THE LOCAL UNIVERSES MODEL: AN OVERLOOKED COHERENCE CONSTRUCTION FOR DEPENDENT TYPE THEORIES

PETER LEFANU LUMSDAINE AND MICHAEL A. WARREN

Abstract. We present a new coherence theorem for comprehension categories, providing strict models of dependent type theory with all standard constructors, including dependent products, dependent sums, identity types, and other inductive types.

Precisely, we take as input a "weak model": a comprehension category, equipped with structure corresponding to the desired logical constructions. We assume throughout that the base category is close to locally Cartesian closed: specifically, that products and certain exponentials exist. Beyond this, we require only that the logical structure should be *weakly stable*—a pure existence statement, not involving any specific choice of structure, weaker than standard categorical Beck–Chevalley conditions, and holding in the now standard homotopy-theoretic models of type theory.

Given such a comprehension category, we construct an equivalent split one, whose logical structure is strictly stable under reindexing. This yields an interpretation of type theory with the chosen constructors.

The model is adapted from Voevodsky's use of universes for coherence, and at the level of fibrations is a classical construction of Giraud. It may be viewed in terms of local universes or delayed substitutions.

Contents

Date: April 6, 2015.

During the preparation of this paper, Lumsdaine was supported by NSF grant DMS-1128155 and the Fondation Sciences Mathématiques de Paris.

Warren was supported by the Oswald Veblen fund and NSF grant DMS-0635607.

1. Introduction

Constructing interpretations of dependent type theory from scratch is a laborious, bureaucratic, and error-prone task. Various algebraic axiomatizations of such models (contextual categories and their relatives) abstract away many of the technicalities, allowing constructions of models to concentrate, for the most part, on their real substance—the logical constructions which genuinely constitute the model.

One issue remains, however, which feels like it ought to be another mere technicality, but which has not been so successfully abstracted away: the so-called coherence problem. In most concrete models, it is not hard to resolve directly, by more or less ad hoc methods. In a few, however (notably, Voevodsky's simplicial model [KLV12] and similar homotopy-theoretic models), it is much less straightforward to deal with. It has also proven problematic for more abstract theorems on model existence, in for instance the weak factorization system models of [AW09].

The most powerful coherence result to date is that of Hofmann [Hof95], which provides coherence for a wide range of models. However, it does not apply to the homotopy-theoretic models mentioned above. The present paper gives a new coherence construction, with slightly different hypotheses from Hofmann's, applying in particular to a wide range of homotopy-theoretic models of intensional type theory.

Specifically, the present work arose from a careful reading of Voevodsky's model in simplicial sets [Voe11, KLV12]. Universes are used there for two distinct purposes: firstly to obtain coherence of the model; and secondly to become type-theoretic universes within the model. It turned out that not only may the two aspects be entirely disentangled, but moreover, the coherence construction may be modified to work without a universe.

The precise sense of *coherence* in question is that substitution/pullback should be strictly functorial, and should strictly preserve all the logical structure under consideration. This holds in the syntax of type theory; so it must hold in any direct model of the theory.

However, categorically such strictness is rather unnatural, involving on-the-nose equality of objects, and rarely arises automatically in nature. We of course expect to find *some* kind of stability—some preservation of logical structure under pullback, such as stability up to isomorphism, or similar—and expect this to be necessary for modelling the type theory. So the outstanding question is: what is some good notion of stability, categorically natural and satisfied in as many of the known models as possible, but strong enough that structures satisfying it can be transformed into ones with strictly stable logical structure?

We investigate this within the framework of *comprehension categories* [Jac93] (one of the many algebraic approaches to modelling type theory). Roughly, a comprehension category consists of a fibration of categories, encoding the contexts, types, and terms of the theory, together with further algebraic structure

implementing the logical rules. Direct models of syntax are given by *split* comprehension categories with *strictly stable* logical structure.

The coherence construction we consider, transforming weak models into strict, is at the level of underlying fibrations a classical one, due to Giraud [Gir65, I, 2.4.3]. Given a comprehension category C, we replace it with an equivalent split one $C_!$, in which the objects/contexts are as in C, but where a type A over Γ consists of another object $V_A \in C$, together with a type E_A over V_A in the sense of C, and a map $\ulcorner A \urcorner : \Gamma \longrightarrow V_A$.

One may view this as a "delayed substitution", i.e. as a surrogate for the pullback $[A] := E_A[\ulcorner A \urcorner]$, and see the pair (V_A, E_A) as a "local universe"—some family of types, not necessarily terribly large or satisfying any closure conditions, from which the map $\ulcorner A \urcorner : \Gamma \longrightarrow V_A$ then picks out the types that actually occur in $[A]$.

$$\begin{array}{ccc} & & E_A \\ & & \vdots \\ \Gamma & \xrightarrow{\ulcorner A \urcorner} & V_A. \end{array}$$

Given a universe (V, E) (not merely a set, but itself an object of C), closed under operations corresponding to the logical constructors, one can simply restrict to the case where (V_A, E_A) is (V, E), so that types are just maps into V. Logical constructions are then modelled by composing these maps with the appropriate operations on V. This is Voevodsky's approach in [KLV12, §1], also previously considered by Hofmann and Streicher [HS9?].

In the absence of such a universe, however, one can implement logical constructions by allowing the local universes to vary.

For instance, suppose we are given types A and B, and wish to construct their sum $A + B$. We do not expect that either of the families $E_A \longrightarrow V_A$, $E_B \longrightarrow V_B$ will include the sum types we need. However, the product $V_A \times V_B$ can be used to parametrize all possible sums of a type from V_A and a type from V_B. So we take this as the new local universe V_{A+B}, together with the family of such sums $E_A[\pi_1] + E_B[\pi_2]$ over it (a sum type over V_{A+B}) as E_{A+B}. Then for $\ulcorner A + B \urcorner$ we take $(\ulcorner A \urcorner, \ulcorner B \urcorner) : \Gamma \longrightarrow V_{A+B}$, picking out the specific sums required.

Substitution in $C_!$ corresponds to precomposition with $\ulcorner A \urcorner$, happening entirely to the left of Γ; and logical constructions are implemented entirely in terms of the local universes $E_A \longrightarrow V_A$, all on the right of Γ. There is no interference between these operations, so strict stability is obtained.

The stability conditions on C required to make this work then turn out to be quite weak, and quite categorically natural. Suppose we are considering a type-constructor, *widgets*, defined by some universal property. Then what we need to assume is that C has widgets whose pullbacks are again widgets—call such things *weakly stable widgets*. Given these, widgets in $C_!$ may be implemented by choosing weakly stable widgets over the local universes, secure in the knowledge that pulled back to the actual contexts involved, they will have the required universal

property. So, typically, if C has weakly stable widgets, then $C_!$ will have strictly stable widgets.

In case the universal property is strong enough to determine widgets up to isomorphism, weak stability is equivalent to a standard Beck-Chevalley condition; but of course, connectives in intensional type theory are often defined by rather weaker properties.

Note also that this stability condition is simply a matter of existence, and does not depend on any specific choices of structure in C.

One more ambient hypothesis is required, for the manipulation of local universes (for instance, the use of $V_A \times V_B$ above): we will require some products and exponentials, in the categorical sense (stronger than the type-theoretic sense), to exist in C.

Our main theorem is therefore:

Theorem. *Let C be a full comprehension category, whose underlying category has (a) finite products, and (b) exponentials along display maps into display maps and product projections (condition (LF) below).*

Then there is an equivalent full split comprehension category $C_!$; and if C has weakly stable binary sums (resp. Π-types, Σ-types, identity types, W-types, ...), then $C_!$ has strictly stable binary sums (Π-types, ...), and hence models the syntax of type theory with binary sums (Π-types, ...).

The paper is organized as follows.

First, in Section 2, we survey some background: the general setting of comprehension categories, some existing split replacement constructions, a range of stability conditions for logical structure, and existing results on when such structure lifts to split replacements.

In Section 3, we give our main construction, in several stages. We first set up the "local universes" split replacement $C_!$, and roughly preview how logical structure will lift to it. With this as motivation, we set up some technical tools for manipulating universes. Equipped with these, we are then ready for the full details of the construction. In Section 3.4, we set out for each logical constructor in turn the precise statement and construction of its lifting to $C_!$. Taken together, these constitute the main theorem.

Finally, in Section 4, we discuss how the construction may be generalized to further logical constructors, and conclude by listing some applications.

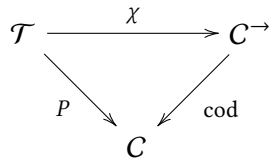
Acknowledgements. During the rather protracted preparation of this paper, the authors have benefited from the support of several institutions: the excellently supportive department at Dalhousie University, Halifax; the perhaps distractingly stimulating special year on Univalent Foundations at the Institute for Advanced Study, Princeton; and, most recently, the special trimester on Proofs and Programmes at the Institut Henri Poincaré, Paris.

We are also especially indebted to Pierre-Louis Curien, Thomas Streicher, Martin Hofmann, Steve Awodey, Mike Shulman, and Vladimir Voevodsky for detailed and useful discussions on the present work.

2. Setting

2.1. **Comprehension categories.** We present models of type theory via the formalism of comprehension categories. We recall here the key points of this approach; for more detailed background, see [Jac93].

Definition 2.1.1. A *comprehension category* consists of a category C together with a (cloven) Grothendieck fibration $P : \mathcal{T} \longrightarrow C$ and a functor $\chi : \mathcal{T} \longrightarrow C^{\to}$ (the *comprehension*), sending cartesian arrows to pullback squares, and such that



commutes (strictly). We say that a comprehension category is *split* if P is a split fibration, and *full* if χ is full and faithful.

In the present work (as in much of the literature), all comprehension categories are taken to be full.

Split comprehension categories are models of an essentially algebraic theory, so carry an evident notion of homomorphism, forming a category **SplCompCat**. Restricting to some fixed base category C (and morphisms acting as the identity on C) yields a subcategory $\mathbf{SplCompCat}(C)$.

Morphisms of non-split comprehension categories are subtler. For the present paper, we take such a morphism to consist of a functor $F : C' \longrightarrow C$ of base categories, and a cartesian functor $\bar{F} : \mathcal{T}' \longrightarrow \mathcal{T}$, strictly over F, and commuting strictly with the comprehension functors. We write **CompCat** for the resulting category; and, again, $\mathbf{CompCat}(C)$ for the category of comprehension categories on a fixed base C.

(The strict commutation with comprehension is, for many purposes, rather unnatural. We take it here just for simplicity, since we make little use of morphisms of general comprehension categories.)

(For readers more familiar with other algebraic approaches, note that full split comprehension categories are precisely equivalent to categories with attributes, categories with families, type-categories, and the like; contextual categories are also closely comparable to all of these, but not quite equivalent.)

Example 2.1.2. The canonical example is given by the syntax of type theory itself.

Take $\mathbb{T}$ to be any type theory with the judgement forms and structural rules of Martin-Löf type theory. Write $C_{\mathbb{T}}$ for the *category of contexts* of $\mathbb{T}$: objects are the contexts of $\mathbb{T}$, and arrows are substitutions between contexts, all up to judgemental equality. Over this, take $\mathcal{T}_{\mathbb{T}}$ to be the category of types-in-context of $\mathbb{T}$, with $p : \mathcal{T}_{\mathbb{T}} \longrightarrow C_{\mathbb{T}}$ sending a type-in-context to its context; so the fiber $\mathcal{T}_{\mathbb{T}}(\Gamma)$ is the category of types over Γ. Reindexing is given by substitution in types; since this is strictly functorial, it makes p into a split fibration. Finally, the comprehension

operation sends a type-in-context $\Gamma \vdash A$ to the context extension $\Gamma, x{:}A$ together with its dependent projection $\pi_{\Gamma,A} : (\Gamma, x{:}A) \longrightarrow \Gamma$.

Together, these form a (full) split comprehension category $C_{\mathbb{T}}$.

This example motivates much of the terminology and notation for general comprehension categories $(\mathcal{C}, \mathcal{T}, p, \chi)$.

The objects of the base category $\mathcal{C}$ are thought of as *contexts*; and given such an object Γ, we consider objects of the fiber $\mathcal{T}(\Gamma)$ as *types over* Γ.

Given a type $A \in \mathcal{T}(\Gamma)$, its comprehension $\chi(A)$ is an arrow with codomain Γ. We denote the domain of $\chi(A)$ by $\Gamma.A$; it may be seen as the context extension of Γ by some new variable of type A, with $\chi(A)$ the resulting dependent projection:

$$\begin{array}{c} \Gamma.A \\ \big\downarrow{\scriptstyle \chi(A)} \\ \Gamma. \end{array}$$

We refer to composites of such maps as *display maps*.

Next, given a map $\sigma : \Delta \longrightarrow \Gamma$ (which we think of as a context morphism, or substitution) and an object A of $\mathcal{T}(\Gamma)$, we write $\sigma_A : A[\sigma] \longrightarrow \Gamma$ for the lift provided by the cleaving of the fibration P. Diagrammatically:

$$\begin{array}{ccc} A[\sigma] & \xrightarrow{\sigma_A} & A \\ \vdots & & \vdots \\ \Delta & \xrightarrow{\sigma} & \Gamma \end{array} \qquad \begin{array}{c} \mathcal{T} \\ \big\downarrow{\scriptstyle P} \\ \mathcal{C}. \end{array}$$

The type $A[\sigma]$ may be seen as the result of applying the substitution σ to A. Following our notation for context extensions, we denote the comprehension $\chi(\sigma_A)$ by $\sigma.A : \Delta.A[\sigma] \longrightarrow \Gamma.A$.

Given our standing assumption of fullness, we will also sometimes silently conflate maps in fibers of $\mathcal{T}$ with maps in slices of $\mathcal{C}$.

In syntactic categories $C_{\mathbb{T}}$, terms of a type A in context Γ correspond to sections $t : \Gamma \longrightarrow \Gamma.A$ of the dependent projection $\Gamma.A$; such sections occur frequently when working with comprehension categories. We will therefore often write just "a section" to mean "a section of a dependent projection", unless specified otherwise.

$$\begin{array}{ccc} & & \Gamma.A \\ & \overset{a}{\nearrow} & \big\downarrow{\scriptstyle \chi(A)} \\ \Gamma & & \\ & \underset{1_\Gamma}{\searrow} & \Gamma \end{array}$$

Finally, for a map $\sigma : \Delta \longrightarrow \Gamma$, we extend the reindexing notation $A[\sigma]$ in several ways. Most straightforwardly, it denotes the reindexing functor $\mathcal{T}(\Gamma) \longrightarrow \mathcal{T}(\Delta)$,

with action on objects provided by the cleaving of $\mathcal{T}$. More generally, given a map $f : A \longrightarrow B$ in $\mathcal{T}(\Gamma)$, and *any* specified reindexings A', B' of A and B (precisely, cartesian lifts $\sigma_A : A' \longrightarrow A$, $\sigma_B : B' \longrightarrow B$ of σ), we write $f[\sigma]$ for the induced map $A' \to B'$ in $\mathcal{T}(\Delta)$. Finally, we write $(-)[\sigma]$ also for the induced pullback functor $\mathcal{C}//\Gamma \longrightarrow \mathcal{C}//\Delta$, where $\mathcal{C}//\Gamma$ denotes the full subcategory of $\mathcal{C}/\Gamma$ with just display maps as objects.

