B : \mathcal{M}.\mathsf{Ty}(1_\mathcal{M}.A))\ (B^\bullet : \forall(a : \mathcal{M}.\mathsf{Tm}(1_\mathcal{M}, A))(a^\bullet : \mathsf{Tm}^\bullet(A^\bullet, a)) \to \mathsf{Ty}^\bullet(B[a]))\\
&\qquad \to \mathsf{Ty}^\bullet(\Pi(A, B)),\\
&\mathsf{app}^\bullet : \forall A\ A^\bullet\ B\ B^\bullet\ (f : \mathcal{M}.\mathsf{Tm}(1_\mathcal{M}, \Pi(A, B)))\ (f^\bullet : \mathsf{Tm}^\bullet(\Pi^\bullet(A^\bullet, B^\bullet), f))\\
&\qquad (a : \mathcal{M}.\mathsf{Tm}(1_\mathcal{M}, A))\ (a^\bullet : \mathsf{Tm}^\bullet(A^\bullet, a))\\
&\qquad \to \mathsf{Tm}^\bullet(B^\bullet(a^\bullet), \mathsf{app}(f, a)),\\
&\mathsf{lam}^\bullet : \forall A\ A^\bullet\ B\ B^\bullet\ (b : \mathcal{M}.\mathsf{Tm}(1_\mathcal{M}.(a : A), B[a]))\\
&\qquad (b^\bullet : \forall(a : \mathcal{M}.\mathsf{Tm}(1_\mathcal{M}, A))(a^\bullet : \mathsf{Tm}^\bullet(A^\bullet, a)) \to \mathsf{Tm}^\bullet(B^\bullet(a^\bullet), b[a]))\\
&\qquad \to \mathsf{Tm}^\bullet(\Pi^\bullet(A^\bullet, B^\bullet), \mathsf{lam}(b)),
\end{aligned}
$$

such that the following equalities hold:

$$\mathsf{app}^\bullet(\mathsf{lam}^\bullet(b^\bullet), a^\bullet) = b^\bullet(a^\bullet), \qquad \mathsf{lam}^\bullet(\lambda a^\bullet \mapsto \mathsf{app}^\bullet(f^\bullet, a^\bullet)) = f^\bullet.$$ ⌟

Most of the components of a displayed higher-order model only depend on the closed types and terms of $\mathcal{M}$; only the binders need to refer to open types and terms.

Note that the data of a displayed higher-order model over the terminal first-order model is equivalent to the data of a non-displayed higher-order model.

3.2 Displayed contextualization

Given any displayed higher-order model $\mathbb{M}^\bullet$ over $\mathcal{M}$, we construct a displayed first-order model over $\mathcal{M}$. This construction is a displayed generalization of the **Set**-contextualization.

The underlying displayed category of this construction is the *Sierpinski cone*, or scone, of $\mathcal{M}$. The scone of a category $\mathcal{C}$ with a terminal object is the comma category $(\mathbf{Set} \downarrow \Gamma_\mathcal{C})$, where $\Gamma_\mathcal{C} : \mathcal{C} \to \mathbf{Set}$ is the global section functor $\Gamma_\mathcal{C} = \mathcal{C}(1_\mathcal{C}, -)$.

▶ **Construction 11** (Displayed contextualization)**.** *Fix a displayed higher-order model $\mathbb{M}^\bullet$ over a first-order model $\mathcal{M}$. We construct a displayed first-order model $\mathbf{Scone}_{\mathbb{M}^\bullet}$ over $\mathcal{M}$, called the* ***displayed contextualization*** *of $\mathbb{M}^\bullet$.*

- *An object of $\mathbf{Scone}_{\mathbb{M}^\bullet}$ over $\Gamma \in \mathcal{M}$ is a family*

  $$\Gamma^\dagger : \mathcal{M}(1_\mathcal{M}, \Gamma) \to \mathrm{Set}$$

  over the global elements (i.e. closing substitutions) of Γ.

- *A morphism of* $\mathbf{Scone}_{\mathbb{M}^\bullet}$ *from* $\Gamma^\dagger$ *to* $\Delta^\dagger$ *over a base morphism* $f : \mathcal{M}(\Gamma, \Delta)$ *is a family*

  $$f^\dagger : \forall(\gamma : \mathcal{M}(1_\mathcal{M}, \Gamma)) \to \Gamma^\dagger(\gamma) \to \Delta^\dagger(f \circ \gamma).$$

  The identity displayed morphism is given by

  $$\mathsf{id}^\dagger \triangleq \lambda\gamma\ \gamma^\bullet \mapsto \gamma^\bullet,$$

  whereas the composition of two displayed morphisms $f^\dagger$ *and* $g^\dagger$ *is*

  $$f^\dagger \circ^\dagger g^\dagger \triangleq \lambda\gamma\ \gamma^\bullet \mapsto f^\dagger(g^\dagger(\gamma^\bullet)).$$

- *A type of* $\mathbf{Scone}_{\mathbb{M}^\bullet}$ *over an object* $\Gamma^\dagger$ *and a type* $A : \mathcal{M}.\mathsf{Ty}(\Gamma)$ *is a function*

  $$A^\dagger : \forall\gamma\ (\gamma^\dagger : \Gamma^\dagger(\gamma)) \to \mathsf{Ty}^\bullet(A[\gamma]).$$

  The restriction of $A^\dagger$ *along a displayed morphism* $f^\dagger$ *is*

  $$A^\dagger[f^\dagger] \triangleq \lambda\gamma^\bullet \mapsto A^\dagger(f^\dagger(\gamma^\bullet)).$$

- *A term of* $\mathbf{Scone}_{\mathbb{M}^\bullet}$ *of type* $A^\dagger$ *over an object* $\Gamma^\dagger$ *and a term* $a : \mathcal{M}.\mathsf{Tm}(\Gamma, A)$ *is a function*

  $$a^\dagger : \forall\gamma\ (\gamma^\bullet : \Gamma^\dagger(\gamma)) \to \mathsf{Tm}^\bullet(A^\dagger(\gamma^\bullet), a[\gamma]).$$

  The restriction of $a^\dagger$ *along a displayed morphism* $f^\dagger$ *is*

  $$a^\dagger[f^\dagger] \triangleq \lambda\gamma^\bullet \mapsto a^\dagger(f^\dagger(\gamma^\bullet)).$$

- *The empty displayed context is the family*

  $$1^\dagger \triangleq \lambda\_ \mapsto \mathbf{1}.$$

- *The extension of a displayed context* $\Gamma^\dagger$ *by a displayed type* $A^\dagger$ *is the family*

  $$(\Gamma^\dagger.A^\dagger) \triangleq \lambda\langle\gamma, a\rangle \mapsto (\gamma^\bullet : \Gamma^\dagger(\gamma)) \times \mathsf{Tm}^\bullet(A^\dagger(\gamma^\bullet), a).$$

- *All type- and term- formers are defined pointwise using the corresponding component of* $\mathbb{M}^\bullet$*:*

  $$\begin{aligned}
  \Pi^\dagger(A^\dagger, B^\dagger) &\triangleq \lambda\gamma^\bullet \mapsto \Pi^\bullet(A^\dagger(\gamma^\bullet), \lambda a^\bullet \mapsto B^\dagger(\gamma^\bullet, a^\bullet)),\\
  \mathsf{app}^\dagger(f^\dagger, a^\dagger) &\triangleq \lambda\gamma^\bullet \mapsto \mathsf{app}^\bullet(f^\dagger(\gamma^\bullet), a^\dagger(\gamma^\bullet)),\\
  \mathsf{lam}^\dagger(b^\dagger) &\triangleq \lambda\gamma^\bullet \mapsto \mathsf{lam}^\bullet(\lambda a^\bullet \mapsto b^\dagger(\gamma^\bullet, a^\bullet)).
  \end{aligned}$$

- *The* β*- and* η*-rules hold as a consequence of the* β*- and* η*-rules of* $\mathbb{M}^\bullet$*.*
- *All naturality conditions are trivial.* ⌟

Note that when $\mathbb{M}$ is a higher-order model seen as a displayed higher-order model over the terminal first-order model, then $\mathbf{Scone}_{\mathbb{M}}$ is equivalent to $\mathbf{Set}_{\mathbb{M}}$.

3.3 Sections of a displayed higher-order model

The notion of displayed higher-order model corresponds to the motives and methods of an induction principle. We now define the notion of section of a displayed higher-order model, corresponding to the result of applying an induction principle: it is simply defined as a section of the displayed contextualization.

▶ **Definition 12.** *A **section** of a displayed higher-order model $\mathbb{M}^\bullet$ is a section $[\![-]\!]$ of its displayed contextualization $\mathbf{Scone}_{\mathbb{M}^\bullet}$ (in $\mathbf{Mod}_{\mathcal{T}}$).* ⌟

The definition of section of a displayed higher-order model $\mathbb{M}^\bullet$ over $\mathcal{M}$ can be unfolded to the following components:

- For every object $\Gamma : \mathcal{M}$, a family

$$[\![\Gamma]\!] : \mathcal{M}(1_{\mathcal{M}}, \Gamma) \to \mathbf{Set}$$

of *environments*.

