SJÄLVSTÄNDIGA ARBETEN I MATEMATIK

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET

Dependent products as relative adjoints

av

Michael Lindgren

2021 - No K43

Dependent products as relative adjoints

Michael Lindgren

Självständigt arbete i matematik 15 högskolepoäng, grundnivå

Handledare: Peter LeFanu Lumsdaine

2021

Dependent products as relative adjoints

Michael Lindgren

October 6, 2021

Abstract

Universal quantification in logic correspond, in categorical models, to right adjoints to weakening. However, in certain weak models of dependent type theory, known as comprehension categories, this is not always the case.

In dependent type theory, universal quantification is given by the dependent product type. In some comprehension categories, those called *full*, dependent products do correspond to right adjoints to weakening, but this does not hold in general.

In this thesis we present a new description of dependent products in comprehension categories as *relative adjoints*, and show that this holds in arbitrary comprehension categories.

Acknowledgements

I would like to thank my thesis advisor, Peter LeFanu Lumsdaine, for suggesting this topic and for providing me with invaluable feedback along the way.

Contents

1 Introduction

It was first recognised by Lawvere in the 1960s [Law70] that quantifiers in first-order logic (or generally, in intuitionistic higher-order logic) correspond to adjoints in toposes. We illustrate the syntactic presentation of these adjunctions with the following example. Let $\bar{\mathbf{x}}$ denote a list of variables $\bar{\mathbf{x}} = x_1, x_2, \ldots, x_n$. Let $*$ be the operation taking a formula $\varphi(\bar{\mathbf{x}})$, with at most the variables in $\bar{\mathbf{x}}$ free, to the same formula considered as a formula with at most $(\bar{\mathbf{x}}, y)$ free (where y is a fresh variable). The rules for $\forall$ give the following two-way rule

$$\frac{*\varphi(\bar{\mathbf{x}}) \vdash \psi(\bar{\mathbf{x}}, y)}{\varphi(\bar{\mathbf{x}}) \vdash \forall y.\psi(\bar{\mathbf{x}}, y)}$$

where the direction top-to-bottom is the $\forall$-introduction rule, and the converse is $\forall$-elimination. Essentially this example demonstrates, without going into all the details, that $\forall$ is a *right adjoint* to the weakening operation $*$:

$$* \dashv \forall.$$

Likewise one can show that the existential quantifier, $\exists$, is a *left adjoint* to weakening: $\exists \dashv *$. Full details of this example can be found in [Awo10].

In the 1970s and 1980s Per Martin-Löf developed his intuitionistic dependent type theory (MLTT), with dependent products Π and dependent sums Σ which under the Curry-Howard correspondence are analogous to the quantifiers $\forall$ and $\exists$ respectively (see [Mar84] for an introduction). Hofmann [Hof95], improving on work by Seely in [See84], constructed a model of dependent type theory (a Category with Attributes [Car86]) from locally cartesian closed categories, allowing one to interpret the syntax of type theory in a LCCC.

A categorical structure more general than LCCCs, Categories with Attributes and Toposes, that has proven to be useful when constructing models of dependent type theory, is *Comprehension Categories*, introduced by Jacobs in [Jac93]. Comprehension categories utilizes the language of fibred category theory to interpret contexts, types and type dependencies of dependent type theory as *structure* on a fibration of categories.

Comprehension categories are in general not strict models of dependent type theory. However, it is often possible to obtain a CwA from a comprehension category, that is a model of the intended type theory.

Jacobs showed, as Lawvere did for quantifiers in topoi, that quantifiers of dependent type theory in *full* comprehension categories are given by fibred adjoint functors to the weakening functor $\langle \chi \rangle$. For example universal quantification, denoted by Π, is a fibred right adjoint:

$$\langle \chi \rangle \dashv \Pi.$$

This holds for comprehension categories with the extra assumption that they are *full*, meaning that morphisms between types correspond bijectively to morphisms between projections of extended contexts [Jac93].

In this thesis we generalise Jacobs' description of quantifiers as adjoints, to comprehension categories *without* this extra assumption, and show that universal quantification in arbitrary comprehension categories correspond to a fibred *relative* right adjoint

$$\mathcal{W} \dashv_{\chi} \chi \circ \Pi.$$

2 Preliminaries

2.1 Relative adjunctions

In this section we look at a generalisation of adjoint functors between categories in what is called *relative* adjoint functors, introduced by Ulmer in [Ulm68].

Definition 2.1.0.1 ([Ulm68, Definition 2.2])**.** Let $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ be categories, and $F\colon \mathbf{A} \to \mathbf{B}$, $J\colon \mathbf{A} \to \mathbf{C}$ and $U\colon \mathbf{B} \to \mathbf{C}$ be functors. F is called a left adjoint to U relative to J if for each pair of objects $a \in \mathbf{A}$, $b \in \mathbf{B}$ there is a bijection

$$\Phi_{a,b}\colon \mathbf{B}(Fa, b) \cong \mathbf{C}(Ja, Ub) \tag{1}$$

natural in a and b. We say that F is J-left adjoint to U and use the notation $F \dashv_J U$.

Dually, F is said to be J-right adjoint to U if for each pair $a \in \mathbf{A}$, $b \in \mathbf{B}$ there is a bijection

$$\Theta_{a,b}\colon \mathbf{B}(b, Fa) \cong \mathbf{C}(Ub, Ja) \tag{2}$$

natural in a and b, and we write $U \dashv_J F$ in this case.

Note that the definitions above are asymmetrical. F being a J-left adjoint of U does not imply that U is a J-right adjoint of F. It is however the case when one take J to be the identity functor, where one obtain the usual hom-set definition of adjoint functors.

While adjoint functors induce two natural transformations - the unit and the counit - relative adjunctions induce only one:

$F \dashv_J U$	determines a relative unit	$\eta\colon J \Rightarrow UF$
$U \dashv_J F$	determines a relative counit	$\epsilon\colon UF \Rightarrow J$,

obtained by evaluating (1) and (2) at Id_{Fa}.

The following lemma will be useful later, and is proved in the same way as the corresponding lemma for adjoint functors. A similar statement exists for J-right relative adjoint functors.

Lemma 2.1.0.2 ([Ulm68, Lemma 2.7])**.** *Let* $\mathbf{A}$, $\mathbf{B}$ *and* $\mathbf{C}$ *be categories and* $F\colon \mathbf{A} \to \mathbf{B}$, $J\colon \mathbf{A} \to \mathbf{C}$ *and* $U\colon \mathbf{B} \to \mathbf{C}$ *functors. Then* F *is* J*-left relative adjoint of* U *iff there exists a natural transformation* $\psi\colon J \Rightarrow UF$ *such that for each pair* $a \in \mathbf{A}$, $b \in \mathbf{B}$ *the composed map*

$$\mathbf{B}(Fa, b) \xrightarrow{U} \mathbf{C}(UFa, Ub) \xrightarrow{\psi} \mathbf{C}(Ja, Ub)$$

is a bijection.

A noteworthy difference between adjunctions and *relative* adjunctions is that the latter does not generalise to 2-categories. The definition 2.1.0.1 refers to a natural isomorphism of hom-sets, just as a natural isomorphism is given as a definition of adjoint functors. This makes sense in the 2-category **Cat** where every object is a category, and we can refer to their internal collections of objects and arrows when defining adjunctions and relative adjunctions. In a general 2-category the objects are not categories, however. To define adjunctions in a 2-category one can observe that equivalent to the hom-set definition of adjoint functors is to require two natural transformations $\eta\colon \mathrm{Id} \Rightarrow LR$, $\epsilon\colon RL \Rightarrow \mathrm{Id}$ that

pastes together appropriately (the "triangle identities"). This can be translated to the language of 2-categories in such a way that it captures both the essence of adjunctions, and recovers adjoint functors when specialised to **Cat**. We will give yet another, equivalent, definition of adjunctions in 2-categories in terms of *absolute liftings* which we will define next.

Definition 2.1.0.3 ([SW78])**.** For 1-cells $f : a \to c, p : b \to c$ in a 2-category, a left lifting of f through p is a pair $\mathrm{Lift}_p f = (\hat{f}, \eta)$ consisting of a 1-cell $\hat{f} : a \to b$ and a 2-cell $\eta : f \Rightarrow p \circ \hat{f}$ such that for any other pair $(g : a \to b, \gamma : f \Rightarrow p \circ g)$ there is a unique 2-cell $\chi : \hat{f} \Rightarrow g$ such that $\gamma = (p \triangleleft \chi) \bullet \eta$, where $\triangleleft$ stands for horizontal composition of a 2-cell with a 1-cell (whiskering) to the right. A left lifting is *absolute* if for any $h : x \to a$ the lift of the composite $f \circ h$ is the lift of f composed with h, that is $\mathrm{Lift}_p(f \circ h) = (\hat{f} \circ h, \eta \triangleright h)$.

One can observe that a functor $\mathrm{R} : \mathrm{X} \to \mathrm{Y}$ is a right adjoint iff there is an absolute left lifting $(\mathrm{L}, \eta) = \mathrm{Lift}_{\mathrm{R}} \mathrm{Id}_{\mathrm{Y}}$. Similarly one can verify that given functors F, J, U as in 2.1.0.1 that F is a J-left relative adjoint of U if and only if there is a natural transformation (the relative counit) $\eta : \mathrm{J} \Rightarrow \mathrm{UF}$ that exhibits F as an absolute left lifting of J through U.

It is tempting to define relative adjunctions in 2-categories as absolute left liftings, but there is a caveat, namely that while the objects of a 2-category are not in general categories, there is a way to describe the structure that a 2-category must have to possess at least some of the properties that we associate with a 2-category of categories. Such a structure, called a Yoneda Structure, was introduced in [SW78], and includes a notion of "local smallness", presheaf objects and Yoneda embeddings. For 1-cells $j : \mathrm{A} \to \mathrm{C}$, $s : \mathrm{A} \to \mathrm{B}$, $t : \mathrm{B} \to \mathrm{C}$ in a 2-category with Yoneda structure one can define a relative adjunction as a 2-cell isomorphism $\pi : \mathrm{B}(s, 1) \Rightarrow \mathrm{C}(j, t)$. A consequence of the axioms of a Yoneda structure is that a relative adjunction, as defined by [SW78], implies that s is an absolute left lifting of j through t. However, the converse does not always hold. There are examples (see discussion at [Cam14]) of 2-categories with a Yoneda structure where an absolute left lifting s of j through t does not imply a 2-cell isomorphism $\mathrm{B}(s, 1) \Rightarrow \mathrm{C}(j, t)$. Therefore the notions of *absolute left lifting* and *relative adjoint* are not always compatible. In short, a *lifting* is definable in any 2-category, a *relative adjoint* is definable only in a 2-category with a Yoneda structure, a relative adjoint always implies an absolute lifting, an absolute lifting does *not* imply a relative adjoint.

2.2 Fibred category theory

This is a short introduction to fibrations and fibred category theory, following [Jac93] and [Jac99]. We include these definitions and constructions in order to be able to define comprehension categories in the next section. For the remainder of this section let $\mathbf{E}$ and $\mathbf{B}$ be categories and $p : \mathbf{E} \to \mathbf{B}$ a functor.

Definition 2.2.0.1 (Cartesian morphism)**.** A morphism $u : a \to b$ in $\mathbf{E}$ is *cartesian* over $f : x \to y$ in $\mathbf{B}$ if $p(u) = f$ and for each $v : c \to b$ in $\mathbf{E}$ such that $p(v) = f \circ g$ for some $g : pc \to x$ there exists a unique morphism $w : c \to a$ such that $p(w) = g$ and $v = u \circ w$, as illustrated in figure 1.

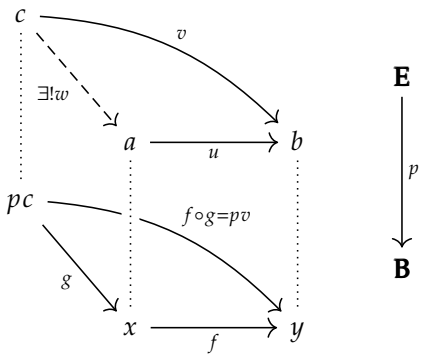


Figure 1: For u cartesian above f and arbitrary v we can lift any factorization of $p(v)$ through f in $\mathbf{B}$ to a factorization of v through u in $\mathbf{E}$.

The diagram in figure 1 is suggestive of the following terminology, which will be used throughout this thesis. We say that an object x in $\mathbf{E}$ is *above* an object b in $\mathbf{B}$ if $px = b$. Similarly, a morphism u is *above* f if $p(u) = f$. If $p(u) = \mathrm{Id}_y$ for some object y in $\mathbf{B}$ we say that the morphism u is *vertical* over y, or that u is above y.

Definition 2.2.0.2 (Fibration)**.** A functor $p : \mathbf{E} \to \mathbf{B}$ is a *fibration* if for every object a in $\mathbf{E}$ and morphism $f : x \to pa$ in $\mathbf{B}$ there is a cartesian morphism above f.

Note that the definition of fibration only gives mere existence of cartesian lifts, which necessitates the following definition.