All the above notation and terminology suggests that the syntactic comprehension categories $\mathcal{C}_{\mathbb{T}}$ may be seen as typical. This is indeed the case; specifically, they turn out to enjoy certain universal properties, arguably the *raison d'être* of comprehension categories.

Starting with $\mathbb{T}_\emptyset$, the theory given by just the structural rules, one has:

Proposition 2.1.3 (Cartmell [Car86, §15])**.** *$\mathcal{C}_{\mathbb{T}_\emptyset}$ is the initial split comprehension category.*

In other words, any split comprehension category $\mathcal{C}$ admits a canonical map $[\![-]\!] : \mathcal{C}_{\mathbb{T}_\emptyset} \longrightarrow \mathcal{C}$; that is, an interpretation of the syntax of $\mathbb{T}_\emptyset$ in $\mathcal{C}$. This justifies taking split comprehension categories as a definition of *models of* $\mathbb{T}_\emptyset$.

Extending this correspondence to less trivial type theories, one considers comprehension categories with extra structure. Take, for instance, the theory $\mathbb{T}_\otimes$ given by the structural rules together with a single type-forming rule:

$$\frac{\Gamma \vdash A \text{ type} \qquad \Gamma \vdash B \text{ type}}{\Gamma \vdash A \otimes B \text{ type}}$$

Say that *(strictly stable)* $\otimes$*-structure* on a comprehension category $\mathcal{C}$ consists of an operation giving, for any objects $\Gamma \in \mathcal{C}$ and $A, B \in \mathcal{T}(\Gamma)$, an object $A \otimes_\Gamma B \in \mathcal{T}(\Gamma)$, strictly stable under reindexing, i.e. such that for any such Γ, A, B and map $\sigma : \Gamma' \longrightarrow \Gamma$, we have $(A \otimes_\Gamma B)[\sigma] = A[\sigma] \otimes_{\Gamma'} B[\sigma]$. Then $\mathcal{C}_{\mathbb{T}_\otimes}$ carries an evident $\otimes$-structure (with its strict stability arising inevitably from the inductive definition of substitution, $(A \otimes B)[\sigma] := A[\sigma] \otimes B[\sigma]$); and indeed, we have:

Proposition 2.1.4. *The syntactic category $\mathcal{C}_{\mathbb{T}_\otimes}$ is initial in the category of split comprehension categories with $\otimes$-structure, and functors strictly preserving the comprehension functor and $\otimes$-structure.*

In other words, any split comprehension category $\mathcal{C}$ with strictly stable $\otimes$-structure carries a canonical interpretation $[\![-]\!] : \mathcal{C}_{\mathbb{T}_\otimes} \longrightarrow \mathcal{C}$ of the syntax of $\mathbb{T}_\otimes$. This justifies once again taking such comprehension categories as an algebraic definition of *models of* $\mathbb{T}_\otimes$.

Similarly, this correspondence extends to all the other usual constructors and rules. Each new constructor corresponds to an extra operation, and each new judgemental equality rule to an algebraic axiom. The syntactic category $\mathcal{C}_{\mathbb{T}}$ is then initial among split comprehension categories with the appropriate strictly stable structure. See [Str91, Ch. 3, p. 181] for a detailed treatment of the case of the Calculus of Constructions.

The task of modelling type theory thus amounts to the construction of split comprehension categories with suitable strictly stable algebraic structure.

Unfortunately, many constructions of models do not directly give this: they give comprehension categories with the appropriate algebraic structure, but they are not split, and the operations are not strictly stable. This occurs particularly in abstract categorical constructions, such as the model of extensional type theory (ETT) in an arbitrary LCCC [See84, Hof95], or of intensional type theory (ITT) using a suitable weak factorization system [AW09]. In such models, $\mathcal{T}(\Gamma)$ usually consists of all maps into Γ satisfying some property, or carrying some extra structure. Reindexing is then given by pullback; but this is defined only up to canonical isomorphism, and so is only functorial up to isomorphism, not on the nose. By the same token, the logical operations are typically characterized at most up to canonical isomorphism (and in some homotopy-theoretic models, only up to homotopy equivalence), and so are not automatically strictly stable.

In concrete models, a solution can often be found: an equivalent comprehension category, split, and with some strictly stable choice of structure. For the model in **Sets**, for instance, one takes $\mathcal{T}(\Gamma)$ not as $\mathbf{Sets}/\Gamma$, but as the equivalent category $\mathbf{Sets}^{\Gamma}$. Related solutions exist for the (higher) groupoid models [HS98, War08, War11], presheaf models of ETT, and various other concrete examples.

However, for general categorical constructions, and some homotopy-theoretic models (e.g. **SSets**), the problem remains: when can one replace a comprehension category, carrying some kind of logical structure, by an equivalent split one with strictly stable structure? This is the *coherence problem* for dependent type theory.

In the remainder of this section, we survey existing results and lay out the setting for our own, taking the problem in two steps: first the splitness, then the strict stability.

2.2. **Split replacements.** The split replacement constructions for fibrations are classical, due to Giraud [Gir65, I, 2.4.3] and Bénabou [Str14]. Let $\mathbf{Fib}(C)$ denote the (1-)category of (cloven) fibrations and cartesian functors (not assumed to preserve the cleaving) over a fixed base C, and similarly $\mathbf{SplFib}_{\mathrm{cl}}(C)$ the category of split fibrations over C, with morphisms *cloven* functors, i.e. preserving the splitting on the nose. Then the inclusion functor $i : \mathbf{SplFib}_{\mathrm{cl}}(C) \longrightarrow \mathbf{Fib}(C)$ possesses both left and right adjoints:

$$\mathbf{SplFib}_{\mathrm{cl}}(C) \quad \begin{array}{c} \uparrow \\ (-)_! \dashv \downarrow \dashv (-)_* \\ \uparrow \end{array} \quad \mathbf{Fib}(C).$$

Explicitly, let $p : \mathcal{T} \longrightarrow C$ be a cloven fibration. An object of $\mathcal{T}_*$ over $\Gamma \in C$ consists of an object A of $\mathcal{T}(\Gamma)$, together with for each $f : \Gamma' \longrightarrow \Gamma$ some cartesian lifting $\overline{f} : A_f \longrightarrow A$, such that $A_{1_\Gamma} = A$, $\overline{1_\Gamma} = 1_A$. On the other hand, an object of $\mathcal{T}_!$ over Γ consists of objects $V \in C$, $E \in \mathcal{T}(V)$, and a map $f : \Gamma \longrightarrow V$ (as discussed in detail in Section 3.1 below).

The unit and counit maps of these adjunctions are not in general isomorphisms. However, considering $\mathbf{Fib}(C)$ as a 2-category (with 2-cells natural transformations over C), they are equivalences.

In other words, every fibration over a given base has two canonical split replacements.

Moreover, much of this situation lifts from fibrations to comprehension categories.

For the right adjoint $(-)_*$, this is described in [Hof95, CGH14]. Given a comprehension category $\mathcal{C} = (C, \mathcal{T}, p, \chi)$, the comprehension χ_* on $\mathcal{T}_*$ is given by sending an object-with-chosen-reindexings simply to the comprehension in C of the object itself.

Similarly, the left adjoint lifts to a functor

$$(-)_! : \mathbf{CompCat}(C) \longrightarrow \mathbf{SplCompCat}(C),$$

described in full in Section 3.1 below, with the comprehension on $\mathcal{T}_!$ sendings an object $f : \Gamma \twoheadrightarrow V, A \in \mathcal{T}(V)$ to $\chi(A[f])$. This is no longer a left adjoint, however: the putative unit map $(C, \mathcal{T}, p, \chi) \longrightarrow (C, \mathcal{T}_!, p_!, \chi_!)$ will strictly preserve the comprehension just when p is a *normal* fibration, i.e. when the cleaving lifts identity maps to identities.

We retain, however, the fact that the maps $C_* \longrightarrow C$ and $C \longrightarrow C_!$ are equivalences in suitable 2-categories.

In sum, we find that simply for bare comprehension categories, the coherence problem is satisfactorily solved: $(-)_*$ and $(-)_!$ provide two ways to replace an arbitrary comprehension category by an equivalent split one.

2.3. **Stability conditions.** The real fun starts when one wishes to model a non-trivial type theory; that is, when one has some logical structure on the original comprehension category, and wishes to lift it to strictly stable structure on a split replacement.

It is reasonable to expect that *some* kind of stability condition will be needed for the operations of the original category. We set out here a range of possible such conditions, from stronger ones which will lift more easily, to weaker ones which are satisfied more often in nature.

We do not give general definitions of them, for arbitrary logical operations: no appropriate generality of operations exists in the literature, and giving one is beyond the scope of this paper. Instead, we define them here in the illustrative case of identity types; then in Section 3.4 below, we define them for other specific type-constructors as we require them.

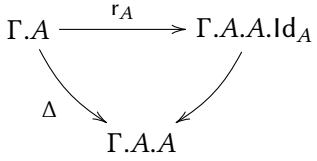
Fix, for the remainder of this section, a comprehension category $\mathcal{C} = (C, \mathcal{T}, p, \chi)$.

Definition 2.3.1. Given objects $\Gamma \in C$ and $A \in \mathcal{T}(\Gamma)$, an *identity type for* A consists of:

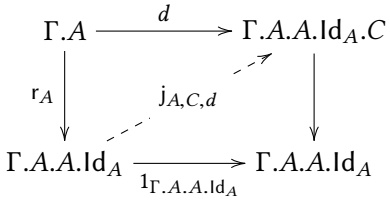
- a type $\mathsf{Id}_A \in \mathcal{T}(\Gamma.A.A)$[1];

[1]Strictly, $\mathcal{T}(\Gamma.A.A[\chi(A)])$; here and elsewhere, we suppress such "weakenings", i.e. reindexings along dependent projections, where they are unambiguous.

- a factorization



  of the diagonal map $\Delta_A : \Gamma.A \longrightarrow \Gamma.A.A$;
- for each $C \in \mathcal{T}(\Gamma.A.A.\mathsf{Id}_A)$ and $d : \Gamma.A \longrightarrow \Gamma.A.A.\mathsf{Id}_A.C$ such that the outer square of



  commutes, a diagonal filler $\mathsf{j}_{A,C,d} : \Gamma.A.A.\mathsf{Id}_A \longrightarrow C$ as indicated, making the resulting triangles commute.

A *choice of identity types on C* is a function giving, for each appropriate Γ, A in C, an identity type $(\mathsf{Id}_A, \mathsf{r}_A, \mathsf{j}_A)$ for A.

To give a direct model of syntactic identity types, one requires stability conditions corresponding to the recursive definition of syntactic substitution:

Definition 2.3.2. A choice of identity types on C is *strictly stable* if for each $\sigma : \Gamma' \longrightarrow \Gamma$, and all appropriate A, C, d,

$$\mathsf{Id}_A[\sigma] = \mathsf{Id}_{A[\sigma]}$$
$$\mathsf{r}_A[\sigma] = \mathsf{r}_{A[\sigma]}$$
$$\mathsf{j}_{A,C,d}[\sigma] = \mathsf{j}_{A[\sigma],C[\sigma],d[\sigma]}.$$

The most problematic aspect of this definition, categorically, is that it requires an on-the-nose equality of types. Also, it does not necessarily respect isomorphism of types. Making the minimal modification to allay these objections, we obtain:

Definition 2.3.3. A choice of identity types on C is *(fully) pseudo-stable* if it is equipped with a cartesian functorial action on cartesian maps. That is, for each $\sigma : \Gamma' \longrightarrow \Gamma$, and cartesian map $\sigma_A : A' \longrightarrow A$ over σ, a cartesian map

$$\mathsf{Id}_{\sigma_A} : \mathsf{Id}_{A'} \longrightarrow \mathsf{Id}_A$$

over $\sigma.\sigma_A.\sigma_A : \Gamma'.A'.A' \longrightarrow \Gamma.A.A$, such that $\mathsf{Id}_{1_A} = 1_{\mathsf{Id}_A}$, $\mathsf{Id}_{\tau_A \circ \sigma_A} = \mathsf{Id}_{\tau_A} \circ \mathsf{Id}_{\sigma_A}$, and moreover commuting appropriately with values of r and j.

(Full details of the "commuting appropriately" are spelled out in [Waro8, Def. 2.38], where these are called *coherent identity types*; we omit them here, since our main results do not involve this condition.)

This action may be seen as combining two parts: comparison isomorphisms $\mathsf{Id}_{A[\sigma]} \cong \mathsf{Id}_A[\sigma]$ for the reindexings given by the cleaving, and functoriality in isomorphisms $A \cong B$ within each fiber $\mathcal{T}(\Gamma)$. An earlier version of this paper incorrectly defined pseudo-stability using just the first of these components.

In homotopy-theoretic models, the fillers j are usually not specified, but given merely by an existence condition. This suggests the further weakening:

Definition 2.3.4. A choice of partly-specified identity types (i.e. operations giving chosen Id_A, r_A, such that there exist elimination fillers j making these identity types), is *partially pseudo-stable* if it is equipped with a cartesian action of Id on cartesian maps (as above), commuting with values r (but not necessarily j).

(Again, see [War08, Def. 2.33] for full details; these are the *stable identity types* there.)

A more categorically familiar property is the *Beck–Chevalley* condition:

Definition 2.3.5. Say a choice of identity types on $\mathcal{C}$ satisfies the *Beck–Chevalley condition* if for each $\sigma : \Gamma' \to \Gamma$ and $A \in \mathcal{T}(\Gamma)$, the canonical map

$$\mathsf{j}_{A[\sigma],\mathsf{Id}_A[\sigma],\mathsf{r}_A[\sigma]} : \mathsf{Id}_{A[\sigma]} \to \mathsf{Id}_A[\sigma]$$

is an isomorphism. This depends on these particular fillers, but not on other values of j.

Our final condition is a pure existence condition on $\mathcal{C}$, not dependent at all on a choice of identity types:

Definition 2.3.6. Given Γ in $\mathcal{C}$ and A in $\mathcal{T}(\Gamma)$, a *weakly stable identity type* for A is a pair (Id,r) as above such that, for all $\sigma : \Gamma' \to \Gamma$, there is some j making $(\mathsf{Id}_A[\sigma],\mathsf{r}_A[\sigma],\mathsf{j})$ an identity type for $A[\sigma]$.

