Università
di Genova

Scuola di Scienze Matematiche, Fisiche e Naturali

Dipartimento di Matematica

Tesi di dottorato

Categorical structures for deduction

CANDIDATO:
Greta Coraglia

RELATORE:
Chiar.mo Prof. Giuseppe Rosolini

ANNO ACCADEMICO 2021–22

Contents

Introduction

One could argue [Eve17, Chapter 2] that counting is one of the first instances of abstraction in the history of the human species. At some point, not later than 10.000 years ago[1], humans discovered that they could *represent* a certain amount of people with fingers on their hand, sheep with marks on a piece of clay, days with incisions on a stick. This representation process took the form of a bijection, so that one mark would be a *stand in* for a sheep and each sheep would be recorded as a mark, and its convenience must have been immediately clear as one could much easily look at a hundred marks than at a hundred sheep. After that, noticing that both a hundred sheep and a hundred days could be represented by the same marks, humans decided that perhaps studying the *thing* "one hundred" would bring them some benefit in and of itself - and it did.

A bunch of thousands of years later, (some) humans still seem to be convinced of the convenience of abstraction, and a few have invented category theory. Category theory has proved to be an interesting tool to reason about common phenomena in mathematics: its level of generality allows us to forget everything but the property we want to consider each time, and this often leaves room for a detached perspective that is hard to otherwise achieve. For example, the Yoneda lemma tells us that to better understand an object one can look at how it interacts with others.

Moreover, when one decides on some properties that they are interested in, for example describing the notion of product of two sets, category theory allows to see what happens when the same notion is transported in another context, say for example topological spaces, and new interesting things are discovered, such as that the "right" topology on the product of two topological spaces is in fact the product topology. The marks on the stick representing the product can be themselves studied, and this brings benefit.

Even though category theory was born [EML45] out of an abstraction of concepts that are exquisitely geometrical, of vector spaces, and topological groups and such, some people soon realized that its language, and its marks, would nicely represent logical concepts as well. The content of this thesis is simply the result of reading the marks on the stick resulting of this process.

In other words: category theory has developed some tools and structures, and marked them on a very nice stick; then, a bunch of people have realized that these marks finely describe some logical notions; we come and take these idea to the extreme, and with logic in mind, read whatever we can on that very

[1] This it the approximate dating of a marked reindeer antler discovered in Little Salt Spring, Florida [GW07], and of markings made in a cave in Monte Alegre [BBD+97]. The author is neither an archaeologist nor an anthropologist, but thought framing the discussion that follows in a *broad* perspective to be quite helpful.

same stick. This process brings some benefits.

Marks of categorical logic

We do not dare put together the history of categorical logic, but we could not start our journey without mentioning a few points that have had a particular influence on the present work. A first insight into the way in which categories could help in the study of mathematical logic is contained in F. W. Lawvere's Ph.D. thesis [Law63]. In a later reflection upon its intent, he writes:

> (5) The dialectical contrast between presentations of abstract concepts and the abstract concepts themselves, as also the contrast between word problems and groups, polynomial calculations and rings, etc. can be expressed as an explicit construction of a new adjoint functor out of any given adjoint functor. Since in practice many abstract concepts (and algebras) arise by means other than presentations, it is more accurate to apply the term "theory", not to the presentations as had become traditional in formalist logic, but rather to the more invariant abstract concepts themselves which serve a pivotal role, both in their connection with the syntax of presentations, as well as with the semantics of representations. [Law04, Seven ideas introduced in the 1963 thesis]

Therefore the notion of theory ought to be reframed within this new language, as it possesses the tools to better describe the relation between syntax and semantics. The reader might find that our discussion on marks and sticks is very much in line with Lawvere's perspective, and in fact he next writes:

> (6) The leap from particular phenomenon to general concept, as in the leap from cohomology functors on spaces to the concept of cohomology operations, can be analyzed as a procedure meaningful in a great variety of contexts and involving functorality and naturality, a procedure actually determined as the adjoint to semantics and called extraction of "structure" (in the general rather than the particular sense of the word). [Law04, Seven ideas introduced in the 1963 thesis]

This breaking shift was followed by a period of intense advance, not only in the study of Lawvere's algebraic theories [Law65, EW67, Fre66, Isb64], but also in those that would later become the theories of topoi [Law69, Law64, Fre72], of fibrations [Law70, Bén68], and of monads [EM65, BB66]. Of course this distinction is merely artificial, as one would imagine in such a fast-growing environment: monads are closely related to pairs of adjoint functors, which in turn are related to certain algebraic theories, which in turn are related to certain fibered categories, some of which give rise to topoi, and so on. In fact, all of these will appear in the following chapters.

Nevertheless, if we were asked to single out the concept that has influenced the present work the most, it would be that of fibration. Again, fibrations were introduced in [Gro61] with quite a different purpose in mind, namely to abstract various descent phenomena in geometry. In [Law69, Law70] Lawvere observed that many key concepts in first-order logic (such as connectives and quantifiers)

could be expressed as adjoint pairs of functors between different categories, each representing formulae in a fixed context, and that this way of organizing logical data could be synthesized in a fibered category. An in-depth introduction to fibrations will be the beginning of this exposition and the content of Chapter 1.

The problem with equality

Dealing with fibrations, and in particular with the universal property of cartesian liftings, poses a few problems that go otherwise unnoticed in other branches of mathematics, we refer in particular to the problem of choice, of coherence, and, above all, of equality. In particular, to have some sort of functoriality, hence a given choice of cartesian lifting for each map, either some form of the axiom of choice is needed, or in some sense a canonical choice ought to be known *a priori* or by other means and from the start. Moreover, the moment one wants to make such a choice, some necessary *coherence* conditions are required so that calculations get more laborious, and new conditions have to be introduced. These issues are dealt with in the study of split fibrations, and come into play several times in the course of this work. But while dealing with this fleeting notion of equality - that entertained between two cartesian liftings of the same map - might appear as a complication of the theory, A. Grothendieck seems to have thought it to be one of its main features. He writes:

> Bien entendu, il y a intérêt le plus souvent à raisonner directement sur des catégories fibrées sans utiliser des clivages explicites, ce qui dispense en particulier de faire appel, pour la notion simple de [...] foncteur cartesién, à une interprétation pesante comme ci-dessus. C'est pour éviter des lourdeurs insupportables, et pour obtenir des énoncés plus intrinsèques, que nous avons dû renoncer à partir de la notion de catégorie clivée [...], qui passe au second rang au profit de celle de catégories fibrée.
>
> Il est d'ailleurs probable que, contrairement à l'usage encore prépondérant maintenant, lié à d'anciennes habitudes de pensée, il finira par s'avérer plus commode dans les problèms universels, de ne pas mettre l'accent sur *une* solution supposée choisie une fois pour toutes mais de mettre toutes les solutions sur un pied d'egalité.[2] [Gro60]

We will repeatedly point out implications of such a "problem", as it will follow us throughout the text. An interesting approach that isolates this phenomenon and makes it particularly clear is that of *displayed categories* [AL17]: we recommend §1 thereof for an inspired account.

[2] "Of course, it is most often useful to reason directly about fibered categories without using explicit cleavages, without the need in particular to appeal, for the simple notion of [...] cartesian functor, to a heavy interpretation as above. It is to avoid unbearable heaviness, and to obtain more intrinsic enunciations, that we had to renounce (or depart) from the notion of split categories [...], which takes second place with respect to that of fibered categories.

It is moreover probable that, contrary to the use still prevalent now, linked to old ways of thinking, it will end up being more convenient for universal problems, not to put the emphasis on a supposed solution chosen once and for all, but to put all solutions on an equal footing."

The author dearly thanks her sister for the translation.

Content of this thesis

In Chapter 2 we introduce categorized judgemental theories and their calculi as a general framework to present and study deductive systems. As an exemplification of their expressivity, we approach dependent type theory and natural deduction as special kinds of categorized judgemental theories. Our analysis sheds light on both the topics, providing a new point of view. In the case of type theory, we provide an abstract definition of type constructor featuring the usual formation, introduction, elimination and computation rules. For natural deduction we offer a deep analysis of structural rules, describing some of their properties, and putting them into context.

In Chapter 3 we put one of the main constructions introduced in Chapter 2, namely that of categorized judgemental dependent type theories, to the test: we frame it in the general context of categorical models for dependent types, describe a few examples, study its properties, and use it to model subtyping and as a tool to prove intrinsic properties hidden in other models.

In 2 and 3 we generalize known models, in Chapter 4 we transport one from set-based categories to enriched categories, and use the information naturally encoded into it to describe a theory of fuzzy types. We recover structural rules, observe new phenomena, and study different possible enrichments and their interpretation. We open the discussion to include different takes on the topic of definitional equality.

Chapter 1

Fibrations

> Category theory starts with the observation that many properties of mathematical systems can be unified and simplified by a presentation with diagrams of arrows.
>
> [ML78, Introduction]

1.1 Categories: notation and underlying theory

Our discussion takes place in the world of categories: we here recall some basic definitions, fix notation, and describe the underlying theory we are working in.

This account follows many introductory texts, above all [ML78] and [Rie17].

1.1.1 Categories

Definition 1.1.1.1 (Category)**.** A *category* $\mathcal{C}$ is the data of

- a collection of *objects* $A, B, C, \dots$, and
- a collection of *morphisms*, or *arrows* or *maps*, $f, g, h, \dots$

where

- each morphism has given *domain* and *codomain* objects: we write $f\colon A \to B$ to mean that f is a morphism with domain A and codomain B;
- for each object A there is a designated *identity morphism* $\mathrm{id}_A\colon A \to A$;
- for each pair of *composable* morphisms f, g, *i.e.* morphisms such that the codomain of f is the domain of g, there exists a specified *composition morphism* gf or $g \circ f$ with domain the domain of f and codomain the codomain of g, meaning

$$f\colon A \to B,\ g\colon B \to C \quad \rightsquigarrow \quad gf\colon A \to C;$$

such that

- for any $f\colon A \to B$, the composites $\mathrm{id}_B f$ and $f\mathrm{id}_A$ are both equal to f;
- for any composable triple f, g, h, the composites $h(gf)$ and $(hg)f$ are equal and from now on denoted hgf.

In other words, composition is associative and has identities for (left and right) units.

Notation 1.1.1.2. We usually write $\mathcal{C}(A, B)$ for the class of morphisms from A to B in $\mathcal{C}$.

Example 1.1.1.3. One traditionally refers to a category with the name of its objects.

1. Call **Set** the category having for objects sets[1], for morphism functions with their domain and codomain, composition of functions and identity functions.
2. Call **Top** the category of topological spaces and continuous functions. Composition and identity are those of functions.
3. Call **Grp** the category of groups and group homomorphisms. Again, composition and identity are those of functions.
4. Similarly, we can define a category **Ab** of abelian groups, **Ring** for rings, R-**Mod** for R-modules, **Pos** for posets, and for many other algebraic structures.
5. Call $\mathbf{Set}_\bullet$ the category of pointed sets and functions preserving the points. Similarly define $\mathbf{Top}_\bullet$.
6. Call **Meas** the category of measurable spaces and measurable functions.
7. Call **1** the category having one object and one (necessarily identity) morphism.

Notation 1.1.1.4. Usually we use the letters $\mathcal{C}, \mathcal{D}, \mathcal{E}$ for unspecified categories, and **Set**, **Top**, **Grp** for given categories.

Example 1.1.1.5. We have two paradigmatic examples with opposite "minimal" information.

- Let $(P, \leq)$ a poset. It induces a category $\overline{P}$ with objects the elements of the poset, and between two object there is at most one map, and there is one $p \to p'$ if and only if $p \leq p'$. Composition is given by transitivity and identity by reflexivity.
- Let $(M, \cdot, 1)$ a monoid. It induces a category $\underline{M}$ with one object and morphisms the elements of the monoid. Composition and identities are provided by, respectively, $\cdot$ and 1.

[1] By "set" we mean a set according to any choice of foundation, such as ZF, ZFC, NBG, and others. See Section 1.1.4 for more on the meta-theory we are working in.