Definition 2.2.0.3 (Cleavage)**.** A *cleavage* for a fibration $p : \mathbf{E} \to \mathbf{B}$ is a collection of chosen cartesian liftings $\{f_{\mathrm{A}}\}$; that is, for each object $\mathrm{A} \in \mathbf{E}$ and morphism $f : x \to p(\mathrm{A})$ in $\mathbf{B}$ a *chosen* cartesian morphism f_{A} above f. A fibration is *cloven* if it is equipped with a cleavage.

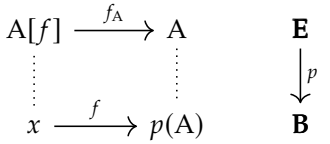


Figure 2: f_{A} is the chosen cartesian lifting of f into A.

Fibrations will henceforth be assumed to be cloven. Figure 2 illustrates some of the notation that will be used frequently from now on. When the name of a morphism occurs with the name of an object above its codomain as a subscript then what is referred to is the *chosen* cartesian lifting of that morphism. For example: f_{A} refers to the chosen cartesian lifting of f into the object A above the codomain of f. Similarly, $\mathrm{A}[f]$ is notation for the domain of the chosen cartesian lifting of f into A above the domain of f.

Definition 2.2.0.4 (Split fibration)**.** A fibration $p : \mathbf{E} \to \mathbf{B}$ is called *split* if it comes equipped with a cleavage such that

- the chosen cartesian liftings of identity morphisms are identity morphisms;
- for morphisms $a \xrightarrow{f} b \xrightarrow{g} c$ in $\mathbf{B}$ and an object A above c in $\mathbf{E}$,

$$(g \circ f)_{\mathrm{A}} = g_{\mathrm{A}[f]} \circ f_{\mathrm{A}}.$$

A cleavage satisfying the above is called a *splitting* of p. It is possible to modify any cleavage to satisfy the first of the properties above, but the second property is in general not as easily obtainable.

Definition 2.2.0.5 (Fiber)**.** For a fibration $p\colon \mathbf{E} \to \mathbf{B}$ we call $\mathbf{B}$ the *base* category and $\mathbf{E}$ the *total* category. For an object b in $\mathbf{B}$, the *fiber* $\mathbf{E}_b$ is the category with objects above b and *vertical* morphisms between them.

Definition 2.2.0.6 (Reindexing functor)**.** Let $p\colon \mathbf{E} \to \mathbf{B}$ be a (cloven) fibration. Then every morphism $f : \Delta \to \Gamma$ in $\mathbf{B}$ determines a functor $f^*\colon \mathbf{E}_\Gamma \to \mathbf{E}_\Delta$ between fiber categories, defined on objects A in $\mathbf{E}_\Gamma$ as $f^*(\mathrm{A}) = \mathrm{A}[f]$ and on morphisms $\phi\colon \mathrm{A} \to \mathrm{C}$ as the unique mediating morphism $\mathrm{A}[f] \to \mathrm{C}[f]$ given by the universal property of the cartesian morphism f_{C}. It follows from the uniqueness property that f^* preserves identities and composition.

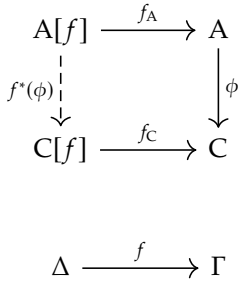


Figure 3: Action of the reindexing functor f^* on morphisms.

Definition 2.2.0.7 (Fibred functor)**.** Let $p\colon \mathbf{E} \to \mathbf{B}$ and $q\colon \mathbf{D} \to \mathbf{B}$ be fibrations over the same base category $\mathbf{B}$. A functor $\mathrm{F}\colon \mathbf{E} \to \mathbf{D}$ is called a *fibred functor* (also called a *cartesian functor*) if it preserves cartesian arrows and $q \circ \mathrm{F} = p$ strictly.

Definition 2.2.0.8 (Fibrations over a base)**.** For a category $\mathbf{B}$ let $\mathbf{Fib}(\mathbf{B})$ be the 2-category with fibrations into $\mathbf{B}$ as objects, fibred functors as morphisms and vertical natural transformations. Similarly, let $\mathbf{SplFib}(\mathbf{B})$ be the 2-category of *split* fibrations over $\mathbf{B}$.

In general a cloven fibration is not isomorphic to a split fibration, but a result due to Bénabou (detailed in [Str20]) is that any fibration $p\colon \mathbf{E} \to \mathbf{B}$ is *equivalent* to a split fibration $p_\star\colon \mathbf{E}_\star \to \mathbf{B}$ via a right adjoint $(-)_\star\colon \mathbf{Fib}(\mathbf{B}) \to \mathbf{SplFib}(\mathbf{B})$ to the inclusion $\mathbf{SplFib}(\mathbf{B}) \hookrightarrow \mathbf{Fib}(\mathbf{B})$. For a more details the reader is referred to the source material by Bénabou, or one of the more recent expositions in for example [Str20] and [Jac99].

Definition 2.2.0.9 (Right-adjoint splitting)**.** Given a cloven fibration $p\colon \mathbf{E} \to \mathbf{B}$ define $\mathbf{E}_\star$ to be the category where

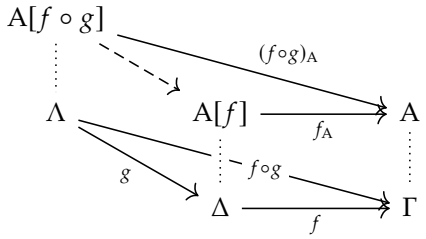
- objects are pairs $(A, A[-])$ where A is an object in $\mathbf{E}$ over some Γ in $\mathbf{B}$ and $A[-]$ is the operation assigning to each morphism f into Γ the cartesian lifting of f into A given by the cleaving of p;
- morphisms $(A, A[-]) \to (D, D[-])$ are morphisms $A \to D$ in $\mathbf{E}$.

Let $p_\star$ be the fibration $\mathbf{E}_\star \to \mathbf{B}$ given by applying p to the first component of the pairs in $\mathbf{E}_\star$. This fibration is split with splitting defined as follows.

- For $f : \Delta \to \Gamma$ and $(A, A[-])$ over Γ, let the cartesian lift of f into $(A, A[-])$ be given by $f_A : A[f] \to A$, i.e. we take

$$f_{(A,A[-])} = f_A : (A[f], A[f][-]) \to (A, A[-]);$$

- for $g : \Lambda \to \Delta$, $f : \Delta \to \Gamma$ and $(A, A[-])$ above Γ define the cartesian lift of g into $(A[f], A[f][-])$ as the unique cartesian morphism above g completing the triangle in the diagram below.



Definition 2.2.0.10 (Fibred adjunction)**.** An adjunction in $\mathbf{Fib}(\mathbf{B})$ is given by fibred functors F, U as in

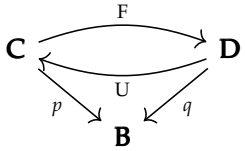


Figure 4

such that $F \dashv U$ as functors, and the natural transformations $\eta : \mathrm{Id}_{\mathbf{C}} \Rightarrow UF$ and $\epsilon : FU \Rightarrow \mathrm{Id}_{\mathbf{D}}$ given by the adjunction are *vertical*.

For $p : \mathbf{E} \to \mathbf{B}$ a fibration, $u : A \to B$ in $\mathbf{B}$, $X \in \mathbf{E}_A$ and $Y \in \mathbf{E}_B$, let

$$\mathbf{E}_u(X, Y) = \{f : X \to Y \mid pf = u\}$$

denote the collection of morphisms above u with specified domain X and codomain Y.

We make the observation that an adjunction of fibred functors is a fibred adjunction precisely when the hom-set isomorphism of the adjunction can be restricted to hom-sets above morphisms in the base category.

Lemma 2.2.0.11. *Let* $\mathrm{F} \dashv \mathrm{U}$ *be a fibred adjunction where* F *and* U *are fibred functors as in figure 4. Then the induced natural isomorphism of hom-sets* $\phi\colon \mathbf{D}(\mathrm{F}(-),-) \cong \mathbf{C}(-,\mathrm{U}(-))$ *is* fibred *in the sense that it restricts to isomorphisms of hom-sets over morphisms in the base, i.e. for each morphism* $u\colon \Delta \to \Gamma$ *in* $\mathbf{B}$ *and objects* c *in* $\mathbf{C}_\Delta$ *and* d *in* $\mathbf{D}_\Gamma$

$$\phi_u\colon \mathbf{D}_u(\mathrm{F}c, d) \cong \mathbf{C}_u(c, \mathrm{U}d).$$

Proof. Let $f\colon \mathrm{F}c \to d$ in $\mathbf{D}$. Since the unit component η_c given by the fibred adjunction is vertical and U is a fibred functor it follows that

$$p(\phi(f)) = p(\mathrm{U}(f) \circ \eta_c) = p(\mathrm{U}(f)) = q(f).$$

Applying the same argument to the counit gives the result. □

Lemma 2.2.0.12. *An adjunction of fibred functors* F *and* U *is a fibred adjunction if the natural isomorphism of hom-sets* $\Phi\colon \mathbf{D}(\mathrm{F}c, d) \cong \mathbf{C}(c, \mathrm{U}d)$ *given by the adjunction is an isomorphism of hom-sets over morphisms* $u\colon pc \to qd$ *in the base category.*

$$\Phi_{c,d}\colon \mathbf{D}_u(\mathrm{F}c, d) \cong \mathbf{C}_u(c, \mathrm{U}d)$$

Proof. The components of the unit is constructed by evaluating $\eta_c = \Phi_{c,\mathrm{F}c}(\mathrm{Id}_{\mathrm{F}c})$, but $\mathrm{Id}_{\mathrm{F}c}$ is above $\mathrm{Id}_{q\mathrm{F}c} = \mathrm{Id}_{pc}$ since q is a functor and F fibred. Using that Φ preserves morphisms above morphisms in the base it follows that $\eta_c\colon c \to \mathrm{UF}c$ is above Id_{pc}. The proof for the counit is analogous. □

The next proposition introduces an important method to obtain new fibrations from old, by changing the base of a fibration.

Proposition 2.2.0.13 (Change-of-base)**.** *The pullback in* **Cat** *of a fibration* $p\colon \mathbf{T} \to \mathbf{C}$ *and an arbitrary functor* $\mathrm{F}\colon \mathbf{A} \to \mathbf{C}$ *yields a fibration* $\mathrm{F}^\star(p)\colon \mathbf{A} \underset{\mathrm{F},p}{\times} \mathbf{T} \to \mathbf{A}$.

$$\begin{array}{ccc} \mathbf{A} \underset{\mathrm{F},p}{\times} \mathbf{T} & \longrightarrow & \mathbf{T} \\ {\scriptstyle \mathrm{F}^*(p)}\downarrow & \lrcorner & \downarrow{\scriptstyle p} \\ \mathbf{A} & \xrightarrow{\mathrm{F}} & \mathbf{C} \end{array}$$

Figure 5: Change of base of fibration.

This is the just the ordinary pullback of categories. For a proof that the pullback of a fibration is a fibration see for example [Jac99, Lemma 1.5.1].

2.3 Type theory

The following is not a systematic or full presentation of the syntax of dependent type theory, nor is it an exposition of how type theory can be used informally to do mathematics. Instead, this section aims to introduce the *formal* syntax of type theory, but only to such a level that the reader becomes familiar with the terminology used when we later introduce comprehension categories. In particular we hope that the reader, after finishing this section, has some informal intuition of types, terms, contexts, weakening, context extension and substitution.

Example 2.3.0.1. We begin with an informal (and hopefully motivating) example. Let $\mathtt{Mat}(3,4)$ be the type of 3×4-matrices. We can use this type to prove interesting theorems about 3×4-matrices, define an addition operation on it, and relate it to other types, such as the types of 4×3-, 3×3- and 4×4-matrices. The obvious issue is that we would have to repeat all of this work when we want to prove similar theorems, or define similar operations, on matrices of other dimensions.

Naturally we would much rather work with matrices of dimension $n \times m$, where n and m are *variables* of the natural numbers type. And we can, if we take the *dependent* type $\mathtt{Mat}(n, m)$, the family of types indexed by $\mathbb{N} \times \mathbb{N}$. When we prove theorems for $\mathtt{Mat}(n, m)$ we do so in the context of two variables of the natural numbers type. When we then substitute n and m for natural numbers in these theorems, say 3 and 4, we obtain theorems specific to 3×4-matrices.

In the section on dependent products we will expand on this example.

For a good introduction on how to use dependent type theory to do mathematics we refer the reader to [Uni13]. A shorter, more digestible, exposition of Martin-Löfs dependent type theory can be found in [Mar84], which includes explanations on how Π- and Σ-types can be understood as quantifiers. Another good introduction to MLTT can be found in [NPS90], in which they follow the style used in [Mar84] but provide many more examples and give an arguably more pedagogical exposition. For a more detailed account of the formal syntax of type theory, including the treatment of contexts, the appendix of [Uni13] gives a more thorough treatment of the subject than we give here.

The reader is assumed to be familiar with the meaning of the turnstile $\vdash$, i.e. that an expression such as $\mathrm{P} \vdash \mathrm{Q}$ is understood as "Q is a consequence of P". Similarly an inference rule

$$\frac{\mathrm{P}_1 \qquad \cdots \qquad \mathrm{P}_n}{\mathrm{Q}}$$

should be understood as "if all of $\mathrm{P}_1 \cdots \mathrm{P}_n$ then also Q".