Say that $\mathcal{C}$ *has weakly stable identity types* if for every Γ, A, there is some weakly stable identity type (Id,r).

The conditions above may be usefully compared in terms of their impliciations for the Beck–Chevalley maps $\mathsf{j}_{A[\sigma],\mathsf{Id}_A[\sigma],\mathsf{r}_A[\sigma]} : \mathsf{Id}_{A[\sigma]} \to \mathsf{Id}_A[\sigma]$, and for the stability of values of j modulo these maps.

choice of Id-types	(Id,r)	j
strictly stable	$=$	$=$
pseudo-stable	$\cong$	$=$
partially pseudo-stable	$\cong$	$\sim$
weakly stable	$\simeq$	$\sim$
arbitrary		

Here $\sim$ denotes homotopy (pointwise propositional equality), and $\simeq$ homotopy equivalence, in the sense of [Uni13, Ch. 4].

Analogous definitions of these levels of stability may be made for the other usual type and term constructors; as mentioned already, we will define these in full as they are required, in Sec. 3.4 below. Briefly, they are obtained for inductive types (+, Σ, 1, etc.) by replacing (Id,r) in the definitions above by the type former in question and its constructors, and replacing j by the eliminator; and for

Π-types, by replacing $(\mathsf{Id}, \mathsf{r})$ by the Π-type and its application map, and j by the λ-abstraction operation.

Existence conditions and the axiom of choice. The various existence conditions above—in particular, weak stability—may each be interpreted in two ways: classically, as mere existence, or according to the constructive tradition, with each forall–exists statement witnessed by some function (but with no conditions on the function assumed).

Assuming the axiom of choice, the two are of course equivalent. In the absence of AC, however, the witnessed form is stronger, and is the form required for the results of Section 3 below. (Compare the use of *cloven* fibrations in the definition of comprehension categories.) We will for the most part elide this distinction; where necessary, we will speak of *witnessed weakly stable identity types*, and the like.

2.4. **Lifting logical structure.** Equipped with these definitions, we can now state when logical structure lifts from C to its strict replacements C_*, $C_!$.

For the right-handed strictification C_*, the known results require either restrictions on the type theory in question, or strong stability conditions.

Theorem 2.4.1 (Hofmann [Hof95, Thm. 2,4], analysed further in [CGH14])**.** *Suppose C is a comprehension category, equipped with structure corresponding to the logical constructions of* extensional *Martin-Löf type theory, including in particular the reflection rule for identity types*[2]*, and all satisfying the appropriate Beck–Chevalley conditions.*

Then C_* may be equipped with strictly stable Π*-structure,* Σ*-structure, etc.*

In particular, if $\mathcal{E}$ is a locally cartesian closed category, then $(\mathcal{E}, \mathcal{E}^{\to}, \mathrm{cod}, 1)_*$ is a model of extensional type theory.

With the terminology above, this may be read as factoring into two lemmas. The first characterizes when logical structure (not just that corresponding to ETT) lifts to C_*.

Lemma 2.4.2. *Suppose C is a comprehension category, equipped with pseudo-stable* Id*-types (resp.* Π*-types, +-types, etc.).*

Then C_* carries strictly stable Id*-types (*Π*-types, +-types, etc.).*

Proof. Just as in the proof of [Hof95, Thm. 2] □

The second shows why for extensional type theory, and similar theories, pseudo-stability is a very reasonable condition to expect.

Lemma 2.4.3. *Suppose C is a comprehension category, with identity types satisfying the reflection rule, and with* Π*-types (resp. +-types, etc.), all weakly stable (or,* a fortiori*, satisfying the Beck–Chevalley condition), and in the case of* Π*-types, satisfying the η-rule.*

[2] That is, for each $A \in \mathcal{T}(\Gamma)$, there is a type Id_A over $\Gamma.A.A$ whose comprehension is isomorphic over $\Gamma.A.A$ to the diagonal map $\Gamma.A \twoheadrightarrow \Gamma.A.A$.

Then all this structure is in fact pseudo-stable, with comparison isomorphisms corresponding to the Beck–Chevalley maps.

Proof. Using the identity types with the reflection rule, one may show that the maps produced by eliminators of inductive types (e.g. j), or by λ-abstraction for Π-types with η, are unique; e.g. given identity-elimination data Γ, A, C, d, and an Id-type $(\mathsf{Id}_A, \mathsf{r}_A, \mathsf{j}_A)$ for A, then any section f of $\chi(C)$ with $f \circ \mathsf{r}_A = d$ must be equal to $\mathsf{j}_{A,C,d}$. From this, it follows that the various type formers have categorical universal properties, either as certain initial algebras or as exponential objects. Weak stability then implies that the Beck–Chevalley maps, as algebra maps between two initial/terminal objects, are isomorphisms, and satisfy the appropriate axioms to witness that the logical structure is pseudo-stable. □

For theories such as intensional Martin-Löf type theory, however, pseudo-stability is often difficult to obtain.

Most obviously, type constructors such as Id-types, or sum types without η-rules, are not automatically determined up to isomorphism, only up to equivalence. This is not often an obstacle in practice, though, since the specific constructions used usually are stable up to coherent isomorphism after all.

More problematically, however, the Id-elimination operation (and some other inductive eliminators) is not canonically determined, but merely given by existence conditions, so does not commute with substitution and the coherence isomorphisms; and even in concrete cases (e.g. **SSets**), it may not be clear how to make choices that do [Waro8, §2.3].

In the terminology from above, identity types in comprehension categories coming from homotopy-theoretic models are usually partially pseudo-stable, but are often not fully so. It is not possible, therefore, to apply Lemma 2.4.2 to obtain strictifications of such models.

Theorem 3.4.1 and Heuristic 4.1.1 below resolve this situation, stating that for lifting logical structure to the *left-handed* strictification $C_!$, only weak stability is required, provided that certain products and exponentials exist in the base category C.

The next section is devoted to the full statement and proof of this theorem.

3. The local universes model

3.1. **The comprehension category $C_!$.** Throughout this section, assume given a fixed comprehension category $C = (C, \mathcal{T}, p, \chi)$.

Definition 3.1.1. Define the new comprehension category $C_! = (C_!, \mathcal{T}_!, p, \chi_!)$ as follows:

Base category: We set $C_! := C$; the base category does not change.
Types: An object of $\mathcal{T}_!$ over $\Gamma \in C$ consists of a tuple $(V_A, E_A, \ulcorner A \urcorner)$, where V_A is an object of C, E_A an object of $\mathcal{T}(V_A)$, and $\ulcorner A \urcorner$ an arrow $\Gamma \longrightarrow V_A$ in C.

One may view this diagrammatically as follows:

$$\begin{array}{ccc} & & E_A \\ & & \vdots \\ \Gamma & \xrightarrow{\ulcorner A \urcorner} & V_A. \end{array}$$

Given such an object, write $[A]$ for its reindexing $E_A[\ulcorner A \urcorner]$ in $\mathcal{T}(\Gamma)$.

An arrow $(V_B, E_B, \ulcorner B \urcorner) \longrightarrow (V_A, E_A, \ulcorner A \urcorner)$ in $\mathcal{T}_!$ over $\sigma : \Delta \longrightarrow \Gamma$ is just a map $[B] \longrightarrow [A]$ over σ in $\mathcal{T}$:

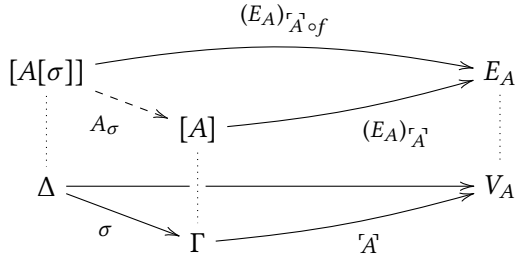
$$\begin{array}{ccc} [B] & \longrightarrow & [A] \\ \vdots & & \vdots \\ \Delta & \xrightarrow{\sigma} & \Gamma \end{array}$$

Together, this gives the category $\mathcal{T}_!$ together with a projection $p_! : \mathcal{T}_! \longrightarrow C_!$.

Reindexing: Cartesian lifts for $p_!$ are given as follows. Let $\sigma : \Delta \longrightarrow \Gamma$ be an arrow in C, and $A = (V_A, E_A, \ulcorner A \urcorner)$ an object of $\mathcal{T}_!(\Gamma)$. Set

$$A[\sigma] := (V_A, E_A, \ulcorner A \urcorner \circ \sigma)$$

and take the map $A_\sigma : A[\sigma] \longrightarrow A$ over σ to be the canonical map $[A[\sigma]] \longrightarrow [A]$ over σ in $\mathcal{T}$ given by the cartesianness of $(E_A)_{\ulcorner A \urcorner}$ for p:



This is straightforwardly seen to make $p_!$ a *split* fibration.

Comprehension: Given an object $(V_A, E_A, \ulcorner A \urcorner)$ of $\mathcal{T}_!(\Gamma)$, take

$$\chi_!(V_A, E_A, \ulcorner A \urcorner) := \chi([A]).$$

Cartesianness of the functor $\chi_!$ follows directly from that of χ.

Intuitively, we think of V_A together with E_A as a kind of "local universe". (By abuse of notation, we often refer to the pair (V_A, E_A) just as V_A, leaving E_A understood.) Following this intuition, the map $\ulcorner A \urcorner : \Gamma \longrightarrow V_A$ picks out the actual type family $[A]$ from the local universe V_A.

In general, $C_!$ may not support the interpretation of any interesting constructors, even when C does. However, provided that the underlying category C comes equipped with a modest amount of additional structure it will be possible to lift

the interpretations of various constructors from C to $C_!$. We first recall some definitions.

Definition 3.1.2. If $Z \xrightarrow{g} Y \xrightarrow{f} X$ are maps in a category C such that all pullbacks of f exist, a *(categorical, dependent) exponential* for f and g is an object $\prod[f,g]$ of C/X together with a natural isomorphism

$$C/X(W, \textstyle\prod[f,g]) \cong C/Y(W \times_X Y, Z),$$

for all $h : W \longrightarrow X$.

A map $f : Y \longrightarrow X$ possessing all pullbacks is *(dependently) exponentiable* if for every $g : Z \longrightarrow Y$, some dependent product $\prod[f,g]$ exists; equivalently, if the pullback functor $f^* : C/X \longrightarrow C/Y$ has a right adjoint.

Then the precise ambient hypothesis required, for lifting structure to $C_!$, is as follows (named by analogy with the logical framework presentation):

Definition 3.1.3. Say that C satisfies *condition (LF)* if its underlying category has finite products, and given maps $Z \xrightarrow{g} Y \xrightarrow{f} X$, if f is a display map and g is either a display map or a product projection, then a dependent exponential $\prod[f,g]$ exists.

Proposition 3.1.4. *Each of the following (simpler) conditions implies (LF):*

- ***(LFa)**: C has finite products; and every display map is exponentiable.*
- ***(LFb)**: Every map $X \longrightarrow 1$ is a display map; and for each $A \in \mathcal{T}(\Gamma)$, the reindexing functor $\chi(A)^* : \mathcal{T}(\Gamma) \longrightarrow \mathcal{T}(\Gamma.A)$ has a right adjoint.*
- ***(LCCC)**: C is locally Cartesian closed.* □

The exponentials required by (LF) can be essentially independent of any function types one may consider in the type theory. On the one hand, they are not required to themselves be display maps. On the other, they *are* required to be categorical exponentials, not merely type-theoretic function types (in general slightly weaker). Compare how in logical framework presentations of the type theory, one usually asks for strong function types in the ambient logical framework, independently of what function types the object theory may possess [NPS90, Ch. 19].

Given this assumption, all standard type-constructors will lift from C to $C_!$ essentially independently. We consider them one by one in Sections 3.4.1–3.4.4, which all have roughly the same form: we define precisely what it means for a comprehension category to have weakly stable widgets, and to have a strictly stable choice of widgets; and we show that if C has weakly stable widgets and satisfies (LF), then $C_!$ has a strictly stable choice of widgets, and hence models type theory with widgets. Before embarking on this, however, we first set out the general template for the construction of the structure on $C_!$, and set up some machinery that it requires.

We assume throughout the following discussion that C satisfies condition (LF); however, we explicitly re-state this hypothesis in all theorems, as required.

3.2. **Template for structure on $C_!$.** To illustrate the pattern we will follow, first consider the operation corresponding to the (+)-formation rule:

$$\frac{\Gamma \vdash A \ \mathsf{type} \qquad \Gamma \vdash B \ \mathsf{type}}{\Gamma \vdash A + B \ \mathsf{type}}$$

Assuming that C is equipped with such an operation, we wish to lift it to $C_!$.

The result should send any object $\Gamma \in C_!$ and pair of types $A_1, A_2 \in \mathcal{T}_!(\Gamma)$ to some type $A_1 + A_2 \in \mathcal{T}_!(\Gamma)$. These inputs correspond, in C, to objects $\Gamma, V_{A_i} \in C$, types $E_{A_i} \in \mathcal{T}(V_{A_i})$, and maps $\ulcorner A_i \urcorner : \Gamma \longrightarrow V_{A_i}$.

One cannot directly take the sum $E_{A_1} + E_{A_2}$ in C, since these live in different fibers of $\mathcal{T}$. One could pull both E_{A_1}, E_{A_2} back to $\mathcal{T}(\Gamma)$ and take their sum there, but since this involves Γ, the resulting operation would not be strictly stable unless (+) were already so in C.

Instead, note that taken together, the maps $\ulcorner A_i \urcorner$ correspond to the single map $(\ulcorner A_1 \urcorner, \ulcorner A_2 \urcorner) : \Gamma \longrightarrow V_{A_1} \times V_{A_2}$, and factor through it via the projections π_1, π_2. We can thus pull both E_{A_1}, E_{A_2} back to $\mathcal{T}(V_{A_1} \times V_{A_2})$, and take their sum there. Putting this together, we set $V_{A_1+A_2} := V_{A_1} \times V_{A_2}$, $E_{A_1+A_2} := E_{A_1}[\pi_2] + E_{A_2}[\pi_1]$, and $\ulcorner A_1 + A_2 \urcorner := (\ulcorner A_1 \urcorner, \ulcorner A_2 \urcorner)$.

This is strictly stable in Γ, since the only part involving Γ is the definition of $\ulcorner A_1 + A_2 \urcorner$, which (as an element of a *set*, not an object of a category) is, as one would hope, strictly natural in Γ.