Definition 1.1.1.6 (Opposite category)**.** For a category $\mathcal{C}$, one can define its *opposite category* $\mathcal{C}^{\mathsf{op}}$ to be that having the same objects and maps going in the opposite direction: $\mathcal{C}^{\mathsf{op}}(A, B) = \mathcal{C}(B, A)$.

In the very spirit of categories, putting the accent of morphisms over objects, we can define a notion of morphism between categories.

1.1.2 Functors and natural transformations

Definition 1.1.2.1 (Functor)**.** A *functor* F from $\mathcal{C}$ to $\mathcal{D}$, written $F\colon \mathcal{C} \to \mathcal{D}$, is the data of

- for each object A in $\mathcal{C}$, an object $F(A)$ in $\mathcal{D}$, and
- for each morphism $f\colon A \to B$ in $\mathcal{C}$, a morphism $Ff\colon F(A) \to F(B)$ in $\mathcal{D}$,

so that for any composable pair f, g, the composition $Fg \circ Ff$ is equal to $F(gf)$, and for all A, $F(\mathrm{id}_A)$ is equal to $\mathrm{id}_{F(A)}$.

Example 1.1.2.2. One can define many functors between the categories in Example 1.1.1.3.

1. There is a functor $U\colon \mathbf{Grp} \to \mathbf{Set}$, called *forgetful*, sending each group to its underlying set. There are similar functors with domain $\mathbf{Top}, \mathbf{Ring}, \mathbf{Set}_\bullet$ and so on. Similarly, one can define a functor $U\colon R\text{-}\mathbf{Mod} \to \mathbf{Ab}$ forgetting the multiplicative structure.
2. The *fundamental group* defines a functor $\pi_1\colon \mathbf{Top}_\bullet \to \mathbf{Grp}$.
3. The operation of taking the *free group* over a given set produces a functor $F\colon \mathbf{Set} \to \mathbf{Grp}$.
4. For an object A in a locally small category $\mathcal{C}$ we can define a functor $\mathcal{C}(A, -)\colon \mathcal{C} \to \mathbf{Set}$ that to each object B maps the set of morphisms $A \to B$, and at each morphism acts as postcomposition. Similarly, one can define a functor $\mathcal{C}(-, A)\colon \mathcal{C}^{\mathsf{op}} \to \mathbf{Set}$.

Of course, we could wonder why should we stop at maps between categories when we have new objects to relate, namely functors.

Definition 1.1.2.3 (Natural transformation)**.** A *natural transformation* α from F to G, written $\alpha\colon F \Rightarrow G$, is the data of

- for each object A in $\mathcal{C}$, a morphism $\alpha_A\colon F(A) \to G(A)$ in $\mathcal{D}$, called the *component* of α at A,

such that at any $f\colon A \to B$ the square

$$\begin{array}{ccc} F(A) & \xrightarrow{\alpha_A} & G(A) \\ {\scriptstyle Ff}\downarrow & & \downarrow{\scriptstyle Gf} \\ F(B) & \xrightarrow[\alpha_B]{} & G(B) \end{array}$$

commutes.

Example 1.1.2.4. Let $\mathcal{C}$ be a locally small category, then every morphism $f\colon A \to B$ produces a natural transformation $f^*\colon \mathcal{C}(B,-) \Rightarrow \mathcal{C}(A,-)$ that acts by precomposition.

Functors and natural transformations can be composed in different ways, depending on how they combine.

Lemma 1.1.2.5 (Vertical composition, [ML78, II.5])**.** Suppose $\alpha\colon F \Rightarrow G$ and $\beta\colon G \Rightarrow H$ are natural transformations between parallel $F, G, H\colon \mathcal{C} \to \mathcal{D}$. Then there is a natural transformation $\beta \cdot \alpha\colon F \Rightarrow G$ defined as $(\beta \cdot \alpha)_A = \beta_A \circ \alpha_A$.

Lemma 1.1.2.6 (Horizontal composition, [ML78, II.5])**.** Suppose $\alpha\colon F \Rightarrow G$ and $\beta\colon H \Rightarrow K$ are natural transformations where $F, G\colon \mathcal{C} \to \mathcal{D}$ and $H, K\colon \mathcal{D} \to \mathcal{E}$. Then there is a natural transformation $\beta * \alpha\colon HF \Rightarrow KG$ where $(\beta * \alpha)_A$ is provided by the following composition.

$$\begin{array}{ccc} HF(A) & \xrightarrow{\beta_{F(A)}} & KF(A) \\ {\scriptstyle H\alpha_A}\downarrow & \searrow & \downarrow{\scriptstyle K\alpha_A} \\ HG(A) & \xrightarrow[\beta_{G(A)}]{} & KG(A) \end{array}$$

The horizontal composition of α and β is called their *whiskering*. When many are computed, we might omit writing the "$*$": that is the case, for example, in Section 3.2.

Corollary 1.1.2.7. As one could consider either α or β to be the identity, this produces a way to compose natural transformations with functors, as well.

1.1.3 2-categories

Of course one could go on and try describing morphisms of natural transformations, and then morphisms of such morphisms, and so on. For the purpose of this thesis, we only need information up to level 2, meaning concerning objects, functors, and natural transformations. Actually, considering them all together encourages a new definition to be given.

Definition 1.1.3.1 (2-category)**.** A *2-category* $\mathcal{C}$ is the data of

- a collection of *objects* or *0-cells*, and
- a collection of *morphisms* or *1-cells* between pairs of 0-cells, and
- a collection of *2-cells* between parallel pairs of 1-cells

such that

- 0-cells and 1-cells form a category,
- for each pair of objects A, B, $\mathcal{C}(A,B)$ is a category with objects the 1-cells of the form $f\colon A \to B$ and morphisms the 2-cells between them, with composition $\cdot$ called *vertical composition*,
- for any object A there is a functor $\mathbf{1} \to \mathcal{C}(A,A)$ that picks out the *identity 1-cell* id_A and its *identity 2-cell* $\mathrm{Id}_{\mathrm{id}_A}$,

- for all objects A, B, C there is a functor $*\colon \mathcal{C}(B,C) \times \mathcal{C}(A,B) \to \mathcal{C}(A,C)$ called *horizontal composition* which is associative and satisfies $\mathrm{Id}_{\mathrm{id}_H} * \mathrm{Id}_{\mathrm{id}_F} = \mathrm{Id}_{\mathrm{id}_{HF}}$.

Example 1.1.3.2. The category **Cat** of locally small categories is a 2-category, with vertical and horizontal composition as in Lemma 1.1.2.5 and Lemma 1.1.2.6.

One can naturally define the appropriate notion of a functor between 2-categories, namely a *2-functor*, and extend many related concepts, such as that of adjoint pair, equivalence, and so on. We will see it all applied in the following chapter, but point the reader to expository texts for a formal introduction, see for example [KS74] for an early presentation of the topic.

1.1.4 Logical environment

As this aims to be a thesis in logic, above all, we ought to spend a few words on what is the logical environment the theory of (fibered) categories lives in. Different scholars have taken different approaches, some considering an extension of ZF(C) to include classes, some building a suitable hierarchy of Grothendieck universes, some using NBG. We do not dwell on their similarities or differences, but present perhaps the most commonly used, which the reader might find a detailed account of in [Shu08]. Other notable resources are [FK69] and [Bén85].

The reason why the theories described take into account at least *two* types of entities, commonly called *sets* and *classes*, is that some key categorical constructions that are interesting for mathematicians only make sense when one is allowed to restrict sizes to a smaller universe than the one one usually works in, or, conversely, one wishes to include constructions that inevitably explode the size over to a second, bigger, universe. Of course Cantor's paradox is an occurrence of the second, while one of the most famous examples of the first is the following.

Theorem 1.1.4.1 ([McL92, Ch. 24])**.** If a category $\mathcal{C}$ has products which are indexed by the collection of morphisms in $\mathcal{C}$, then $\mathcal{C}$ is a preorder. In particular, any small complete category is a preorder, and no large category that is not a preorder can admit products indexed by proper classes.

This work is meant to be read in ZFC, meaning Zermelo-Fraenkel set theory (a comprehensive account of which can be found in [Jec07]) with the axiom of choice. We consider a universe V whose elements we call *sets*, then any category having objects and arrows being sets in V we call *small* (with respect to V). If one wishes to consider the category of sets and functions, already they end up needing a new, bigger, universe. This can be achieved in the language of ZFC, still, because the axiom of replacement itself guarantees that the image of a set under any "class function" is a set. There exists a class of all sets, namely V, while there might be things, such as V, which are not sets: these we call *proper classes*. We say that a category is *large* if its collection of objects and arrows are proper classes, and say that it is *locally small* if the collection of arrows with fixed domain and codomain is a set.

Now to choice. There are different non-equivalent formulations of it, but in classical ZF it is expressed for sets. Given that we have classes, too, we might need to consider a version for classes: this is usually called the axiom of *global choice*. Since assuming them both remains a delicate matter, we will express

explicitly when we use each. As a staring point, we refer to [Jec07, I.5] for a technical presentation of choice in ZF, comprising of the principle of unique choice, and to [Mai17] for the case of dependent type theory.

Of course we are making our life easy here, and perhaps we are in the right, provided that discussing foundations of category theory is not the aim of the present work. Nevertheless, given the spirit of our approach to both logic and categories, entertaining the idea that categories could be *the* foundations, and foregoing sets completely is perhaps not so far-fetched. In fact, as [Bén85, §2] puts it, why should we exclude the possibility of having for sets the objects of an elementary topos, instead of "finding" **Set** itself, together with its properties? Well, one could argue [Bén85, §3] that the moment one decides to do that, and use categories and their most peculiar constructions, one cannot help but stumble on fibrations.

This is why we are here. We hope that the present work will convince the reader of what benefits using these techniques can bring to the study of logic.

1.2 Motivation

Fibrations were introduced by A. Grothendieck in [Gro61] following the observation of different phenomena arising in geometric and algebraic contructions. In particular, J. Bénabou writes:

> Fibrations were introduced in [Gro61] to axiomatize in terms of descent all the "gluing-processes". [Bén85, §12]

In order to describe a few, we first need to point out a simple consequence of the way in which categories are built: if the main notion of categories is that of arrows, it most makes sense to compare objects depending on the arrows relating them. In many ways, this is well-known in mathematical practice, because strict *identities* are often too restrictive to say something meaningful, and this is why such a big part of math involves *isomorphisms* instead. This has had many repercussions in logic as well, see for example the works based on sheaves [Sco79, Pit99], and the techniques of homotopy type theory [Uni13]. This suggests we reformulate many definitions from basic category theory asking for isomorphism instead of identity in certain equations. We will do it quite a lot in the following chapters, for the moment we only do it for the definition of functor.

Definition 1.2.0.1 (Pseudofunctor)**.** A *pseudofunctor* F from a category $\mathcal{C}$ into the category **Cat**, written $F\colon \mathcal{C} \rightsquigarrow \mathbf{Cat}$, is the data of

- for each object A in $\mathcal{C}$, an object $F(A)$ in **Cat**, and
- for each morphism $f\colon A \to B$ in $\mathcal{C}$, a morphism $Ff\colon F(A) \to F(B)$ in **Cat**,

so that for any composable pair f, g, the composition $Fg \circ Ff$ is isomorphic to $F(gf)$, and for all A, $F(\mathrm{id}_A)$ is isomorphic to $\mathrm{id}_{F(A)}$. Moreover, the isomorphisms above are all natural, associative and respect identities in the sense that certain diagrams commute.

The notion of pseudofunctor is nowadays mostly considered in the context of bicategories [Bén67], and a suitable definition can be expressed between any pair of bicategories or 2-categories, see for example [Lei98] for a thorough description of the necessary axioms. Throughout this presentation, instead, given that the mathematical gadget we are really interested in are Grothendieck fibrations (1.3.0.2), we simply consider pseudofunctors from 2-categories with trivial 2-cells (containing no more data than a 1-category) into **Cat**, and only use **Cat** second dimension to describe the necessary coherences. This is also the main interest of the original work introducing fibrations, [Gro61], where the perspective is that of providing a generalization of the notion of (set-valued) presheaf.