For the purpose of this thesis we consider a type theory to be a formal system in which the primitive objects are *types* and *terms* of types. We write

$$\mathrm{A}\ \textsc{type} \qquad\qquad x : \mathrm{A}$$

to assert that "A is a type" and "x is a term of type A" respectively.

A *context* is a (possibly empty) finite list of assumptions in the form of typing declarations, $[x_1{:}\mathrm{A}_1, \ldots, x_n{:}\mathrm{A}_n]$. Each variable x_k should be distinct, and may depend on any variable declared to the left of it. Similarly every type A_k occurring in a context may depend on variables declared to the left of it, but the types themselves need not be distinct. We write

$$\Gamma\ \textsc{ctx}$$

to assert that Γ is a well-formed context, and we typically use capital Greek letters to refer to arbitrary (well-formed) contexts.

If $\mathcal{J}$ is a valid expression that holds assuming a variable $x : \mathrm{A}$ we write

$$x{:}\mathrm{A} \vdash \mathcal{J}.$$

However, often we will include an arbitrary context to the left of the turnstile, e.g.

$$\Gamma, x{:}\mathrm{A} \vdash \mathcal{J},$$

to assert that $\mathcal{J}$ follows from any context with at least a variable $x : \mathrm{A}$. The intended interpretation is that the expression $\mathcal{J}$ (possibly) depends on variables declared *explicitly* anywhere to the left of the turnstile (in this case only x:A), and not on variables present in opaque contexts occurring before the turnstile symbol.

If A is a type in context Γ, we can form a new context by *extending* Γ with a fresh variable of type A, and if $\mathcal{J}$ is any expression derivable from Γ then it is also derivable from this extended context. That is, we *weaken* the assertion that $\mathcal{J}$ follows from the assumptions made in context Γ by adding an assumption of type A.

$$\frac{\Gamma \vdash \mathrm{A}\ \textsc{type}}{\Gamma, x{:}\mathrm{A}\ \textsc{ctx}}\ \textsc{ctx-ext} \qquad\qquad \frac{\Gamma \vdash \mathrm{A}\ \textsc{type} \qquad \Gamma \vdash \mathcal{J}}{\Gamma, x{:}\mathrm{A} \vdash \mathcal{J}}\ \textsc{weak}$$

Figure 6: Context extension and weakening.

In *dependent* type theory types may depend on other types. Following the syntax established above this will be written as

$$\Gamma, x{:}\mathrm{A} \vdash \mathrm{B}\ \textsc{type}$$

which declare B to be a type that may depend on the variable x of type A, or equivalently that B is a *family* of types indexed by A. In other literature it is not uncommon to annotate dependent types with the variables that they depend on. For example one could write

$$\Gamma, x{:}\mathrm{A} \vdash \mathrm{B}(x)\ \textsc{type}$$

to emphasise that B is indexed by A. In this thesis we mostly use the variable free notation, but may sometimes choose to annotate types with their dependencies when doing so increases readability or when giving specific examples of dependent types.

Given a type B dependent on a variable x of type A, and a term $t : \mathrm{A}$, we can substitute all occurrences of x in B with t to get the type associated with t, which we denote as the type $\mathrm{B}[t/x]$. For *terms* of dependent types we need to perform substitution in both the dependent type and in the term. For example if we have a term $b : \mathrm{B}$ depending on the variable $x : \mathrm{A}$, and a term $t : \mathrm{A}$, then we need to replace x with t in both b and B.

$$\frac{\Gamma, x{:}\mathrm{A} \vdash \mathrm{B}\ \textsc{type} \qquad \Gamma \vdash t : \mathrm{A}}{\Gamma \vdash \mathrm{B}[t/x]\ \textsc{type}}\ \textsc{subst-ty} \qquad\qquad \frac{\Gamma, x{:}\mathrm{A} \vdash b : \mathrm{B} \qquad \Gamma \vdash t : \mathrm{A}}{\Gamma \vdash b[t/x] : \mathrm{B}[t/x]}\ \textsc{subst-tm}$$

Figure 7: Substitution in types and terms.

Finally we mention *judgemental equality*, which is an essential component in type theory. In any proper introduction to type theory judgemental equality would be introduced at the same time as types and terms in order to justify the computational properties of the type theory, and in particular it is important when defining substitution.

In this thesis judgemental equality, denoted by the symbol $\equiv$, is used (explicitly) only in the computation- and uniqueness rules of types. In the categorical interpretation, which we introduce in the section on comprehension categories, this will correspond to equality in the meta-theory of category theory.

For types A and B we write $A \equiv B : \textsc{type}$ to assert that A and B are equal types, and for terms $a, a' : A$ we write $a \equiv a' : A$ for the judgement that a and a' are equal as terms of type A.

As an example of where judgemental equality is used, suppose we want to define a term of type $\mathbb{B} \to \mathbb{N}$ of functions from the booleans to the natural numbers. Since $\mathbb{B}$ has two canonical elements, `false` and `true`, the *elimination rule* for $\mathbb{B}$ states that we need to provide two terms of $\mathbb{N}$, say $0 : \mathbb{N}$ and $1 : \mathbb{N}$. In return we obtain a function $f : \mathbb{B} \to \mathbb{N}$. The *computation rule* for $\mathbb{B}$ then tells us that if we apply this f to the canonical elements of $\mathbb{B}$ we get the specified elements of $\mathbb{N}$ back:

$$f(\texttt{false}) \equiv 0$$
$$f(\texttt{true}) \equiv 1.$$

It might seem obvious that if we define a function from one type to another by specifying a mapping of the canonical elements of the source type, that when we apply that function to canonical elements that we the specified elements of the target type back, and that is precisely the point. The computation rules uses judgemental equality to assert the expected behaviour of mappings out of types.

2.4 Comprehension Categories

Seely gave in [See84] the first interpretation of dependent type theory, in locally cartesian closed categories. In this interpretation, contexts of the type theory are interpreted as objects of a LCCC $\mathbf{C}$. Types in context, i.e. assertions of the form $\Gamma \vdash A \textsc{type}$, are interpreted as morphisms with codomain Γ, terms $\Gamma \vdash x : A$ are interpreted as sections of types, and substitution is interpreted as pullback.

This construction contained a coherence issue in that pullbacks are only unique up to unique isomorphism, giving a non-strict interpretation of substitution, whereas substitution in type theory is strict up to judgemental equality.

This issue was addressed by Hofmann in [Hof95], by applying Bénabou's right-adjoint splitting construction to the codomain fibration associated with a locally cartesian closed category, to obtain an equivalent split fibration. From this he construct a Category with Attributes [Car86] and showed that the CwA was equipped with strictly stable Π-, Σ- and extensional identity types, hence was a model of dependent type theory. Strict stability means, e.g. for Π-types, that if $(\Pi_A B, \mathrm{app}_{A,B}, \lambda_{A,B})$ is the chosen dependent product for (Γ, A, B), and $\sigma : \Gamma' \to \Gamma$, then $(\Pi_A B[\sigma], \mathrm{app}_{A,B}[\sigma], \lambda_{A,B}[\sigma])$ is the chosen dependent product for $(\Gamma', A[\sigma], B[\sigma.A])$, where $[-]$ denotes the substitution operation.

Comprehension Categories were introduced by [Jac93] as a categorical structure to describe type dependency. Essentially, a comprehension category is

a fibration with extra structure, allowing one not only to interpret contexts and types over contexts, but also *context extension*. That is, a comprehension category provides a way to pass from a judgement $\Gamma \vdash \mathrm{A}\ \textsc{type}$ to the extended context $\Gamma, x{:}\mathrm{A}$. They can be understood as the minimal structure needed on a category to describe type dependencies. For this reason, many categorical structures used to interpret dependent type theory are instances of comprehension categories. Generic examples include Categories with Attributes, Toposes and LCCCs.

The logical structure on a comprehension category is typically not strictly stable under reindexing. Therefore, to obtain a model of dependent type theory, one needs some coherence theorem.

In [LW15] they provide one such coherence theorem, by generalising Hofmanns result for LCCCs to comprehension categories. They give the stability conditions required of logical structure on a comprehension category in order to obtain strictly stable structure on the resulting CwA.

Definition 2.4.0.1 (Comprehension category, [Jac93, Def. 4.1])**.** A comprehension category is a functor $\chi \colon \mathbf{T} \to \mathbf{C}^{\to}$ for which

- the composite $p = \mathbf{cod} \circ \chi \colon \mathbf{T} \to \mathbf{C}$ is a fibration;
- χ preserves cartesian morphisms.

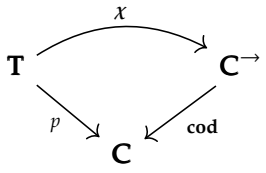


Figure 8

The second condition above means that χ sends cartesian morphisms to pullback squares in $\mathbf{C}$. Note that, even though χ preserves cartesian morphisms and the diagram in figure 8 commutes strictly, χ is not a fibred functor (2.2.0.7) in the technical sense, since **cod** is a fibration only when $\mathbf{C}$ has all pullbacks.

Definition 2.4.0.2 (Full and split comprehension categories)**.** A comprehension category $\chi \colon \mathbf{T} \to \mathbf{C}^{\to}$ is called

- *full* if χ is a full and faithful functor, and
- *split* if $\mathbf{cod} \circ \chi$ is a split fibration.

Notation 2.4.0.3. For a comprehension category $\chi : \mathbf{T} \to \mathbf{C}^{\to}$ we write $p = \mathbf{cod} \circ \chi$ for the fibration. We use the notation $\{-\}$ for the functor $\mathbf{dom} \circ \chi : \mathbf{T} \to \mathbf{C}$. For an object $\mathrm{A} \in \mathbf{T}$ above Γ we adopt the convention of writing $\Gamma.\mathrm{A}$ for the object $\{\mathrm{A}\}$, and we call the morphism $\chi(\mathrm{A}) : \Gamma.\mathrm{A} \to \Gamma$ a *dependent projection* or *display map*. We call reindexing functors induced by dependent projections $\chi(\mathrm{A}) : \Gamma.\mathrm{A} \to \Gamma$ *weakening functors* $\chi(\mathrm{A})^* : \mathbf{T}_\Gamma \to \mathbf{T}_{\Gamma.\mathrm{A}}$.

As an illustration of the notation consider a morphism $f : \Delta \to \Gamma$ in $\mathbf{C}$ and A an object over Γ, we use the notation $f_A : A[f] \to A$ for the (chosen) cartesian lifting of f into A in $\mathbf{T}$. Applying χ to f_A gives us a pullback square in $\mathbf{C}$:

$$\begin{array}{ccc} \Delta.A[f] & \xrightarrow{\{f_A\}} & \Gamma.A \\ {\scriptstyle \chi(A[f])}\downarrow & \lrcorner & \downarrow{\scriptstyle \chi(A)} \\ \Delta & \xrightarrow{f} & \Gamma \end{array}$$

Example 2.4.0.4 (Categories with Attributes, [Jac93])**.** A Category with Attributes is a *discrete* fibration $p\colon \mathbf{T} \to \mathbf{C}$ and a functor $\{-\}\colon \mathbf{T} \to \mathbf{C}$ with a natural transformation $\pi\colon \{-\} \Rightarrow p$ such that for every morphism $f\colon \Delta \to \Gamma$ in $\mathbf{C}$ and every object A in $\mathbf{T}$ above Γ the naturality square (9) is a pullback

$$\begin{array}{ccc} \{A[f]\} & \xrightarrow{\{f_A\}} & \{A\} \\ {\scriptstyle \pi_{A[f]}}\downarrow & \lrcorner & \downarrow{\scriptstyle \pi_A} \\ \Delta & \xrightarrow[f]{} & \Gamma \end{array}$$

Figure 9

where f_A is the (unique) cartesian morphism above f into A.

A Category with Attributes can be seen as a comprehension category where the fibration is discrete.

The original definition of Categories with Attributes is given by [Car86]. However, the definition by Cartmell is not quite equivalent to how CwAs have been defined in the literature since then. The commonly used definition is equivalent to the one we gave above, but use presheaves instead of fibrations. We chose the definition given by Jacobs because it presents a CwA as a comprehension category with discrete fibres.

Blanco showed in [Bla91] that CwAs are equivalent to full, split comprehension categories.

Example 2.4.0.5 (Term model)**.** Given a dependent type theory $\mathbb{T}$ we form a category of contexts $\mathbf{C}_\mathbb{T}$ with objects equivalence classes $[\![\Gamma]\!]$ of contexts Γ. Morphisms $[\![\Delta]\!] \to [\![\Gamma]\!]$ are substitutions. Form the category $\mathbf{T}_\mathbb{T}$ of types over $\mathbf{C}_\mathbb{T}$ with objects equivalence classes of types in context $[\![\Gamma \vdash A \ \text{TYPE}]\!]$ and arrows $[\![\Delta \vdash D \ \text{TYPE}]\!] \to [\![\Gamma \vdash A \ \text{TYPE}]\!]$ are pairs $([\![\sigma]\!], [\![d]\!])$ where $[\![\sigma]\!]\colon [\![\Delta]\!] \to [\![\Gamma]\!]$ and $\Delta, x{:}D \vdash d : A[\sigma]$. The canonical functor $p\colon \mathbf{T}_\mathbb{T} \to \mathbf{C}_\mathbb{T}$ sending a type over a context Γ to the interpretation of Γ is a fibration. Define $\chi\colon \mathbf{T}_\mathbb{T} \to \mathbf{C}_\mathbb{T}^{\to}$ as the functor that sends a type in context Γ to the canonical projection $[\![\Gamma, x{:}A]\!] \to [\![\Gamma]\!]$.