The key point is that once the local universes V_{A_i}, E_{A_i} are chosen, the object $V_{A_1} \times V_{A_2}$ *represents* the premises of (+)-form: instances of the premises over a given Γ, with these universes, correspond to maps $\Gamma \longrightarrow V_{A_1} \times V_{A_2}$. One may see $V_{A_1} \times V_{A_2}$ as parametrizing all possible sums of a type from V_{A_1} and a type from V_{A_2}.

This pattern will be followed for all rules and constructors. Given universes for all type premises of the rule, we construct an object V representing the rest of the data of the premises.

A specific instance of the premises over some context Γ then corresponds to a map $\Gamma \longrightarrow V$. In particular, there is a universal instance over V itself. To perform the operation on a particular instance, we first apply the operations of C to this universal instance over V. (Here, we may rely on weak stability of structure in C to know that reindexed from their own universes to V, all types involved retain any universal properties required.)

In the case of a type constructor, we are now done, using V as the local universe of the new type, and the map $\Gamma \longrightarrow V$ corresponding to the rest of the data as the new name map. In the case of a term constructor, we further need to actually perform the reindexing from $\mathcal{T}(V)$ to $\mathcal{T}(\Gamma)$ to obtain an appropriate map in $C_!$.

In either case, the constructions depend on Γ only via operations that are strictly natural in Γ: use of the universal property of V, and reindexing of *maps* (not objects!) between fibers of $\mathcal{T}$. They are thus strictly stable in Γ, regardless of the stability of the structure in C.

3.3. **Manipulating local universes.** In the example above, it was straightforward to construct the representing object $V_A \times V_B$ for the premises of the rule in question. For more complex rules, however, construction of this representing object—the new universe—may be rather more involved. Indeed, this is the main technical work of the proof.

We establish here some tools for constructing such objects, beginning with one construction, in particular, which recurs for several different logical constructors. The operations corresponding to Π-formation, Σ-formation, and W-formation all take as input an object Γ, a type A over Γ, and a dependent type B over $\Gamma.A$.

These data in $\mathcal{C}_!$, over a given object Γ, consist of $A = (V_A, E_A, \ulcorner A \urcorner)$ in $\mathcal{T}_!(\Gamma)$ and $B = (V_B, E_B, \ulcorner B \urcorner)$ in $\mathcal{T}_!(\Gamma.A)$, which amount in $\mathcal{C}$ to the following configuration:

$$\begin{array}{ccccc} & & & & E_B \\ & & \overset{\ulcorner B \urcorner}{\longrightarrow} & & \vdots \\ \Gamma.A & & & V_B & \quad E_A \\ \downarrow & & & & \vdots \\ \Gamma & & \overset{\ulcorner A \urcorner}{\longrightarrow} & & V_A \end{array}$$

(Here $\Gamma.A \longrightarrow A$ is the comprehension $\chi_!(A)$, taken in $\mathcal{C}_!$; in terms of $\mathcal{C}$, this is $\chi([A]) : \Gamma.[A] \longrightarrow \Gamma$.)

Now, fixing (V_A, E_A) and (V_B, E_B), pairs of maps $\ulcorner A \urcorner$, $\ulcorner B \urcorner$ as above correspond by adjunction to maps from Γ into the object

$$(V_A, E_A) \triangleleft (V_B, E_B) := \sum_{V_A} \prod_{E_A} \begin{pmatrix} (V_A.E_A) \times V_B \\ \downarrow \\ V_A.E_A \end{pmatrix},$$

where $\prod_{E_A} : \mathcal{C}/V_A.E_A \longrightarrow \mathcal{C}/V_A$ denotes the right adjoint to $\chi(E_A)^*$, which exists by hypothesis (LF), and $\sum_{V_A} : \mathcal{C}/V_A \longrightarrow \mathcal{C}$ simply sends a map to its codomain. (By the usual abuse of notation, we will often denote this object simply by $V_A \triangleleft V_B$.) Moreover, this correspondence is natural in Γ.

In particular, the identity map of $V_A \triangleleft V_B$ corresponds to the universal such pair of maps, which we denote by

$$\pi_A : V_A \triangleleft V_B \longrightarrow V_A, \qquad \pi_B : (V_A \triangleleft V_B).(E_A[\pi_A]) \longrightarrow V_B.$$

$V_A \triangleleft V_B$ may thus be considered as the object of "families of types in V_B, indexed by a type in V_A".

The definition of $V_A \triangleleft V_B$ may alternatively be presented in a type-theoretic internal language for $\mathcal{C}$: not the arbitrary type theory that we are trying to model, but a specific type theory with just Π-types, satisfying the judgemental η-rule, to handle the substitution and exponentiation in $\mathcal{C}$ and its slices. In this language, the definition becomes:

$$V_A \triangleleft V_B := [a{:}V_A,\ b{:}{V_B}^{E_A(a)}].$$

For complex constructions, this notation scales somewhat more readably than using categorical combinators. For instance, in the operation corresponding to the (+)-elimination rule,

$$\frac{\Gamma \vdash A, B \text{ type} \qquad \Gamma, w : A + B \vdash C(w) \text{ type} \qquad \Gamma, x{:}A \vdash d_1 : C(\mathsf{inl}(x)) \qquad \Gamma, y{:}B \vdash d_2 : C(\mathsf{inr}(y))}{\Gamma, w : A + B \vdash \langle x.d_1, y.d_2 \rangle(w) : C(w)}$$

once universes V_A, V_B, V_C are chosen, the representing object for the premises is given by

$$[a{:}V_A,\ b{:}V_B,\ c{:}{V_C}^{E_A(a)+E_B(b)},\ d_1{:}\Pi_{x:E_A(a)}E_C(c(\nu_1(x))),\ d_2{:}\Pi_{y:E_B(b)}E_C(c(\nu_2(y)))]$$

which in categorical combinators is

$$\Sigma_{V_A}\Sigma_{\Delta_{V_A}V_B}\Sigma_{(\Delta_{\Delta_{V_A}V_B}V_C)^{(E_A[\pi_1]+E_B[\pi_2])}} \Big(\Pi_{E_A[\pi_1\circ\pi_1]}E_C[\mathrm{ev}\circ\nu_1[\pi_1]] \times \Pi_{E_B[\pi_2\circ\pi_1]}E_C[\mathrm{ev}\circ\nu_2[\pi_1]]\Big).$$

For this reason, we use the internal language to present such objects below. Unfortunately, this is somewhat laborious to formally justify. Since we do not restrict the local universes V_A to "fibrant objects" (i.e. with $V_A \twoheadrightarrow 1$ a display map), nor assume that exponentiation preserves display maps, we need "types" of this internal language to include arbitrary maps of C, or at least something more general than the types of $\mathcal{T}$. Π-types between them may therefore not always be defined; so we cannot take the language to be the Π-fragment of ETT, and thus cannot quite apply Theorem 2.4.1 to justify its interpretation.

To thoroughly address this question, one could consider type theory with an extra judgement "$\Gamma \vdash A$ fib" added to the syntax (cf. Voevodsky's system HTS [Voe13]), and with Π-formation restricted to the case where the domain is given by this judgement. Correspondingly, one would consider comprehension categories equipped with a subfibration $\mathcal{F} \subseteq \mathcal{T}$, and extend Theorem 2.4.1 to this setting.

For the present purposes, however, it is simpler to regard the internal language merely as a notational shorthand, since we do not require the full interpretation function, but only finitely many instances of it, which the scrupulous reader may unwind into the algebraic language of products, pullbacks, and exponentials as required.

3.4. **Logical structure on $C_!$.** With the machinery set up, we are now ready to lift logical structure from C to $C_!$, one constructor at a time.

Taken together, the following lemmata constitute our main result:

Theorem 3.4.1. *Let C be a full comprehension category satisfying condition (LF).*

If C has weakly stable binary sums (resp. Π-types, identity types, Σ-types, zero types, unit types, or W-types relative to a stable class of Π-types), then its split replacement $C_!$ has strictly stable binary sums (Π-types, …), and hence models the syntax of type theory with binary sums (Π-types, …).

See also Section 4.1 for a discussion of how this extends to other rules and constructors.

3.4.1. **Binary sums.** First, we return in full to the case of binary sums. (Note that we consider general type-theoretic (weak) binary sums, not necessarily assumed to be categorical coproducts.)

Definition 3.4.1.1. Given a comprehension category C, an object $\Gamma \in C$, and types $A_1, A_2 \in \mathcal{T}(\Gamma)$, a *binary sum* for A_1 and A_2 consists of:

- a type $A_1 + A_2 \in \mathcal{T}(\Gamma)$;
- maps $\nu_i : \Gamma.A_i \longrightarrow \Gamma.A_1 + A_2$ in C over Γ (for $i = 1, 2$) (the *sum inclusions*);
- such that for any type $C \in \mathcal{T}(\Gamma.(A_1 + A_2))$ and sections $t_i : \Gamma.A_i \longrightarrow \Gamma.A_i.C[\nu_i]$, there is some section $\langle t_1, t_2 \rangle : \Gamma.(A_1 + A_2) \longrightarrow \Gamma.(A_1 + A_2).C$, such that $\langle t_1, t_2 \rangle \circ \nu_i = \nu_i.C \circ t_i$.

Definition 3.4.1.2. A comprehension category C has *weakly stable binary sums* if each Γ, A_1, A_2 as above has some binary sum $(A_1 + A_2, \nu_1, \nu_2)$, such that for every $\sigma : \Delta \longrightarrow \Gamma$, $((A_1 + A_2)[\sigma], \nu_1[\sigma], \nu_2[\sigma])$ is a binary sum for $A_1[\sigma]$ and $A_2[\sigma]$ over Δ.

(Note that this condition is independent of the choice of reindexings used, i.e. of the cleaving of $\mathcal{T}$.)

Definition 3.4.1.3. A split comprehension category C has *strictly stable binary sums* it it is equipped with functions giving for each Γ, A_1, A_2 some chosen binary sum $(A_1 + A_2, \nu_1, \nu_2)$, and moreover for each suitable C, t_1, t_2 some chosen copair $\langle t_1, t_2 \rangle$, such that for every $\sigma : \Delta \longrightarrow \Gamma$,

$$\begin{aligned} (A_1 + A_2)[\sigma] &= A_1[\sigma] + A_2[\sigma] \\ \nu_i[\sigma] &= \nu_i : A_i[\sigma] \longrightarrow A_1[\sigma] + A_2[\sigma] \\ \langle t_1, t_2 \rangle[\sigma] &= \langle t_1[\sigma], t_2[\sigma] \rangle. \end{aligned}$$

(By contrast, this is certainly not independent of the choice of reindexings given by the splitting of $\mathcal{T}$.)

The components of this definition—the sum type, the inclusion maps, the copair map, and the copair equations—correspond precisely to the type-theoretic rules for binary sums [ML84, p. 55]. A split comprehension category C with strictly stable binary sums is thus precisely what is needed to interpret the syntax of type theory with these rules (cf. [Hof97, §3.3]).

Lemma 3.4.1.4. *If C has weakly stable binary sums and satisfies condition (LF), then $C_!$ has strictly stable binary sums.*

Proof. We take each component of the definition in turn.

Formation. Suppose we are given suitable Γ, A_1, A_2 in $C_!$, and wish to form $A_1 + A_2$. These correspond to local universes $V_{A_i} \in C$, $E_{A_i} \in \mathcal{T}(V_{A_i})$, and maps $\ulcorner A_i \urcorner : \Gamma \longrightarrow V_{A_i}$.

Set $V_{A_1+A_2} := V_{A_1} \times V_{A_2}$. As indicated previously, this may be seen as the object of "(formal) sums of a type from V_{A_1} with a type from V_{A_2}". Precisely, for any Γ, pairs $\ulcorner A_1 \urcorner$, $\ulcorner A_2 \urcorner$ as above correspond to maps $(\ulcorner A_1 \urcorner, \ulcorner A_2 \urcorner) : \Gamma \longrightarrow V_{A_1+A_2}$, naturally in Γ.

In particular, the identity map of $V_{A_1+A_2}$ corresponds to the projections $\pi_i : V_{A_1+A_2} \longrightarrow V_{A_i}$. Pulling back the types E_{A_i} along these, we obtain types $E_{A_i}[\pi_i] \in \mathcal{T}(V_{A_1+A_2})$. Take $E_{A_1+A_2}$ to be the sum $E_{A_1}[\pi_1] + E_{A_2}[\pi_2]$.

Finally, take $\ulcorner A_1 + A_2 \urcorner := (\ulcorner A_1 \urcorner, \ulcorner A_2 \urcorner) : \Gamma \longrightarrow V_{A_1+A_2}$, picking out the appropriate specific pairs of types.

For strict stability, suppose in addition to the above data we have some $\sigma : \Gamma' \longrightarrow \Gamma$; we need to show that $(A_1 + A_2)[\sigma] = A_1[\sigma] + A_2[\sigma]$. It is immediate that their universes are equal—i.e. that $V_{(A_1+A_2)[\sigma]} = V_{A_1[\sigma]+A_2[\sigma]}$ and $E_{(A_1+A_2)[\sigma]} = E_{A_1[\sigma]+A_2[\sigma]}$—since the universe of the sum depends only on the universes of the summands, and substitution in the summands does not change their universes. On the other hand,

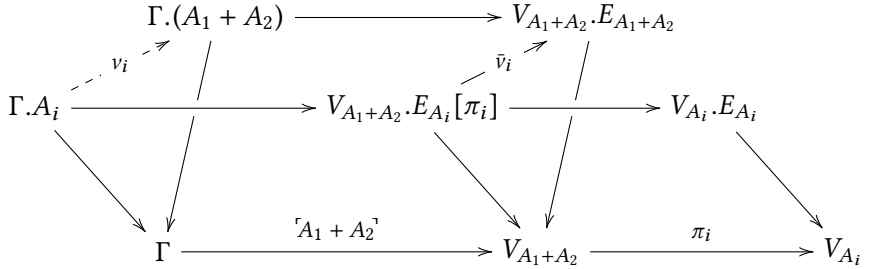
$$\begin{aligned} \ulcorner (A_1 + A_2)[\sigma] \urcorner &= (\ulcorner A_1 \urcorner, \ulcorner A_2 \urcorner) \circ \sigma \\ &= (\ulcorner A_1 \urcorner \circ \sigma, \ulcorner A_2 \urcorner \circ \sigma) \\ &= \ulcorner A_1[\sigma] + A_2[\sigma] \urcorner; \end{aligned}$$

that is, strict stability of $(+)$ comes exactly from the (strict) naturality in Γ of the universal property of $V_{A_1} \times V_{A_2}$.

Introduction. For the sum inclusions, suppose again we have Γ, A_1, A_2 in $\mathcal{C}_!$. Having constructed $A_1 + A_2$ as above, note that since $E_{A_1+A_2}$ was chosen as a weakly stable sum, it comes with inclusion maps $\bar{\nu}_i : E_{A_i}[\pi_i] \longrightarrow E_{A_1+A_2}$. Now set $\nu_i := \bar{\nu}_i[\ulcorner A_1 + A_2 \urcorner] : [A_i] \longrightarrow [A_1 + A_2]$.