- For every morphism $f : \mathcal{M}(\Gamma, \Delta)$, a family

$$[\![f]\!] : \forall\gamma \to [\![\Gamma]\!](\gamma) \to [\![\Delta]\!](f \circ \gamma)$$

of maps between environments.

- For every type $A : \mathcal{M}.\mathsf{Ty}(\Gamma)$, a family

$$[\![A]\!] : \forall\gamma\ (\gamma^\bullet : [\![\Gamma]\!](\gamma)) \to \mathsf{Ty}^\bullet(A[\gamma])$$

of displayed types over closures of A.

- For every term $a : \mathcal{M}.\mathsf{Tm}(\Gamma, A)$, a family

$$[\![a]\!] : \forall\gamma\ (\gamma^\bullet : [\![\Gamma]\!](\gamma)) \to \mathsf{Tm}^\bullet([\![A]\!](\gamma^\bullet), a[\gamma])$$

of displayed terms over closures of a.

- Subject to functoriality and naturality equations:

$$[\![\mathsf{id}]\!](\gamma^\bullet) = \gamma^\bullet,$$
$$[\![f \circ g]\!](\gamma^\bullet) = [\![f]\!]([\![g]\!](\gamma^\bullet)),$$
$$[\![A[f]]\!](\gamma^\bullet) = [\![A]\!]([\![f]\!](\gamma^\bullet)),$$
$$[\![a[f]]\!](\gamma^\bullet) = [\![A]\!]([\![f]\!](\gamma^\bullet)).$$

- Such that context extensions are preserved:

$$[\![1_{\mathcal{M}}]\!](\star) = \{\star\},$$
$$[\![\Gamma.A]\!](\gamma, a) = (\gamma^\bullet : [\![\Gamma]\!](\gamma)) \times (a^\bullet : \mathsf{Tm}^\bullet([\![A]\!](\gamma^\bullet), a)),$$
$$[\![\lambda\gamma \mapsto (\delta(\gamma), a(\gamma))]\!](\gamma) = ([\![\delta]\!](\gamma), [\![a]\!](\gamma)).$$

- With computation rules for every type and term former:

$$[\![\lambda\gamma \mapsto \Pi(A(\gamma), \lambda a \mapsto B(\gamma, a))]\!](\gamma^\bullet) = \Pi^\bullet([\![A]\!](\gamma^\bullet), \lambda a^\bullet \mapsto [\![B]\!](\gamma^\bullet)),$$
$$[\![\lambda\gamma \mapsto \mathsf{app}(f(\gamma), a(\gamma))]\!](\gamma^\bullet) = \mathsf{app}^\bullet([\![f]\!](\gamma^\bullet), [\![a]\!](\gamma^\bullet)),$$
$$[\![\lambda\gamma \mapsto \mathsf{lam}(b(\gamma))]\!](\gamma^\bullet) = \mathsf{lam}^\bullet(\lambda a^\bullet \mapsto [\![b]\!](\gamma^\bullet, a^\bullet)).$$

When x is a closed type or term of $\mathcal{M}$, we write $[\![x]\!]$ for the interpretation $[\![x]\!](\star)$ of x in the empty environment. We may use underlined names to distinguish the variable of open terms. For instance, we may write $[\![\mathsf{app}(\underline{f}, \underline{a})]\!][\underline{f} \mapsto f', \underline{a} \mapsto a']$ instead of $[\![\lambda(f, a) \mapsto \mathsf{app}(f, a)]\!](f', a')$.

▶ Remark 13. Let $\mathbb{S}^\bullet$ be a displayed higher-order model over the first-order model $F^*(\mathsf{Tele}_\mathbb{S})$ in $\mathbf{Psh}(\mathcal{C})$, for some functor $F : \mathcal{C} \to \mathcal{S}$. The displayed contextualization $\mathbf{Scone}_{\mathbb{S}^\bullet}$ is a displayed first-order model over $F^*(\mathsf{Tele}_\mathbb{S})$. Then its externalization $1^*_\mathcal{C}(\mathbf{Scone}_{\mathbb{S}^\bullet})$ is an external displayed first-order model lying over $1^*_\mathcal{C}(F^*(\mathsf{Tele}_\mathbb{S})) = 1^*_\mathcal{S}(\mathsf{Tele}_\mathbb{S})$. Up to the isomorphism $1^*_\mathcal{S}(\mathsf{Tele}_\mathbb{S}) \cong \mathcal{S}$, the externalized **Scone**-contextualization $1^*_\mathcal{C}(\mathbf{Scone}_{\mathbb{S}^\bullet})$ coincides with gluing. Its underlying category is the comma category $(\mathcal{S} \downarrow N_F)$, where $N_F : \mathcal{S} \to \mathbf{Psh}(\mathcal{C})$ is the nerve functor $\mathcal{S} \xrightarrow{\text{よ}} \mathbf{Psh}(\mathcal{S}) \xrightarrow{F^*} \mathbf{Psh}(\mathcal{C})$. ⌟

4 Example: canonicity proof

As a first example of a relative induction principle and its application, we prove canonicity for $\mathcal{T}$ extended with booleans (given by a type former Bool with constructors true and false and a dependent eliminator $\mathsf{elim}_\mathsf{Bool}$ with two computation rules).

We use the induction principle relative to the functor $1_\mathcal{S} : 1_\mathbf{Cat} \to \mathcal{S}$ that selects the terminal object in the syntax $\mathcal{S}$. It turns out that proving this specific relative induction principle is trivial.

▶ **Theorem 14** (Induction principle for $\mathbb{S}$ relative to $1_\mathcal{S} : 1_\mathbf{Cat} \to \mathcal{S}$)**.**
Let $\mathbb{S}^\bullet$ be a displayed higher-order model over the initial model $\mathcal{S}$, or equivalently over $1^*_\mathcal{S}(\mathsf{Tele}_\mathbb{S})$. Then $\mathbf{Scone}_{\mathbb{S}^\bullet}$ admits a section $[\![-]\!]$ over $\mathcal{S}$.

Proof. By initiality of $\mathcal{S}$. ◀

We now construct the displayed higher-order model $\mathbb{S}^\bullet$ over $1^*_\mathcal{S}(\mathsf{Tele}_\mathbb{S})$ that will be used to prove canonicity. A displayed type $A^\bullet$ over a closed type $A : \mathcal{S}.\mathsf{Ty}(1_\mathcal{S})$ is a Set-valued logical predicate over the closed terms of type A:

$$\mathsf{Ty}^\bullet(A) \triangleq \mathcal{S}.\mathsf{Tm}(1_\mathcal{S}, A) \to \mathrm{Set}.$$

A displayed term $a^\bullet$ of type $A^\bullet$ over a closed term $a : \mathcal{S}.\mathsf{Tm}(1_\mathcal{S}, A)$ is an element of the logical predicate $A^\bullet$ evaluated at a:

$$\mathsf{Tm}^\bullet(A^\bullet, a) \triangleq A^\bullet(a).$$

Given logical predicates $A^\bullet$ and $B^\bullet$, the logical predicate $\Pi^\bullet(A^\bullet, B^\bullet)$ expresses the fact that functions $f : \mathcal{S}.\mathsf{Tm}(1_\mathcal{S}, \Pi(A, B))$ should preserve the logical predicates.

$$\begin{aligned}
\Pi^\bullet(A^\bullet, B^\bullet) &\triangleq \lambda(f : \mathcal{S}.\mathsf{Tm}(1_\mathcal{S}, \Pi(A, B))) \mapsto (\forall a\ a^\bullet \to B^\bullet(a^\bullet, \mathsf{app}(f, a))),\\
\mathsf{app}^\bullet(f^\bullet, a^\bullet) &\triangleq f^\bullet(a^\bullet),\\
\mathsf{lam}^\bullet(b^\bullet) &\triangleq \lambda a^\bullet \mapsto b^\bullet(a^\bullet).
\end{aligned}$$

It is easy to check that the displayed β- and η-rules hold. The logical predicate $\mathsf{Bool}^\bullet : \mathcal{S}.\mathsf{Tm}(1_\mathcal{S}, \mathsf{Bool}) \to \mathrm{Set}$ is defined as an inductive family with two constructors $\mathsf{true}^\bullet : \mathsf{Bool}^\bullet(\mathsf{true})$ and $\mathsf{false}^\bullet : \mathsf{Bool}^\bullet(\mathsf{false})$. The displayed eliminator $\mathsf{elim}^\bullet_\mathsf{Bool}$ is defined using the elimination principle of $\mathsf{Bool}^\bullet$ and the displayed β-laws hold. This concludes the definition of all components of $\mathbb{S}^\bullet$.