Morphisms in $\mathbf{T}_\mathbb{T}$ is defined using substitution in the type theory $\mathbb{T}$, which is strictly functorial, which makes p a *split* fibration. It follows that the comprehension category $\chi\colon \mathbf{T}_\mathbb{T} \to \mathbf{C}_\mathbb{T}^{\to}$ is a full split comprehension category.

Objects in the base category will be referred to as *contexts*, and given a context $\Gamma \in \mathbf{C}$ objects of the fibre $\mathbf{T}_\Gamma$ will be called *types over* Γ. For a type A over Γ the

object $\{A\} = \Gamma.A$ can be seen as the context Γ extended with a variable of type A, and we refer to sections of the dependent projection $\chi(A)$ as *terms* of type A.

Definition 2.4.0.6 (Terms)**.** For A above Γ write

$$\mathbf{Tm}\,(A) = \{s : \Gamma \to \Gamma.A \mid \chi(A) \circ s = \mathrm{Id}_\Gamma\}$$

for the collection of terms of type A. For every $f : \Delta \to \Gamma$ and A above Γ there is a canonical morphism

$$\mathbf{Tm}\,(A) \to \mathbf{Tm}\,\big(A[f]\big)$$

defined on $s \in \mathbf{Tm}\,(A)$ by the universal property of the pullback $\chi(f_A)$ for the span $(\mathrm{Id}_\Gamma, s \circ f_A)$ in figure 10. We use the notation $s[f_A]$ to refer to the term s reindexed by the cartesian morphism f_A, similarly to how substitution is denoted in syntax. When it is clear from the context and unambiguous we may drop the subscript and write simply $s[f]$.

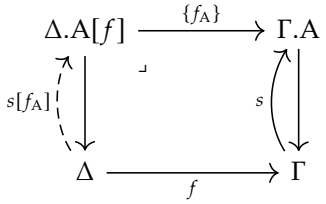
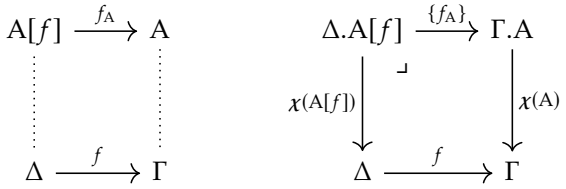


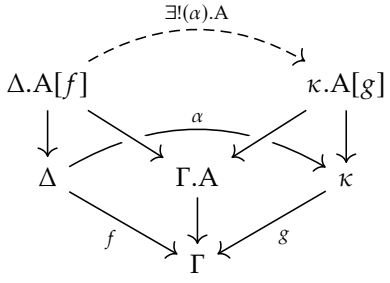
Figure 10: Reindexing of terms.

Definition 2.4.0.7. [Jac93, Definition 4.20] A comprehension category $\chi : \mathbf{T} \to \mathbf{C}^{\to}$ is called *nonempty* if in every context there is at least one inhabited type, i.e. for every Γ in $\mathbf{C}$ there is at least some F in $\mathbf{T}$ above Γ, such that $\mathbf{Tm}\,(F) \neq \emptyset$.

Definition 2.4.0.8 (Pullback functor)**.** Any type A over a context Γ determines a pullback functor between slice categories $(-).A : \mathbf{C}/_\Gamma \to \mathbf{C}/_{\Gamma.A}$, defined on objects $f : \Delta \to \Gamma$ as $(f).A \coloneqq \{f_A\}$; that is, the comprehension of the chosen cartesian lifting of f into A.



The action on morphisms $\alpha : f \to g$ is induced by the universal property of the pullback square given by $\chi(g_A)$ in the right hand side of the following diagram.



It follows from the universal property of the pullback that this operation respects identity morphisms and composition, hence $(-).\mathrm{A}$ is a functor.

Lemma 2.4.0.9 ([Jac93]). *For* $\Gamma \in \mathbf{C}$, $\mathrm{A} \in \mathbf{T}_\Gamma$ *and* $f : \Delta \to \Gamma$ *there is an isomorphism*

$$\mathbf{C}/_\Gamma(f, \chi(\mathrm{A})) \cong \mathbf{C}/_\Delta(\mathrm{Id}_\Delta, \chi(\mathrm{A}[f])).$$

Definition 2.4.0.10. A comprehension category determines a category

$$\mathrm{Cart}(\mathbf{T}) \hookrightarrow \mathbf{T}$$

with only cartesian morphisms. By restriction one obtains two functors

$$|p| : \mathrm{Cart}(\mathbf{T}) \to \mathbf{C}$$
$$|\{-\}| : \mathrm{Cart}(\mathbf{T}) \to \mathbf{C}.$$

By taking the pullback of the fibration $|p| : \mathrm{Cart}(\mathbf{T}) \to \mathbf{C}$ along these functors one obtains two fibrations

$$|p|_p : \mathrm{Cart}(\mathbf{T}) \underset{p}{\times} \mathrm{Cart}(\mathbf{T}) \to \mathrm{Cart}(\mathbf{T})$$
$$|\{-\}|_p : \mathrm{Cart}(\mathbf{T}) \underset{\{-\}}{\times} \mathrm{Cart}(\mathbf{T}) \to \mathrm{Cart}(\mathbf{T}).$$

The above fibrations let us define the structure of *weakening* next. The category $\mathrm{Cart}(\mathbf{T}) \underset{\{-\}}{\times} \mathrm{Cart}(\mathbf{T})$ has as objects pairs (A, B) where A is above some object $\Gamma \in \mathbf{C}$ and B is above $\{\mathrm{A}\} = \Gamma.\mathrm{A}$, and similarly for morphisms. An object (A, B) represents a type family B indexed by A. Objects in $\mathrm{Cart}(\mathbf{T}) \underset{p}{\times} \mathrm{Cart}(\mathbf{T})$ are pairs $(\mathrm{A}, \mathrm{A}')$ both over some object Γ in the base category.

Recall that weakening in type theory is the rule that states that if we have some valid expression $\mathcal{J}$ that follows from a context Γ, and A is a type over Γ, then $\mathcal{J}$ also follows from Γ extended by a fresh variable of type A. If we take $\mathcal{J}$ to be the judgement A' TYPE then the weakening rule take the following form.

$$\frac{\Gamma \vdash \mathrm{A} \text{ TYPE} \qquad \Gamma \vdash \mathrm{A}' \text{ TYPE}}{\Gamma, x{:}\mathrm{A} \vdash \mathrm{A}' \text{ TYPE}}$$

It is precisely this that the next definition captures.

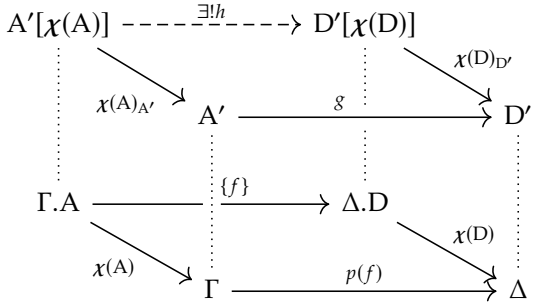
Definition 2.4.0.11 (Weakening, [Jac93]). The functor $\chi : \mathbf{T} \to \mathbf{C}^{\to}$ can be lifted to a fibred functor $\langle \chi \rangle$

$$\begin{array}{ccc} \mathrm{Cart}(\mathbf{T}) \underset{p}{\times} \mathrm{Cart}(\mathbf{T}) & \xrightarrow{\langle \chi \rangle} & \mathrm{Cart}(\mathbf{T}) \underset{\{-\}}{\times} \mathrm{Cart}(\mathbf{T}) \\ & \searrow \quad \swarrow & \\ & \mathrm{Cart}(\mathbf{T}) & \end{array}$$

defined

- on objects $(\mathrm{A}, \mathrm{A}')$ by $\langle \chi \rangle(\mathrm{A}, \mathrm{A}') \coloneqq (\mathrm{A}, \mathrm{A}'[\chi(\mathrm{A})])$

- on morphisms $(f, g)\colon (\mathrm{A}, \mathrm{A}') \to (\mathrm{D}, \mathrm{D}')$ as (f, h) where h is the unique mediating morphism above $\{f\}$ induced by the cartesian morphism $\chi(\mathrm{D})_{\mathrm{D}'}$ in the following diagram.



Functoriality follows from the uniqueness property of cartesian morphisms. The analogue of the two-pullback-lemma for cartesian morphisms imply that if g in the diagram above is cartesian, then so is h, hence $\langle \chi \rangle$ is a fibred functor.

In a *full* comprehension category dependent products and dependent sums, which we will look at in the section on categorical semantics, are described as fibred right- and left adjoints to $\langle \chi \rangle$. In the section on categorical semantics we will investigate dependent products and sums in non-full comprehension categories, and to do that we need to introduce a few more categories and fibrations.

We obtain the following two functors by taking the pullback of **cod** along $|p|$ and $|\{-\}|$:

$$|p|_{\mathbf{cod}}\colon \mathrm{Cart}(\mathbf{T}) \underset{p}{\times} \mathbf{C}^{\to} \to \mathrm{Cart}(\mathbf{T})$$
$$|\{-\}|_{\mathbf{cod}}\colon \mathrm{Cart}(\mathbf{T}) \underset{\{-\}}{\times} \mathbf{C}^{\to} \to \mathrm{Cart}(\mathbf{T}).$$

We can construct an analogue of $\langle \chi \rangle$ for these functors.

Definition 2.4.0.12. Let $\mathcal{W}\colon \mathrm{Cart}(\mathbf{T}) \underset{p}{\times} \mathbf{C}^{\to} \to \mathrm{Cart}(\mathbf{T}) \underset{\{-\}}{\times} \mathbf{C}^{\to}$ be the functor defined on objects as

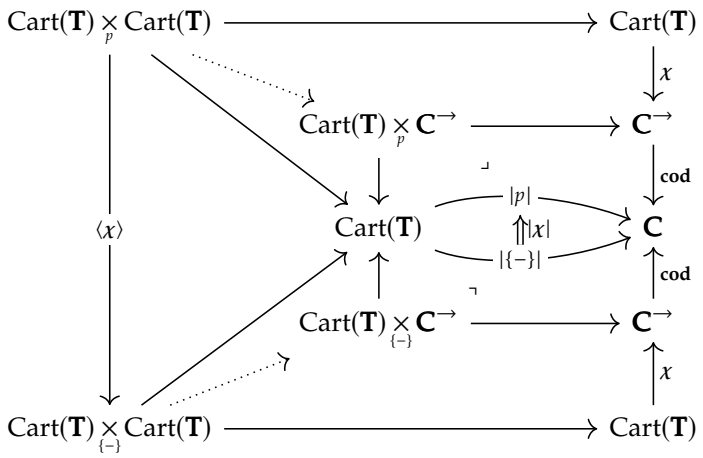
$$\mathcal{W}\left(\mathrm{A}, \Gamma' \xrightarrow{f} \Gamma\right) = \left(\mathrm{A}, \Gamma'.\mathrm{A}[f] \xrightarrow{\{f_{\mathrm{A}}\}} \Gamma.\mathrm{A}\right)$$

and on arrows $\mathcal{W}(\sigma_{\mathrm{D}}, (\sigma, g)\colon f' \to f) = (\sigma_{\mathrm{D}}, (\{\sigma_{\mathrm{D}}\}, h))$ as illustrated below

$$\begin{array}{ccc} \Delta' & \xrightarrow{g} & \Gamma' \\ {\scriptstyle f'}\downarrow & & \downarrow{\scriptstyle f} \\ \Delta & \xrightarrow[\sigma]{} & \Gamma \end{array} \quad \overset{\mathcal{W}}{\longmapsto} \quad \begin{array}{ccc} \Delta'.\mathrm{D}[f'] & \xrightarrow{h} & \Gamma'.\mathrm{A}[f] \\ {\scriptstyle \{f'_{\mathrm{D}}\}}\downarrow & & \downarrow{\scriptstyle \{f_{\mathrm{A}}\}} \\ \Delta.\mathrm{D} & \xrightarrow[\{\sigma_{\mathrm{D}}\}]{} & \Gamma.\mathrm{A} \end{array}$$

where h is induced by the pullbacks involved. By the two-pullback lemma the right hand square is a pullback if the square to the left is. Note that this functor makes use of the specific cleavage of the fibration.

In summary we can describe the categories and fibrations presented above obtained by change-of-base, as pullbacks illustrated in the diagram below.



3 Types in non-full comprehension categories

3.1 Unit types

A unit type in our dependent type theory is a type with a single unique term. Formally this is given by the following inference rules.

$$\frac{\Gamma \ \text{CTX}}{\Gamma \vdash \mathbf{1} \ \text{TYPE}} \ \textbf{1-FORM} \qquad \frac{\Gamma \ \text{CTX}}{\Gamma \vdash \star : \mathbf{1}} \ \textbf{1-INTRO} \qquad \frac{\Gamma \vdash z : \mathbf{1}}{\Gamma \vdash z \equiv \star : \mathbf{1}} \ \textbf{1-UNIQ}$$

Remark 3.1.0.1. We could have presented the unit type with an elimination rule and a computation rule instead of the uniqueness rule, and from those derived the rule **1-uniq** using the extensional identity type. Assuming **1-uniq** we can instead derive an elimination rule, and the computation rule becomes trivial.