(We are using here the convention that maps may be reindexed to arbitrary reindexings of their domain and codomain. We will do so in future without comment.)

Breaking down this definition a little: we first consider the introduction maps in the universal case, $\bar{\nu}_i : E_{A_i}[\pi_i] \longrightarrow E_{A_1+A_2}$, over $V_{A_1+A_2}$. We then reindex this to $\mathcal{T}(\Gamma)$ along $\ulcorner A_1 + A_2 \urcorner$.



Elimination. Defining copairing in $\mathcal{C}_!$ holds a subtle pitfall for the unwary—one worth looking at explicitly, since it will recur later for other constructors.

Consider the data Γ, A_1, A_2, C, d_1, d_2 in $\mathcal{C}_!$, for forming a copair. In $\mathcal{C}$, these correspond to Γ, V_{A_i}, E_{A_i}, $\ulcorner A_i \urcorner$ as above, together with another local universe $E_C \in \mathcal{T}(V_C)$, a name map $\ulcorner C \urcorner : \Gamma.(A_1 + A_2) \longrightarrow V_C$, and sections $d_i : \Gamma.A_i \longrightarrow \Gamma.A_i.E_C[\ulcorner C \urcorner \circ \nu_i]$.

We require a copair for C, d_1, d_2 in $\mathcal{C}_!$; that is, a certain section $\Gamma.(A_1 + A_2) \longrightarrow \Gamma.(A_1 + A_2).C$, commuting appropriately with ν_i, d_i. This corresponds to a section $\Gamma.[A_1 + A_2] \longrightarrow \Gamma.[A_1 + A_2].[C]$ in $\mathcal{C}$, commuting there with ν_i, d_i.

The obvious way to obtain such a section is simply as a copair in $\mathcal{C}$. We chose $E_{A_1+A_2}$ as a weakly stable (+)-type over $V_{A_1+A_2}$, so $[A_1 + A_2]$ is a (+)-type over Γ, and the data $[C]$, d_1, d_2 are just right for forming a copair there.

However, the resulting operation would *not* necessarily be strictly stable, since the copair in $\mathcal{C}$ was taken over Γ.[3] We therefore resist this tempting shortcut and keep to the general approach prescribed above, first taking a "universal copair" depending just on the universes V_{A_1}, V_{A_2}, V_C. Only having done this do we pull it back (strictly naturally) to the specific context Γ in question.

Precisely, fix universes V_{A_i}, E_{A_i}, V_C, E_C, and set:

$$V_{\langle d_1,d_2\rangle} := [a_1{:}V_{A_1},\ a_2{:}V_{A_2},\ c{:}{V_C}^{E_{A_1+A_2}(a_1,a_2)},\\ d_1{:}\Pi_{x:E_{A_1}(a_1)}E_C(c(\nu_1(x))),\ d_2{:}\Pi_{y:E_{A_2}(a_2)}E_C(c(\nu_2(y)))]$$

The remaining data $\ulcorner A_i \urcorner$, $\ulcorner C \urcorner$, d_i correspond to maps $\Gamma \longrightarrow V_{\langle d_1,d_2\rangle}$, naturally in Γ. In particular, the identity $1_{V_{\langle d_1,d_2\rangle}}$ corresponds to maps

$$\pi_{A_i} : V_{\langle d_1,d_2\rangle} \longrightarrow V_{A_i} \quad \pi_C : V_{\langle d_1,d_2\rangle}.E_{A_1+A_2}[(\pi_{A_1},\pi_{A_2})] \longrightarrow V_C$$
$$\pi_{d_i} : V_{\langle d_1,d_2\rangle}.E_{A_i}[\pi_{A_i}] \longrightarrow V_{\langle d_1,d_2\rangle}.E_{A_i}[\pi_{A_i}].E_C[\pi_C \circ \nu_i[(\pi_{A_1},\pi_{A_2})]].$$

Now, as in the direct approach, since $E_{A_1+A_2}$ was a weakly stable sum, its reindexing $E_{A_1+A_2}[(\pi_{A_1},\pi_{A_2})]$ together with the inclusion maps $\nu_i[(\pi_{A_1},\pi_{A_2})]$ is a sum for $E_{A_1}[\pi_{A_1}]$ and $E_{A_2}[\pi_{A_2}]$ over $V_{\langle d_1,d_2\rangle}$. So we may form there the copair section

$$\langle \pi_{d_1},\pi_{d_2}\rangle : V_{\langle d_1,d_2\rangle}.E_{A_1+A_2}[(\pi_{A_1},\pi_{A_2})] \longrightarrow V_{\langle d_1,d_2\rangle}.E_{A_1+A_2}[(\pi_{A_1},\pi_{A_2})].E_C[\pi_C].$$

Pulling this back along

$$(\ulcorner A_1 \urcorner, \ulcorner A_2 \urcorner, \ulcorner C \urcorner, d_1, d_2) : \Gamma \longrightarrow V_{\langle d_1,d_2\rangle}$$

then gives us a section

$$\Gamma.(A_1 + A_2) \longrightarrow \Gamma.(A_2 + A_2).C$$

which we take as the copair $\langle d_1, d_2\rangle$ in $\mathcal{C}_!$.

Strict stability of this operation follows from the fact that the only involvement of $\Gamma, \ulcorner A_1 \urcorner, \ulcorner A_2 \urcorner, \ulcorner C \urcorner, d_1, d_2$ was via the map $\Gamma \longrightarrow V_{\langle d_1,d_2\rangle}$, and the pullback of a section along this map, both of which are strictly natural in Γ. In particular, the copair used in $\mathcal{C}$, which a priori may not satisfy any naturality condition, was taken over $V_{\langle d_1,d_2\rangle}$, and so is unaffected by any reindexing $\Gamma' \longrightarrow \Gamma$.

[3] If the (+)-types of $\mathcal{C}$ are pseudo-stable—for instance, if they satisfy the η-rule, making them categorical coproducts—then this direct definition of copairing is strictly stable after all; and in the case of (+)-types, this would be a reasonably mild extra condition to demand. However, for identity types (and more general inductive families), the analogous hypothesis would be much less innocuous, implying in particular the reflection rule, and hence UIP.

Computation. Finally, the copair-inclusion equations for $\langle d_1, d_2\rangle$ follow directly from the equations in C for $\langle \pi_{d_1}, \pi_{d_2}\rangle$, pulled back along $(\ulcorner A_1\urcorner, \ulcorner A_2\urcorner, \ulcorner C\urcorner, d_1, d_2)$. □

3.4.2. **Dependent products.**

Definition 3.4.2.1. Given $\Gamma \in C$, $A \in \mathcal{T}(\Gamma)$, $B \in \mathcal{T}(\Gamma.A)$, a *dependent product* for Γ, A, B is given by:

- a type $\prod[A,B] \in \mathcal{T}(\Gamma)$;
- a map $\mathrm{app}_{A,B} : \prod[A,B][\chi(A)] \longrightarrow B$ in $\mathcal{T}(\Gamma.A)$;
- an operation giving for each section $t : \Gamma.A \longrightarrow \Gamma.A.B$, a section $\lambda(t) : \Gamma \longrightarrow \Gamma.\prod[A,B]$, such that $(\Gamma.A.\mathrm{app}_{A,B}) \circ (1_{\Gamma.A}, \lambda(t)) = t$.

Definition 3.4.2.2. C has *weakly stable dependent products* if every Γ, A, B has some $(\prod[A,B], \mathrm{app}_{A,B})$ as above, such that for every $\sigma : \Delta \longrightarrow \Gamma$, there is some operation λ making $(\prod[A,B][\sigma], \mathrm{app}_{A,B}[\sigma], \lambda)$ a dependent product for Δ, $A[\sigma]$, $B[\sigma]$. More specifically, call such $(\prod[A,B], \mathrm{app}_{A,B})$ a *weakly stable dependent product* for Γ, A, B.

(Note again that this is independent of the cleaving of $\mathcal{T}$.)

Definition 3.4.2.3. C has *strictly stable dependent products* if it is equipped with operations giving $\prod(A,B)$, $\mathrm{app}_{A,B}$, and $\lambda(t)$ for all appropriate Γ, A, B and t as above, such that for every $\sigma : \Delta \longrightarrow \Gamma$,

$$\begin{aligned} \textstyle\prod[A,B][\sigma] &= \textstyle\prod\big[A[\sigma], B[\sigma]\big] \\ \mathrm{app}_{A,B}[\sigma] &= \mathrm{app}_{A[\sigma],B[\sigma]} \\ (\lambda(t))[\sigma] &= \lambda(t[\sigma]). \end{aligned}$$

(Again, this is by contrast entirely dependent on the chosen cleaving.)

Lemma 3.4.2.4. *If C has weakly stable dependent products and satisfies condition (LF), then $C_!$ has strictly stable dependent products.*

Proof. Again, we consider the components of the definition—the four rules for Π-types—one by one.

Formation. Suppose we have $A = (V_A, E_A, \ulcorner A\urcorner)$ in $\mathcal{T}_!(\Gamma)$ and $B = (V_B, E_B, \ulcorner B\urcorner)$ in $\mathcal{T}_!(\Gamma.A)$, and wish to form $\prod[A,B]$.

We begin by setting

$$V_{\prod[A,B]} := V_A \triangleleft V_B.$$

(Recall $V_A \triangleleft V_B$ is the object given in the internal language by $[a{:}V_A,\ b{:}{V_B}^{E_A(a)}]$.)

As described in Section 3.3, maps $\Gamma \longrightarrow V_A \triangleleft V_B$ correspond to data $\ulcorner A\urcorner : \Gamma \longrightarrow V_A$, $\ulcorner B\urcorner : \Gamma.E_A[\ulcorner A\urcorner] \longrightarrow V_B$ as above. In particular, there is the universal case

$$\pi_A : V_A \triangleleft V_B \longrightarrow V_A, \qquad \pi_B : (V_A \triangleleft V_B).(E_A[\pi_A]) \longrightarrow V_B$$

over $V_{\prod[A,B]}$ itself. To obtain $E_{\prod[A,B]}$, we choose a weakly stable dependent product in C for this universal case:

$$E_{\prod[A,B]} := \prod [E_A[\pi_A], E_B[\pi_B]] \qquad \in \mathcal{T}(V_{\prod[A,B]}).$$

Finally, we take $\ulcorner \prod[A,B] \urcorner : \Gamma \longrightarrow V_{\prod[A,B]}$ to be the map corresponding to the pair $\ulcorner A \urcorner$, $\ulcorner B \urcorner$ under the universal property of $V_{\prod[A,B]}$.

Together, these define the type $\prod[A,B]$ in $\mathcal{C}_!$. To see that the resulting operation is moreover strictly stable, suppose we have Γ, A, B as above, and additionally some $\sigma : \Gamma' \longrightarrow \Gamma$. We need to check that $\prod[A,B][\sigma] = \prod[A[\sigma], B[\sigma]]$.

It is immediate that the local universes of these two products are the same, since they depend only on the local universes V_A, V_B, which are unaffected by the reindexing.

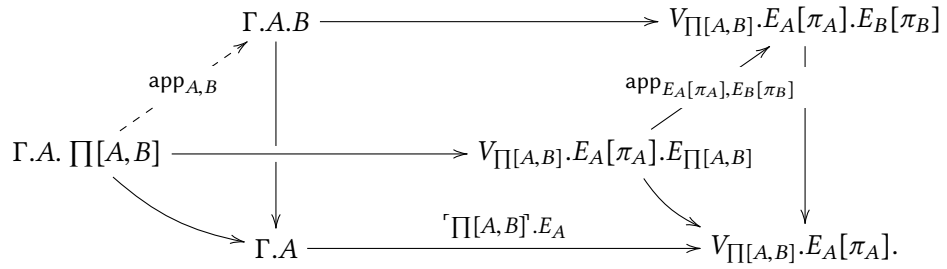
It therefore only remains to show that $\ulcorner \prod[A,B][\sigma] \urcorner = \ulcorner \prod[A[\sigma], B[\sigma]] \urcorner$; but this follows just from the (strict) naturality in Γ of the universal property of $V_A \triangleleft V_B$.

Application. By the definition of $E_{\prod[A,B]}$ as a weakly stable dependent product, it comes with a map

$$\mathsf{app}_{E_A[\pi_A], E_B[\pi_B]} : V_{\prod[A,B]}.E_A[\pi_A].E_{\prod[A,B]} \longrightarrow V_{\prod[A,B]}.E_A[\pi_A].E_B[\pi_B]$$

over $V_{\prod[A,B]}.E_A[\pi_A]$.

We define $\mathsf{app}_{A,B}$ just as the reindexing of this map to $\mathcal{T}(\Gamma.A)$:



Once again, strict stability of this follows directly by construction. The universal case $\mathsf{app}_{E_A[\pi_A], E_B[\pi_B]}$ depends only on V_A, V_B; and the subsequent reindexing is strictly natural in Γ.

Introduction. As in the case of copairing above, there is a direct approach to defining λ-abstraction, which however may fail to be strictly stable. We therefore take once again a two-stage approach. First, we define the object $V_{\lambda_{A,B}}$ of all possible λ-abstractions into $E_{\prod A,B}$, and choose a universal λ-abstraction over that; then, we pick out the λ-abstractions in $\mathcal{C}_!$ as pullbacks of that universal one.

Let $V_{\lambda_{A,B}}$ be the object:

$$[a : V_A,\ b : {V_B}^{E_A(a)},\ t : \overline{\prod}_{x:E_A(a)} E_B(b(x))]$$

Here we write $\overline{\prod}$ to emphasize that this description is interpreted using the categorical exponentials in $\mathcal{C}$ provided by condition (LF), not the weakly stable dependent products of types in $\mathcal{T}$ used for $E_{\prod[A,B]}$. Modulo that difference, this is exactly analogous to the object $V_{\prod[A,B]}.E_{\prod[A,B]}$.

Write $\pi_{\lambda_{A,B}}$ for the evident projection $V_{\lambda_{A,B}} \longrightarrow V_{\prod[A,B]}$.