▶ **Theorem 15.** *The initial model $\mathbb{S}$ satisfies canonicity: any closed boolean term $b : \mathcal{S}.\mathsf{Tm}(1_\mathcal{S}, \mathsf{Bool})$ is canonical, i.e. equal to exactly one of true or false.*

Proof. By the relative induction principle Theorem 14, the displayed higher-order model $\mathbb{S}^\bullet$ admits a section $[\![-]\!]$. Now given a closed boolean term $b : \mathcal{S}.\mathsf{Tm}(1_{\mathcal{S}}, \mathsf{Bool})$, we have $[\![b]\!] : \mathsf{Tm}^\bullet([\![\mathsf{Bool}]\!], b)$. By the computation rule of the section for Bool, $\mathsf{Tm}^\bullet([\![\mathsf{Bool}]\!], b) = \mathsf{Bool}^\bullet(b)$. Thus, $[\![b]\!] : \mathsf{Bool}^\bullet(b)$ witnesses the fact that b is canonical.

Since $[\![\mathsf{true}]\!] = \mathsf{true}^\bullet$, $[\![\mathsf{false}]\!] = \mathsf{false}^\bullet$ and $\mathsf{true}^\bullet \neq \mathsf{false}^\bullet$, we know that $\mathsf{true} \neq \mathsf{false}$. ◀

Note that in the canonicity proof, we have not needed to evaluate the section $[\![-]\!]$ on non-closed types or terms. Evaluating the section on open types and terms is usually only needed when encountering binders: the evaluation of the section on a closed binder depends on the evaluation of the section on an open type or term. The following is an example of the computation of the evaluation of the section $[\![-]\!]$ on the application of the boolean negation function $\mathsf{lam}(\lambda b \mapsto \mathsf{elim}_{\mathsf{Bool}}(\mathsf{Bool}, \mathsf{false}, \mathsf{true}, b))$ to true.

$$
\begin{aligned}
&[\![\mathsf{app}(\mathsf{lam}(\lambda b \mapsto \mathsf{elim}_{\mathsf{Bool}}(\mathsf{Bool}, \mathsf{false}, \mathsf{true}, b)), \mathsf{true})]\!] \\
&\quad = [\![\mathsf{lam}(\lambda b \mapsto \mathsf{elim}_{\mathsf{Bool}}(\mathsf{Bool}, \mathsf{false}, \mathsf{true}, b))]\!]([\![\mathsf{true}]\!]) \\
&\quad = (\lambda b^\bullet \mapsto [\![\mathsf{elim}_{\mathsf{Bool}}(\mathsf{Bool}, \mathsf{false}, \mathsf{true}, \underline{b})]\!][\underline{b} \mapsto b^\bullet])(\mathsf{true}^\bullet) \\
&\quad = [\![\mathsf{elim}_{\mathsf{Bool}}(\mathsf{Bool}, \mathsf{false}, \mathsf{true}, \underline{b})]\!][\underline{b} \mapsto \mathsf{true}^\bullet] \\
&\quad = \mathsf{elim}^\bullet_{\mathsf{Bool}}(\mathsf{Bool}^\bullet, \mathsf{false}^\bullet, \mathsf{true}^\bullet, \mathsf{true}^\bullet) \\
&\quad = \mathsf{false}^\bullet.
\end{aligned}
$$

5 Example: normalization proof

In this section we prove normalization for the initial model $\mathcal{S}$ of $\mathcal{T}$ using an induction principle relative to $F : \mathbf{Ren}_{\mathcal{S}} \to \mathcal{S}$, where $\mathbf{Ren}_{\mathcal{S}}$ is the category of renamings of $\mathcal{S}$, i.e. the category whose morphisms are the substitutions of $\mathcal{S}$ that are built out of variables. We use the alternative definition from [10] of $\mathbf{Ren}_{\mathcal{S}}$ as the initial object in a category of first-order renaming algebras.

5.1 The category of renamings

▶ **Definition 16.** *Let $\mathcal{C}$ be a first-order model of $\mathcal{T}$. A **higher-order renaming algebra** $\mathbb{C}$ over $\mathcal{C}$ consists of:*

$$
\begin{aligned}
&\mathbb{C}.\mathsf{Var} : \mathcal{C}.\mathsf{Ty}(1_{\mathcal{C}}) \to \mathcal{U}, \\
&\mathbb{C}.\mathsf{var} : (A : \mathcal{C}.\mathsf{Ty}(1_{\mathcal{C}})) \to \mathbb{C}.\mathsf{Var}(A) \to \mathcal{C}.\mathsf{Tm}(1_{\mathcal{C}}, A).
\end{aligned}
$$

⌟

▶ **Definition 17.** *Let $\mathcal{D}$ be a first-order model of $\mathcal{T}$. A **first-order renaming algebra** over $\mathcal{D}$ is a category $\mathcal{C}$ with a terminal object along with a functor $F : \mathcal{C} \to \mathcal{D}$ that preserves the terminal object and with the structure of a global higher-order renaming algebra $\mathbb{C}$ over $F^*(\mathsf{Tele}_{\mathbb{D}})$ such that $\mathbb{C}.\mathsf{Var}$ is locally representable and $\mathbb{C}.\mathsf{var}$ strictly preserves context extensions.* ⌟

Equivalently, a first-order renaming algebra over $\mathcal{D}$ is a CwF $\mathcal{C}$ together with a CwF morphism $F : \mathcal{C} \to \mathcal{D}$ whose action on types is bijective. (There is a cofibrantly generated factorization system with such morphisms as its right class. The renaming algebras are the objects in the left class.) The category of first-order renaming algebras over $\mathcal{S}$ is locally finitely presentable, and there is an initial first-order renaming algebra $\mathbf{Ren}_{\mathcal{S}}$. The category $\mathbf{Ren}_{\mathcal{S}}$ is the category of renamings of $\mathcal{S}$; in this section we write F for the functor $F : \mathbf{Ren}_{\mathcal{S}} \to \mathcal{S}$.

5.2 Relative induction principle

We pose $\mathcal{S}_F \triangleq F^*(\mathbf{Tele}_{\mathbb{S}})$; $\mathcal{S}_F$ is an internal first-order model in $\mathbf{Psh}(\mathbf{Ren}_{\mathcal{S}})$.

▶ **Theorem 18** (Induction principle for $\mathbb{S}$ relative to $F : \mathbf{Ren}_{\mathcal{S}} \to \mathcal{S}$)**.**
Let $\mathbb{S}^\bullet$ be a displayed higher-order model over $\mathcal{S}_F$. Given the additional data of

$$\mathsf{var}^\bullet : \forall(A : \mathcal{S}_F.\mathsf{Ty}(1_{\mathcal{S}_F}))\ (A^\bullet : \mathsf{Ty}^\bullet(A))\ (x : \mathsf{Var}(A)) \to \mathsf{Tm}^\bullet(A^\bullet, \mathsf{var}(x)),$$

the displayed contextualization $\mathbf{Scone}_{\mathbb{S}^\bullet}$ admits a section $[\![-]\!]$ that satisfies the additional computation rule $[\![\mathsf{var}_A(x)]\!] = \mathsf{var}^\bullet([\![A]\!], x)$.

Proof. See Appendix A.4. ◀

Note that $\mathsf{var}_A(x)$ is always a closed term of $\mathcal{S}_F$, thus $[\![\mathsf{var}_A(x)]\!]$ does not depend on any environment.

5.3 Normal forms

Neutrals and normal forms are defined internally to $\mathbf{Psh}(\mathbf{Ren}_{\mathcal{S}})$, as inductive families

$$\mathsf{Ne}, \mathsf{Nf} : \forall(A : \mathcal{S}_F.\mathsf{Ty}(1_{\mathcal{S}_F}))\ (a : \mathcal{S}_F.\mathsf{Tm}(1_{\mathcal{S}_F}, A)) \to \mathrm{Set},$$

generated by the following constructors:

$$\mathsf{var}^{\mathsf{ne}} : (x : \mathsf{Var}(A)) \to \mathsf{Ne}_A(\mathsf{var}(x)),$$
$$\mathsf{app}^{\mathsf{ne}} : \mathsf{Ne}_{\Pi(A,B)}(f) \to \mathsf{Nf}_A(a) \to \mathsf{Ne}_{B[a]}(\mathsf{app}(f, a)),$$
$$\mathsf{lam}^{\mathsf{nf}} : ((a : \mathsf{Var}(A)) \to \mathsf{Nf}_{B[a]}(b[a])) \to \mathsf{Nf}_{\Pi(A,B)}(\mathsf{lam}(b)).$$

The goal of normalization is to prove that every term has a unique normal form:

$$\forall(A : \mathcal{S}_F.\mathsf{Ty}(1_{\mathcal{S}_F}))\ (a : \mathcal{S}_F.\mathsf{Tm}(1_{\mathcal{S}_F}, A)) \to \mathrm{isContr}(\mathsf{Nf}_A(a)).$$

This is accomplished in two steps. First a normalization function is obtained from the relative induction principle, witnessing the existence of normal forms. Then the uniqueness of normal forms is derived from the stability of the normalization; a fact that is proven by mutual induction on neutrals and normal forms.

5.4 Normalization displayed model

We now construct the normalization displayed higher-order model $\mathbb{S}^\bullet$ over $\mathcal{S}_F$.