Example 1.2.0.2 (Right actions of a monoid)**.** Let **Mon** be the category of monoids and monoid homomorphisms. To every monoid M one can associate the category $H(M) = \mathbf{Set}^{M^{\mathsf{op}}}$ of right actions of M on (some) set, and for each $h\colon N \to M$ a functor

$$H(h) = h^*\colon \mathbf{Set}^{M^{\mathsf{op}}} \to \mathbf{Set}^{N^{\mathsf{op}}},$$

such that for each set X and M-action α, $h^*(X,\alpha)\colon X \times N \to X$ acts as $(x,n) \mapsto \alpha(x,h(n))$. This assembles into a pseudofunctor $H\colon \mathbf{Mon}^{\mathsf{op}} \rightsquigarrow \mathbf{Cat}$.

Example 1.2.0.3 (Modules for a ring)**.** Let **CRing** be the category of commutative rings with units and ring homomorphisms. To every ring R one can associate the category $H(R) = R\text{-}\mathbf{Mod}$ of R-modules and their morphisms, and for each $h\colon S \to R$ a functor

$$H(h)\colon R\text{-}\mathbf{Mod} \to S\text{-}\mathbf{Mod},$$

that "restricts" scalars, meaning that for an R-module M, $H(h)(M)$ is the S-module with the same addition as M, and multiplication $r \cdot x = h(r) \cdot_M x$. This assembles into a pseudofunctor $H\colon \mathbf{CRing}^{\mathsf{op}} \rightsquigarrow \mathbf{Cat}$.

Example 1.2.0.4 (Families of objects)**.** Let $\mathcal{C}$ be a category, and consider the functor

$$\mathsf{Fam}(\mathcal{C})\colon \mathbf{Set}^{\mathsf{op}} \to \mathbf{Cat}$$

that maps each set I to $\mathcal{C}^I$, the category of I-indexed objects in $\mathcal{C}$, and each function $u\colon J \to I$ to the reindexing along u. More precisely, an object in $\mathcal{C}^I$ is a family of the form $\{X_i\}_{i\in I}$ with X_i an object of $\mathcal{C}$ for each $i \in I$. The image of u is a functor $\mathcal{C}^I \to \mathcal{C}^J$ that maps a family $\{X_i\}_{i\in I}$ to $\{X_{u(j)}\}_{j\in J}$. This example is paradigmatic in the work of Bénabou and is greatly discussed in [Str22] and in [Jac99].

Example 1.2.0.5 (Slice category)**.** Consider $\mathcal{C}$ a category with pullbacks and define

$$P\colon \mathcal{C}^{\mathsf{op}} \rightsquigarrow \mathbf{Cat}$$

that maps an object A to the slice category $\mathcal{C}_{/A}$ and a morphism $f\colon B \to A$ to the functor $f^*\colon \mathcal{C}_{/A} \to \mathcal{C}_{/B}$ computing the pullback along f. This is not functorial but only pseudofunctorial because in general we only have $P(fg) \cong P(g) \circ P(f)$ by the universal property of pullbacks.

See [Jac99, Prop. 1.2.2] for a proof that when $\mathcal{C} = \mathbf{Set}$ this is equivalent (in a suitable sense) to Example 1.2.0.4.

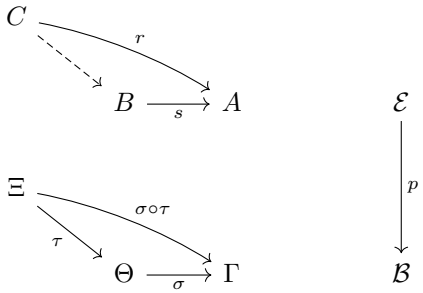
Example 1.2.0.6 (Restricted slice category)**.** Of course one can restrict the construction in Example 1.2.0.5 to any class of maps in a category $\mathcal{C}$, provided that a pullback exists and that such class is closed under taking it. A classical example is that of monomorphisms, which is strictly related to Example 1.7.1.5.

All of the above describe for each object a category, called the *fiber* over such object, and for each morphism a way to move from one category to the other. This is why pseudofunctors in **Cat** are often also called *indexed categories*. The key intuition in [Gro61] was that of turning this perspective around, so that one could collect all of the data of all of these categories into one *total* category, in such a way that one could always retrieve the original index of each fiber. This process is described in Section 1.5.

1.3 Main definitions

The result of the change in perspective described at the end of the previous section is that of a functor with the special property that one can move through fibers in a coherent way. Such a coherence is identified in the notion of *cartesian morphism*.

Definition 1.3.0.1 (Cartesian morphism)**.** Let $p\colon \mathcal{E} \to \mathcal{B}$ a functor and $s\colon B \to A$ a morphism in $\mathcal{E}$. We say that s is *p-cartesian* or *cartesian* over $\sigma\colon \Theta \to \Gamma$ if $p(s) = \sigma$ and for any other $r\colon C \to A$ and τ such that $p(r) = \sigma \circ \tau$ there is a unique $t\colon C \to B$ in $\mathcal{E}$ such that $p(t) = \tau$ and $s \circ t = r$.

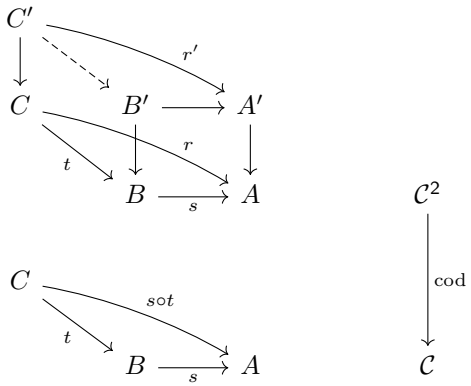


We say that s is a *(p-)cartesian lifting* of σ.

Definition 1.3.0.2 (Fibration)**.** A functor $p\colon \mathcal{E} \to \mathcal{B}$ is a *fibration* if for all A in $\mathcal{E}$, each $\sigma : \Theta \to pA$ has a cartesian lifting. We also say that $\mathcal{E}$ is *fibered over* $\mathcal{B}$ or that $\mathcal{E}$ is *over* $\mathcal{B}$. Oftentimes $\mathcal{B}$ is called the *base category* and $\mathcal{E}$ the *total category* of p.

Example 1.3.0.3 (The fundamental fibration)**.** One of the main examples of fibration is that underlying Example 1.2.0.5. Consider a category $\mathcal{C}$ and the codomain functor $\mathrm{cod}\colon \mathcal{C}^2 \to \mathcal{C}$: the universal property of cartesian liftings becomes that of pullbacks, so that a square over a map in $\mathcal{C}$ is cod-cartesian iff

it is a pullback. In fact, the very term *cartesian* is inspired by this example.



More generally, it is possible to show that

- cod is an opfibration (1.8.0.2), and
- cod is a fibration iff $\mathcal{C}$ has pullbacks.

Dual results hold for the domain functor dom.

Example 1.3.0.4 (The simple slice, [Jac99, 1.3])**.** Another slice-like fibration is the so called *simple slice*, which is of much interest to the study of simple type theory, and will be thoroughly discussed in Chapter 3.

Let $\mathcal{B}$ be a category with products, and consider the category $s(\mathcal{B})$ whose

- objects are pairs (I, X) of objects in $\mathcal{B}$;
- arrows $(J, Y) \to (I, X)$ are pairs (u, f) of maps in $\mathcal{B}$, with $u\colon J \to I$ and $f\colon J \times Y \to X$.

Composition can be defined using the universal property of products, meaning that the composite of maps

$$(K, Z) \overset{(v,g)}{\to} (J, Y) \overset{(u,f)}{\to} (I, X)$$

is defined as the pair $(u \circ v, f \circ \langle g, v\mathrm{pr}_1 \rangle)$, and identities are of the form $(\mathrm{id}, \mathrm{pr}_2)$. The obvious projection functor $s(\mathcal{B}) \to \mathcal{B}$ sending

$$(I, X) \mapsto I \quad \text{and} \quad (u, f) \mapsto u$$

is a fibration. Its fibers are also denoted $s(\mathcal{B})_I = \mathcal{B}_{/\!/ I}$, they have objects X in $\mathcal{B}$ and maps $X' \to X$ are $f\colon I \times X' \to X$ in $\mathcal{B}$: one can think of these as I-indexed families $f_i\colon X' \to X$ with fixed domain and codomain. Perhaps a comparison makes the action more evident, so recall that in the case of Fam from Example 1.2.0.4 these are $f_i\colon X_i' \to X_i$. If the definition of vertical maps makes the reader think of a co-Kleisli construction, they are right.

Lemma 1.3.0.5 (Uniqueness up to iso of the cartesian lifting)**.** Two cartesian liftings of a $\sigma\colon \Theta \to pA$ are isomorphic in $\mathcal{E}_{/A}$.

If cartesian maps allow to move from fiber to fiber, it is natural to give a name to maps that do the opposite.

Definition 1.3.0.6 (Vertical morphism, fibers)**.** A morphism $f\colon A' \to A$ such that $p(f) = \mathrm{id}$ is called *vertical*. For Γ in $\mathcal{B}$ we write $\mathcal{E}_\Gamma$ for the subcategory of $\mathcal{E}$ of objects and vertical maps over Γ: this is the *fiber over* Γ.

We might also denote $\mathcal{E}_\sigma$ the subcategory of $\mathcal{E}$ of objects and maps that are over σ in $\mathcal{B}$.

Many properties that one usually lists of both cartesian and vertical maps, such as that of being classes that are closed under composition, are contained in the following result.

Proposition 1.3.0.7 (Vertical/cartesian factorization system, [Mye20])**.** Consider $p\colon \mathcal{E} \to \mathcal{B}$ a fibration. The classes of vertical and cartesian morphisms form a orthogonal factorization system on $\mathcal{E}$. It additionally has the following properties:

1. if g and gf are vertical, then so is f;
2. pullbacks of vertical maps along cartesian ones exist and are vertical.

Remark 1.3.0.8 (Fibrations in any 2-category)**.** We shall notice that many of the constructions here are heavily **Cat**-based, but could be reproduced in a different 2-category (or bicategory), see for example [Str80] and others. In the present work, we only use the term *fibration* to mean a Grothendieck fibration, *i.e.* a strict fibration in **Cat**.

1.3.1 On the fibers

One can produce a classification for fibrations based on the structure of their fibers. The simplest example is that of fibrations where fibers are just sets.

Definition 1.3.1.1 (Discrete fibration)**.** A fibration p is said to be *discrete* if all vertical maps are identities.

Remark 1.3.1.2 (Uniqueness of cartesian lifting)**.** In a discrete fibration, cartesian liftings are unique: in fact, they are generally unique up to vertical isomorphism (see 1.3.0.5), and isomorphisms can only be identities in the discrete case.

Remark 1.3.1.3 (Cartesian wrt to discrete)**.** With respect to a discrete fibration, all morphisms are cartesian. This follows, from example, from Proposition 1.3.0.7.

One can see that a discrete fibration ends up having on each context not any category, but a category that is a set (*modulo* size issues). While we will discuss more on this in Proposition 1.5.0.4, we care to stress that this has proven to be crucial in describing models for dependent types, as for a given context there is traditionally supposed to be only a *set* (again, *modulo* size issues) of type-judgements in that context. We will come back to this in Chapter 3.

Additionally, discrete fibrations fit into an orthogonal factorization system on **Cat** with an interesting logical interpretation.

Theorem 1.3.1.4 (Comprehensive factorization)**.** Every functor has an orthogonal factorization into a final functor followed by a discrete fibration.