As a categorical dependent product, $V_{\lambda_{A,B}}$ has application map

$$\overline{\mathsf{app}} : V_{\lambda_{A,B}}.E_A[\pi_A \circ \pi_{\lambda_{A,B}}] \longrightarrow V_{\lambda_{A,B}}.E_A[\pi_A \circ \pi_{\lambda_{A,B}}].E_B[\pi_A \circ \pi_{\lambda_{A,B}}.E_A].$$

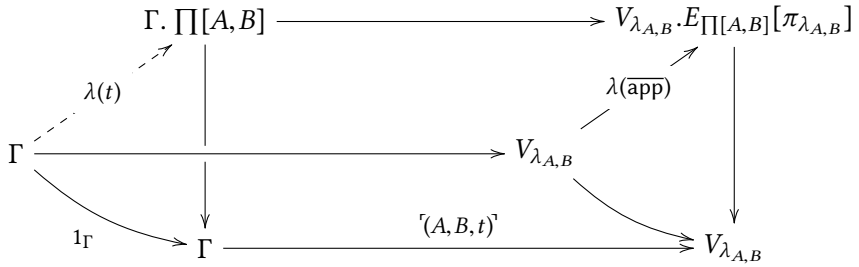
Since $E_{\prod[A,B]}$ was a weakly stable dependent product, $\overline{\mathsf{app}}$ induces a map

$$\lambda(\overline{\mathsf{app}}) : V_{\lambda_{A,B}} \longrightarrow V_{\lambda_{A,B}}.E_{\prod[A,B]}[\pi_{\lambda_{A,B}}]$$

with $\mathsf{app}_{E_A[\pi_A],E_B[\pi_B]}[\pi_{\lambda_{A,B}}] \circ \lambda(\overline{\mathsf{app}})[\chi(E_A[\pi_A \circ \pi_{\lambda_{A,B}}])] = \overline{\mathsf{app}}$.

Now, suppose we are given the inputs for λ-abstraction in $\mathcal{C}_!$. That is, in addition to Γ, A, B as before, we have a section $t : \Gamma.A \longrightarrow \Gamma.A.B$.

By the universal property of the categorical dependent product, such tuples $(\ulcorner A \urcorner, \ulcorner B \urcorner, t)$ correspond naturally to maps $\ulcorner (A,B,t) \urcorner : \Gamma \longrightarrow V_{\lambda_{A,B}}$. So, we may take $\lambda(t) : \Gamma \longrightarrow \Gamma.\prod[A,B]$ to be the reindexing of $\lambda(\overline{\mathsf{app}})$ along $\ulcorner (A,B,t) \urcorner$:



Strict stability is immediate by construction, just as for $\mathsf{app}_{A,B}$.

Computation. Finally, the β-reduction equation for $\lambda(t)$ in $\mathcal{C}_!$ follows from the corresponding equation for the universal case $\lambda(\overline{\mathsf{app}})$, which holds by its construction as a λ-abstraction into $E_{\prod[A,B]}$. □

Often, one may want to restrict the Π-types used to some well-behaved or well-understood subclass—typically, the categorical dependent products. Indeed, one might want the same for other constructors as well; we spell out the case of Π-types since we will need it for setting up weakly stable W-types below.

Definition 3.4.2.5. A *stable class of* Π*-types* on $\mathcal{C}$ consists of:

- for each Γ, A, B, a non-empty family $\mathcal{G}_\Pi(\Gamma, A, B)$ of Π-types (Π, app) for A, B, stable under reindexing, in that for all $\sigma : \Delta \longrightarrow \Gamma$ and $(\Pi, \mathsf{app}) \in \mathcal{G}_\Pi(\Gamma, A, B)$, and any reindexings A', B', Π', app' of these along σ, we have $(\Pi', \mathsf{app}') \in \mathcal{G}_\Pi(\Gamma', A', B')$;
- and, for each Γ, A, B as before, $(\Pi, \mathsf{app}) \in \mathcal{G}_\Pi(\Gamma, A, B)$, and section $t : \Gamma.A \longrightarrow \Gamma.A.B$, a non-empty family of sections $\mathcal{G}_\lambda(\Gamma, \ldots, t)$, similarly stable under reindexing.

Given such a class, we refer to an element of $\mathcal{G}_\Pi(\Gamma, A, B)$ (resp. $\mathcal{G}_\lambda(\Gamma, \ldots, t)$) as a *good* Π-type for A, B (resp. a good λ-abstraction of t). Stability says just that any reindexing of a good Π-type or λ-abstraction is again good.

Scholium 3.4.2.6. *If $\mathcal{C}$ is equipped with a stable class of dependent products, then $\mathcal{C}_!$ has strictly stable dependent products, always chosen from the given stable class.*

Proof. Immediate from the proof of Lemma 3.4.2.4. □

Proposition 3.4.2.7. *C has weakly stable Π-types if and only if it can be equipped with a stable class of Π-types.*

Proof. The "if" direction is immediate. For the "only if", note that if C has weakly stable Π-types, then the class of *all* weakly stable Π-types and λ-abstractions forms a stable class. □

Finally, we pause to consider the pseudo-stable level, again for later use in the presentation of W-types:

Definition 3.4.2.8. C has *pseudo-stable dependent products* if it is equipped with operations Π, app, λ as above, together with a cartesian functorial action on cartesian maps; that is, for each $\sigma : \Gamma' \longrightarrow \Gamma$, and cartesian maps $\sigma_A : A' \to A$ over σ and $\sigma_B : B' \to B$ over $\sigma.\sigma_A$, a cartesian map $\Pi[\sigma_A, \sigma_B] : \Pi[A', B'] \to \Pi[A, B]$ over σ, such that

$$\begin{aligned} \Pi[1_A, 1_B] &= 1_{\Pi[A,B]} \\ \Pi[\tau_A \circ \sigma_A, \tau_B \circ \sigma_B] &= \Pi[\tau_A, \tau_B] \circ \Pi[\sigma_A, \sigma_B] \\ \sigma.\sigma_B \circ \mathsf{app}_{A',B'} &= \mathsf{app}_{A,B} \circ \sigma.\sigma_A.\Pi[\sigma_A, \sigma_B] \\ \sigma.\Pi[\sigma_A, \sigma_B] \circ \lambda_{A',B'}(t[\sigma]) &= \lambda_{A,B}(t) \circ \sigma \end{aligned}$$

(for all suitable $\sigma, \sigma_A, \sigma_B, \tau, \tau_A, \tau_B, t$).

Proposition 3.4.2.9. *If C is equipped with pseudo-stable dependent products, then it carries a stable class of dependent products, consisting of all Π-types equipped with isomorphisms to those supplied by the pseudo-stable structure, and all λ-abstractions corresponding under those isomorphisms to the ones provided by the pseudo-stable structure.* □

3.4.3. **Identity types.** The identity types we consider in this section will be slightly stronger than those set out in Section 2.3 above. Specifically, we consider structure corresponding to the elimination rule:

$$\frac{\begin{array}{c}\Gamma,\ x,y{:}A,\ u{:}\mathsf{Id}_A(x,y),\ \Delta \vdash C(x,y,u)\ \mathsf{type} \\ \Gamma,\ z{:}A,\ \Delta[z/x, z/y, \mathsf{r}(z)/u] \vdash d(z) : C(z,z,\mathsf{r}(z))\end{array}}{\Gamma,\ a,b{:}A,\ p{:}\mathsf{Id}_A(x,y),\ \Delta[a/x, b/y, p/u] \vdash \mathsf{J}_{x,y,u.C;\,z.d}(a,b,p) : C(a,b,p)}$$

where Δ may be an arbitrary context extension.

Often, Δ is omitted in the basic definition of identity types, and the version with it is called *strong* or *Frobenius* identity types. In the presence of Π-types, the two forms are interderivable, so the weak form suffices. In general, however, the Frobenius form is the more important and more natural; so that is the form we consider here, and throughout this section, *identity types* will always refer to the Frobenius form. (As ever, though, the construction works for either set of rules.)

Given this, we make some slight modifications to the definitions of Sec. 2.3.

Definition 3.4.3.1. A *(Frobenius) identity type* for $\Gamma \in \mathcal{C}$, $A \in \mathcal{T}(\Gamma)$ consists of Id_A, r_A as in Definition 2.3.1, together with for each sequence $\Delta = (B_1, \ldots, B_n)$

such that $B_i \in \mathcal{T}(\Gamma.A.A.\mathsf{Id}_A.B_1.\ \ldots\ .B_{i-1})$, each $C \in \mathcal{T}(\Gamma.A.A.\mathsf{Id}_A.\Delta)$, and each $d : \Gamma.A.\Delta[\mathsf{r}_A] \longrightarrow \Gamma.A.A.\mathsf{Id}_A.\Delta.C$ such that the square

$$\begin{array}{ccc} \Gamma.A.\Delta[\mathsf{r}_A] & \xrightarrow{d} & \Gamma.A.A.\mathsf{Id}_A.\Delta.C \\ {\scriptstyle \mathsf{r}_A}\downarrow & \nearrow_{\mathsf{j}_{A,\Delta,C,d}} & \downarrow \\ \Gamma.A.A.\mathsf{Id}_A.\Delta & \xrightarrow[1]{} & \Gamma.A.A.\mathsf{Id}_A.\Delta \end{array}$$

commutes, a diagonal filler $\mathsf{j}_{A,\Delta,C,d}$, which we call an *elimination section* for the data A, Δ, C, d.

A choice of identity types on C is *strictly stable* if for each $\sigma : \Gamma' \longrightarrow \Gamma$, and all appropriate A, Δ, C, d,

$$\begin{aligned} \mathsf{Id}_A[\sigma] &= \mathsf{Id}_{A[\sigma]} \\ \mathsf{r}_A[\sigma] &= \mathsf{r}_{A[\sigma]} \\ \mathsf{j}_{A,\Delta,C,d}[\sigma] &= \mathsf{j}_{A[\sigma],\Delta[\sigma],C[\sigma],d[\sigma]}. \end{aligned}$$

A *weakly stable identity type* for $A \in \mathcal{T}(\Gamma)$ is $(\mathsf{Id}, \mathsf{r})$ as above such that, for all $\sigma : \Gamma' \longrightarrow \Gamma$, there exists some j making $(\mathsf{Id}[\sigma], \mathsf{r}[\sigma], \mathsf{j})$ an identity type for $A[\sigma]$. Say C *has weakly stable identity types* if for each Γ, A, there exists some weakly stable identity type.

Lemma 3.4.3.2. *If C has weakly stable identity types and satisfies condition (LF), then $C_!$ has strictly stable identity types.*

Proof. As usual, we consider the rules one by one.

Formation. Given $A \in \mathcal{T}_!(\Gamma)$, choose some weakly stable identity type $(\mathsf{Id}_{E_A}, \mathsf{r}_{E_A})$ for E_A over V_A, and define:

$$\begin{aligned} V_{\mathsf{Id}_A} &:= V_A.E_A.E_A \\ E_{\mathsf{Id}_A} &:= \mathsf{Id}_{E_A} \\ \ulcorner \mathsf{Id}_A \urcorner &:= \ulcorner A \urcorner.E_A.E_A : \Gamma.A.A \longrightarrow V_{\mathsf{Id}_A} \end{aligned}$$

As usual, this is strictly stable just since V_{Id_A} and E_{Id_A} do not depend on Γ, $\ulcorner A \urcorner$, while the construction of $\ulcorner \mathsf{Id}_A \urcorner$ is strictly natural in Γ.

Introduction. For the reflexivity map, take

$$\mathsf{r}_A := \mathsf{r}_{E_A}[\ulcorner A \urcorner]$$

where r_{E_A} is the reflexivity map of the chosen weakly stable identity type Id_{E_A}.

Again, strict stability is immediate.

Elimination, computation. Let Γ, A, $\Delta = (B_1, \ldots, B_n)$, C, d be instances of the premises of Id-elimination in $C_!$. (In particular, $B_i \in \mathcal{T}_!(\Gamma.A.A.\mathsf{Id}_A.B_1.\ldots.B_{i-1})$.)

As usual, we work by first fixing the universes V_A, $V_{B_1}, \ldots, V_{B_n}$, V_C, and constructing a representing object V for "data as above, with the given universes". Due to the Frobenius premise Δ, this is slightly more involved than other rules we have considered.

In the internal language, it may be expressed as

$$\begin{aligned} V := [a : V_A,\ & \\ & b_1 : \prod x,x' {:} E_A(a),\, y {:} \mathsf{Id}_{E_A}(a,x,x').\ V_{B_1}, \\ & b_2 : \prod x,x' {:} E_A(a),\, y {:} \mathsf{Id}_{E_A}(a,x,x'),\, z_1 {:} E_{B_1}(b_1(x,x',y)).\ V_{B_2}, \\ & \dots \\ & c : \prod x,x',y,z_1,\dots,z_n.\ V_C \\ & d : \prod x : E_A,\, z_1 : E_{B_1}(b_1(x,x,\mathsf{r}_{E_A}(a,x))),\ \dots, \\ & \qquad\qquad z_n : E_{B_n}(b_n(x,x,\mathsf{r}_{E_A}(a,x),z_1,\dots,z_{n-1})), \\ & \quad E_C(c(x,x,\mathsf{r}_{E_A}(a,x),z_1,\dots,z_n))] \end{aligned}$$

(Here $(\mathsf{Id}_{E_A},\mathsf{r}_{E_A})$ are the identity type used for Id_A, r_A above.)

Maps $\ulcorner A,\Delta,C,d\urcorner : \Gamma \longrightarrow V$ now correspond, naturally in Γ, to tuples over Γ $(\ulcorner A\urcorner,\ulcorner B_1\urcorner,\dots,\ulcorner B_n\urcorner,\ulcorner C\urcorner,d)$ as in the original data. In particular, the identity 1_V corresponds to such data over V itself. Since $(\mathsf{Id}_{E_A},\mathsf{r}_{E_A})$ was weakly stable, we may choose some universal elimination section j for this data:

$$\begin{aligned} \mathsf{j} : [(a,b_1,\dots,c,d){:}V,\ & x,x' {:} E_A(a),\, y {:} \mathsf{Id}_{E_A}(a,x,x'),\, z_1,\dots,z_n] \\ & \longrightarrow [(a,b_1,\dots,c,d),\, x,x',y,\, z_1,\dots,z_n,\, c {:} E_C(c(x,\dots,z_n))]. \end{aligned}$$

Returning to the original specific inputs Γ, A, Δ, C, d, we can now pull this universal j back along the representing map $\ulcorner A,\Delta,C,d\urcorner : \Gamma \longrightarrow V$ to give the required elimination section for d:

$$\mathsf{j}_{A,\Delta,C,d} := \mathsf{j}[\ulcorner A,\Delta,C,d\urcorner] : (\Gamma.A.A.\mathsf{Id}_A.\Delta) \longrightarrow (\Gamma.A.A.\mathsf{Id}_A.\Delta.C).$$

Strict stability follows, as usual, from the fact that this depends on Γ only via the universal property of V and the action on morphisms of a pullback functor, both of which are suitably natural.

In particular, the choice of an elimination section j—the one operation which is *not* strictly natural in many models, and cannot easily be made so—was made over V, and so depends only on the universes involved, not on Γ, or on anything else affected by reindexing in $\mathcal{C}_!$. □

3.4.4. **Other constructors.** The three cases above illustrate essentially all the issues that arise in constructing structure on $\mathcal{C}_!$.