Had we instead assumed a type theory with *intensional* identity types à la Martin-Löf and postulated a unit type together with an elimination and computation rules, then the judgemental uniqueness rule would not be derivable within the type theory, but it could optionally be added. One would still be able to derive a *propositional* uniqueness rule in this setting.

Compare the above syntactic definition of the unit type with the following categorical description.

Definition 3.1.0.2. A unit type for an object $\Gamma \in \mathbf{C}$ in a comprehension category $\chi \colon \mathbf{T} \to \mathbf{C}^{\to}$ consists of:

- an object 1_Γ in the fibre above Γ;
- a *unique* section $s_\Gamma \colon \Gamma \to \Gamma.1$ of the dependent projection $\chi(1_\Gamma)$.

The definition above defines a unit type for a specific context $\Gamma \in \mathbf{C}$. To say what it means for a comprehension category to have unit types for every $\Gamma \in \mathbf{C}$ we need to also give some stability condition, defining how they behave with

respect to reindexing. We define pseudo-stable unit types in a comprehension category next, since this condition is strong enough to be able to obtain strictly stable unit types via the right adjoint splitting functor.

Definition 3.1.0.3. A comprehension category has *pseudo-stable* unit types if there is a fibred functor $\mathbf{1}\colon \mathbf{C} \to \mathbf{T}$ as in figure 11, and for each Γ in $\mathbf{C}$ there is a unique section $s(\Gamma)\colon \Gamma \to \Gamma.1$ of the dependent projection $\chi(\mathbf{1}(\Gamma))$.

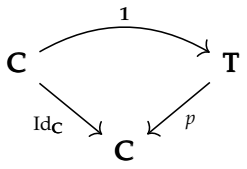


Figure 11

Next we show that pseudo-stable unit types can be described as a fibred adjunction.

Proposition 3.1.0.4. *A comprehension category $\chi\colon \mathbf{T} \to \mathbf{C}^{\to}$ has pseudo-stable units if there is a fibred functor $\mathbf{1}\colon \mathbf{C} \to \mathbf{T}$ (as in figure 11) and an adjunction*

$$\mathbf{cod} \dashv \chi \circ \mathbf{1}$$

that is fibred in the sense that the bijection restricts to hom-sets over morphisms in the base. That is, for each morphism $u\colon \Delta \to \Gamma$ in $\mathbf{C}$ and morphism f with $\mathbf{cod}(f) = \Delta$ an isomorphism

$$\mathbf{C}_u(\mathbf{cod}(f), \Gamma) \cong \mathbf{C}^{\to}_u(f, \chi(\mathbf{1}(\Gamma))). \tag{3}$$

Remark 3.1.0.5. The adjunction is not a fibred adjunction in the proper sense, i.e. an adjunction in $\mathbf{Fib}(\mathbf{C})$, since $\mathbf{cod}$ is not a fibration unless $\mathbf{C}$ has all pullbacks. However, all functors involved preserve cartesian morphisms, and the relevant diagrams commute strictly.

Remark 3.1.0.6. It is worth spelling out what the isomorphism (3) implies. Fix a morphism $u : \Delta \to \Gamma$ in $\mathbf{C}$. Since the left homset is fibred over $\mathbf{C}$ by the identity fibration there is only one morphism above u, namely u itself. The isomorphism then implies that for each $f : \Lambda \to \Delta$ there is a unique morphism $\Lambda \to \Gamma.1$ making the square in figure 12 commute.

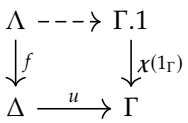


Figure 12

Proof of 3.1.0.4. Assuming $1 : \mathbf{C} \to \mathbf{T}$ fibred and a fibred adjunction $\mathbf{cod} \dashv \chi \circ \mathbf{1}$, we need to show that for each Γ in $\mathbf{C}$ there is a unique section of the unit

projection $\chi(\mathbf{1}(\Gamma))$. Take $u = f = \mathrm{Id}_\Gamma$ in (3) and apply the isomorphism from left to right to obtain a unique map

$$\begin{array}{ccc} \Gamma & \dashrightarrow & \Gamma.1 \\ \| & & \downarrow{\scriptstyle \chi(1_\Gamma)} \\ \Gamma & = & \Gamma \end{array}$$

which is a section of the unit projection. □

Corollary 3.1.0.7. *For $f : \Delta \to \Gamma$ a morphism between contexts in a comprehension category with pseudo-stable unit types, $s(\Gamma) \circ f = \{1(f)\} \circ s(\Delta)$, where $s(-)$ is the unique section of the unit type in the given context.*

Proof.

$$\begin{array}{ccc} \Delta.1 & \xrightarrow{\{1(f)\}} & \Gamma.1 \\ {\scriptstyle \chi(1_\Delta)}\downarrow & \lrcorner & \downarrow{\scriptstyle \chi(1_\Gamma)} \\ \Delta & \xrightarrow{f} & \Gamma \end{array} \tag{4}$$

The pullback above determines a unique morphism $\phi : \Delta \to \Delta.1$ for the span $(\mathrm{Id}_\Delta, s(\Gamma) \circ f)$ such that $\{1(f)\} \circ \phi = s(\Gamma) \circ f$ and $\chi(1_\Delta) \circ \phi = \mathrm{Id}_\Delta$, which makes ϕ a term of $\Delta.1$. By uniqueness of terms in the unit type this implies that $\phi = s(\Delta)$ and $s(\Gamma) \circ f = \{1(f)\} \circ s(\Delta)$. □

Proposition 3.1.0.8. *Pseudo-stable unit types in a comprehension category determines an adjunction $\mathbf{cod} \dashv \chi \circ \mathbf{1}$ fibred over $\mathbf{C}$, i.e. a bijection*

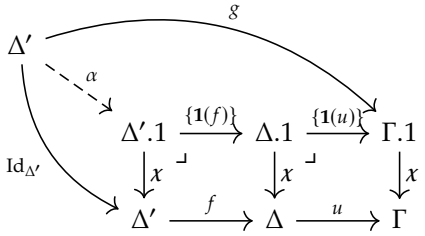
$$\Phi : \mathbf{C}_u(\mathbf{cod}(f), \Gamma) \cong \mathbf{C}^{\to}_u(f, \chi(\mathbf{1}(\Gamma)) \tag{5}$$

natural in f and Γ.

Proof. We construct Φ explicitly and prove that it is a natural. Notice that $\mathbf{C}_u(\mathbf{cod}(f), \Gamma)$ is a singleton set, for $u : \Delta \to \Gamma$. We need $\Phi(u)$ to be an arrow in $\mathbf{C}$ making the diagram below commute.

$$\begin{array}{ccc} \Delta' & \xrightarrow{\Phi(u)} & \Gamma.1 \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle \chi(\mathbf{1}(\Gamma))} \\ \Delta & \xrightarrow{u} & \Gamma \end{array} \tag{6}$$

Take $\Phi(u) := s(\Gamma) \circ u \circ f$. This map is unique by the following argument. Let $g : \Delta' \to \Gamma.1$ be an arrow that makes the above square commute. Then there is a unique $\alpha : \Delta' \to \Delta'.1$ given by the pullbacks: This α is clearly a section of $\Delta'.1$, and by uniqueness this implies $\alpha = s(\Delta')$. By functoriality of $\mathbf{1}$ and $\{-\}$ and corollary 3.1.0.7 it follows that $g = s(\Gamma) \circ u \circ f = \Phi(u)$. From uniqueness it also follows that Φ is an isomorphism, since for given u in $\mathbf{C}$ the collection of morphisms in (5) are singletons.



(7)

For naturality, we need that for $h : \Gamma \to \Lambda$ in $\mathbf{C}$ and $(g, \bar{g}) = \mathrm{G} : m \implies f$ in $\mathbf{C}^{\to}$ the following square commutes

$$\begin{array}{ccc} \mathbf{C}_u(\mathbf{cod}(f), \Gamma) & \xrightarrow{\Phi(u)} & \mathbf{C}^{\to}_u(f, \chi(1(\Gamma))) \\ \downarrow{\scriptstyle \mathrm{Hom}(g,h)} & & \downarrow{\scriptstyle \mathrm{Hom}(\mathrm{G},\chi(1(h)))} \\ \mathbf{C}_{h\circ u\circ g}(\mathbf{cod}(m), \Lambda) & \xrightarrow{\Phi(h\circ u\circ g)} & \mathbf{C}^{\to}_{h\circ u\circ g}(m, \chi(1(\Lambda))) \end{array}$$

which it does by necessity: all the sets involved are singletons, by applying the argument in (7). □

3.2 Dependent products

Dependent product types, also called dependent function types and Π-types, is a generalisation of the non-dependent function type $\mathrm{A} \to \mathrm{B}$ by allowing B to be a *family* of types dependent on A. For a dependent function $f : \Pi_{x:\mathrm{A}}\mathrm{B}(x)$ and a term $a : \mathrm{A}$ we can apply f to a to get a term $f(a) : \mathrm{B}(a)$. The key observation to make is that the codomain of f may vary depending on its input, so if a and a' are different elements of A then $f(a)$ and $f(a')$ may not only be different as elements of some type, they might even be elements of *different* types!

Consider the dependent type of $n \times n$-matricies indexed by natural numbers, $n{:}\mathbb{N} \vdash \mathtt{Mat}(n, n)$. To introduce a term of the dependent product type

$$\prod_{n:\mathbb{N}} \mathtt{Mat}(n, n)$$

we need to define a term of $\mathtt{Mat}(n, n)$ while assuming a variable $n : \mathbb{N}$. For example, assuming $n : \mathbb{N}$ we can construct $\mathrm{I}(n) : \mathtt{Mat}(n, n)$ to be the $n \times n$-matrix with entries 0 everywhere except on the diagonal where it is 1. This allows us to introduce the dependent function

$$\mathrm{I} : \prod_{n:\mathbb{N}} \mathtt{Mat}(n, n)$$

which for each natural number returns an $n \times n$-matrix. To apply I to a natural number, say $3 : \mathbb{N}$, we do so by substituting 3 for n in both our construction $\mathrm{I}(n)$ and the dependent type $\mathtt{Mat}(n, n)$ to get a term

$$\mathrm{I}(3) : \mathtt{Mat}(3, 3).$$

Note that we apply I to 3 by substituting 3 for n, not in I itself, but in $\mathrm{I}(n)$ which we used to construct I. This is an example of using the *computation* rule for dependent products, which we may use when we have constructed a dependent function explicitly. In this case we defined I by constructing an identity matrix for each natural number n, so it makes sense that when applying I to a natural number we get the corresponding identity matrix back.

Under the Curry-Howard and propositions-as-types interpretation, dependent products play the role of universal quantification. The Π-symbol would then be understood as a "for all" statement. In the example with square matricies

above we would read the type $\Pi_{n:\mathbb{N}}\mathtt{Mat}(n,n)$ as the proposition "for all natural numbers n there is a $n \times n$-matrix", and the term $\mathrm{I} : \Pi_{n:\mathbb{N}}\mathtt{Mat}(n,n)$ is a proof of that proposition, which produces an $n \times n$-matrix for each natural number n.

We begin by defining the formal syntax of Π-types, then we define and study the corresponding logical structure on comprehension categories.

3.2.1 Syntax of dependent products

Formally the syntax is given by the following inference rules.

$$\frac{\Gamma \vdash \mathrm{A}\ \textsc{type} \qquad \Gamma, x{:}\mathrm{A} \vdash \mathrm{B}\ \textsc{type}}{\Gamma \vdash \Pi_{x:\mathrm{A}}\mathrm{B}\ \textsc{type}}\ \prod\textsc{-form} \qquad \frac{\Gamma, x{:}\mathrm{A} \vdash b : \mathrm{B}}{\Gamma \vdash \lambda(x{:}\mathrm{A}).b : \Pi_{x:\mathrm{A}}\mathrm{B}}\ \prod\textsc{-intro}$$

$$\frac{\Gamma \vdash f : \Pi_{x:\mathrm{A}}\mathrm{B} \qquad \Gamma \vdash a : \mathrm{A}}{\Gamma \vdash \mathrm{app}(f,a) : \mathrm{B}[a/x]}\ \prod\textsc{-elim}$$

$$\frac{\Gamma, x{:}\mathrm{A} \vdash b : \mathrm{B} \qquad \Gamma \vdash a : \mathrm{A}}{\Gamma \vdash \mathrm{app}(\lambda(x{:}\mathrm{A}).b, a) \equiv b[a/x] : \mathrm{B}[a/x]}\ \prod\textsc{-comp}$$

We will also assume the following uniqueness rule, also known as the η-rule. Π-types satisfying the uniqueness rule will be called *strong* dependent products.

$$\frac{\Gamma \vdash f : \Pi_{x:\mathrm{A}}\mathrm{B}}{\Gamma \vdash f \equiv \lambda(x{:}\mathrm{A}).\mathrm{app}(f,x) : \Pi_{x:\mathrm{A}}\mathrm{B}}\ \prod\textsc{-uniq}$$

We prefer to not include the name of variables that type families depend on in the notation. Instead of writing $\Gamma, x{:}\mathrm{A} \vdash \mathrm{B}(x)\ \textsc{type}$ for a type family that may depend on $x : \mathrm{A}$ we write simply $\Gamma, x{:}\mathrm{A} \vdash \mathrm{B}\ \textsc{type}$. This allows us to consistently use the bracket-notation for substitution, for example in **∏-elim** and **∏-comp**. We also prefer the notation $\mathrm{app}(f,a)$ over writing $f(a)$ or fa to denote application of dependent functions. This is also to be more compatible with the notation we introduce when describing the logical structure of Π-types in comprehension categories.