Sketch of the proof. For a functor $F\colon \mathcal{C} \to \mathcal{D}$ define $\mathcal{E}(F)$ to be the category having for objects pairs $(D, [g])$ with $g\colon FC \to D$ and $[g]$ representing the connective component of (g, C, D) in $(F/\mathrm{id}_{\mathcal{D}})$. One can define a functor $\mathcal{C} \to \mathcal{E}(F)$ mapping C to $(FC, [\mathrm{id}_{FC}])$ and show that this is final, also the forgetful $\mathcal{E}(F) \to \mathcal{D}$ is a fibration and is clearly discrete. See [SW73] for the full proof and more context. □

The reader might have noticed the familiar construction of $\mathcal{E}(F)$ in the proof above: we will talk about it extensively in Section 1.5. Unfortunately, being discrete is not a categorical notion, in the sense that its not a property that is preserved under equivalence of categories, but we can identify other properties of the fibers that end up being both interesting on their own and leading to the next best notion of discreteness.

Definition 1.3.1.5 (Groupoidal, faithful fibration)**.** A fibration p is said to be

- *groupoidal* if all vertical arrows are isomorphisms;
- *faithful* if it is a faithful functor;
- *essentially discrete* if it is both faithful and groupoidal.

Remark 1.3.1.6 (Essential discreteness)**.** A fibration p is equivalent to a discrete one iff it is essentially discrete. A fibration p is essentially discrete iff p is faithful and reflects isomorphisms. See [Str22, Sec. 9].

1.3.2 On cartesian liftings

Recall from Lemma 1.3.0.5 that the cartesian lifting of a given map is unique only up to isomorphism. This poses the question of what happens when one considers a particular *choice* of liftings.

Definition 1.3.2.1 (Cleavage)**.** A *cleavage* S for a fibration $p\colon \mathcal{E} \to \mathcal{B}$ is a choice for each A in $\mathcal{E}$ and $\sigma\colon \Theta \to pA$ of a cartesian lifting of σ at A. We denote it $s_{A,\sigma}\colon S(A,\sigma) \to A$. A cleavage induces for each $\sigma : \Theta \to \Gamma$ in $\mathcal{B}$ a functor $\mathcal{E}_\Gamma \to \mathcal{E}_\Theta$ by cartesianness of $s_{A,\sigma}$.

$$\begin{array}{ccc} S(A',\sigma) & \xrightarrow{s_{A',\sigma}} & A' \\ \vdots & & \downarrow f \\ S(A,\sigma) & \xrightarrow[s_{A,\sigma}]{} & A \end{array}$$

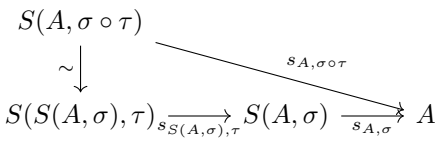
$$\Theta \xrightarrow[\sigma]{} \Gamma$$

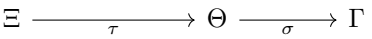
We call this *reindexing functor* and denote it with $S(-,\sigma)$ or σ^*, as it is often done in the literature.

Though this process seems functorial, it is not always the case, as one would hope, that for composable σ and τ

$$S(-, \sigma \circ \tau) = S(-,\tau) \circ S(-,\sigma),$$

though one can show that they are canonically isomorphic.





One can see such a behavior in Example 1.2.0.5 and in many other examples that occur naturally in the mathematical practice. Similarly, one does not necessarily have $S(-, \mathrm{id}) = \mathrm{id}$, but only a canonical isomorphism.

Notation 1.3.2.2 (Cartesian liftings)**.** If a cleavage S is not specified, we write $\overline{\sigma}\colon A[\sigma] \to A$ or $\overline{\sigma}\colon \sigma^* A \to A$ or $\overline{\sigma}\colon A\sigma \to A$ for a cartesian lifting of σ at A.

Definition 1.3.2.3 (Split fibration)**.** A cleavage S for a fibration p is called *split* or *splitting* if

1. $s_{A,\mathrm{id}_\Gamma} = \mathrm{id}_A$;
2. $s_{A,\sigma\circ\tau} = s_{A,\sigma} \circ s_{S(A,\sigma),\tau}$.

A fibration is called *split* if it is endowed with a split cleavage.

As it will be hopefully clear from the rest of this work – starting from Section 1.7 – we are interested in using fibrations because they can reasonably model the relation that is entertained between types/formulae and their contexts. In particular, the process of computing the cartesian lifting of a given map at a given type/formula amounts to computing substitution, hence split fibrations will be of main importance because they interpret a logical system in which computing substitution with (along) two suitably composable terms (maps) amounts to computing it with the resulting composite term (map). This is sometimes referred to as having substitution *on-the-nose*.

1.3.3 Characterizing fibrations

The geometric inspiration of fibrations is perhaps evident in the following result.

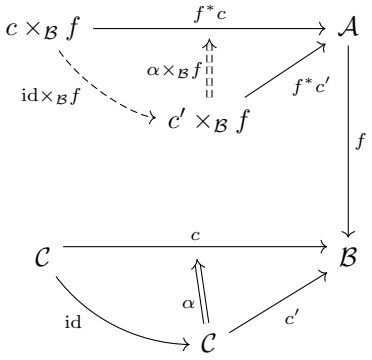
Theorem 1.3.3.1 ([Gra66, Theorem 2.10])**.** The following are equivalent for a functor $p\colon \mathcal{E} \to \mathcal{B}$:

1. p is a fibration;
2. for each A in $\mathcal{E}$, $p\colon \mathcal{E}_{/A} \to \mathcal{B}_{/pA}$ has a right adjoint right inverse.

Intuition. The right adjoint performs the action of picking a cartesian lifting. □

In the style of [Gra66] we can provide a new characterization of fibrations by means of lifting of 2-cells. We will make abundant use of it in Chapter 2.

Definition 1.3.3.2 (♯-lifting)**.** Consider a functor $f\colon \mathcal{A} \to \mathcal{B}$ and a 2-cell $\alpha\colon c' \Rightarrow c\colon \mathcal{C} \to \mathcal{B}$, and compute the pullback of c and c' along f. A *sharp lifting* or ♯*-lifting* of α along f is a pair $(\mathrm{id} \times_{\mathcal{B}} f, \alpha \times_{\mathcal{B}} f)$ of a functor and a natural transformation as below,



so that all "vertical" squares commute.

Theorem 1.3.3.3 (Characterizing fibrations via ♯-lifting)**.** The following are equivalent for a functor $p\colon \mathcal{E} \to \mathcal{B}$:

1. p is a fibration with a cleavage S;
2. each 2-cell $\alpha\colon c' \Rightarrow c\colon \mathcal{C} \to \mathcal{B}$ admits a terminal ♯-lifting along p, meaning that provided another (g, β) ♯-lifting of α along p, we have a unique vertical $\overline{\beta}$ such that $\beta = \alpha \times_{\mathcal{B}} p * \overline{\beta}$.

Proof. If p is a fibration with cleavage S, we can define the functor

$$\mathrm{id} \times_{\mathcal{B}} p\colon c \times_{\mathcal{B}} p \to c' \times_{\mathcal{B}} p, \quad (X, A) \mapsto (X, S(A, \alpha_X))$$

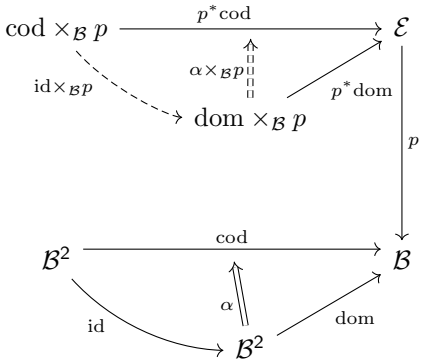
reindexing A along $\alpha_X : c'X \to cX = pA$. All desired squares commute. Moreover, we have a natural transformation $\alpha \times_{\mathcal{B}} p\colon p^* c' \circ \mathrm{id} \times_{\mathcal{B}} p \Rightarrow p^* c$ that on components is defined as follows

$$(\alpha \times_{\mathcal{B}} p)_{(X,A)} = s_{A,\alpha_X}\,.$$

In particular, since each s_{A,α_X} is cartesian we have that the pair $(\mathrm{id} \times_{\mathcal{B}} p, \alpha \times_{\mathcal{B}} p)$ enjoys the desired universal property: the induced unique vertical arrows assemble into the necessary $\overline{\beta}$.

Conversely, let p a functor and consider the trivial 2-cell dom $\Rightarrow$ cod, then

there exists a terminal sharp lifting as below,



therefore id $\times_{\mathcal{B}} p$ maps a pair $(A, \sigma\colon \Theta \to pA)$ to a pair $(B, \sigma\colon \Theta \to pA)$ with $pB = \Theta$, and there is a morphism

$$(\alpha \times_{\mathcal{B}} p)_{(A,\sigma)}\colon B \to A$$

in $\mathcal{E}$ over σ. We denote $(\mathrm{id} \times_{\mathcal{B}} p)(A, \sigma) = S(A, \sigma)$ and $(\alpha \times_{\mathcal{B}} p)_{(A,\sigma)} = s_{A,\sigma}$. It is cartesian because any other map over $\sigma\colon \Theta \to pA$ is part of another $\pitchfork$-lifting (g, β) of α along p and since $(\mathrm{id} \times_{\mathcal{B}} p, \alpha \times_{\mathcal{B}} p)$ is terminal with respect to this property, the unique induced $\overline{\beta}$ produces a suitable unique vertical map into $S(A, \sigma)$. □

1.4 The 2-category of fibrations

We can gather fibrations into a 2-category.

Definition 1.4.0.1 (The 2-category of fibrations)**.** Call **Fib** the 2-category having

- for 0-cells fibrations;
- for 1-cells *strict fibration morphisms* $p \to p'$ *i.e.* pairs of functors (E, B) making the following square commute

$$\begin{array}{ccc} \mathcal{E} & \xrightarrow{E} & \mathcal{E}' \\ {\scriptstyle p}\downarrow & & \downarrow{\scriptstyle p'} \\ \mathcal{B} & \xrightarrow[B]{} & \mathcal{B}' \end{array}$$

  and such that if a map s is p-cartesian over σ, then Es is p'-cartesian over $B\sigma$;
- for 2-cells $(E_1, B_1) \to (E_2, B_2)$ pairs of natural transformations (ϕ, ψ) with $\phi\colon E_1 \Rightarrow E_2$ and $\psi\colon B_1 \Rightarrow B_2$ such that $p' * \phi = \psi * p$.

Most of the times one is particularly interested in categories over a fixed base $\mathcal{B}$, for instance in Theorem 1.5.0.2. Specifying the definition above, one gets the following.

Definition 1.4.0.2 (The 2-category of fibrations over $\mathcal{B}$)**.** Call $\mathbf{Fib}(\mathcal{B})$ the 2-category having

- for 0-cells fibrations over $\mathcal{B}$;
- for 1-cells functors such that $p'E = p$ and preserving cartesian maps, also called *cartesian functors*;
- for 2-cells natural transformations with vertical components.

Definition 1.4.0.3 (Cartesian functors)**.** More generally, for a lax-commutative square

$$\begin{array}{ccc} \mathcal{E} & \xrightarrow{E} & \mathcal{E}' \\ {\scriptstyle p}\downarrow & \swarrow^{\alpha} & \downarrow{\scriptstyle p'} \\ \mathcal{B} & \xrightarrow[B]{} & \mathcal{B}' \end{array}$$

involving fibrations p, p' and functors E, B, we will say that E is *cartesian* if for each p-cartesian map f in $\mathcal{E}$ (over some σ), $E(f)$ is p'-cartesian (over $B(\sigma)$). This will come in handy in Chapter 3.