For the remaining constructors, therefore, we give just the definitions of the appropriate strictly/weakly stable structure, and precise statements of the lifting lemmas. We omit their proofs, since they follow exactly the same template as the cases above.

The definitions, too, contain just the same components as the cases above, with one exception, in the case of W-types. Since their rules refer to Π-types, the definition of weakly stable W-types must be given relative to some form of Π-types—most naturally and flexibly, to a chosen stable class thereof. This is the only new twist appearing in the definitions below, and indicates more generally

how one might extend the present results to other type-formers whose rules make reference to other previously-defined types.

Dependent sums.

Definition 3.4.4.1. For $\Gamma \in \mathcal{C}$, $A \in \mathcal{T}(\Gamma)$, and $B \in \mathcal{T}(\Gamma.A)$, a *dependent sum* for B consists of:

- a type $\Sigma_A B \in \mathcal{T}(\Gamma)$;
- a "pairing" map $\langle -,- \rangle : \Gamma.A.B \longrightarrow \Gamma.\Sigma_A B$; such that
- for any type $C \in \mathcal{T}(\Gamma.\Sigma_A B)$ and section $d : \Gamma.A.B \longrightarrow \Gamma.A.B.C[\langle -,- \rangle]$, there is a section $\mathrm{split}_{C,d} : \Gamma.\Sigma_A B \longrightarrow \Gamma.\Sigma_A B.C$, with $\mathrm{split}_{C,d} \circ \langle -,- \rangle = (\langle -,- \rangle.C) \circ d$.

$\mathcal{C}$ has *weakly stable dependent sums* if for each Γ, A, B as above, there exists some $(\Sigma_A B, \langle -,- \rangle)$, such that for each $\sigma : \Delta \longrightarrow \Gamma$, $(\Sigma_A B[\sigma], \langle -,- \rangle[\sigma])$ is a dependent sum for $A[\sigma]$ and $B[\sigma]$ over Δ.

A split comprehension category $\mathcal{C}$ has *strictly stable dependent sums* if it is equipped with functions giving for each Γ, A, B a dependent sum $(\Sigma_A B, \langle -,- \rangle)$, and moreover for each suitable C, d some appropriate section $\mathrm{split}_{C,d}$, all commuting on the nose with reindexing in $\mathcal{C}$.

Lemma 3.4.4.2. *If $\mathcal{C}$ has weakly stable dependent sums and satisfies condition (LF), then $\mathcal{C}_!$ has strictly stable dependent sums.* □

Zero types.

Definition 3.4.4.3. Given $\Gamma \in \mathcal{C}$, a *zero type over* Γ consists of:

- a type $0 \in \mathcal{T}(\Gamma)$;
- for any type $C \in \mathcal{T}(\Gamma.0)$, a section of C.

$\mathcal{C}$ has *weakly stable zero types* if for each Γ, there exists some type 0 over Γ such that for every $\sigma : \Delta \longrightarrow \Gamma$, $0[\sigma]$ is a zero type over Δ.

A split comprehension category $\mathcal{C}$ has *strictly stable zero types* if it is equipped with functions giving for each Γ a zero type 0_Γ, and for each $C \in \mathcal{T}(\Gamma.0_\Gamma)$ a section, both commuting strictly with reindexing.

Lemma 3.4.4.4. *If $\mathcal{C}$ satisfies condition (LF) and has weakly stable zero types, then $\mathcal{C}_!$ has strictly stable zero types.* □

Unit types.

Definition 3.4.4.5. Given $\Gamma \in \mathcal{C}$, a *unit type over* Γ consists of:

- a type $1 \in \mathcal{T}(\Gamma)$;
- a section $\mathtt{tt} : \Gamma \longrightarrow \Gamma.1$;
- for any type $C \in \mathcal{T}(\Gamma.1)$ and section d of $C[\mathtt{tt}]$, a section $\mathrm{urec}_{C,d}$ of C, such that $\mathrm{urec}_{C,d} \circ \mathtt{tt} = d$.

$\mathcal{C}$ has *weakly stable unit types* if for each Γ, there is some $(1, \mathtt{tt})$ over Γ such that for every $\sigma : \Delta \longrightarrow \Gamma$, elimination sections can be chosen making $(1[\sigma], \mathtt{tt}[\sigma])$ a unit type over Δ.

A split comprehension category C has *strictly stable unit types* if it is equipped with functions giving unit types 1_Γ, tt_Γ, and elimination sections $\mathrm{urec}_{C,d}$, all commuting strictly with reindexing.

Lemma 3.4.4.6. *If C satisfies condition (LF) and has weakly stable unit types, then $C_!$ has strictly stable unit types.* □

W-types. W-types (also known as inductive types, or types of well-founded trees), are the most powerful of the standard type-constructors [ML84, p. 79].

Since their rules involve Π-types, any kind of W-type structure on a comprehension category C must be relative to some form of Π-type structure on C. This dependence introduces an extra subtlety into the definition of weakly stable W-types. We will therefore consider two different weak forms: a simpler form, assuming that the Π-types of C are pseudo-stable (for instance, if they are categorical exponentials); and a more involved but more general form, allowing that the Π-types themselves may be only weakly stable .

Definition 3.4.4.7. Suppose C is equipped with some choice of dependent products. Given $\Gamma \in \mathcal{C}$, $A \in \mathcal{T}(\Gamma)$, $B \in \mathcal{T}(\Gamma.A)$, a *W-type* $(\mathsf{W}, \mathrm{fold}, \mathrm{wrec})$ for Γ, A, B consists of:

- a type $\mathsf{W} \in \mathcal{T}(\Gamma)$;
- a map $\mathrm{fold} : \Gamma.A.\Pi[B, \mathsf{W}[\chi(A)]] \longrightarrow \Gamma.\mathsf{W}$, over Γ;
- together with, for any type C over $\Gamma.\mathsf{W}$ and square of the form

$$\begin{array}{ccc} \Gamma.\mathsf{W}.\Pi[B,\mathsf{W}].\Pi[B, C[\mathrm{app}'_{B,\mathsf{W}}]] & \xrightarrow{d} & \Gamma.\mathsf{W}.C \\ \downarrow & (1) & \downarrow \\ \Gamma.\mathsf{W}.\Pi[B,\mathsf{W}] & \xrightarrow[\mathrm{fold}]{} & \Gamma.\mathsf{W}, \end{array}$$

a section $\mathrm{wrec}_{C,d} : \Gamma.\mathsf{W} \longrightarrow \Gamma.\mathsf{W}.C$, such that the square

$$\begin{array}{ccc} \Gamma.\mathsf{W}.\Pi[B,\mathsf{W}].\Pi[B, C[\mathrm{app}'_{B,\mathsf{W}}]] & \xrightarrow{d} & \Gamma.\mathsf{W}.C \\ {\scriptstyle \lambda(\mathrm{wrec}_{C,d} \circ \mathrm{app}'_{B,\mathsf{W}})}\uparrow & (2) & \uparrow{\scriptstyle \mathrm{wrec}_{C,d}} \\ \Gamma.\mathsf{W}.\Pi[B,\mathsf{W}] & \xrightarrow[\mathrm{fold}]{} & \Gamma.\mathsf{W}, \end{array}$$

commutes.

(Here app′ is app with its arguments flipped; and we suppress several weakenings, so e.g. $\Pi[B, \mathsf{W}]$ is strictly speaking $\Pi[B, \mathsf{W}[\chi(A)]]$.)

For the strictly stable case, of course, the Π-types too must be strictly stable, in order for the stability equations for fold and wrec to typecheck.

Definition 3.4.4.8. Suppose C has strictly stable dependent products. We say C has *strictly stable* W-*types* (over the given dependent products) if it is equipped

with operations providing, for all Γ, A, B as above, a W-type $(\mathsf{W}_{A,B}, \mathsf{fold}_{A,B}, \mathsf{wrec}_{A,B})$, such that for $\sigma : \Gamma' \to \Gamma$ and all appropriate A, B, C, d,

$$\begin{aligned} \mathsf{W}_{A,B}[\sigma] &= \mathsf{W}_{A[\sigma],B[\sigma]} \\ \mathsf{fold}_{A,B}[\sigma] &= \mathsf{fold}_{A[\sigma],B[\sigma]} \\ \mathsf{wrec}_{A,B;C,d}[\sigma] &= \mathsf{wrec}_{A[\sigma],B[\sigma];C[\sigma],D[\sigma]}. \end{aligned}$$

If we assume pseudo-stable dependent products, then we can give a simple definition of weakly stable W-types, along the same lines as the other definitions of weakly stable constructors so far.

Definition 3.4.4.9. Suppose C is equipped with pseudo-stable dependent products. A *weakly stable* W-type for Γ, A, B (over these dependent products) is $(\mathsf{W}, \mathsf{fold})$ as above, such that for each $\sigma : \Delta \longrightarrow \Gamma$, there is some wrec making $(\mathsf{W}[\sigma], \mathsf{fold}[\sigma], \mathsf{wrec})$ a W-type for Δ, $A[\sigma]$, $B[\sigma]$.

(Here the reindexing $\mathsf{fold}[\sigma]$ is taken with domain $\Delta.\mathsf{W}[\sigma].\Pi[B[\sigma], \mathsf{W}[\sigma]]$, which by pseudo-stability is a valid reindexing of $\Gamma.\mathsf{W}.\Pi[B, \mathsf{W}]$.)

We say C has *weakly stable* W-types (over the given dependent products) if, for each Γ, A, B, there exists some weakly stable W-type.

Lemma 3.4.4.10. *If C has pseudo-stable dependent products, and weakly stable* W*-types over these, then $C_!$ has strictly stable* W*-types over the strictly stable dependent products given by Scholium 3.4.2.6 together with Prop. 3.4.2.7.*

In maximal generality, one might not want to assume that the Π-types are any more than weakly stable themselves. Defining W-types over these, that can lift to $C_!$, is a little more involved.

(In fact, dependent products are pseudo-stable in all examples we know of, so for W-types this more general definition is never really needed. However, it is illustrative of a more general situation that does arise in practice: constructors that depend on others previously defined, where the earlier ones (identity types, perhaps) are only weakly stable. See Heuristic 4.1.1 below for more on this point.)

Definition 3.4.4.11. Suppose C is equipped with a stable class $\mathcal{G}$ of Π-types. C has *weakly stable* W-types over $\mathcal{G}$, if:

- for each Γ, A, B as above, there is some $\mathsf{W} \in \mathcal{T}(\Gamma)$ such that
- for each $\sigma : \Gamma' \longrightarrow \Gamma$, any reindexings A', B', W' of A, B, W along σ, and any good Π-type $(\Pi[B', \mathsf{W}'], \mathsf{app}_{B',\mathsf{W}'})$, there is some map $\mathsf{fold} : \Gamma'.A'\Pi[B', \mathsf{W}'] \longrightarrow \Gamma'.\mathsf{W}'$, over Γ, such that
- for each further $\sigma' : \Gamma'' \longrightarrow \Gamma'$, reindexings A'', B'', W'', type $C \in \mathcal{T}(\Gamma''.\mathsf{W}'')$, good Π-type $(\Pi[B'', C[\mathsf{app}'_{B',\mathsf{W}'}[\sigma]]],\ \mathsf{app}_{B'', C[\mathsf{app}'_{B',\mathsf{W}'}[\sigma]]})$, and map $d : \Gamma''.A''.\Pi[B', \mathsf{W}'][\sigma].\Pi[B'', C[\mathsf{app}'_{B',\mathsf{W}'}[\sigma]]] \to \Gamma''.\mathsf{W}''.C$ over $\mathsf{fold}[\sigma']$ as in the square (1) above, a section wrec of C, such that
- for each further reindexing of everything along some map $\sigma'' : \Gamma''' \longrightarrow \Gamma''$, and good λ-abstraction $\lambda(\mathsf{wrec}[\sigma''] \circ \mathsf{app}'_{B',\mathsf{W}'}[\sigma'][\sigma''])$, the square corresponding to (2) above commutes.

As one would hope, these definitions of weakly stable W-types agree:

Proposition 3.4.4.12. *Suppose C is equipped with pseudo-stable Π-types. Then C has weakly stable W-types in the sense of Def. 3.4.4.9 over this pseudo-stable structure if and only if it has weakly stable W-types in the sense of Def. 3.4.4.11 over the corresponding stable class defined in Prop. 3.4.2.7.* □

We may now lift weakly stable structure to $C_!$, using the techniques established above. Once again, nothing surprising occurs, and no new subtleties arise.

Lemma 3.4.4.13. *Suppose C satisfies (LF), and is equipped with a stable class of Π-types, and weakly stable W-types relative to these. Then $C_!$ carries strictly stable W-types, over the strictly stable Π-types from Scholium 3.4.2.6.* □

This completes the proof of Theorem 3.4.1. □

4. Further notes

4.1. **Generalization to other rules.** We have given Theorem 3.4.1 just for (a selection of) the standard constructors and rules of Martin-Löf type theory. However, one of the hallmarks of type theory is its extensibility. One usually wants to consider at least some other rules and constructors beyond these; a technique applying only to this standard core would be highly limited in its utility. A word is therefore in order on how the present results extend to rules and constructors beyond those considered above.

Heuristic 4.1.1. *The definition of* strictly stable *structure and the initiality of the syntactic category extend straightforwardly to all reasonable constructors and rules of type theory.*

The definition of weakly stable *structure seems to extend to all reasonable constructors and rules, in general along the (slightly involved) lines of the case of W-types in Section 3.4.4 above.*

Given these, the lifting of weakly stable structure on C to strictly stable structure on $C_!$ extends straightforwardly to all reasonable finitary rules and constructors, except for type equality rules*, exactly along the lines of the cases given in Section 3.4. The finitariness condition may be removed by strengthening (LF) appropriately.*

For the case of strictly stable structure, this is well-understood and generally believed in the community; but, to our knowledge, no precise statement of it has been formulated. Given such a formulation, one would hope that the present heuristics could be made precise, and the results of this paper given in some more satisfying generality. However, setting up such a general framework is beyond the scope of this paper; so for now we treat the general case informally.

The case of weak stability is less clear; extending the definition is in general rather trickier than most of the cases considered in this paper might suggest.

The complication comes from the dependence of later rules on earlier ones. Since most standard constructors fall into independent groups, we have "cheated" slightly within these groups, and given weak stability in a slightly simpler form than the general approach would provide. In the case of W-types, however, we see the complications that arise with successive dependency of rules.