A displayed type $A^\bullet : \mathsf{Ty}^\bullet(A)$ over a type $A : \mathcal{S}_F.\mathsf{Ty}(1_{\mathcal{S}_F})$ is a triple $(A^\bullet_p, A^\bullet_u, A^\bullet_q)$ consisting of a logical *predicate* $A^\bullet_p : \mathcal{S}_F.\mathsf{Tm}(1_{\mathcal{S}_F}, A) \to \mathrm{Set}$, over the terms of type A, valued in the universe of sets; an *unquoting* (or reflection) function $A^\bullet_u : (a : \mathcal{S}_F.\mathsf{Tm}(1_{\mathcal{S}_F}, A)) \to \mathsf{Ne}_A(a) \to A^\bullet_p(a)$, witnessing the fact that any neutral term satisfies the logical predicate $A^\bullet_p$; a *quoting* (or reification) function $A^\bullet_q : (a : \mathcal{S}_F.\mathsf{Tm}(1_{\mathcal{S}_F}, A)) \to A^\bullet_p(a) \to \mathsf{Nf}_A(a)$, witnessing the fact a term satisfying the logical predicate $A^\bullet_p$ admits a normal form. A displayed term $a^\bullet$ of type $A^\bullet$ over a term $a : \mathcal{S}_F.\mathsf{Tm}(1_{\mathcal{S}_F}, A)$ is an element of $A^\bullet_p(a)$: $\mathsf{Tm}^\bullet(A^\bullet, a) \triangleq A^\bullet_p(a)$. The logical predicate for Π-types is defined in the same way as in the canonicity model.

$$\Pi^\bullet_p(A^\bullet, B^\bullet)(f) \triangleq (\forall a\ a^\bullet \to B^\bullet_p(a^\bullet, \mathsf{app}(f, a))).$$

The unquoting function relies on the unquoting function of the codomain and the quoting function of the domain.

$$\Pi^\bullet_u(A^\bullet, B^\bullet)(f^{\mathsf{ne}}) \triangleq \lambda a^\bullet \mapsto B^\bullet_u(a^\bullet, \mathsf{app}^{\mathsf{ne}}(f^{\mathsf{ne}}, A^\bullet_q(a^\bullet))).$$

The quoting function says that any element of a Π-type is a lambda, as implied by the η-rule. It relies on the quoting function of the codomain and the unquoting function of the domain, and on the fact that every variable is neutral.

$$\Pi^\bullet_q(A^\bullet, B^\bullet)(f^\bullet) \triangleq \mathsf{lam}^{\mathsf{nf}}(\lambda a \mapsto \mathsf{let}\ a^\bullet = A^\bullet_u(\mathsf{var}^{\mathsf{ne}}(a))\ \mathsf{in}\ B^\bullet_q(a^\bullet, f^\bullet(a^\bullet))).$$

This completes the definition of the displayed higher-order model $\mathbb{S}^\bullet$. It remains to check the last hypothesis of the relative induction principle:

$$\mathsf{var}^\bullet \quad : \quad \forall A\ A^\bullet\ (x : \mathsf{Var}(A)) \to \mathsf{Tm}^\bullet(A^\bullet, \mathsf{var}(x)),$$
$$\mathsf{var}^\bullet(A^\bullet, x) \triangleq A^\bullet_u(\mathsf{var}^{\mathsf{ne}}(x)).$$

By the relative induction principle (Theorem 18), we obtain a section $[\![-]\!]$ of $\mathbf{Scone}_{\mathbb{S}^\bullet}$. We can then define the normalization function as follows:

$$\mathsf{norm} \quad : \quad \forall A\ (a : \mathcal{S}_F.\mathsf{Tm}(1_{\mathcal{S}_F}, A)) \to \mathsf{Nf}_A(a),$$
$$\mathsf{norm}_A(a) \triangleq [\![A]\!]_q([\![a]\!]).$$

5.5 Stability of normalization and uniqueness of normal forms

Finally, we show the uniqueness of normal forms following [26]: we prove that normalization is stable, that is every normal form for a term a is equal to the normal form of a obtained from the normalization function. As the proof relies on most of the computation rules of the section $[\![-]\!]$, it is a good example of computations with a section.

▶ **Lemma 19** (Stability). *Given any normal form $a^{\mathsf{nf}} : \mathsf{Nf}_A(a)$, we have $a^{\mathsf{nf}} = \mathsf{norm}_A(a)$.*

Proof. We prove the following two facts, by mutual induction on neutrals and normal forms:

$$(a^{\mathsf{ne}} : \mathsf{Ne}_A(a)) \to [\![a]\!] = [\![A]\!]_u(a^{\mathsf{ne}}), \qquad (a^{\mathsf{nf}} : \mathsf{Nf}_A(a)) \to a^{\mathsf{nf}} = [\![A]\!]_q([\![a]\!]).$$

Each case involves some of the computation rules of $[\![-]\!]$.

Case $a^{\mathsf{ne}} = \mathsf{var}^{\mathsf{ne}}(A, x)$

$$\begin{aligned}
&[\![\mathsf{var}_A(x)]\!] \\
&\quad = \mathsf{var}^\bullet([\![A]\!], x) && \text{(by the computation rule for } [\![\mathsf{var}(-)]\!]) \\
&\quad = [\![A]\!]_u(a^{\mathsf{ne}}). && \text{(by definition of } \mathsf{var}^\bullet)
\end{aligned}$$

Case $a^{\mathsf{ne}} = \mathsf{app}^{\mathsf{ne}}(f^{\mathsf{ne}}, a^{\mathsf{nf}})$

$$\begin{aligned}
&[\![\mathsf{app}(f, a)]\!] \\
&\quad = \mathsf{app}^\bullet([\![f]\!], [\![a]\!]) && \text{(by the computation rule for } [\![\mathsf{app}(-)]\!]) \\
&\quad = [\![f]\!]([\![a]\!]) && \text{(by definition of } \mathsf{app}^\bullet) \\
&\quad = [\![\Pi(A, B)]\!]_u(f^{\mathsf{ne}}, [\![a]\!]) && \text{(by the induction hypothesis for } f^{\mathsf{ne}}) \\
&\quad = [\![B[a]]\!]_u(\mathsf{app}^{\mathsf{ne}}(f^{\mathsf{ne}}, a^{\mathsf{nf}})). \\
& && \text{(by definition of } \Pi^\bullet_u \text{ and the induction hypothesis for } a^{\mathsf{nf}})
\end{aligned}$$

Case $a^{\mathsf{nf}} = \mathsf{lam}^{\mathsf{nf}}(b^{\mathsf{nf}})$

$$
\begin{aligned}
&[\![\Pi(A,B)]\!]_q([\![\mathsf{lam}(b)]\!]) \\
&\quad = \Pi^\bullet_q([\![A]\!], \lambda a^\bullet \mapsto [\![B(\underline{a})]\!][\underline{a} \mapsto a^\bullet])(\lambda a^\bullet \mapsto [\![b(\underline{a})]\!][\underline{a} \mapsto a^\bullet]) \\
&\qquad\qquad \text{(by the computation rules for } [\![\Pi(-)]\!] \text{ and } [\![\mathsf{lam}]\!]) \\
&\quad = \mathsf{lam}^{\mathsf{nf}}(\lambda a \mapsto \mathsf{let}\ a^\bullet = [\![A]\!]_u(\mathsf{var}^{\mathsf{ne}}(a))\ \mathsf{in}\ ([\![B(\underline{a})]\!][\underline{a} \mapsto a^\bullet])_q([\![b(\underline{a})]\!][\underline{a} \mapsto a^\bullet]) \\
&\qquad\qquad \text{(by definition of } \Pi^\bullet_q) \\
&\quad = \mathsf{lam}^{\mathsf{nf}}(\lambda a \mapsto ([\![B(\underline{a})]\!][\underline{a} \mapsto [\![\mathsf{var}(a)]\!]])_q([\![b(\underline{a})]\!][\underline{a} \mapsto [\![\mathsf{var}(a)]\!]])) \\
&\qquad\qquad \text{(by the computation rule for } [\![\mathsf{var}(a)]\!]) \\
&\quad = \mathsf{lam}^{\mathsf{nf}}(\lambda a \mapsto [\![B[\mathsf{var}(a)]]\!]_q([\![b[\mathsf{var}(a)]]\!])) \qquad \text{(by the naturality of } [\![-]\!]) \\
&\quad = \mathsf{lam}^{\mathsf{nf}}(b^{\mathsf{nf}}). \qquad \text{(by the induction hypothesis for } b^{\mathsf{nf}}) \quad \blacktriangleleft
\end{aligned}
$$

6 The category of sections of a displayed first-order model

The last tool that is needed for the proofs of relative induction principles is the *category of sections* of an internal displayed higher-order model. This replaces the use of a displayed inserter in our previous work [10].