Lemma 1.4.0.4 (Properties of cartesian functors, [Jac93, Lemma 2.5])**.** Let $p\colon \mathcal{E} \to \mathcal{B}$ and $q\colon \mathcal{D} \to \mathcal{B}$ be two fibrations, and let $F\colon \mathcal{E} \to \mathcal{D}$ be a cartesian functor. Then:

1. for each Γ in $\mathcal{B}$, F induces a $F|_\Gamma\colon \mathcal{E}_\Gamma \to \mathcal{D}_\Gamma$, and
$$F \text{ is full (faithful)} \quad \text{iff} \quad \text{every } F|_\Gamma \text{ is full (faithful)};$$
2. if F is fully faithful, then
$$s \text{ is } p\text{-cartesian} \quad \text{iff} \quad Fs \text{ is } q\text{-cartesian}.$$

Let us now call $\mathbf{Cat}^2$ the 2-category having for objects functors, with $\mathbf{Cat}^2(F, G)$ the category of pairs of functors (H_0, H_1), (K_0, K_1) and of natural transformations (α_0, α_1) as below,

$$\begin{array}{ccc} \mathcal{C} & \overset{H_0}{\underset{K_0}{\Downarrow \alpha_0}} & \mathcal{C}' \\ {\scriptstyle F}\downarrow & & \downarrow{\scriptstyle G} \\ \mathcal{D} & \overset{H_1}{\underset{K_1}{\Downarrow \alpha_1}} & \mathcal{D}' \end{array}$$

with $GH_0 = H_1F$, $GK_0 = K_1F$ and $G * \alpha_0 = \alpha_1 * F$.

One can quickly see that **Fib** is a 2-full subcategory of $\mathbf{Cat}^2$. Not only that, but the composition of such inclusion with $\mathrm{cod}\colon \mathbf{Cat}^2 \to \mathbf{Cat}$ yields a fibration itself, with fibers precisely $\mathbf{Fib}(\mathcal{B})$'s.

Proposition 1.4.0.5 (The fibration of fibrations, [Jac93, Proposition 2.6])**.** The functor $\mathbf{Fib} \to \mathbf{Cat}$ sending each fibration to its base is itself a fibration. Moreover $\mathbf{Fib}_{\mathcal{B}} = \mathbf{Fib}(\mathcal{B})$.

Sketch of the proof. Consider a fibration $p\colon \mathcal{E} \to \mathcal{B}$ and a functor $H\colon \mathcal{C} \to \mathcal{B}$. The pullback of p along H in **Cat** produces a functor which is itself a fibration, and the resulting square is a strict fibration morphism. □

One could of course consider the 2-subcategories of all fibrations that are discrete, groupoidal, faithful. In the case of split ones, we additionally ask that cartesian functors preserve the cleavage, hence the respective 2-subcategory is not 1-full. We will come back to this with details when needed.

1.5 Fibrations and pseudofunctors

We have been saying that a fibration induces for each object of the base a category, namely the fiber over such object, and that (provided a cleavage) for each morphism a functor between such fibers. A whole lot of functoriality is happening, and this Section details how, expressing a correspondence between fibrations and pseudofunctors, sometimes also called *indexed categories*.

Remark 1.5.0.1 (On the existence of a cleavage)**.** The results that follow assume either one of the two:

1. to each fibration it is possible to associate at least one cleavage: this heavily depends on the meta-theory one is working in;
2. each fibration comes equipped with a choice of cleavage: this implies slightly changing the definition of fibration, and relaxing the definition of morphism of fibrations.

One could argue that the definition of fibration is made in order for Theorem 1.5.0.2 to hold, so we ask the reader to commit to either one of the two possibilities. More on this and the existence of cleavages, especially split ones, will follow in Section 1.6.

Theorem 1.5.0.2 (Grothendieck construction, [Gro61])**.** There exists a 2-equivalence

$$\mathbf{Fib}(\mathcal{B}) \cong \mathbf{Psd}[\mathcal{B}^{\mathsf{op}}, \mathbf{Cat}]$$

between the 2-category of fibrations (with base $\mathcal{B}$), cartesian functors, and natural transformations, and that of contravariant pseudofunctors (from $\mathcal{B}$) in the sense of 1.2.0.1, pseudonatural transformations, and modifications. Here **Cat** stands for the 1-category of categories and functors.

Sketch of the proof. Provided a fibration $\mathcal{E} \to \mathcal{B}$, we can define a pseudofunctor $\mathcal{B}^{\mathsf{op}} \rightsquigarrow \mathbf{Cat}$ mapping each Γ in $\mathcal{B}$ to its fiber category, and each $\Theta \to \Gamma$ to the functor $\mathcal{E}_\Gamma \to \mathcal{E}_\Theta$ described in Section 1.3.2. The universal property of cartesian liftings provides functoriality only up to isomorphism.

Conversely, a pseudofunctor $F\colon \mathcal{B}^{\mathsf{op}} \rightsquigarrow \mathbf{Cat}$ gives rise to a fibration $p\colon \int F \to \mathcal{B}$, with $\int F$ the category where

- objects are pairs (Γ, A) with Γ in $\mathcal{B}$ and A in $F(\Gamma)$,
- arrows $(\Theta, B) \to (\Gamma, A)$ are pairs (σ, s) with $\sigma\colon \Theta \to \Gamma$ in $\mathcal{B}$ and $s\colon B \to F(\sigma)(A)$ in $F(B)$,

and p is the projection on the first component. Such a construction, called *the Grothendieck construction for* F, can also be seen as the result of a lax-colimit of the functor along a certain universal map.

Cartesian functors are in a 1-to-1 correspondence with natural transformations between the relative pseudofunctors, since for a given context we have a functor between the respective fibers if and only if the functor between the total categories is cartesian.

One can read more on this proof in [Jac99, Section 1.10] or, for an account following the original work of Grothendieck, [Vis04, Section 3.1]. □

Remark 1.5.0.3 (Cartesian maps for a Grothendieck fibration)**.** With respect to $p\colon \int F \to \mathcal{B}$, cartesian maps are precisely those (σ, s) where s is an isomorphism.

Of course, additional properties of fibrations translate to properties of the corresponding pseudofunctors, and one could almost trivially restrict the 2-equivalence above to the respective 2-subcategories.

Proposition 1.5.0.4 (Restricting the equivalence)**.** Let p a fibration and F its correspondent under Theorem 1.5.0.2. We have that:

- if p is discrete then F is a functor $\mathcal{B}^{\mathsf{op}} \to \mathbf{Set}$;
- if p is faithful then F is a functor $\mathcal{B}^{\mathsf{op}} \to \mathbf{Pos}$;
- if p is groupoidal then F is a pseudofunctor $\mathcal{B}^{\mathsf{op}} \rightsquigarrow \mathbf{Gpd}$;
- p is split iff F is a functor.

1.6 Split fibrations and the fibered Yoneda lemma

As we have mentioned, when one is interested in logic, split fibrations bear a special interest because they interpret substitution on-the-nose. As discussed in Remark 1.5.0.1, one can reasonably hope to find a cleavage for each fibration, and maybe it is possible to achieve that such a cleavage picks identities on identities, but this is not the case for composition, as would be instead required by Definition 1.3.2.3, and splittings can behave in unexpected ways with respect to their transpositions to pseudofunctors. This discussion closely follows [Str22, Section 3]. The main results are due to J. Bénabou and J. Giraud.

Remark 1.6.0.1 (Counterexamples in fibrations)**.**

1. Not all fibrations admit a choice of a split cleavage. Consider for example the (categories corresponding to) groups $\mathcal{B} = (\mathbb{Z}_2, +_2)$ and $\mathcal{E} = (\mathbb{Z}, +)$, and the fibration $p\colon a \mapsto a(mod\, 2)$. If a splitting for p existed, it would produce a functor $s\colon \mathcal{B} \to \mathcal{E}$ such that $p \circ s = \mathrm{Id}$, but there is no group homomorphism $h\colon (\mathbb{Z}_2, +_2) \to (\mathbb{Z}, +)$ with $h(1)$ odd.

2. Different splittings of the same fibration may give rise to the same **Cat**-valued functor. For example, consider $H\colon \mathbf{2}^{\mathsf{op}} \to \mathbf{Ab}$ mapping 1 to the zero group and 0 to some non-trivial group A. It follows that every $a \in A$ induces a different splitting of $\int H$ simply by picking a to be the choice of lifting of $0 \to 1$, but these all produce the same functor $\mathbf{2}^{\mathsf{op}} \to \mathbf{Ab}$, namely H.

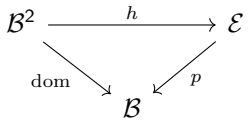
Nevertheless, one can always produce a split fibration out of a given one. In fact, one can do it in *two* canonical ways: consider $\mathbf{Sp}(\mathcal{B})$ the 2-subcategory of $\mathbf{Fib}(\mathcal{B})$ of split fibrations, cartesian functors preserving cleavages, meaning for a $F\colon p \to p'$ that

$$F(S(A,\sigma)) = S'(FA,\sigma),$$

and natural transformations, then there are both a left and a right 2-adjoint to the 2-forgetful $U\colon \mathbf{Sp}(\mathcal{B}) \to \mathbf{Fib}(\mathcal{B})$.

1.6.1 Right adjoint splitting

Remark 1.6.1.1 (Intuition)**.** The domain fibration $\underline{\Gamma} = \mathrm{dom}\colon \mathcal{B}_{/\Gamma} \to \mathcal{B}$ is the discrete fibration associated to the representable presheaf $\yoneda_\Gamma = \mathcal{B}(-,\Gamma)$. Additionally, to each $\sigma\colon \Theta \to \Gamma$ we can match a cartesian functor $\underline{\sigma}\colon \underline{\Theta} \to \underline{\Gamma}$ which acts as postcomposition with σ: this corresponds to Yoneda's presheaf morphism $\yoneda_\Theta \to \yoneda_\Gamma$. Functors h as



coherently pick objects of the form $A(\sigma) \to A$ out of (and over) a σ in $\mathcal{B}^2$, so that when one asks for h to be cartesian, one gets a split cleavage.

Theorem 1.6.1.2 (Fibered Yoneda lemma)**.** The 2-functor forgetting the splitting $U\colon \mathbf{Sp}(\mathcal{B}) \to \mathbf{Fib}(\mathcal{B})$ has a right 2-adjoint. Moreover, the corresponding counit is an equivalence.

Sketch of proof. We can define a 2-functor $S\colon \mathbf{Fib}(\mathcal{B}) \to \mathbf{Sp}(\mathcal{B})$ that on objects acts as follows: consider a fibration $p\colon \mathcal{E} \to \mathcal{B}$ and define the following

$$Spr(p)\colon \mathcal{B}^{\mathsf{op}} \to \mathbf{Cat}, \quad \Gamma \mapsto \mathbf{Fib}(\underline{\Gamma}, p),$$

where $\underline{\Gamma}$ is $\mathrm{dom}\colon \mathcal{B}_{/\Gamma} \to \mathcal{B}$. Since this is strictly functorial, the Grothendieck construction provides a split fibration.

The counit $US(p) \to p$ is obtained by mapping a pair $(\Gamma, h\colon \underline{\Gamma} \to p)$ to $h(\mathrm{id}_\Gamma)$. One can show that this in fact satisfies the appropriate universal property, and that it is fully faithful. If one assumes the axiom of choice for classes, then it is possible to show that it is also surjective on objects and that is therefore an equivalence of categories. □

Corollary 1.6.1.3 (Canonical equivalence)**.** Every fibration is equivalent to a distinguished split one.

Remark 1.6.1.4 (On the meta-theory)**.** Theorem 1.6.1.2 uses the axiom of choice for classes to prove that the counit is an equivalence, but that $U \dashv S$ is independent from it.

1.6.2 Left adjoint splitting

Theorem 1.6.2.1 (Left adjoint splitting)**.** The 2-forgetful functor U has a left 2-adjoint. Moreover, the corresponding unit is an equivalence.