Note that if one takes B to be a non-dependent type in the inference rules above, then one get precisely the non-dependent function type $\mathrm{A} \to \mathrm{B}$ as one would expect.

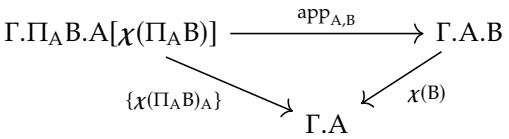
3.2.2 Logical structure of dependent products

The following is the definition of dependent products given in [LW15], but with minor modifications.

Definition 3.2.2.1 ([LW15, Def. 3.4.2.1])**.** Let $\chi\colon \mathbf{T} \to \mathbf{C}^{\to}$ be a comprehension category. For objects $\Gamma \in \mathbf{C}$, $\mathrm{A} \in \mathbf{T}_\Gamma$ and $\mathrm{B} \in \mathbf{T}_{\Gamma.\mathrm{A}}$, a *dependent product* for $(\Gamma, \mathrm{A}, \mathrm{B})$ consists of

- a type $\Pi_{\mathrm{A}}\mathrm{B} \in \mathbf{T}_\Gamma$;

- a morphism $\mathrm{app}_{A,B} \colon \Gamma.\Pi_A B.A[\chi(\Pi_A B)] \to \Gamma.A.B$ in $\mathbf{C}$ such that



  commutes;

- an operation giving for each section $t \colon \Gamma.A \to \Gamma.A.B$, a section $\lambda(t) \colon \Gamma \to \Gamma.\Pi_A B$, such that $\mathrm{app}_{A,B} \circ (\mathrm{Id}_{\Gamma.A}, \lambda(t) \circ \chi(A)) = t$.

The definition given in [LW15] is identical except that $\mathrm{app}_{A,B}$ is taken to be a map $\Pi_A B[\chi(A)] \to B$ in $\mathbf{T}_{\Gamma.A}$. Dependent products according to the definition in [LW15] imply dependent products as given here, and in a *full* comprehension category the two are equivalent. We do not assume that comprehension categories are full, therefore we use the more general definition of dependent products given here.

Definition 3.2.2.1 immediately gives a categorical description of the formation- and introduction rule of dependent products, and maybe less immediate the computation rule, which we explain next.

Definition 3.2.2.2. A dependent product in a comprehension category induces a map of terms

$$app \colon \mathbf{Tm}\,(\Pi_A B) \to \mathbf{Tm}\,(B)$$

by taking a section $f \colon \Gamma \to \Gamma.\Pi_A B$ to

$$app(f) := \mathrm{app}_{A,B} \circ f.A : \Gamma.A \to \Gamma.A.B.$$

This operation is a retraction of the λ-operation: for every section $t \colon \Gamma.A \to \Gamma.A.B$,

$$app(\lambda(t)) = t$$

by the third condition of definition 3.2.2.1.

The uniqueness rule (also called the η-rule) is not included in definition 3.2.2.1, but we may add it as an assumption.

Definition 3.2.2.3 (Strong dependent products)**.** A comprehension category with dependent products has *strong* dependent products if, for each Γ in $\mathbf{C}$, A above Γ and B above $\Gamma.A$, the dependent product satisfies the η-rule; i.e. that the λ-operation induced by the dependent product is an inverse of the app-operation, meaning that for every section $s \colon \Gamma \to \Gamma.\Pi_A B$,

$$\lambda(app(s)) = s.$$

This gives a bijection of terms

$$\mathbf{Tm}\,(\Pi_A B) \cong \mathbf{Tm}\,(B)\,.$$

Lemma 3.2.2.4. *A comprehension category which for each context $\Gamma \in \mathbf{C}$ with objects $A \in \mathbf{T}_\Gamma$, $B \in \mathbf{T}_{\Gamma.A}$ has an object $\Pi_A B \in \mathbf{T}_\Gamma$ and an isomorphism natural in f*

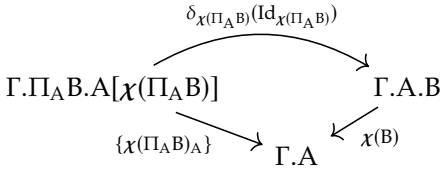
$$\delta_f \colon \mathbf{C}/_\Gamma(f, \chi(\Pi_A B)) \cong \mathbf{C}/_{\Gamma.A}(f.A, \chi(B)) \tag{8}$$

has Π-types satisfying the η-rule.

Proof. Take $f = \chi(\Pi_A B)$ in (8) and evaluate at the identity to get a morphism

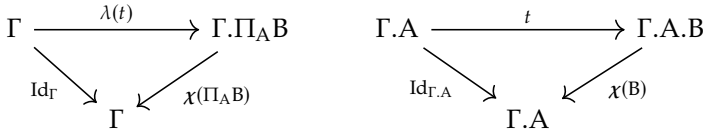
$$\delta_{\chi(\Pi_A B)}(\mathrm{Id}_{\chi(\Pi_A B)}) \colon \Gamma.\Pi_A B.A[\chi(\Pi_A B)] \to \Gamma.A.B$$

such that



commutes. Define $\mathrm{app}_{A,B} := \delta_{\chi(\Pi_A B)}(\mathrm{Id}_{\chi(\Pi_A B)})$. For a section $t \colon \Gamma.A \to \Gamma.A.B$, we define

$$\lambda(t) := \delta^{-1}_{\mathrm{Id}_\Gamma}(t) \colon \Gamma \to \Gamma.\Pi_A B.$$



Recall from 3.2.2.2 that the operation $app \colon \mathbf{Tm}(\Pi_A B) \to \mathbf{Tm}(B)$ is defined as $app(f) = \mathrm{app}_{A,B} \circ f.A$. From this follows that for every section $t \colon \Gamma.A \to \Gamma.A.B$ of the projection $\chi(B)$

$$\begin{aligned} app(\lambda(t)) &= \mathrm{app}_{A,B} \circ \lambda(t).A \\ &= \delta_{\mathrm{Id}_\Gamma}(\lambda(t)) && \text{(by naturality)} \\ &= \delta_{\mathrm{Id}_\Gamma}(\delta^{-1}_{\mathrm{Id}_\Gamma}(t)) \\ &= t. \end{aligned}$$

This construction also satisfies the η-rule: let $s \colon \Gamma \to \Gamma.\Pi_A B$ be any section of the projection $\chi(\Pi_A B)$. By naturality and the definition $\mathrm{app}_{A,B} := \delta_{\chi(\Pi_A B)}(\mathrm{Id}_{\chi(\Pi_A B)})$ it follows that

$$app(s) = \mathrm{app}_{A,B} \circ s.A = \delta_{\mathrm{Id}_\Gamma}(s),$$

and because the λ-operation is defined as $\delta^{-1}_{\mathrm{Id}_\Gamma}$ the identity $\lambda(app(s)) = s$ holds. □

Remark 3.2.2.5. Naturality of (8) gives the identity

$$\delta_f(h) = \mathrm{app}_{A,B} \circ h.A$$

for every $f \colon \Gamma' \to \Gamma$ and $h \colon \Gamma' \to \Gamma.\Pi_A B$ such that $\chi(\Pi_A B) \circ h = f$.

Definition 3.2.2.6 (Pseudo-stable Π-types, [LW15, Def. 3.4.2.8])**.** A comprehension category has *pseudo-stable* dependent products if there is a fibred functor

$$\Pi \colon \mathrm{Cart}(\mathbf{T}) \underset{\{-\}}{\times} \mathrm{Cart}(\mathbf{T}) \to \mathrm{Cart}(\mathbf{T}) \underset{p}{\times} \mathrm{Cart}(\mathbf{T})$$

such that, for all cartesian morphisms $\sigma_D \colon D \to A$ above $\sigma \colon \Delta \to \Gamma$ and $\sigma_E \colon E \to B$ above $\{\sigma_D\} \colon \Delta.D \to \Gamma.A$, and sections $t \colon \Gamma.A \to \Gamma.A.B$, the diagrams below commute.

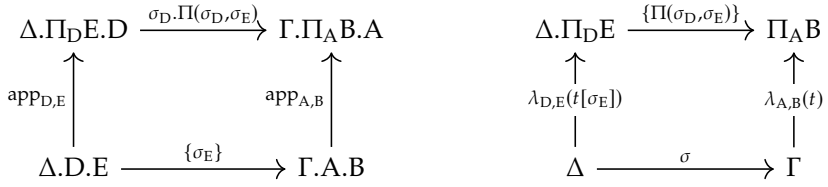


Figure 13

3.2.3 Dependent products in full comprehension categories

In [Jac93], a comprehension category $\chi\colon \mathbf{T} \to \mathbf{C}^{\to}$ *has products* if there is a fibred right adjoint to weakening

$$\langle\chi\rangle \dashv \Pi.$$

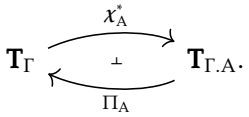
One could hope this to be equivalent to a comprehension category having type-theoretic dependent products, but this is unfortunately not the case.

In this section we recount what logical structure one finds in a comprehension category with products, and give some sufficient conditions for a comprehension category with products, to have type-theoretic dependent products, following Jacobs.

One sufficient condition will be that a (nonempty) comprehension category with products is *full*; that is, there is a bijection between morphisms of types and morphisms of dependent projections.

We begin by describing what logical structure a right adjoint to weakening give us in an arbitrary comprehension category.

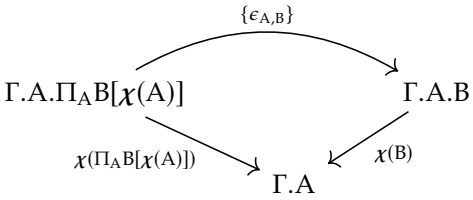
First, observe that a fibred adjunction $\langle\chi\rangle \dashv \Pi$ gives an adjunction between fibres, i.e. for every type A above Γ there is an adjunction



For each type B above $\Gamma.A$ the counit is a vertical morphism $\chi_A^*(\Pi_A B) \to B$, or to be consistent with notation, a morphism

$$\epsilon_{A,B}\colon \Pi_A B[\chi(A)] \to B$$

above $\Gamma.A$. If we apply χ to the counit we get a commutative triangle



which motivates us to take as a definition

$$\mathrm{app}_{A,B} := \{\epsilon_{A,B}\}.$$

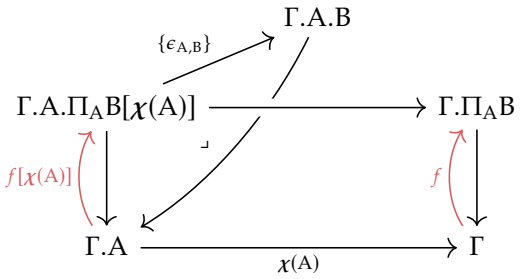
Note that there is a difference between how we define $\mathrm{app}_{A,B}$ here and how it is defined in 3.2.2.1, in that $\{\epsilon_{A,B}\}$ has dependent projections both as domain and codomain, whereas the one in 3.2.2.1 has $\{\chi(\Pi_A B)_A\}$ as domain. They are however canonically isomorphic as objects in slices over $\Gamma.A$. The difference will be relevant when we later assume fullness, in that $\mathrm{app}_{A,B}$, as defined here, will lift uniquely to a morphism between $\Pi_A B[\chi(A)]$ and B.

There is a function $app\colon \mathbf{Tm}(\Pi_A B) \to \mathbf{Tm}(B)$ that takes a section

$$f\colon \Gamma \to \Gamma.\Pi_A B$$

to

$$\{\epsilon_{A,B}\} \circ f[\chi(A)]\colon \Gamma.A \to \Gamma.A.\Pi_A B[\chi(A)]$$



This is how far we get by only assuming a fibred right adjoint to weakening; that is, we get type formation and an app-operation, but not λ-abstraction or computation and uniqueness.

As Jacobs points out, for a comprehension category with products to have type-theoretic dependent products, one expects the app-operation to be an isomorphism, with inverse given by λ-abstraction. This is the case precisely when the comprehension category preserves products on its own projections in the slices of the base category, which is the statement of the following lemma.