Let $\mathcal{M}^\bullet$ be a global internal displayed first-order model over a first-order model $\mathcal{M}$, internally to some presheaf category $\mathbf{Psh}(\mathcal{C})$. We see $\mathcal{M}$ as a functor $\mathcal{M} : \mathcal{C} \to \mathbf{Mod}^{\mathsf{op}}_{\mathcal{T}}$, as justified by Proposition 6. Write $\mathbf{DispMod}_{\mathcal{T}}$ for the external category of a first-order model of $\mathcal{T}$ with a displayed first-order model over it. There is a forgetful functor $U : \mathbf{DispMod}_{\mathcal{T}} \to \mathbf{Mod}_{\mathcal{T}}$. Since the notion of displayed first-order model is also essentially algebraic, we can also view $\mathcal{M}^\bullet$ as a functor $\mathcal{C} \to \mathbf{DispMod}^{\mathsf{op}}_{\mathcal{T}}$ such that $U \circ \mathcal{M}^\bullet = \mathcal{M}$. Similarly, writing $\mathbf{Sect}_{\mathcal{T}}$ for the category of a displayed first-order model of $\mathcal{T}$ with a section, a section of $\mathcal{M}^\bullet$ can be identified with a functor $[\![-]\!] : \mathcal{C} \to \mathbf{Sect}^{\mathsf{op}}_{\mathcal{T}}$ such that $V \circ [\![-]\!] = \mathcal{M}^\bullet$, where V is the forgetful functor $\mathbf{Sect}_{\mathcal{T}} \to \mathbf{DispMod}_{\mathcal{T}}$.

▶ **Definition 20.** *The category* $\mathbf{Sect}^{\mathsf{op}}_{\mathcal{T}}[\mathcal{M}^\bullet]$ *of* ***sections*** *of* $\mathcal{M}^\bullet$ *is the pullback (in* $\mathbf{Cat}$*)*

$$
\begin{array}{ccc}
\mathbf{Sect}^{\mathsf{op}}_{\mathcal{T}}[\mathcal{M}^\bullet] & \xrightarrow{[\![-]\!]_0} & \mathbf{Sect}^{\mathsf{op}}_{\mathcal{T}} \\
\downarrow{\scriptstyle \pi_0} & \lrcorner & \downarrow{\scriptstyle V} \\
\mathcal{C} & \xrightarrow{\mathcal{M}^\bullet} & \mathbf{DispMod}^{\mathsf{op}}_{\mathcal{T}}
\end{array}
$$

By the universal property of the pullback, the data of a section of $\mathcal{M}^\bullet$ is equivalent to the data of a section of π_0 in $\mathbf{Cat}$. The category $\mathbf{Sect}^{\mathsf{op}}_{\mathcal{T}}[\mathcal{M}^\bullet]$ is itself equipped with a section $[\![-]\!]_0$ of $\pi_0^*(\mathcal{M}^\bullet)$, which is called the *generic section* of $\mathcal{M}^\bullet$.

In order to prove an induction principle such as Theorem 18, we want to use the initiality of $\mathcal{C}$ in some category to obtain section of π_0. For example, when $\mathcal{C}$ is the category of renamings, it suffices to equip $\mathbf{Sect}^{\mathsf{op}}_{\mathcal{T}}[\mathcal{M}^\bullet]$ with the structure of a renaming algebra that is preserved by π_0.

This typically involves lifting the terminal object and context extensions of $\mathcal{C}$ to $\mathbf{Sect}^{\mathsf{op}}_{\mathcal{T}}[\mathcal{M}^\bullet]$. Conditions for the lifting of these finite limits are given in Appendix A.3; the initiality of the syntax is needed to lift the terminal object.

References

1 Andreas Abel, Joakim Öhman, and Andrea Vezzosi. Decidability of conversion for type theory in type theory. *Proc. ACM Program. Lang.*, 2(POPL):23:1–23:29, 2018. doi:10.1145/3158111.

2 Thorsten Altenkirch, Martin Hofmann, and Thomas Streicher. Categorical reconstruction of a reduction free normalization proof. In David H. Pitt, David E. Rydeheard, and Peter T. Johnstone, editors, *Category Theory and Computer Science, 6th International Conference, CTCS '95, Cambridge, UK, August 7-11, 1995, Proceedings*, volume 953 of *Lecture Notes in Computer Science*, pages 182–199. Springer, 1995. doi:10.1007/3-540-60164-3_27.

3 Thorsten Altenkirch, Martin Hofmann, and Thomas Streicher. Reduction-free normalisation for a polymorphic system. In *Proceedings, 11th Annual IEEE Symposium on Logic in Computer Science, New Brunswick, New Jersey, USA, July 27-30, 1996*, pages 98–106. IEEE Computer Society, 1996. doi:10.1109/LICS.1996.561309.

4 Thorsten Altenkirch and Ambrus Kaposi. Normalisation by Evaluation for Dependent Types. In Delia Kesner and Brigitte Pientka, editors, *1st International Conference on Formal Structures for Computation and Deduction (FSCD 2016)*, volume 52 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:16, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.FSCD.2016.6.

5 Thorsten Altenkirch and Ambrus Kaposi. Type theory in type theory using quotient inductive types. In *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '16, pages 18–29, New York, NY, USA, 2016. ACM. doi:10.1145/2837614.2837638.

6 Thorsten Altenkirch, Ambrus Kaposi, and Michael Shulman. Towards higher observational type theory. In Delia Kesner and Pierre-Marie Pédrot, editors, *28th International Conference on Types for Proofs and Programs (TYPES 2022)*. University of Nantes, 2022. URL: https://types22.inria.fr/files/2022/06/TYPES_2022_paper_37.pdf.

7 Danil Annenkov, Paolo Capriotti, Nicolai Kraus, and Christian Sattler. Two-level type theory and applications. *CoRR*, abs/1705.03307, 2019. arXiv:1705.03307.

8 Steve Awodey. Natural models of homotopy type theory. *Mathematical Structures in Computer Science*, 28(2):241–286, 2018. doi:10.1017/S0960129516000268.

9 Jean-Philippe Bernardy, Patrik Jansson, and Ross Paterson. Proofs for free — parametricity for dependent types. *Journal of Functional Programming*, 22(02):107–152, 2012. doi:10.1017/S0956796812000056.

10 Rafaël Bocquet, Ambrus Kaposi, and Christian Sattler. Relative induction principles for type theories. *CoRR*, abs/2102.11649, 2021. arXiv:2102.11649.

11 Simon Castellan, Pierre Clairambault, and Peter Dybjer. Categories with families: Unityped, simply typed, and dependently typed. *CoRR*, abs/1904.00827, 2019. arXiv:1904.00827.

12 Thierry Coquand. Canonicity and normalization for dependent type theory. *Theor. Comput. Sci.*, 777:184–191, 2019. doi:10.1016/j.tcs.2019.01.015.

13 Thierry Coquand, Simon Huber, and Christian Sattler. Canonicity and homotopy canonicity for cubical type theory, 2021. arXiv:1902.06572.

14 Thierry Coquand, Simon Huber, and Christian Sattler. Canonicity and homotopy canonicity for cubical type theory. *Log. Methods Comput. Sci.*, 18(1), 2022. doi:10.46298/lmcs-18(1:28)2022.

15 Marcelo P. Fiore. Semantic analysis of normalisation by evaluation for typed lambda calculus. In *Proceedings of the 4th international ACM SIGPLAN conference on Principles and practice of declarative programming, October 6-8, 2002, Pittsburgh, PA, USA (Affiliated with PLI 2002)*, pages 26–37. ACM, 2002. doi:10.1145/571157.571161.

16 Marcelo P. Fiore and Ola Mahmoud. Second-order algebraic theories. *CoRR*, abs/1308.5409, 2013. arXiv:1308.5409.

17 Daniel Gratzer. Normalization for multimodal type theory. In *Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science*, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3531130.3532398.

18 Daniel Gratzer and Jonathan Sterling. Syntactic categories for dependent type theory: sketching and adequacy. *CoRR*, abs/2012.10783, 2020. arXiv:2012.10783.

19 Robert Harper. An equational logical framework for type theories. *CoRR*, abs/2106.01484, 2021. arXiv:2106.01484.

20 Robert Harper, Furio Honsell, and Gordon D. Plotkin. A framework for defining logics. *J. ACM*, 40(1):143–184, 1993. doi:10.1145/138027.138060.

21 Martin Hofmann. Syntax and semantics of dependent types. In *Semantics and Logics of Computation*, pages 79–130. Cambridge University Press, 1997.

22 Martin Hofmann. Semantical analysis of higher-order abstract syntax. In *14th Annual IEEE Symposium on Logic in Computer Science, Trento, Italy, July 2-5, 1999*, pages 204–213. IEEE Computer Society, 1999. doi:10.1109/LICS.1999.782616.

23 Jason Z. S. Hu, Brigitte Pientka, and Ulrich Schöpp. A category theoretic view of contextual types: from simple types to dependent types. *CoRR*, abs/2206.02831, 2022. doi:10.48550/arXiv.2206.02831.

24 Peter T Johnstone. *Sketches of an elephant: a Topos theory compendium*. Oxford logic guides. Oxford Univ. Press, New York, NY, 2002. URL: https://cds.cern.ch/record/592033.

25 Achim Jung and Jerzy Tiuryn. A new characterization of lambda definability. In Marc Bezem and Jan Friso Groote, editors, *Typed Lambda Calculi and Applications, International Conference on Typed Lambda Calculi and Applications, TLCA '93, Utrecht, The Netherlands, March 16-18, 1993, Proceedings*, volume 664 of *Lecture Notes in Computer Science*, pages 245–257. Springer, 1993. doi:10.1007/BFb0037110.