Sketch of the proof. One can define a functor $L\colon \mathbf{Fib}(\mathcal{B}) \to \mathbf{Sp}(\mathcal{B})$ acting on objects as follows. Let $p\colon \mathcal{E} \to \mathcal{B}$ a fibration with a cleavage (recall Remark 1.5.0.1): without loss of generality, we can pick it in such a way that it satisfies

$$s_{A,\mathrm{id}} = \mathrm{id}_A,$$

if not, we can just switch the original one with one that picks the desired map at the identities. A cleavage with this property is sometimes said to be *normalized*. Out of this we may construct a functor $Spl(p)\colon \mathcal{B}^{\mathsf{op}} \to \mathbf{Cat}$, which in turn gives rise to a split fibration, $L(p)$. We define $Spl(p)(\Gamma)$ to be the category with objects pairs (σ, A), where A is in $\mathcal{E}$ and $\sigma\colon \Gamma \to pA$, and morphisms $(\tau, B) \to (\sigma, A)$ are vertical maps as below,

$$\begin{array}{ccc} B & \longleftarrow & S(B,\tau) \\ & & \downarrow \\ & & S(A,\sigma) \longrightarrow A \end{array}$$

$$pB \xleftarrow[\tau]{} \Gamma \xrightarrow[\sigma]{} pA$$

meaning that $Spl(p)(\Gamma)\,(\,(\tau, B), (\sigma, A)\,) = \mathcal{E}_\Gamma\,(\,S(B,\tau), S(A,\sigma)\,)$. With a bit of care, one can define $Spl(p)$ on base morphisms, too, repeatedly using cartesianness of maps, and show that it is strictly functorial by construction.

The unit $p \to UL(p)$ maps A to (id_{pA}, A) and satisfies the desired universal property. Since our starting cleavage is normalized, it is possible to show that this produces an equivalence of fibrations. □

Remark 1.6.2.2 (On the meta-theory)**.** In this case, it is possible to avoid choosing a cleavage: one can define morphisms between the fibers of $Spl(p)$ to be certain equivalence classes, and see that for each (normalized) cleavage on p there exists precisely one representative of that in said equivalence class. See [Str22, p. 14] for more details.

Remark 1.6.2.3 (It takes a 2-category)**.** Both in Theorem 1.6.1.2 and in Theorem 1.6.2.1, the 2-categorical aspect of **Fib** is fundamental: a fibration is not isomorphic but *equivalent* to its split counterpart.

Both splittings have found interesting uses in logic, and have different interpretations. We will come back to some in Chapter 3.

1.7 Faithful fibrations and the theory of doctrines

A special mention should go to faithful fibrations, as they represent one of the stepping stones in the process of using fibrations, which originally had a geometric flavor, to do logic.

In a series of seminal papers, F.W. Lawvere [Law69, Law70] introduces the notion of *hyperdoctrine* following the intuition that logical languages and theories can be encoded into indexed categories, and that the study of their 2-categorical properties can be both meaningful and fruitful. One striking example is that of connectives and quantifiers, which are determined by adjunctions between

certain fibers. We here revisit some results and definitions from the literature in increasing order of complexity.

For the rest of this section, we assume that $\mathcal{B}$ has finite products, for we want to be able to produce (non dependent) contexts juxtaposing typing statements. We write **Pos** for the category of partial orders and monotone functions.

1.7.1 Primary

Definition 1.7.1.1 (Primary doctrine, [MR13])**.** A *primary doctrine* is a functor $P\colon \mathcal{B}^{\mathsf{op}} \to \mathbf{Pos}$ which factorizes via a functor $\mathcal{B}^{\mathsf{op}} \to \mathbf{InfSL}$, the category of inf-semilattices, *i.e.* such that

1. for every Γ in $\mathcal{B}$, the partial order $P(\Gamma)$ has infinite infima, and
2. for every $\sigma\colon \Theta \to \Gamma$ in $\mathcal{B}$, the monotone map $P(\sigma) = P_\sigma$ preserves them.

A primary doctrine contains the structure needed to describe a many-sorted logic with conjunction and a constant for "true".

Example 1.7.1.2 (The algebra of formulae for a theory)**.** The prominent example of a primary doctrine is that induced by the Lindenbaum-Tarski algebra of a given first-order theory $\mathcal{T}$ in a language $\mathcal{L}$. We take **ctx** to be the category where

- objects are lists of distinct variables $x = (x_1, \dots, x_n)$,
- arrows are lists of substitution for variables, meaning $[t_1/y_1, \dots, t_m/y_m] = [t/y]\colon x \to y$, with t_j's being $\mathcal{L}$-terms that are built on variables $x_1, \dots, x_n$,

and composition is defined by simultaneous substitution. The product of two lists x and y is a list w with length the sum of the lengths of x and y, and projections on x and y are substitutions with, respectively, the first n and the last m variables in w. Categorically and in the sense of [Law63], this is the free Lawvere theory on the language $\mathcal{L}$.

One can define the functor $LT_{\mathcal{T}}\colon \mathbf{ctx}^{\mathsf{op}} \to \mathbf{InfSL}$ so that to each list x, the category $LT_{\mathcal{T}}(x)$ has for objects equivalence classes of well-formed formulae in $\mathcal{L}$ with free variables at most those that are in x, and with respect to the equivalence relation induced by reciprocal deducibility in $\mathcal{T}$, $\phi \dashv\vdash_{\mathcal{T}} \phi'$. Notice that this makes our treatment *proof-irrelevant*. Maps in $LT_{\mathcal{T}}(x)$ are provable consequences in $\mathcal{T}$. Composition is given by the cut rule of the calculus, and identities are tautologies. Since substitution preserves provability, $LT_{\mathcal{T}}$ can be suitably extended to a functor.

Such a construction was first introduced in [Law70] and is thoroughly explained in [KR77] and [MR13].

Example 1.7.1.3 (The powerset doctrine)**.** Provided a suitable meta-theory, one can define the doctrine $S\colon \mathbf{Set}^{\mathsf{op}} \to \mathbf{InfSL}$ that

- to each set A assigns its poset of subsets, where morphisms are given by inclusions,
- to each function $f\colon B \to A$ assigns a functor between the fibers that acts as the inverse image, $S_f = f^{-1}$.

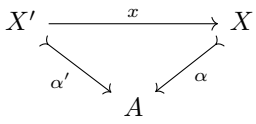
Example 1.7.1.4 (The powerobject doctrine)**.** Example 1.7.1.3 extends to any elementary topos $\mathcal{E}$. Let Ω be its subobject classifier: when computing $\mathcal{E}(-,\Omega)$ one gets a functor

$$\mathcal{E}(-,\Omega)\colon \mathcal{E}^{\mathsf{op}} \to \mathbf{HeytAlg}$$

with values in Heyting algebras and their morphisms. This captures subobjects, see for example [MLM94, Prop. I.3.1].

Example 1.7.1.5 (The subobject doctrine)**.** Actually, a result of the kind of Example 1.7.1.4 holds with much weaker hypotheses: consider a category $\mathcal{C}$ with finite products and pullbacks, and define $S\colon \mathcal{C}^{\mathsf{op}} \to \mathbf{InfSL}$ mapping each object A to the poset $S(A)$ whose

- objects are (equivalence classes of) subobjects $\alpha\colon X \rightarrowtail A$ in $\mathcal{C}$, and
- $\alpha' \leq \alpha$ iff there is a commutative diagram



  for a necessarily unique $x\colon X' \to X$.

Example 1.7.1.6 (The weak subobject doctrine, [MR13, Example 2.9])**.** Example 1.7.1.5 can be generalized to an even weaker context. Consider a category $\mathcal{C}$ with finite products and weak pullbacks, and define the functor $\Psi\colon \mathcal{C}^{\mathsf{op}} \to \mathbf{InfSL}$ given by the poset reflection of each comma category $\mathcal{C}_{/A}$ with A varying in $\mathcal{C}$. Weak (chosen) pullbacks provide the desired reindexing functors.

Doctrines are not only interesting as objects, but also when one consider morphisms between them: in particular, morphisms from $LT_{\mathcal{T}}$ (Example 1.7.1.2) to S (Example 1.7.1.3) have a special interpretation.

Lemma 1.7.1.7 (Models)**.** There is a 1-to-1 correspondence between classical models of $\mathcal{T}$ and indexed-categories morphisms $LT_{\mathcal{T}} \to S$ preserving the structure of a primary doctrine.

We believe that Lemma 1.7.1.7 testifies to the clarity that the theory of doctrines can bring to the study of logic. On top of a primary doctrine, we can put different structure in order to expand our expressive power.

1.7.2 Elementary

Definition 1.7.2.1 (Elementary doctrine, [MR13])**.** A primary doctrine $P\colon \mathcal{B}^{\mathsf{op}} \to \mathbf{InfSL}$ is *elementary* if for every Γ in $\mathcal{B}$ there is an object δ_Γ in $P(\Gamma \times \Gamma)$ such that for all Θ the functor

$$\begin{aligned} Æ_{\Theta,\Gamma}\colon P(\Theta \times \Gamma) &\longrightarrow P(\Theta \times (\Gamma \times \Gamma)) \\ A &\mapsto P_{\langle \mathrm{pr}_1, \mathrm{pr}_2\rangle}(A) \wedge P_{\langle \mathrm{pr}_2, \mathrm{pr}_3\rangle}(\delta_\Gamma) \end{aligned}$$

is left adjoint to $P_{\langle \mathrm{pr}_1, \mathrm{pr}_2, \mathrm{pr}_2\rangle}$.

Remark 1.7.2.2 (Intuition)**.** To clarify Definition 1.7.2.1 we unfold it in the case of Example 1.7.1.2, and get the assignment

$$x, y \vdash A(y, x) \quad \rightsquigarrow \quad x, x', y \vdash A(y, x) \wedge \delta(x, x'),$$

so that when we ask that this is left adjoint to $P_{\langle \mathrm{pr}_1, \mathrm{pr}_2, \mathrm{pr}_2 \rangle}$, we get

$$x, x', y;\ A(y, x) \wedge \delta(x, x') \vdash B(y, x, x') \quad \text{iff} \quad x, y;\ A(y, x) \vdash B(y, x, x).$$

It is clear that this amounts to asking (at each context) the existence of an *equality-like* predicate.

Studying equality in this algebraic guise allows for interesting findings about its "nature" to appear. Notably, it was shown in [EPR20] that the procedure of freely adding equality predicates is comonadic, and that such a result can be applied to the elimination of imaginaries in models of first-order theories.

1.7.3 Existential, universal

Definition 1.7.3.1 (Existential doctrine, [Law69])**.** A primary doctrine $P\colon \mathcal{B}^{\mathrm{op}} \to \mathbf{InfSL}$ is *existential* if for any Γ_1 and Γ_2 in $\mathcal{B}$ and for any $\mathrm{pr}_i\colon \Gamma_1 \times \Gamma_2 \to \Gamma_i$, the reindexing P_{pr_i} has a left adjoint $\exists_{\mathrm{pr}_i}$, this is natural in Γ_i, and together they satisfy

1. the *Beck-Chevalley condition*, meaning that for any pullback of a projection pr along any map σ

$$\begin{array}{ccc} \Theta' & \xrightarrow{\mathrm{pr}'} & \Gamma' \\ {\scriptstyle \sigma'}\downarrow & \lrcorner & \downarrow{\scriptstyle \sigma} \\ \Theta & \xrightarrow[\mathrm{pr}]{} & \Gamma \end{array}$$

   and for any A in $P(\Theta)$, the canonical arrow $\exists_{\mathrm{pr}'} P_{\sigma'}(A) \leq P_\sigma \exists_{\mathrm{pr}}(A)$ in $P(\Gamma')$ is an iso;

2. *Frobenius reciprocity*, meaning that for $\mathrm{pr}\colon \Theta \to \Gamma$, A in $P(\Gamma)$, and B in $P(\Theta)$, the canonical arrow $\exists_{\mathrm{pr}}(P_{\mathrm{pr}}(A) \wedge B) \leq A \wedge \exists_{\mathrm{pr}}(B)$ is an iso.

Remark 1.7.3.2 (Intuition)**.** Again, we turn to Example 1.7.1.2 for a better understanding of the definition. The existential $\exists_1$ acts as

$$x, y \vdash A(x, y) \quad \rightsquigarrow \quad x \vdash \exists y. A(x, y)$$

then adjointness provides

$$x;\ \exists y. A(x, y) \vdash B(x) \quad \text{iff} \quad x, y;\ A(x, y) \vdash B(x).$$

Beck-Chevalley guarantees that the existential commutes with substitution, so that for $A(x, y)$

$$\exists y. (A[t/x]) \dashv\vdash (\exists y. A)[t/x],$$

while Frobenius dictates the following

$$\exists y. (A(x) \wedge B(x, y)) \dashv\vdash A(x) \wedge \exists y. B(x, y).$$

Similarly, one can give an account of the universal quantifier.