Lemma 3.2.3.1 ([Jac93, Lemma 5.2])**.** *For a comprehension category $\chi\colon \mathbf{T} \to \mathbf{C}^{\to}$ with products, the following two statements are equivalent:*

1. $\mathbf{Tm}(\Pi_A B) \cong \mathbf{Tm}(B)$ *for all appropriate* A *and* B*;*
2. χ *preserves products, i.e. for all appropriate* A *above* Γ, B *above* $\Gamma.A$ *and* $u\colon \Delta \to \Gamma$ *one has*

$$\mathbf{C}/_{\Gamma}(u, \chi(\Pi_A B)) \cong \mathbf{C}/_{\Gamma.A}(u.A, \chi(B))$$

where $u.A$ *is the pullback functor in 2.4.0.8 applied to* u*.*

Proof. Note that the direction (2) $\Rightarrow$ (1) is lemma 3.2.2.4. For a full proof see [Jac93]. □

It turns out, as Jacobs shows, that if a comprehension category, with a fibred right adjoint to weakening, is both nonempty and full, then it has strong dependent products. We give part of the proof to illustrate how fullness is used.

Lemma 3.2.3.2 ([Jac93, Lemma 5.4])**.** *A nonempty, full, comprehension category with a fibred adjunction*

$$\langle \chi \rangle \dashv \Pi$$

has strong dependent products.

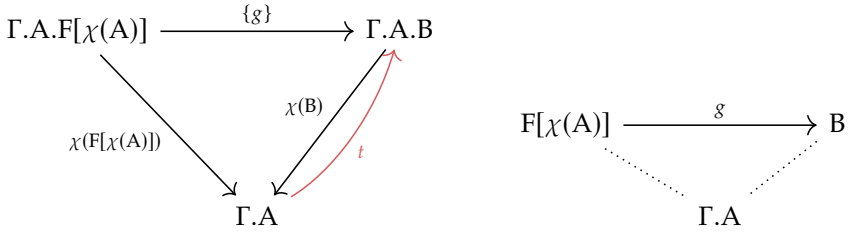
Proof. Let Γ be a context in $\mathbf{C}$ and A a type above Γ, and B a type above $\Gamma.\mathrm{A}$. The counit of the adjunction provides the morphism $\mathrm{app}_{\mathrm{A,B}} := \{\epsilon_{\mathrm{A,B}}\}$, hence also the function

$$app(f) := \mathrm{app}_{\mathrm{A,B}} \circ f[\chi(\mathrm{A})] \colon \mathbf{Tm}\,(\Pi_{\mathrm{A}}\mathrm{B}) \to \mathbf{Tm}\,(\mathrm{B})\,.$$

We need to construct the inverse of this, the λ-operation.

Let $t \colon \Gamma.\mathrm{A} \to \Gamma.\mathrm{A.B}$ be a term of type B. Because the comprehension category is nonempty, there is some type F above Γ and a term $f \colon \Gamma \to \Gamma.\mathrm{F}$. By weakening we obtain a type $\mathrm{F}[\chi(\mathrm{A})]$ above $\Gamma.\mathrm{A}$.

By fullness we can lift the composite $t \circ \chi(\mathrm{F}[\chi(\mathrm{A})])$ to a unique morphism $g \colon \mathrm{F}[\chi(\mathrm{A})] \to \mathrm{B}$ vertical above $\Gamma.\mathrm{A}$ such that $\{g\} = t \circ \chi(\mathrm{F}[\chi(\mathrm{A})])$.



Transposing g across the adjunction yields a morphism

$$\hat{g} = \Pi_{\mathrm{A}}(g) \circ \eta_{\mathrm{A,F}} \colon \mathrm{F} \to \Pi_{\mathrm{A}}\mathrm{B},$$

which, after applying comprehension and composing with the term $f \colon \Gamma \to \Gamma.\mathrm{F}$, gives a term

$$\{\hat{g}\} \circ f \colon \Gamma \to \Gamma.\Pi_{\mathrm{A}}\mathrm{B}.$$

Take this operation as definition of the λ-operation.

We omit the proof that the λ-operation is an inverse of app. The proof is given in full in [Jac93, Lemma 5.4]. □

3.2.4 Dependent products as relative adjoints

In this section we show that dependent products in *arbitrary* comprehension categories correspond to *relative* adjoint functors.

Lemma 3.2.4.1. *Pseudo-stable strong dependent products induce an isomorphism*

$$\delta_{f,\mathrm{B}} \colon \mathbf{C}/_{\Gamma}(f, \chi(\Pi_{\mathrm{A}}\mathrm{B})) \cong \mathbf{C}/_{\Gamma.\mathrm{A}}(f.\mathrm{A}, \chi(\mathrm{B}))$$

for each $\Gamma \in \mathbf{C}$, $f \colon \Delta \to \Gamma$, $\mathrm{A} \in \mathbf{T}_{\Gamma}$ *and* $\mathrm{B} \in \mathbf{T}_{\Gamma.\mathrm{A}}$ *natural in* f *and* B.

Proof. We restrict the functor Π to the fibre category $\mathrm{Cart}(\mathbf{T})_{\Gamma.A}$ and compose with χ to get a functor $\Pi_A \colon \mathrm{Cart}(\mathbf{T})_{\Gamma.A} \to \mathbf{C}/_{\Gamma.A}$. For $f \colon \Delta \to \Gamma$ and $B \in \mathbf{T}_{\Gamma.A}$:

$$\begin{aligned}
\mathbf{C}/_{\Gamma}(f, \chi(\Pi_A B)) &\cong \mathbf{C}/_{\Delta}(\mathrm{Id}_\Delta, \chi(\Pi_A B[f])) && \text{by lemma 2.4.0.9} \\
&\cong \mathbf{C}/_{\Delta}(\mathrm{Id}_\Delta, \chi(\Pi_{A[f]} B[f.A])) && \Pi \text{ is fibred} \\
&\cong \mathbf{C}/_{\Delta.A[f]}(\mathrm{Id}_{\Delta.A[f]}, \chi(B[f.A])) && \eta \\
&\cong \mathbf{C}/_{\Gamma.A}(f.A, \chi(B)) && \text{by lemma 2.4.0.9.}
\end{aligned}$$

Pseudo-stability imply that the above isomorphism can be given explicitly as

$$\delta_{f,B}(g) = \mathrm{app}_{A,B} \circ g.A$$

instead of the more complicated

$$\delta_{f,B}(g) = \{(f.A)_B\} \circ \mathrm{app}_{A[f],B[f.A]} \circ (g.A)^*.$$

Suppose $\tau \colon B \to C$ in $\mathbf{T}_{\Gamma.A}$ is cartesian. For naturality in B we need to show that for any $g \in \mathbf{C}/_{\Gamma}(f, \chi(\Pi_A B))$ we have

$$\{\tau\} \circ \delta_{f,B}(g) = \delta_{f,C}(\{\Pi_A(\tau)\} \circ g)$$

as morphisms $\Delta.A[f] \to \Gamma.A.C$ over $\Gamma.A$. The left-hand side expands to $\{\tau\} \circ \mathrm{app}_{A,B} \circ g.A$ while the right-hand side expands to $\mathrm{app}_{A,C} \circ (\{\Pi_A(\tau)\} \circ g).A$ which by functoriality of $(-).A$ and pseudo-stability are equal (see 14).

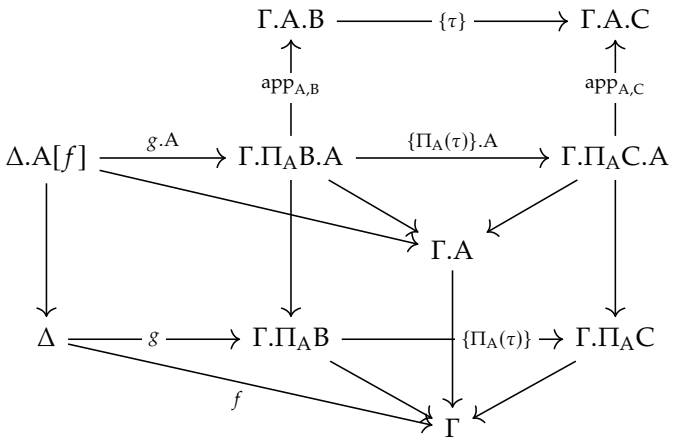


Figure 14

For naturality in f let $h \colon g \to f$ in the slice over Γ and $z \in \mathbf{C}/_{\Gamma}(f, \chi(\Pi_A B))$, then

$$\begin{aligned}
\delta_{g,B}(z \circ h) &= \mathrm{app}_{A,B} \circ (z \circ h).A \\
&= \mathrm{app}_{A,B} \circ z.A \circ h.A \\
&= \delta_{f,B}(z) \circ h.A
\end{aligned}$$

by functoriality of $(-).A$. □

Proposition 3.2.4.2. *A comprehension category with pseudo-stable dependent products satisfying the η-rule induce a relative adjunction*

$$\mathcal{W} \dashv_{\chi} \chi \circ \Pi \tag{9}$$

that is fibred, i.e. all the functors preserve cartesian morphisms and the counit is vertial.

Notation 3.2.4.3. The relative adjunction (9) means that for objects $[\mathrm{D}, f]$ where $\mathrm{D} \in \mathbf{T}_{\Delta}$, $f : \Delta' \to \Delta$, and $[\mathrm{A}, \mathrm{B}]$ where $\mathrm{A} \in \mathbf{T}_{\Gamma}$, $\mathrm{B} \in \mathbf{T}_{\Gamma.\mathrm{A}}$ there is an isomorphism

$$\mathrm{Cart}(\mathbf{T}) \underset{p}{\times} \mathbf{C}^{\to}\left(\left[\mathrm{D}, f\right], [\mathrm{A}, \chi\left(\Pi_{\mathrm{A}}\mathrm{B}\right)]\right) \overset{\Phi}{\cong} \mathrm{Cart}(\mathbf{T}) \underset{\{-\}}{\times} \mathbf{C}^{\to}\left(\mathcal{W}\left[\mathrm{D}, f\right], [\mathrm{A}, \chi(\mathrm{B})]\right)$$

natural in $\left[\mathrm{D}, f\right]$ and $[\mathrm{A}, \mathrm{B}]$ fibred over $\mathrm{Cart}(\mathbf{T})(\mathrm{D}, \mathrm{A})$. Since the isomorphism preserves arrows over homsets in the base category this allows us to introduce the following somewhat shorter notation. For cartesian $\sigma_{\mathrm{D}} : \mathrm{D} \to \mathrm{A}$ we write

$$\mathbf{C}^{\to}_{p\sigma_{\mathrm{D}}}(f, \chi(\Sigma_{\mathrm{A}}\mathrm{B}))$$

for the collection of morphisms in $\mathrm{Cart}(\mathbf{T}) \underset{p}{\times} \mathbf{C}^{\to}([\mathrm{D}, f], [\mathrm{A}, \chi(\Pi_{\mathrm{A}}\mathrm{B})])$ above σ_{D}. This allows us to write the above isomorphism as

$$\Phi : \mathbf{C}^{\to}_{p\sigma_{\mathrm{D}}}(f, \chi(\Sigma_{\mathrm{A}}\mathrm{B})) \cong \mathbf{C}^{\to}_{\{\sigma_{\mathrm{D}}\}}(\mathcal{W}(f), \chi(\mathrm{B}))$$

where the objects A and D are inferred from the codomain and domain of σ_{D}.

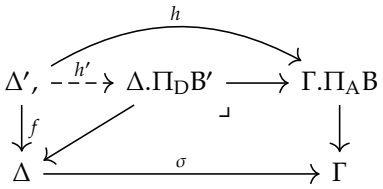
Proof of 3.2.4.2. For cartesian $\sigma_{\mathrm{D}} : \mathrm{D} \to \mathrm{A}$ above $\sigma : \Delta \to \Gamma$ define

$$\Phi : \mathbf{C}^{\to}_{p\sigma_{\mathrm{D}}}(f, \chi(\Pi_{\mathrm{A}}\mathrm{B})) \to \mathbf{C}^{\to}_{\{\sigma_{\mathrm{D}}\}}(\mathcal{W}(f), \chi(\mathrm{B}))$$

for $f : \Delta' \to \Delta$ as follows. An object in the left homset is a commutative square

$$\begin{array}{ccc} \Delta' & \xrightarrow{h} & \Gamma.\Pi_{\mathrm{A}}\mathrm{B} \\ \downarrow{\scriptstyle f} & & \downarrow \\ \Delta & \xrightarrow{\sigma} & \Gamma \end{array}$$

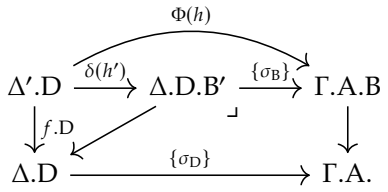
which we can factor as



where the pullback square is given by the cartesian action of Π on $(\sigma_{\mathrm{D}}, \sigma_{\mathrm{B}})$, and $\sigma_{\mathrm{B}} : \mathrm{B}' \to \mathrm{B}$ is the cartesian lift of $\{\sigma_{\mathrm{D}}\}$ into B. With lemma 3.2.4.1 we can transport h' to an arrow in the slice over $\Delta.\mathrm{D}$

$$\begin{array}{ccc} \Delta'.\mathrm{D} & \xrightarrow{\delta(h')} & \Delta.\mathrm{D}.\mathrm{B}' \\ \downarrow & \swarrow & \\ \Delta.\mathrm{D} & & \end{array}$$

and finally postcomposing with $(\{\sigma_D\}, \{\sigma_B\})$ gives $\Phi(h) = \{\sigma_B\} \circ \delta(h')$:



One easily verifies that all the steps involved in constructing Φ are invertible. For naturality, recall that $\delta(h')$ factors as $\delta(h') = \mathrm{app}_{D,B'} \circ (h').D$ so that we can apply the assumption of pseudo-stability to get

$$\begin{aligned}\Phi(h) &= \{\sigma_B\} \circ \mathrm{app}_{D,B} \circ (h'.D) \\ &= \mathrm{app}_{A,B} \circ \sigma_D.\Pi[\sigma_D, \sigma_B] \circ (h'.D) \\ &= \mathrm{app}_{A,B} \circ \mathcal{W}(h).\end{aligned}$$

Showing that the isomorphism is natural in (A, B) in $\mathrm{Cart}(\mathbf{T}) \underset{\{-\}}{\times} \mathrm{Cart}(\mathbf{T})$ amounts to proving that for $\alpha_A \colon A \to A'$ over $\alpha \colon \Gamma \to \Delta$ and $\alpha_B \colon B \to B'$ over $\{\alpha_A\}$

$$\Phi(\{\Pi[\alpha_A, \alpha_B]\} \circ h) = (\{\alpha_A\}, \{\alpha_B\}) \circ \Phi(h).$$

Expanding both sides gives

$$\mathrm{app}_{A',B'} \circ \mathcal{W}(\{\Pi[\alpha_A, \alpha_B]\}) \circ \mathcal{W}(h) = (\{\alpha_A\}, \{\alpha_B\}) \circ \mathrm{app}_{A,B} \circ \mathcal{W}(h)$$

which holds by assumption of pseudo-stability. For naturality in (D, f) consider

$$\begin{array}{ccc} \kappa' & \xrightarrow{h'} & \Delta' \\ \downarrow{\scriptstyle f'} & & \downarrow{\scriptstyle f} \\ \kappa & \xrightarrow{\alpha} & \Delta \end{array}$$

and a cartesian morphism $\alpha_{D'} \colon D' \to D$ over α. We then want $\Phi(h \circ h') = \mathrm{app}_{A,B} \circ \mathcal{W}(h) \circ \mathcal{W}(h')$. Unfolding Φ gives $\Phi(h \circ h') = \mathrm{app}_{A,B} \circ \mathcal{W}(h \circ h')$, naturality follows from functoriality of $\mathcal{W}$. Finally, we observe that Φ is fibred over $\mathrm{Cart}(\mathbf{T})(D,A)$ by construction since the action of Φ on cartesian morphisms in the first component is the identity, and projecting down to $\mathrm{Cart}(\mathbf{T})$ is the identity of the first component. □

Proposition 3.2.4.4. *A comprehension category with a fibred functor*

$$\Pi \colon \mathrm{Cart}(\mathbf{T}) \underset{\{-\}}{\times} \mathrm{Cart}(\mathbf{T}) \to \mathrm{Cart}(\mathbf{T}) \underset{p}{\times} \mathrm{Cart}(\mathbf{T})$$

with an isomorphism

$$\Phi \colon \mathrm{Cart}(\mathbf{T}) \underset{p}{\times} \mathbf{C}^{\to}(f, \chi(\Pi_A B)) \to \mathrm{Cart}(\mathbf{T}) \underset{\{-\}}{\times} \mathbf{C}^{\to}(\mathcal{W}(f), \chi(B))$$

natural in (D, f) *and* (A, B) *and fibred over* $\mathbf{T}(A, D)$ *has pseudo-stable dependent products satisfying the* η*-rule.*

Proof. Following the notation introduced in 3.2.4.3 we take Φ to be an isomorphism

$$\Phi_{\sigma_D,f} : \mathbf{C}^{\rightarrow}_{p\sigma_D}(f, \chi(\Sigma_A B)) \cong \mathbf{C}^{\rightarrow}_{\{\sigma_D\}}(\mathcal{W}(f), \chi(B)) \tag{10}$$

for each suitable cartesian morphism σ_D. When σ_D is an identity morphism we may identify objects of the above homsets with canonical morphisms in corresponding slice categories. For example when $\sigma_D = \mathrm{Id}_A$ above Γ in 10 we write $\mathbf{C}/_\Gamma(f, \chi(\Pi_A B)) \cong \mathbf{C}/_{\Gamma.A}(\mathcal{W}(f), \chi(B))$.

For all objects $\Gamma \in \mathbf{C}$, $A \in \mathbf{T}_\Gamma$ and $B \in \mathbf{T}_{\Gamma.A}$,

- let $\sigma_D = \mathrm{Id}_A$ and $f = \chi(\Pi_A B)$ in (10) to define

  $$\mathrm{app}_{A,B} := \Phi_{\mathrm{Id}_A, \chi(\Pi_A B)}(\mathrm{Id}_{\chi(\Pi_A B)}) : \mathcal{W}(\Pi_A B) \to \chi(B)$$

  in $\mathbf{C}/_{\Gamma.A}$;
- define $\lambda_{A,B} := \Phi^{-1}_{\mathrm{Id}_A, \mathrm{Id}_{\Gamma.A}} : \mathbf{C}/_{\Gamma.A}(\mathrm{Id}_{\Gamma.A}, \chi(B)) \to \mathbf{C}/_\Gamma(\mathrm{Id}_\Gamma, \chi(\Pi_A B))$.

Restricting the isomorphism (10) to $\sigma_D = \mathrm{Id}_A$ for each $A \in \mathbf{T}$ it follows from lemma 3.2.2.4 that the comprehension category has dependent products satisfying the η-rule.

Naturality of Φ implies

$$\{\sigma_E\} \circ \mathrm{app}_{D,E} = \mathrm{app}_{A,B} \circ \mathcal{W}(\{\Pi(\sigma_D, \sigma_E)\})$$

as witnessed by the naturality squares in figure 15.

$$\begin{array}{ccc}
\mathbf{C}^{\rightarrow}_{p\,\mathrm{Id}_A}(\chi(\Pi_A B), \chi(\Pi_A B)) & \xrightarrow{\mathrm{app}_{A,B}} & \mathbf{C}^{\rightarrow}_{\{\mathrm{Id}_A\}}(\chi(\Pi_A B).A, \chi(B)) \\
\Big\downarrow{\scriptstyle -\circ\{\Pi(\sigma_D,\sigma_E)\}} & & \Big\downarrow{\scriptstyle -\circ\mathcal{W}(\{\Pi(\sigma_D,\sigma_E)\})} \\
\mathbf{C}^{\rightarrow}_{p\sigma_D}(\chi(\Pi_D E), \chi(\Pi_A B)) & \xrightarrow{\Phi} & \mathbf{C}^{\rightarrow}_{\{\sigma_D\}}(\chi(\Pi_D E).D, \chi(B)) \\
\Big\uparrow{\scriptstyle \{\Pi(\sigma_D,\sigma_E)\}\circ -} & & \Big\uparrow{\scriptstyle \{\sigma_E\}\circ -} \\
\mathbf{C}^{\rightarrow}_{p\,\mathrm{Id}_D}(\chi(\Pi_D E), \chi(\Pi_D E)) & \xrightarrow{\mathrm{app}_{D,E}} & \mathbf{C}^{\rightarrow}_{\{\mathrm{Id}_D\}}(\chi(\Pi_D E).D, \chi(E))
\end{array}$$

Figure 15: Pseudo-stability of app.

Similarly for the λ-rule. Let $\sigma : \Delta \to \Gamma$, $\sigma_D : D \to A$ cartesian over σ, $\sigma_E : E \to B$ cartesian over $\{\sigma_D\}$. The goal is to show that for every section $t : \Gamma.A \to \Gamma.A.B$

$$\lambda_{A,B}(t) \circ \sigma = \{\Pi(\sigma_D, \sigma_E)\} \circ \lambda_{D,E}(t[\sigma_E]).$$

Consider the two naturality squares in diagram 16.

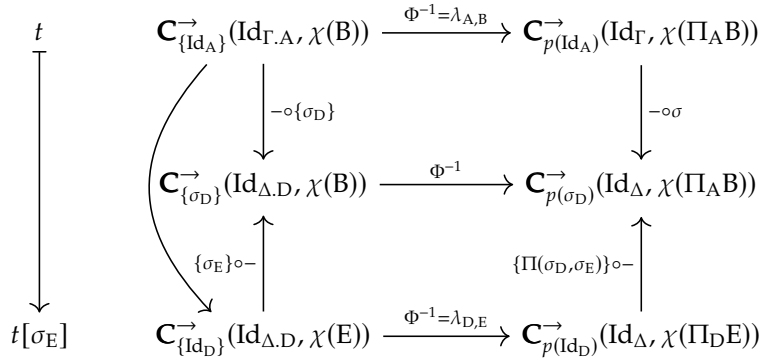


Figure 16: Pseudo-stability of λ.

The leftmost curved arrow is the map taking a section $t : \Gamma.A \to \Gamma.A.B$ to the unique section $t[\sigma_E] : \Delta.D \to \Delta.D.E$ such that

$$\{\sigma_E\} \circ t[\sigma_E] = t \circ \{\sigma_D\} \tag{11}$$

obtained by the pullback square $\chi(\sigma_E)$, thus making the left triangle commute. The result then follows:

$$\begin{aligned} \lambda_{A,B}(t) \circ \sigma &= \Phi^{-1}(t \circ \{\sigma_D\}) && \text{by naturality} \\ &= \Phi^{-1}(\{\sigma_E\} \circ t[\sigma_E]) && \text{by (11)} \\ &= \{\Pi[\sigma_D, \sigma_E]\} \circ \lambda_{D,E}(t[\sigma_E]) && \text{by naturality.} \end{aligned}$$

□

References

[Awo10] Steve Awodey. *Category Theory*. 2nd ed. Oxford Logic Guides 52. Oxford ; New York: Oxford University Press, 2010. 311 pp. ISBN: 978-0-19-923718-0.

[Bla91] Javier Blanco. "Relating Categorical Approaches to Type Dependency". Master Thesis. University of Nijmegen, 1991.

[Cam14] Tim Campion. *An Exegis of Yoneda Structures*. n-Category Café. May 25, 2014. URL: https://golem.ph.utexas.edu/category/2014/03/an_exegesis_of_yoneda_structur.html#c046168 (visited on 10/06/2021).

[Car86] John Cartmell. "Generalised Algebraic Theories and Contextual Categories". In: *Ann. Pure Appl. Logic* 32 (1986), pp. 209–243. ISSN: 0168-0072.

[Hof95] Martin Hofmann. "On the Interpretation of Type Theory in Locally Cartesian Closed Categories". In: *Computer Science Logic*. Ed. by Leszek Pacholski and Jerzy Tiuryn. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1995, pp. 427–441. ISBN: 978-3-540-49404-1. DOI: 10.1007/BFb0022273.

[Jac93] Bart Jacobs. "Comprehension Categories and the Semantics of Type Dependency". In: *Theoretical Computer Science* 107.2 (Jan. 18, 1993), pp. 169–207. ISSN: 0304-3975. DOI: 10.1016/0304-3975(93)90169-T.

[Jac99] Bart Jacobs. *Categorical Logic and Type Theory*. Elsevier, Jan. 14, 1999. 779 pp. ISBN: 978-0-08-052870-0.

[Law70] F William Lawvere. "Quantifiers and Sheaves". In: *Actes Congrès Intern. Math.* 1 (1970), pp. 329–334.

[LW15] Peter LeFanu Lumsdaine and Michael A. Warren. "The Local Universes Model: An Overlooked Coherence Construction for Dependent Type Theories". In: *ACM Transactions on Computational Logic* 16.3 (July 8, 2015), pp. 1–31. ISSN: 1529-3785, 1557-945X. DOI: 10.1145/2754931. arXiv: 1411.1736.

[Mar84] Per Martin-Löf. *Intuitionistic Type Theory*. Vol. 1984. Studies in Proof Theory. Bibliopolis, 1984. ISBN: 88-7088-105-9.

[NPS90] Bengt Nordström, Kent Petersson, and Jan M. Smith. *Programming in Martin-Löf's Type Theory: An Introduction*. International Series of Monographs on Computer Science 7. Oxford : New York: Clarendon Press ; Oxford University Press, 1990. 221 pp. ISBN: 978-0-19-853814-1.

[See84] R. A. G. Seely. "Locally Cartesian Closed Categories and Type Theory". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 95.1 (Jan. 1984), pp. 33–48. ISSN: 0305-0041, 1469-8064. DOI: 10.1017/S0305004100061284.

[Str20] Thomas Streicher. *Fibred Categories a La Jean Benabou*. Nov. 30, 2020. arXiv: 1801.02927 [math]. URL: http://arxiv.org/abs/1801.02927.

[SW78] Ross Street and Robert Walters. "Yoneda Structures on 2-Categories". In: *Journal of Algebra* 50.2 (Feb. 1, 1978), pp. 350–379. ISSN: 0021-8693. DOI: 10.1016/0021-8693(78)90160-6.

[Ulm68] Friedrich Ulmer. "Properties of Dense and Relative Adjoint Functors". In: *Journal of Algebra* 8.1 (Jan. 1, 1968), pp. 77–95. ISSN: 0021-8693. DOI: 10.1016/0021-8693(68)90036-7.

[Uni13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013. URL: https://homotopytypetheory.org/book.