26 Ambrus Kaposi. *Type theory in a type theory with quotient inductive types*. PhD thesis, University of Nottingham, UK, 2017. URL: https://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.713896.

27 Ambrus Kaposi, Simon Huber, and Christian Sattler. Gluing for type theory. In Herman Geuvers, editor, *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24-30, 2019, Dortmund, Germany*, volume 131 of *LIPIcs*, pages 25:1–25:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.FSCD.2019.25.

28 Krzysztof Kapulkin and Peter LeFanu Lumsdaine. The homotopy theory of type theories. *Advances in Mathematics*, 337:1–38, 2018. doi:10.1016/j.aim.2018.08.003.

29 Frank Pfenning and Carsten Schürmann. System description: Twelf - A meta-logical framework for deductive systems. In Harald Ganzinger, editor, *Automated Deduction - CADE-16, 16th International Conference on Automated Deduction, Trento, Italy, July 7-10, 1999, Proceedings*, volume 1632 of *Lecture Notes in Computer Science*, pages 202–206. Springer, 1999. doi:10.1007/3-540-48660-7_14.

30 Loïc Pujet and Nicolas Tabareau. Impredicative Observational Equality. In *POPL 2023 - 50th ACM SIGPLAN Symposium on Principles of Programming Languages*, volume 7 of *Proceedings of the ACM on programming languages*, page 74, Boston, United States, January 2023. doi:10.1145/3571739.

31 Michael Shulman. Towards a third-generation HOTT. Talk series at the Homotopy Type Theory at CMU seminar, 2022.

32 Jonathan Sterling. *First Steps in Synthetic Tait Computability: The Objective Metatheory of Cubical Type Theory*. PhD thesis, Carnegie Mellon University, 2021. Version 1.1, revised May 2022. doi:10.5281/zenodo.6990769.

33 Jonathan Sterling. Naïve logical relations in synthetic Tait computability. Unpublished manuscript, June 2022.

34 Jonathan Sterling and Carlo Angiuli. Normalization for cubical type theory. *CoRR*, abs/2101.11479, 2021. arXiv:2101.11479.

35 Jonathan Sterling and Bas Spitters. Normalization by gluing for free λ-theories. *CoRR*, abs/1809.08646, 2018. arXiv:1809.08646.

36 Taichi Uemura. A general framework for the semantics of type theory. *CoRR*, abs/1904.04097, 2019. arXiv:1904.04097.

37 Jonathan Weinberger, Benedikt Ahrens, Ulrik Buchholtz, and Paige North. Synthetic Tait computability for simplicial type theory. In *28th International Conference on Types for Proofs and Programs (TYPES 2022)*, 2022. URL: https://types22.inria.fr/files/2022/06/TYPES_2022_paper_17.pdf.

A Technical results

A.1 Local representability

We recall the definition of the notion of locally representable dependent presheaf, which encodes the notion of context extension.

▶ **Definition 21.** *Let $\mathcal{C}$ be a category, X be a presheaf over $\mathcal{C}$ and Y be a dependent presheaf over X. Then Y is said to be* ***locally representable*** *if for every element $x : X(\Gamma)$, the presheaf*

$$Y_{|x} : (\mathcal{C}/\Gamma)^{\mathsf{op}} \to \mathbf{Set},$$
$$Y_{|x}(\Delta, \rho) \triangleq Y(\Delta, x[\rho])$$

is representable. The representing object, consisting of an extended context and a projection map, is written $(\Gamma.Y[x], \boldsymbol{p}_x)$ and the generic element is written $\boldsymbol{q}_x : Y(\Gamma.Y[x], x[\boldsymbol{p}_x])$.

Given any object $\Delta \in \mathcal{C}$, map $\rho : \Delta \to \Gamma$ and element $y : Y(\Delta, x[\rho])$, we write $\langle \rho, a \rangle$ for the unique morphism such that $\boldsymbol{p}_x \circ \langle \rho, y \rangle = \rho$ and $\boldsymbol{q}_x[\langle \rho, y \rangle] = y$. ⌟

A.2 Characterization of the telescopic contextualization

We define the contextual slices of a first-order model (which are called fibrant slices in [28]).

▶ **Definition 22** (Contextual slice)**.** *Let $\mathcal{C}$ be a first-order model of $\mathcal{T}$. Given $\Gamma \in \mathcal{C}$, the contextual slice $(\mathcal{C} /\!\!/ \Gamma)$ is the contextual first-order model given by:*

- *Objects of $(\mathcal{C} /\!\!/ \Gamma)$ are telescopes (iterated context extensions) over Γ.*
- *Morphisms from Δ_1 to Δ_2 are morphisms from $\Gamma.\Delta_1$ to $\Gamma.\Delta_2$ in $(\mathcal{C}/\Gamma)$.*
- *The rest of the structure is inherited from $\mathcal{C}$ along the projection*

$$(\mathcal{C} /\!\!/ \Gamma) \ni \Delta \mapsto \Gamma.\Delta \in \mathcal{C}.$$ ⌟

The contextual slice is functorial in both $\mathcal{C}$ and (contravariantly) Γ:

- For any $f : \Delta \to \Gamma$, there is a pullback morphism $f^* : (\mathcal{C} /\!\!/ \Gamma) \to (\mathcal{C} /\!\!/ \Delta)$. Its actions on objects, substitutions, types and terms are all given by substitution along f.
- For any $F : \mathcal{C} \to \mathcal{D}$, there is a morphism $(F /\!\!/ \Gamma) : (\mathcal{C} /\!\!/ \Gamma) \to (\mathcal{D} /\!\!/ F(\Gamma))$. Its actions on objects, substitutions, types and terms are given by the actions of F on telescopes, substitutions, types and terms.
- The following diagrams commute (for any $F : \mathcal{C} \to \mathcal{D}$ and $f : \Delta \to \Gamma$):

$$\begin{array}{ccc} (\mathcal{C} /\!\!/ \Gamma) & \xrightarrow{(F/\!\!/\Gamma)} & (\mathcal{D} /\!\!/ F(\Gamma)) \\ \downarrow{\scriptstyle f^*} & & \downarrow{\scriptstyle f^*} \\ (\mathcal{C} /\!\!/ \Delta) & \xrightarrow{(F/\!\!/\Delta)} & (\mathcal{D} /\!\!/ F(\Delta)) \end{array}.$$

- For any $f : \Theta \to \Gamma$ and object Δ of $(\mathcal{C} /\!\!/ \Gamma)$, we have $(f^* /\!\!/ \Delta) = \langle \boldsymbol{p}^*_\Delta(f), \boldsymbol{q}_\Delta \rangle^*$ as a morphism from $(\mathcal{C} /\!\!/ \Gamma.\Delta)$ to $(\mathcal{C} /\!\!/ \Theta.f^*(\Delta))$. Here $\langle f \circ \boldsymbol{p}_{f^*(\Delta)}, \boldsymbol{q}_{f^*(\Delta)} \rangle$ is a morphism from $\Theta.f^*(\Delta)$ to $\Gamma.\Delta$.

▶ **Lemma 23.** *Let $\mathcal{C}$ be a first-order model of $\mathcal{T}$. Then the telescopic contextualization $\mathbf{Tele}_\mathbb{C}$ corresponds the contextual slice functor*

$$(\mathcal{C} /\!\!/ -) : \mathcal{C} \to \mathbf{Mod}^{\mathsf{op}}_{\mathcal{T}}.$$ ⌟

Proof. Immediate from unfolding the definitions. ◀

▶ **Corollary 24.** *The externalization $1^*_{\mathcal{C}}(\mathbf{Tele}_\mathbb{C})$ is the contextual core of $\mathcal{C}$.*

Proof. By Lemma 23, $1^*_{\mathcal{C}}(\mathbf{Tele}_\mathbb{C})$ is the contextual slice $(\mathcal{C} /\!\!/ 1_{\mathcal{C}})$, the contextual core of $\mathcal{C}$. ◀

▶ **Lemma 25.** *Let $\mathcal{C}$ be a contextual first-order model and $A : \mathcal{C}.\mathsf{Ty}(1_{\mathcal{C}})$ be a closed type. Then the contextual slice $(\mathcal{C} /\!\!/ A)$ satisfies the following universal property: for every model $\mathcal{E}$, morphism $F : \mathcal{C} \to \mathcal{E}$ and element $a : \mathcal{E}.\mathsf{Tm}(F(A))$, there is a unique morphism $\widetilde{F} : (\mathcal{C} /\!\!/ 1_{\mathcal{C}}.A) \to \mathcal{E}$ such that $\widetilde{F} \circ \boldsymbol{p}^*_A = F$ and $\widetilde{F}(\boldsymbol{q}_A) = a$.*

In other words, $(\mathcal{C} /\!\!/ 1_{\mathcal{C}}.A)$ is the free extension of $\mathcal{C}$ by a generic element $\boldsymbol{q}_A$ of type A.

See the Full Version of the paper for the proof.