Definition 1.7.3.3 (Universal doctrine, [MR13])**.** A primary doctrine $P\colon \mathcal{B}^{\mathsf{op}} \to \mathbf{InfSL}$ is *universal* if for any Γ_1 and Γ_2 in $\mathcal{B}$ and for any $\mathrm{pr}_i\colon \Gamma_1 \times \Gamma_2 \to \Gamma_i$, the reindexing P_{pr_i} has a right adjoint $\forall_{\mathrm{pr}_i}$, this is natural in Γ_i, and together they satisfy

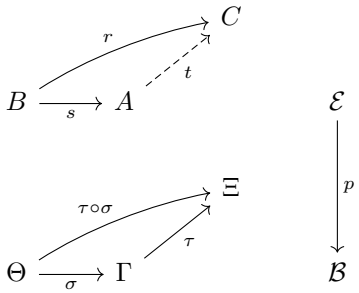
1. the Beck-Chevalley condition;
2. Frobenius reciprocity.

Many other properties and structures can be translated into the language of doctrines, and doctrines with such property or structure can be assembled into suitable 2-categories. Their study provides many insights on the nature of said properties and structures. The reader might be interested in [Pit99, MR13, MR15, MPR17, EPR20, Tro20], and can find how these are related to cartesian bicategories in [BSSS21].

1.8 Opfibrations, bifibrations

As one might expect, it is possible to recalibrate the discussion in Section 1.3, Section 1.4, and Section 1.5 with covariant **Cat**-valued pseudofunctors, liftings with a given domain, and so on.

Definition 1.8.0.1 (Opcartesian morphism)**.** Let $p\colon \mathcal{E} \to \mathcal{B}$ be a functor and $s\colon B \to A$ a morphism in $\mathcal{E}$. We say that s is *p-opcartesian* or *opcartesian* over $\sigma\colon \Theta \to \Gamma$ if it is cartesian over σ for $p\colon \mathcal{E}^{\mathsf{op}} \to \mathcal{B}^{\mathsf{op}}$, meaning that $p(s) = \sigma$ and for any other $r\colon B \to C$ and τ such that $p(r) = \tau \circ \sigma$ there is a unique $t\colon A \to C$ in $\mathcal{E}$ such that $t \circ s = r$.



We say that s is a *(p-)opcartesian lifting* of σ.

Definition 1.8.0.2 (Opfibration)**.** A functor $p\colon \mathcal{E} \to \mathcal{B}$ is an *opfibration* if for all B in $\mathcal{E}$ we have that each $\sigma : pB \to \Gamma$ has a opcartesian lifting.

Theorem 1.8.0.3 (Grothendieck construction)**.**

$$\mathbf{OpFib}(\mathcal{B}) \cong \mathbf{Psd}[\mathcal{B}, \mathbf{Cat}]$$

Definition 1.8.0.4 (Bifibration)**.** A functor $p\colon \mathcal{E} \to \mathcal{B}$ is an *bifibration* if it is both a fibration and an opfibration.

Lemma 1.8.0.5 (Characterizing bifibrations, [Jac99, 9.1.2])**.** A fibration is a bifibration if and only if each reindexing $S(-, \sigma)$ has a left adjoint $\coprod_\sigma$.

Proof. Let $p\colon \mathcal{E} \to \mathcal{B}$ a fibration and $\sigma\colon \Theta \to \Gamma$ a map in $\mathcal{B}$, then for each B in $\mathcal{E}_\Theta$ and A in $\mathcal{E}_\Gamma$ we have

$$\mathcal{E}_\Gamma(\coprod_\sigma B, A) \overset{(1)}{\cong} \mathcal{E}_\Theta(B, S(A,\sigma)) \overset{(2)}{\cong} \mathcal{E}_\sigma(B, A) \overset{(3)}{\cong} \mathcal{E}_\Gamma(\coprod_\sigma B, A)$$

describing the following

$$\begin{array}{ccc} B & & \coprod_\sigma B \\ \downarrow & & \downarrow \\ S(A,\sigma) & \xrightarrow[s_{A,\sigma}]{} & A \end{array}$$

$$\Theta \xrightarrow[\sigma]{} \Gamma$$

where (2) follows from cartesianness of $s_{A,\sigma}$. Then

$$\coprod_\sigma \text{exists} \quad \text{iff} \quad (1) \text{ exists} \quad \text{iff} \quad (3) \text{ exists} \quad \text{iff} \quad p \text{ is an opfibration.}$$

□

Chapter 2

Categorized judgemental theories

> Everything that can be thought at all can be thought clearly.
> Everything that can be said can be said clearly.
>
> [Wit22, 4.116]

This chapter is concerned with the notions of *context, judgement* and *deduction*. These three notions belong to Logic, but different communities with different backgrounds and cultures have quite different perspectives on them. The purpose of this work is to present a mathematical and unified approach that accommodates these diverse takes on the topic of *deduction*. The effort required in order to do so turns out to be extremely fruitful, so much so that it contributes a fresh perspective on deduction itself. In order to show how tightly this new framework captures the motions of deductive systems, we develop two applications of the theory we introduce, and since our choice is intended to pay tribute to two major cultures concerned with the topic, we look at examples from type theory [MS84] and from proof theory [TS00]. They each stand on different *conceptual* grounds, as it is exemplified by the two following rules.

$$\text{(DTy)}\ \frac{\Gamma \vdash a : A \qquad \Gamma.A \vdash B\ \texttt{Type}}{\Gamma \vdash B[a]\ \texttt{Type}} \qquad \text{(Cut)}\ \frac{x;\Gamma \vdash \phi \qquad x;\Gamma,\phi \vdash \psi}{x;\Gamma \vdash \psi}$$

Despite their incredibly similar look, and the somehow parallel development of the theories in the same notational framework, there are some philosophical differences between the interpretation of the symbols above.

TT) In type theories, especially those inspired by the reflections of Martin-Löf, $\Gamma \vdash B\ \texttt{Type}$ is intuitively seen as a *judgement*. A judgement is an act of knowledge [Mar96a, Mar87] bound to a context (Γ) and pertinent to an object (B). For example, $\Gamma \vdash B\ \texttt{Type}$ could be read *Given* Γ, B *is a type*. The ontological status of a context and an object is, in principle, very different. Also, and most notably, judgements can be of different kinds, claiming all sorts of possible things about their objects [Mar96a].

ND) In natural deduction, $x;\Gamma \vdash \psi$ is intuitively seen as a *consecution*. A consecution is a relation between *structured formulae* $(x;\Gamma)$ and *formulae*

(ψ) [Kle67]. For example, the sequent $x;\Gamma \vdash \psi$ could be read *The multiset of formulae* Γ*, in the variables* x*, entails* ψ. Besides the fact that structured formulae are multisets of formulae, there isn't an ontological difference between the glyphs appearing on the left and right side of the entailment.

These differences, though admittedly subtle and not that easy to detect on a technical level, dictate a part of the experts' intuition on the topics (see Section 2.2.1). Of course, one could argue that these different points of view are mostly philosophical, that the oversimplification commanded by the length of this introduction stresses on them in a somewhat artificial way, and that some variations are allowed, for example [NvP08] adopts what we would call a more type theoretic perspective on proof theory, and indeed it is always possible to adopt a judgemental perspective on consecutions. In particular, the deep connection between proof theory and type theory has of course been studied for a while, and its development falls under the paradigm that is mostly known as *propositions-as-types* [Wad15], and how to translate intuitionistic natural deduction into the language of types is beautifully described in [Mar96b]. This work aims at providing a new, perhaps more semantic, argument in the same unifying direction.

Rebooting some ideas from [Jac99], we conciliate the differences in a unified categorical framework that can highlight and clarify in a more precise way the meaning of all these apparently specific phenomena. Going back to the example of (DTy) and (Cut), we intuitively see how they both fit the same paradigm, in the sense that we could read both as instances of the following syntactic string of symbols

$$(\triangle)\quad \frac{\heartsuit \vdash \blacksquare \qquad \square \vdash \clubsuit}{\heartsuit \vdash \spadesuit}$$

which we usually parse as: *by* $\triangle$, *given* $\heartsuit \vdash \blacksquare$ *and* $\square \vdash \clubsuit$ *we deduce* $\heartsuit \vdash \spadesuit$. Our theory allows for a coherent expression of *all* such strings of symbols, and shows how a suitable choice of *context* either produces (DTy) or (Cut). As a necessary biproduct of our effort, we get a theory that has both the advantage of being very versatile, spanning much farther than dependent types and natural deduction, and computationally meaningful in the sense that it has a built-in notion of computation. We also believe it establishes a clear philosophical view on context, judgement, and deduction.

Our contribution

We introduce the notion of categorized judgemental theory using the language of category theory. Judgemental theories are philosophically inspired by Martin-Löf's reflections on the topic of judgement [Mar87, Mar96a], and technically grounded on the most recent developments in the categorical treatment of dependent type theory [Awo18, Uem19]. We believe that our perspective is also very attuned to *type refinement systems* as described in [MZ15].

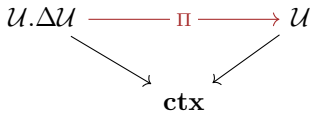
$$\Gamma \vdash H \;\; \mathcal{H} \qquad \lambda H \vdash F \;\; \mathcal{F}$$

Usually, when looking at the premises of a rule, we are confronted with a list of (nested) judgements as above, which then are transformed into another judgement by the rule. The technical advantage of our notion is to allow natively for *nested*

judgements. That is, for us a nested family of judgements is actually a whole judgement per se:

$$\Gamma \vdash H.\lambda F \;\; \mathcal{H}.\lambda \mathcal{F}\,.$$

This flexibility allows for a very algebraic treatment of type constructors (and of connectives), in a fashion that is somewhat inspired by Awodey's natural models. Judgment classifiers will be categories living over contexts, and functors between them will regulate deduction rules. In the example below, which is the formation rule for the Π-constructor in dependent type theory, the category $\mathcal{U}.\Delta\mathcal{U}$ classifies the nested judgement in the premise of the rule (on the right).



$$(\Pi\mathrm{F}) \quad \frac{\Gamma \vdash A\ \mathtt{Type} \qquad \Gamma.A \vdash B\ \mathtt{Type}}{\Gamma \vdash \Pi_A B\ \mathtt{Type}}$$

We expand the original approach à la Jacobs, where some of these ideas were evidently hinted at both in the treatment of propositional logic [Jac99, Chapter 2], and in the treatment of type theories [Jac99, Chapter 10]. It also expands Awodey's natural models [Awo18], taking very seriously his algebraic presentation of some constructors: see for example the discussion at page 9 and later Prop. 2.4 in loc. cit.

It should be noted that Logical Frameworks [HHP93] of Plotkin et al. have a similar purpose, and their system is based on $\lambda\Pi$-calculus. Logical Frameworks do rely on the notion of judgement in a substantial way, as we do, but their approach is somewhat much more syntactic. More recently [Uem19] has provided recipes to transform Logical Frameworks into more categorical gadgets, based on a generalization of Awodey's natural models. One of the advantages of our approach is to avoid the complexity of Logical Frameworks (and thus of Uemura's recipe), substituting it with a native categorical language.

In the next subsection we will discuss in detail all the achievements of this structure on a technical level, though we end this very qualitative discussion with a bird's-eye view on a list of advantages of our system.

1. We provide a very algebraic approach to the notion of rule, which is also suitable for a neat analysis of the proof theory associated to a deductive system.

2. In the case of type theory, this provides a clear definition of (dependent) type constructor, which was actually not available before, if not in a case-by-case form. We put into perspective the usual paradigm of rules (formation, introduction, elimination, β- and η-computation).