The inputs to each operation (or existence condition) are interpretations of the premises of the corresponding rule, which may involve previously given constructors. For instance, the inputs to W-elimination involve Π-types. If those previous constructors are only weakly stable, then the types in the premises will not have canonically chosen interpretations; so one must quantify over all possible choices, or at least over some reasonable class of choices.

So in general, for each operation, one considers C as equipped with a *stable class* of interpretations, called *good*; and in the inputs to each operation, one quantifies over all good interpretations of the types involved in the premises. Since these will in general be derived types, not just primitives, this depends on being able to extend the notion of good interpretations to derived types.

This consideration is subtle enough that, without a precise formulation, we cannot confidently claim that this approach extends to *all* reasonable rules. However, for all the rules that we have investigated in this connection, it extends without further complications.

(If a constructor does not appear in subsequent rules, then there is no need to distinguish a class of good interpretations; one may without loss of generality consider all weakly stable interpretations as good, and hence to ask that for any given input, some such interpretation exists.)

Once weak stability is appropriately defined, the lifting from C to $C_!$ is generally straightforward, modulo two limitations.

Firstly, with (LF) in its present form, one can lift only finitary rules, since we have just finite limits with which to construct representing objects for premises. To lift infinitary rules, one needs to strengthen (LF) by assuming appropriate larger limits in C.

Secondly, and less negotiably, the lifting works just for rules whose conslusions are type constructors, term constructors, or term equalities. It does *not* work for type equality rules. Even when some equality of derived types holds strictly in C, their liftings to $C_!$ will almost always have different local universes.

The most notable type equality rules considered in practice are the constructor commutation rules for universes à la Tarski [ML84, p. 88], and (in the absence of universes) large eliminators for simple inductive types. In each case, one may replace these rules with forms not involving type equalities, which will then lift.

Most straightforwardly, one can simply replace the equalities by (terms representing) equivalences of types.

Alternatively, in the case of operations on universes, one may directly equip the results of operations with the appropriate structure. For instance, if $(\mathsf{V}, \mathsf{el}_{\mathsf{V}})$ is a universe, and $+^{\mathsf{V}} : \mathsf{V} \times \mathsf{V} \longrightarrow \mathsf{V}$ is the sum-types operation on V, then the standard commutation rule states that $\mathsf{el}_{\mathsf{V}}(a +^{\mathsf{V}} b) = \mathsf{el}_{\mathsf{V}}(a) + \mathsf{el}_{\mathsf{V}}(b)$. Instead, one may posit inclusion maps and an eliminator directly exhibiting $\mathsf{el}_{\mathsf{V}}(a +^{\mathsf{V}} b)$ as a sum for $\mathsf{el}_{\mathsf{V}}(a)$ and $\mathsf{el}_{\mathsf{V}}(b)$, independently of any globally defined sum types. This is interestingly analogous to weak stability, replacing an explicit commutation condition by the preservation of the universal property.

In either case, it seems that in replacing type equality rules with these weaker forms, one loses only convenience, not logical strength. Again, however, this is somewhat heuristic and conjectural: generally believed based on practical experience, but not (to our knowledge) known in any precise form.

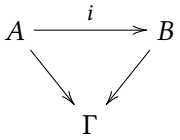
4.2. **Applications.** Our main motivating examples are homotopy-theoretic in nature, along the lines of [AW09, War08, dBG12, Voe11], and similar. The slogan for all such models is *types as fibrations*.

Specifically, any weak factorization system (or algebraic wfs) on a finitely complete category $\mathcal{E}$ gives a comprehension category $\mathcal{T} \longrightarrow \mathcal{E}$ over $\mathcal{E}$, with $\mathcal{T}(\Gamma)$ the category of right maps into Γ. (We will refer to the right maps of the factorization system as *fibrations*.) When it is clear which weak factorization system is under consideration, we refer to the resulting comprehension category again simply as $\mathcal{E}$. Similarly, we write $\mathcal{E}_f$ for the full sub comprehension category on the fibrant objects of $\mathcal{E}$.

If (the underlying maps of) fibrations are exponentiable, then the ambient hypothesis (LF) applies; so to model type theory, we need only show that appropriate weakly stable structure exists. As shown in various recent work—[AW09, War08, AK11]—this structure often follows from well-known homotopy-theoretic facts. Various combinations of homotopy-theoretic properties turn out to suffice. One particularly fruitful combination is the following (the terminology is taken from [AK11]; we modify their definition somewhat):

Definition 4.2.1. A *logical weak factorization system on* $\mathcal{E}$ is a weak factorization system on $\mathcal{E}$, such that:

(a) fibrations are exponentiable: for any fibration $p : Y \longrightarrow X$, the pullback functor $p^* : \mathcal{E}/X \longrightarrow \mathcal{E}/Y$ has a right adjoint Π_p;
(b) left maps are preserved by pullback along fibrations (equivalently, fibrations are preserved by exponentiation along fibrations); and
(c) any left map i between fibrations over a common base Γ



is *substitution-stable in* Γ, i.e. its pullback along any map $f : \Gamma' \longrightarrow \Gamma$ is again a left map.

A *semi-logical weak factorization system* is as above, but with the conditions required only for fibrations over fibrant bases.

A *(semi-)logical model structure* is a model structure whose (trivial cofibration, fibration) weak factorization system is (semi-)logical.

Theorem 4.2.2. *Suppose $\mathcal{E}$ is a finitely complete category, with stable finite coproducts, equipped with a weak factorization system.*

If the weak factorization system is semi-logical, then $(\mathcal{E}_f)_!$ models type theory with Π-, Σ-, *unit,* Id-, *and finite sum types. If it is moreover logical, then so does $\mathcal{E}_!$.*

Proof. The assumption of finite completeness, together with (a), ensures that condition (LF) applies; so by the results of Section 3 it suffices to construct the desired weakly stable structure on $\mathcal{E}$. Identity types are constructed as in [AW09, Thm. 3.1] and [War08, Thm. 2.17] The other structure is along standard categorical lines, with condition (c) providing weak stability (compare [AK11, Thm. 26]). □

For recognizing (semi-)logical weak factorization systems, it is often convenient to replace (c) with a simpler equivalent criterion:

Lemma 4.2.3. *In any weak factorization system satisfying (b) above, the following are equivalent:*

(i) Any left map between fibrations over a common base Γ is substitution-stable in Γ.

(ii) Any map between fibrations over a common base Γ has an $(\mathcal{L},\mathcal{R})$ factorization whose $\mathcal{L}$-factor is substitution-stable in Γ.

(iii) For any fibration $p : X \twoheadrightarrow \Gamma$, the diagonal map $\Delta_p : X \twoheadrightarrow X \times_\Gamma X$ has an $(\mathcal{L},\mathcal{R})$ factorization whose $\mathcal{L}$-factor is substitution-stable in Γ.

Moreover, they remain equivalent when restricted to fibrant bases.

Proof. The implications (i) ⇒ (ii) ⇒ (iii) are immediate. For the converses, (ii) ⇒ (i) is a standard retract argument, while (iii) ⇒ (ii) is analogous to the proof of [GG08, Lemma 4.2.2]. □

A stronger condition, but usually easier to verify when it holds, is:

Lemma 4.2.4. *Let $\mathcal{E}$ be a model category in which cofibrations are stable under pullback. Then condition (c) holds in $\mathcal{E}$.*

Proof. By Ken Brown's lemma, weak equivalences between fibrations over a base are always stable under pullback in the base. Thus, if cofibrations are also stable under such pullbacks, so are trivial cofibrations. □

Examples 4.2.5. The following is a (non-exhaustive) list of examples to which Theorem 4.2.2 is easily seen to apply, using the lemmas above. (Cf. [Shu15, Exs. 2.16].)

1. The "canonical" model structures on **Cat** and **Gpd** are logical (since isofibrations are exponentiable in **Cat**) [JT91].
2. The usual (Kan/Quillen) and quasicategory (Joyal) model structures on **SSets** [Joy08] are both logical.
3. Any right proper Cisinski model structure [Cis06] is logical.
4. Any locally cartesian closed right proper simplicial model category in which the cofibrations are monomorphisms is logical [War08, Cor. 2.46]. (Examples include simplicial presheaves and simplicial sheaves.)
5. For any cofibrantly generated logical model category $\mathcal{C}$ and small category $\mathcal{J}$, the injective and projective model structures on $\mathcal{C}^{\mathcal{J}}$ are logical.
6. If $\mathcal{C}$ is any logical model category, and $\mathcal{J}$ is an inverse category, then the Reedy (equivalently, injective) model structure on $\mathcal{C}^{\mathcal{J}}$ is logical. [Shu15, §11] treats this example in detail, showing moreover that univalent universes in $\mathcal{C}$ lift to univalent universes in $\mathcal{C}^{\mathcal{J}}$, using the results of the present paper to obtain coherence.

More examples of (semi-)logical weak factorization sytems are given (with slightly different terminology) in [Awo14, §3.3].

References

[AK11] Peter Arndt and Krzysztof Kapulkin, *Homotopy-theoretic models of type theory*, Typed Lambda Calculi and Applications, Springer, Berlin, Heidelberg, 2011, pp. 45–60.

[AW09] Steve Awodey and Michael A. Warren, *Homotopy theoretic models of identity types*, Math. Proc. Camb. Phil. Soc. **146** (2009), 45–55.

[Awo14] Steve Awodey, *Natural models of homotopy type theory*, Preprint, 2014, arXiv:1406.3219, doi:10.1017/bsl.2014.9.

[dBG12] Benno van den Berg and Richard Garner, *Topological and simplicial models of identity types*, ACM Trans. Comput. Log. **13** (2012), no. 1, Art. 3, 44, arXiv:1007.4638v1, doi:10.1145/2071368.2071371.

[Car86] John Cartmell, *Generalised algebraic theories and contextual categories*, Ann. Pure Appl. Logic **32** (1986), no. 3, 209–243.

[CGH14] Pierre-Louis Curien, Richard Garner, and Martin Hofmann, *Revisiting the categorical interpretation of dependent type theory*, Theor. Comput. Sci. **546** (2014), 99–119, doi:10.1016/j.tcs.2014.03.003.

[Cis06] Denis-Charles Cisinski, *Les préfaisceaux comme modèles des types d'homotopie*, Astérisque **308** (2006), xxiv–390.

[GG08] Nicola Gambino and Richard Garner, *The identity type weak factorisation system*, Theoret. Comput. Sci. **409** (2008), no. 1, 94–109, arXiv:0803.4349, doi:10.1016/j.tcs.2008.08.030.

[Gir65] Jean Giraud, *Cohomologie non abélienne*, C. R. Acad. Sci. Paris **260** (1965), 2666–2668.

[Hof95] Martin Hofmann, *On the interpretation of type theory in locally Cartesian closed categories*, Computer science logic (Kazimierz, 1994), Lecture Notes in Comput. Sci., vol. 933, Springer, Berlin, Heidelberg, 1995, pp. 427–441, doi:10.1007/BFb0022273.

[Hof97] ———, *Syntax and semantics of dependent types*, Semantics and logics of computation (Cambridge, 1995), Publ. Newton Inst., vol. 14, Cambridge Univ. Press, Cambridge, 1997, pp. 79–130, doi:10.1017/CBO9780511526619.004.

[HS9?] Martin Hofmann and Thomas Streicher, *Lifting Grothendieck universes*, Unpublished note, 199?, http://www.mathematik.tu-darmstadt.de/~streicher/.

[HS98] ———, *The groupoid interpretation of type theory*, Twenty-five years of constructive type theory (Venice, 1995), Oxford Logic Guides, vol. 36, Oxford Univ. Press, New York, 1998, pp. 83–111.

[Jac93] Bart Jacobs, *Comprehension categories and the semantics of type dependency*, Theoret. Comput. Sci. **107** (1993), no. 2, 169–207, doi:10.1016/0304-3975(93)90169-T.

[Joy08] André Joyal, *The theory of quasi-categories and its applications*, Quaderns **45** (2008), no. 2, 151–496, Vol. II of course notes from Simplicial Methods in Higher Categories, http://www.crm.es/HigherCategories/notes.html.

[JT91] André Joyal and Myles Tierney, *Strong stacks and classifying spaces*, Category theory (Como, 1990) (Aurelio Carboni, Maria Cristina Pedicchio, and Guiseppe Rosolini, eds.), Lecture Notes in Math., vol. 1488, Springer, Berlin, Heidelberg, 1991, pp. 213–236, doi:10.1007/BFb0084222.

[KLV12] Chris Kapulkin, Peter LeF. Lumsdaine, and Vladimir Voevodsky, *The simplicial model of univalent foundations*, Submitted, 2012, arXiv:1211.2851.

[ML84] Per Martin-Löf, *Intuitionistic type theory*, Studies in Proof Theory. Lecture Notes, vol. 1, Bibliopolis, Naples, 1984.

[NPS90] Bengt Nordström, Kent Petersson, and Jan M. Smith, *Programming in Martin-Löf's type theory*, International Series of Monographs on Computer Science, vol. 7, Clarendon Press, Oxford University Press, New York, 1990.

[See84] Robert A. G. Seely, *Locally Cartesian closed categories and type theory*, Math. Proc. Cambridge Philos. Soc. **95** (1984), no. 1, 33–48, doi:10.1017/S0305004100061284.

[Shu15] Michael Shulman, *Univalence for inverse diagrams and homotopy canonicity*, Mathematical Structures in Computer Science **FirstView** (2015), 1–75, arXiv:1203.3253, doi:10.1017/S0960129514000565, http://journals.cambridge.org/article_S0960129514000565.

[Str91] Thomas Streicher, *Semantics of type theory*, Progress in Theoretical Computer Science, Birkhäuser Boston Inc., Boston, MA, 1991, Correctness, completeness and independence results.

[Str14] ———, Personal communication, June 2014.

[Uni13] The Univalent Foundations Program, *Homotopy type theory: Univalent foundations of mathematics*, Tech. report, Institute for Advanced Study, 2013.

[Voe11] Vladimir Voevodsky, *Notes on type systems*, Unpublished notes, April 2011, http://www.math.ias.edu/~vladimir/Site3/Univalent_Foundations_files/expressions_current.pdf.

[Voe13] ———, *A simple type system with two identity types*, Ongoing unpublished notes, 2013, https://uf-ias-2012.wikispaces.com/file/view/HTS.pdf/410120566/HTS.pdf.

[War08] Michael A. Warren, *Homotopy theoretic aspects of constructive type theory*, Ph.D. thesis, Carnegie Mellon University, 2008.

[War11] ———, *The strict ω-groupoid interpretation of type theory*, Models, logics, and higher-dimensional categories, CRM Proc. Lecture Notes, vol. 53, Amer. Math. Soc., Providence, RI, 2011, pp. 291–340.

Department of Mathematics, Stockholm University, Stockholm, Sweden
E-mail address: p.l.lumsdaine@gmail.com

Los Angeles, California, USA