▶ **Lemma 26.** *Given any $\Gamma \in \mathcal{C}$ and $A : \mathcal{C}.\mathsf{Ty}(\Gamma)$, then $(\mathcal{C} /\!\!/ \Gamma.A)$ satisfies the following universal property: for every model $\mathcal{E}$, morphism $F : (\mathcal{C} /\!\!/ \Gamma) \to \mathcal{E}$ and element $a : \mathcal{E}.\mathsf{Tm}(F(A))$, there is a unique morphism $\widetilde{F} : (\mathcal{C} /\!\!/ \Gamma.A) \to \mathcal{E}$ such that $\widetilde{F} \circ \boldsymbol{p}^*_A = F$ and $\widetilde{F}(\boldsymbol{q}_A) = a$.*

In other words, $(\mathcal{C} /\!\!/ \Gamma.A)$ is the free extension of $(\mathcal{C} /\!\!/ \Gamma)$ by a generic element $\boldsymbol{q}_A$ of type A.

Proof. We have $(\mathcal{C} /\!\!/ \Gamma.A) \cong ((\mathcal{C} /\!\!/ \Gamma) /\!\!/ 1_{(\mathcal{C} /\!\!/ \Gamma)}.A)$. Then the result follows from Lemma 25. ◀

A.3 Properties of the category of sections

We now prove the properties of the category of sections of a displayed first-order model. Fix a global internal first-order model $\mathcal{M} : \mathcal{C} \to \mathbf{Mod}^{\mathsf{op}}_{\mathcal{T}}$ and a displayed first-order model $\mathcal{M}^\bullet : \mathcal{C} \to \mathbf{DispMod}^{\mathsf{op}}_{\mathcal{T}}$ over $\mathcal{M}$.

We prove conditions that relate the existence of some finite limits in $\mathbf{Sect}^{\mathsf{op}}_{\mathcal{T}}[\mathcal{M}^\bullet]$ to the universal properties of the first-order models $\mathcal{M}(\Gamma)$ for $\Gamma \in \mathcal{C}$.

We have defined the category $\mathbf{Sect}^{\mathsf{op}}_{\mathcal{T}}[\mathcal{M}^\bullet]$ of sections of $\mathcal{M}^\bullet$ as the following pullback

$$\begin{array}{ccc} \mathbf{Sect}^{\mathsf{op}}_{\mathcal{T}}[\mathcal{M}^\bullet] & \xrightarrow{[\![-]\!]_0} & \mathbf{Sect}^{\mathsf{op}}_{\mathcal{T}} \\ \downarrow{\scriptstyle \pi_0} & \lrcorner & \downarrow \\ \mathcal{C} & \xrightarrow{\mathcal{M}^\bullet} & \mathbf{DispModel}^{\mathsf{op}}_{\mathcal{T}} . \end{array}$$

Unfolding the definition, an object of $\mathbf{Sect}^{\mathsf{op}}_{\mathcal{T}}[\mathcal{M}^\bullet]$ is a pair $(\Gamma, [\![-]\!]_\Gamma)$ where $\Gamma \in \mathcal{C}$ and $[\![-]\!]_\Gamma$ is a section of $\mathcal{M}^\bullet(\Gamma)$ over $\mathcal{M}(\Gamma)$. A morphism of from $(\Gamma, [\![-]\!]_\Gamma)$ to $(\Delta, [\![-]\!]_\Delta)$ is a morphism $f : \mathcal{C}(\Gamma, \Delta)$ such that the outer square commutes in the following diagram:

$$\begin{array}{ccc} \mathcal{M}^\bullet(\Delta) & \xrightarrow{\mathcal{M}^\bullet(f)} & \mathcal{M}^\bullet(\Gamma) \\ {\scriptstyle [\![-]\!]_\Gamma}\uparrow\downarrow & & \downarrow\uparrow{\scriptstyle [\![-]\!]_\Delta} \\ \mathcal{M}(\Delta) & \xrightarrow{\mathcal{M}(f)} & \mathcal{M}(\Gamma) . \end{array}$$

▶ **Lemma 27.** *If $\mathcal{C}$ has a terminal object $1_{\mathcal{C}}$ and $\mathcal{M}(1_{\mathcal{C}})$ is the initial model of $\mathcal{T}$, then $\mathbf{Sect}^{\mathsf{op}}_{\mathcal{M}^\bullet}$ has a terminal object that is strictly preserved by π_0.*

Proof. Since $\mathcal{M}(1_{\mathcal{C}})$ is initial, we obtain a section $[\![-]\!]_{1_{\mathcal{C}}}$ of $\mathcal{M}^\bullet(1_{\mathcal{C}})$. We now prove that $(1_{\mathcal{C}}, [\![-]\!]_{1_{\mathcal{C}}})$ is terminal in $\mathbf{Sect}^{\mathsf{op}}_{\mathcal{M}^\bullet}$. Let $(\Gamma, [\![-]\!]_\Gamma)$ be any object of $\mathbf{Sect}^{\mathsf{op}}_{\mathcal{M}^\bullet}$. A morphism from $(\Gamma, [\![-]\!]_\Gamma)$ to $(1_{\mathcal{C}}, [\![-]\!]_{1_{\mathcal{C}}})$ is a morphism $f : \mathcal{C}(\Gamma, 1_{\mathcal{C}})$ such that the following square commutes:

$$\begin{array}{ccc} \mathcal{M}^\bullet(1_{\mathcal{C}}) & \xrightarrow{\mathcal{M}^\bullet(f)} & \mathcal{M}^\bullet(\Gamma) \\ {\scriptstyle [\![-]\!]_{1_{\mathcal{C}}}}\uparrow & & \uparrow{\scriptstyle [\![-]\!]_\Gamma} \\ \mathcal{M}(1_{\mathcal{C}}) & \xrightarrow{\mathcal{M}(f)} & \mathcal{M}(\Gamma)\ . \end{array}$$

Since $1_{\mathcal{C}}$ is terminal, there is only one morphism $f : \mathcal{C}(\Gamma, 1_{\mathcal{C}})$, and the corresponding square commutes by initiality of $\mathcal{M}(1_{\mathcal{C}})$. ◀

▶ **Definition 28.** *Let X be a presheaf over $\mathcal{C}$ and Y be a locally representable dependent presheaf over X. Assume given the data of global elements*

$$f_X : X \to \mathcal{M}.\mathsf{Ty}(1_{\mathcal{M}}),$$
$$f_Y : (x : X) \to Y(x) \to \mathcal{M}.\mathsf{Tm}(1_{\mathcal{M}}, f_X(x))$$

of $\mathbf{Psh}(\mathcal{C})$.

We say that f_Y is compatible with $\mathcal{M}$ if for every element $x : X(\Gamma)$, the external first-order model $\mathcal{M}(\Gamma.Y(x))$ satisfies the following universal property: for every first-order model $\mathcal{E}$, morphism $E : \mathcal{M}(\Gamma) \to \mathcal{E}$ and element $z : \mathcal{E}.\mathsf{Tm}(1_{\mathcal{E}}, E(f_X(x)))$, there is a unique morphism $\widetilde{E} : \mathcal{M}(\Gamma.Y(x)) \to \mathcal{E}$ such that $E = \widetilde{E} \circ \mathcal{M}(\boldsymbol{p}_x)$ and $z = \widetilde{E}(f_Y(x, \boldsymbol{q}_x))$.

In other words, $\mathcal{M}(\Gamma.Y(x))$ should be the free extension of $\mathcal{M}(\Gamma)$ by an element $f_Y(x, \boldsymbol{q}_x)$ of type $f_X(x)$, the extension being witnessed by the morphism $\mathcal{M}(\boldsymbol{p}_x) : \mathcal{M}(\Gamma) \to \mathcal{M}(\Gamma.Y(x))$. ⌟

▶ **Lemma 29.** *Let $F : \mathcal{C} \to \mathcal{D}$ be a CwF morphism.*

Then the condition of Definition 28 is satisfied, with $\mathcal{M} = F^*(\mathsf{Tele}_{\mathbb{D}})$, X and Y being respectively the types and terms of $\mathcal{C}$, and f_X and f_Y being the actions of F on types and terms.

Proof. By Definition 22, $\mathcal{M}$ correspond to the functor

$$\mathcal{C} \ni \Gamma \mapsto (\mathcal{D} /\!\!/ F(\Gamma)) \in \mathbf{Mod}^{\mathsf{op}}_{\mathcal{T}}.$$

Thus, we need to check that given any $\Gamma \in \mathcal{C}$ and $A : \mathcal{C}.\mathsf{Ty}(\Gamma)$, the model $(\mathcal{D} /\!\!/ F(\Gamma.A))$ is the free extension of $(\mathcal{D} /\!\!/ F(\Gamma))$ by a generic element of type $F(A)$. Since F preserves extensions, $F(\Gamma.A) \cong F(\Gamma).F(A)$, and thus the result follows by Lemma 26. ◀

▶ **Lemma 30.** *Let X be a presheaf over $\mathcal{C}$ and Y be a locally representable presheaf family over X equipped with operations f_X and f_Y satisfying the condition of Definition 28. Finally, assume that for every $(\Gamma, [\![-]\!]_\Gamma) \in \mathbf{Sect}^{\mathsf{op}}_{\mathcal{M}^\bullet}$, $\Delta \in \mathcal{C}$, $\gamma : \mathcal{C}(\Delta, \Gamma)$, $x : X(\Gamma)$ and $y : Y(\Delta, x[\gamma])$ we have*

$$f^\bullet_Y(x, y) : \mathcal{M}^\bullet(\Delta).\mathsf{Tm}^\bullet(1^\bullet, \mathcal{M}^\bullet(\gamma)([\![f_X(x)]\!]_\Gamma), f_Y(x[\gamma], y)),$$

naturally in $(\Gamma, [\![-]\!]_\Gamma)$ and Δ.

Consider the presheaf $X_0 \triangleq \pi_0^*(X)$ *over* $\mathbf{Sect}^{\mathsf{op}}_{\mathcal{M}^\bullet}$ *and the dependent presheaf* Y_0 *over* X_0 *specified on objects by:*

$$Y_0((\Gamma, [\![-]\!]_\Gamma), x) \triangleq \{y : Y(\Gamma, x) \mid [\![f_Y(x, y)]\!]_\Gamma = f_Y^\bullet(x, y)\}.$$

Then the presheaf family Y_0 *is locally representable and the action induced by the first projections* $Y_0((\Gamma, [\![-]\!]_\Gamma), x) \to Y(\Gamma, x)$ *strictly preserve context extensions.*

Proof. Let $(\Gamma, [\![-]\!]_\Gamma)$ be an object of $\mathbf{Sect}^{\mathsf{op}}_{\mathcal{M}^\bullet}$ and $x : X(\Gamma)$ be an element of X_0 at this object. We have to prove that the presheaf $Y_{0|x}$ over $(\mathbf{Sect}^{\mathsf{op}}_{\mathcal{M}^\bullet}/(\Gamma, [\![-]\!]_\Gamma))$ is representable.

Consider the diagram:

$$\begin{array}{ccc} \mathcal{M}^\bullet(\Gamma) & \xrightarrow{\mathcal{M}^\bullet(\boldsymbol{p}_x)} & \mathcal{M}^\bullet(\Gamma.Y(x)) \\ {\scriptstyle [\![-]\!]_\Gamma}\uparrow \downarrow & & \downarrow \uparrow {\scriptstyle [\![-]\!]_{\Gamma.Y(x)}} \\ \mathcal{M}(\Gamma) & \xrightarrow{\mathcal{M}(\boldsymbol{p}_x)} & \mathcal{M}(\Gamma.Y(x)) \end{array}$$

We construct a section $[\![-]\!]_{\Gamma.Y(x)}$ of $\mathcal{M}^\bullet(\Gamma.Y(x))$ over $\mathcal{M}(\Gamma.Y(x))$. Using the universal property of $\mathcal{M}(\Gamma.Y(x))$, we define $[\![-]\!]_{\Gamma.Y(x)}$ as the unique extension of $\mathcal{M}^\bullet(\boldsymbol{p}_x) \circ [\![-]\!]_\Gamma$ that sends $f_Y(x, \boldsymbol{q}_x)$ to $f_Y^\bullet(x, \boldsymbol{q}_x)$.

We can check that $\boldsymbol{p}_x$ lifts to a morphism $\boldsymbol{p}_x : (\Gamma.Y(x), [\![-]\!]_{\Gamma.Y(x)}) \to (\Gamma, [\![-]\!]_\Gamma)$ in $\mathbf{Sect}^{\mathsf{op}}_{\mathcal{M}^\bullet}$.

We now show that $((\Gamma.Y(x), [\![-]\!]_{\Gamma.Y(x)}), \boldsymbol{p}_x)$ represents the functor $Y_{0|x}$. Let $(\Delta, [\![-]\!]_\Delta)$ be another object of $\mathbf{Sect}^{\mathsf{op}}_{\mathcal{M}^\bullet}$, with a morphism $\rho : (\Delta, [\![-]\!]_\Delta) \to (\Gamma, [\![-]\!]_\Gamma)$ and an element $y : Y_{0|x}((\Delta, [\![-]\!]_\Delta), \rho)$. Unfolding the definitions, we have $y : Y(\Delta, x[\rho])$ with $[\![f_Y(x[\rho], y)]\!]_\Delta = f_Y^\bullet(x[\rho], y)$.

The local representability of Y implies that there is a unique morphism $\widetilde{\rho} : \Delta \to \Gamma.Y(x)$ in $\mathcal{C}$ such that $\boldsymbol{p}_x \circ \widetilde{\rho} = \rho$ and $\boldsymbol{q}_x[\widetilde{\rho}] = y$. We have to show that this morphism lifts to $\mathbf{Sect}^{\mathsf{op}}_{\mathcal{M}^\bullet}$, i.e. that the following square commutes:

$$\begin{array}{ccc} \mathcal{M}^\bullet(\Gamma.Y(x)) & \xrightarrow{\mathcal{M}^\bullet(\widetilde{\rho})} & \mathcal{M}^\bullet(\Delta) \\ {\scriptstyle [\![-]\!]_{\Gamma.Y(x)}}\uparrow \downarrow & & \downarrow \uparrow {\scriptstyle [\![-]\!]_\Delta} \\ \mathcal{M}(\Gamma.Y(x)) & \xrightarrow{\mathcal{M}(\widetilde{\rho})} & \mathcal{M}(\Delta) \end{array}$$

By the universal property of $\mathcal{M}(\Gamma.Y(x))$, it suffices to show that $f_Y(x, \boldsymbol{q}_x)$ is mapped to the same element by the compositions $\mathcal{M}^\bullet(\widetilde{\rho}) \circ [\![-]\!]_{\Gamma.Y(x)}$ and $[\![-]\!]_\Delta \circ \mathcal{M}(\widetilde{\rho})$. We compute $\mathcal{M}^\bullet(\widetilde{\rho})([\![f_Y(x, \boldsymbol{q}_x)]\!]_{\Gamma.Y(x)}) = \mathcal{M}^\bullet(\widetilde{\rho})(f_Y^\bullet(x, \boldsymbol{q}_x)) = f_Y^\bullet(x[\rho], y)$ and $[\![\mathcal{M}(\widetilde{\rho})(f_Y(x, \boldsymbol{q}_x))]\!]_\Delta = [\![f_Y(x[\rho], y)]\!]_\Delta = f_Y^\bullet(x[\rho], y)$.

This completes the proof that $((\Gamma.Y(x), [\![-]\!]_{\Gamma.Y(x)}), \boldsymbol{p}_x)$ represents the functor $Y_{0|x}$.

We have proven that $Y_{0|x}$ is representable for every x, i.e. that Y_0 is locally representable. The first projections $Y_0((\Gamma, [\![-]\!]_\Gamma), x) \to Y(\Gamma, x)$ strictly preserve the chosen representing objects. ◀

A.4 Proofs of relative induction principles

Proof of Theorem 18. We consider the category $\mathbf{Sect}^{\mathsf{op}}_{\mathcal{T}}[\mathbf{Scone}_{\mathbb{S}^\bullet}]$ of sections of $\mathbf{Scone}_{\mathbb{S}^\bullet}$.

By Lemma 27, the category $\mathbf{Sect}^{\mathsf{op}}_{\mathcal{T}}[\mathbf{Scone}_{\mathbb{S}^\bullet}]$ has a terminal object. We equip it with the structure of a higher-order renaming algebra $(\mathsf{Var}_0, \mathsf{var}_0)$ over $(F \circ \pi_0) : \mathbf{Sect}^{\mathsf{op}}_{\mathcal{T}}[\mathbf{Scone}_{\mathbb{S}^\bullet}] \to \mathcal{S}$ as follows:

$$\mathsf{Var}_0((\Gamma, [\![-]\!]_\Gamma), A) \triangleq \{a : \mathsf{Var}(\Gamma, A) \mid [\![\mathsf{var}(\Gamma, a)]\!]_\Gamma = \mathsf{var}^\bullet(\Gamma, [\![A]\!]_\Gamma, a)\},$$

$$\mathsf{var}_0((\Gamma, [\![-]\!]_\Gamma), A, a) \triangleq \mathsf{var}(\Gamma, a).$$

By Lemma 29, the action of $F : \mathcal{R} \to \mathcal{S}$ on variables is compatible with $\mathcal{S}_F$. By Lemma 30, the presheaf family Var_0 is locally representable and the first projections $\mathsf{Var}_0((\Gamma, [\![-]\!]_\Gamma), A) \to \mathsf{Var}(\Gamma, A)$ strictly preserve context extensions. By initiality of $\mathcal{R}$ among first-order renaming algebras, we obtain a section H of π_0 in the category of first-order renaming algebras.

We thus have a section $[\![-]\!] \triangleq H^*([\![-]\!]_0)$ of $\mathbf{Scone}_{\mathbb{S}^\bullet}$ in $\mathbf{Psh}(\mathcal{R})$. The action of H on variables proves that it satisfies the equality $[\![\mathsf{var}_A(x)]\!] = \mathsf{var}^\bullet([\![A]\!], x)$. ◀