3. In a similar spirit, we provide an in depth analysis of what constitutes what in proof theory are called structural rules. We here see that branches in proof trees are cones in our framework.

4. We introduce the notion of *policy* for a categorized judgemental theory, inspired by the classical Cut of the Gentzen calculus. Surprisingly, type dependency in dependent type theory is precisely a type theoretic form of Cut.

5. Our proofs are computationally meaningful. In a sense, this is due to the structural rigidity and the algebraicity of the framework. Each of our proofs needs to be as atomized and as transparent as possible, this will be particularly evident in our analysis of the proof theory generated by a dependent type theory with Π-types. The whole framework feels like a categorical proof assistant when doing proofs.

While this introduction seems to focus mainly on dependent type theories and natural deduction, the reader will notice that we have only chosen these two specific frameworks as an *exemplum* of the expressive power of this theory. Indeed we could have covered modal logic, infinitary logics and much more.

2.1 Categorized judgemental theories

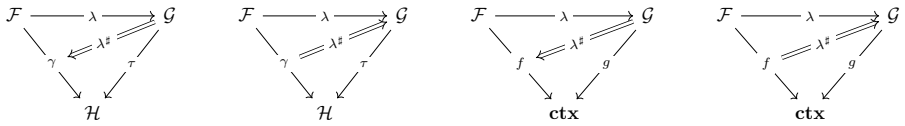
Definition 2.1.0.1 (Categorized pre-judgemental theory)**.** A *categorized pre-judgemental theory* (**ctx**, $\mathscr{J}$, $\mathscr{R}$, $\mathscr{P}$) of (*contexts*, *judgements*, *rules*, *policies*) is specified by the following data:

(**ctx**) a category (with terminal object $\diamond$);

($\mathscr{J}$) a set of functors $f : \mathcal{F} \to \mathbf{ctx}$ over the category of contexts;

($\mathscr{R}$) a set of functors $\lambda : \mathcal{F} \to \mathcal{G}$.

($\mathscr{P}$) a set of 2-dimensional cells filling (some) triangles induced by the rules (functors in $\mathscr{R}$) and the judgements (functors in $\mathscr{J}$), as in the diagrams below.



Notation 2.1.0.2. Let us introduce a bit of terminology:

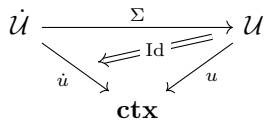
- *contexts* Γ, Θ are objects of **ctx**, morphisms $\sigma : \Theta \to \Gamma$ are *substitutions*;
- the element $g : \mathcal{G} \to \mathbf{ctx}$ of $\mathscr{J}$ is the *classifier of the judgement* $\mathcal{G}$. We will often blur the distinction between the classifier and its judgement. In general we use letters such as $\mathcal{F}, \mathcal{G}, \mathcal{H}$;
- objects G in $\mathcal{G}$ are usually named after corresponding letter;
- a *rule* λ is an element of $\mathscr{R}$;
- a *policy* $\lambda^\sharp$ is an element of $\mathscr{P}$;

and for special categorized judgemental theories that happen to have an established notation we declare a switch of notation in the appropriate section.

This is all the syntactic data needed to describe deduction: judgement classifiers prescribe the status of objects with respect to contexts; rules transform objects into other objects, with the context changing accordingly; and policies allow for the possibility that the context of the premise of the rule and that of the

consequent are somehow naturally related, either covariantly (i.e. $\lambda^\sharp : f \Rightarrow g\lambda$), contravariantly (i.e. $\lambda^\sharp : g\lambda \Rightarrow f$), or constantly (i.e. when the triangle is strictly commutative) with respect to the direction of the rule. A bird's eye view of this first definition and what it might have been can be found in Section 2.5.

Example 2.1.0.3 (Toy Martin-Löf type theory)**.** In order to get acquainted with the definition, let us introduce the categorical syntax to present a toy type theory. Consider a category **ctx** of contexts and substitutions, $\mathcal{U}$ a category (universe) of types and $\dot{\mathcal{U}}$ a category (universe) of terms. For simplicity, we imagine that a term is always registered together with its type, so that objects of $\dot{\mathcal{U}}$ are of the form (a, A) with A an object in $\mathcal{U}$. Define the categorized pre-judgemental theory having $\mathcal{J} = \{u, \dot{u}\}$, $\mathcal{R} = \{\Sigma\}$, $\mathcal{P} = \{\mathrm{Id} : u \circ \Sigma \Rightarrow \dot{u}\}$ as below.



Intuitively, $\dot{u}$ classifies terms with their context, u does the same for types, Σ performs typing, meaning it is the second projection, and Id shows that such an operation preserves the context.

We know that this all looks very unorthodox. We will use this toy example to get a first small impression on how categorized judgemental theories work, see 2.2.3.5, 2.2.4.1, and above all Section 2.3.

On the data expressed by a categorized pre-judgemental theory we wish to impress some deductive power. This is achieved using some 2-categorical constructions and properties.

Definition 2.1.0.4 (Categorized udgemental theory)**.** A *categorized judgemental theory* $(\mathbf{ctx}, \mathscr{J}, \mathscr{R}, \mathscr{P})$ is a categorized pre-judgemental theory such that

1. $\mathscr{R}$ and $\mathscr{P}$ are closed under composition;
2. the judgements are precisely those rules whose codomain is **ctx**;
3. $\mathscr{R}$ and $\mathscr{P}$ are closed under *finite limits* (see 2.1.0.9, 2.1.0.10 and 2.1.0.11), $\sharp$-*liftings* (see 2.1.0.12) and *whiskering* (see 2.1.0.14).

For readability reasons, and when we believe the stressing on their categorical nature is unneeded, we sometimes omit the word "categorized" and refer to them simply as *judgemental theories*.

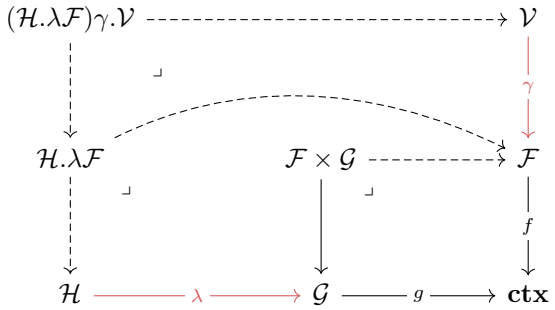
The rest of this section is dedicated to clarifying the technical aspects of this definition. In the next section we will see that these properties influence the inference power of our logical systems. The more we put, the more we infer.

Remark 2.1.0.5. The condition (2) in Definition 2.1.0.4 is actually not needed, yet it is not harmful for the theory and it allows a cleaner axiomatization of (3), which otherwise would not look as pretty.

Remark 2.1.0.6 (Infinitary judgemental theories)**.** We could have allowed λ-small limits for λ a (regular) cardinal, so that we are actually studying *finitary* judgemental theories. In the present work we stick to this choice.

Remark 2.1.0.7 (Economical presentations of categorized judgemental theories)**.** In the majority of concrete instances, a judgemental theory is presented by a pre-judgemental theory $(\mathbf{ctx}, \mathscr{J}, \mathscr{R}, \mathscr{P})$, in the sense that we close the data of judgements, rules and policies under finite limits and $\sharp$-liftings and whiskering. This produces the smallest judgemental theory containing $(\mathbf{ctx}, \mathscr{J}, \mathscr{R}, \mathscr{P})$.

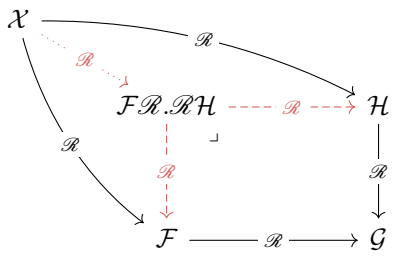
Notation 2.1.0.8. When a classifier $\mathcal{X}$ is obtained by iterated pullback of classifiers along rules, we try to use a notation that keeps in mind this special property of the classifier. Consider thus the diagram below.



- We use the notation $\mathcal{F}\times\mathcal{G}$ when we pullback classifiers along classifiers.
- When we pullback a classifier along a rule, we use the notation $\mathcal{H}.\lambda\mathcal{F}$. We can make sense of this as if we put an additional bound on $\mathcal{F}$, and this is induced from $\mathcal{H}$ via λ. The reader will find more about this in 2.2.3.2.
- When we iterate this procedure, for example as in the diagram, we use the notation $(\mathcal{H}.\lambda\mathcal{F})\gamma.\mathcal{V}$. When $g = f$ we write it $\mathcal{H}\gamma.\lambda\mathcal{V}$.

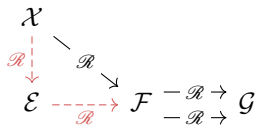
We are aware that this notation is not entirely economical, nor uniquely determined, but in the practical circumstances of this chapter, it will be very useful.

2.1.0.9 (Pullbacks)**.** $\mathscr{R}$ is closed under pullbacks in the sense that, given solid (black) spans and cones in $\mathscr{R}$ as below, we have that all the red arrows belong to $\mathscr{R}$.

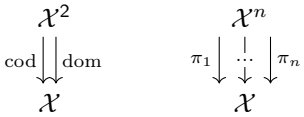


In this definition we see the advantage of including $\mathscr{J}$ in $\mathscr{R}$, otherwise we would have to specify another axiom for the case in which the span is made of judgements. This could have been done without major differences, but would lead to an incredible proliferation of diagrams.

2.1.0.10 (Equalizers)**.** Similarly to the case of pullbacks, we require that the equalizer $\mathcal{E}$, together with its limiting maps, belongs to the rules.



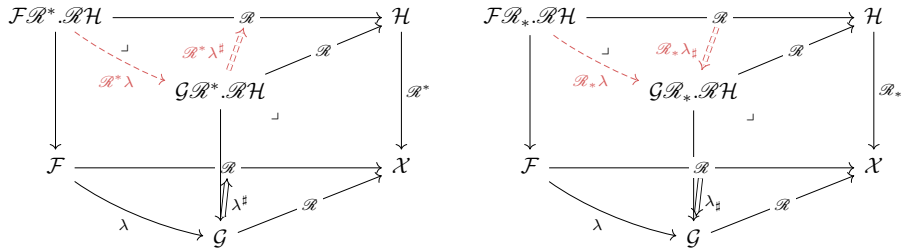
2.1.0.11 (Powers)**.** We also require that, for all rules $\mathcal{X} \to \mathcal{Y}$, we can form the finite powers below in $\mathscr{R}$ and that, as in 2.1.0.10 and 2.1.0.9, all the arrows induced by their universal properties by cones made of rules, are rules too.



Regarding our meta-theory, this only requires that the finite product of sets (or classes, or κ-sets for some inaccessible cardinal κ, depending on the meta-theory of choice) is again a set (or class, or κ-set). See Section 1.1.4 for more on the matter.

2.1.0.12 ($\sharp$-lifting)**.** First of all recall from Definition 1.3.3.2 the notion of $\sharp$-lifting. Notice that it is a construction involving a functor and a 2-cell, and is always possible when said functor is a fibration, so it makes no difference if the functor in the domain of the 2-cell is the composite of different functors or not: in fact, one could restate Theorem 1.3.3.3 for sharp liftings of 2-cells of the following kind. Notice that we slightly change the notation from 1.3.3.2 to isolate λ as below.

Consider a policy $\lambda^\sharp$ as in the diagram below, and a rule $\mathscr{R}^*$ which is a fibration. Then the pair $(\mathscr{R}^*\lambda, \mathscr{R}^*\lambda^\sharp)$ belong to $\mathscr{R}$ and $\mathscr{P}$ (respectively). Similarly, for $\mathscr{R}_*$ an opfibration, we get the op-diagram on the right. Notice that in both cases the square containing λ and $\mathscr{R}^*\lambda$ or $\mathscr{R}_*\lambda$ commutes strictly.



Proof of the existence of the arrows. We only argue for the case of fibrations and only show that a fibration has $\sharp$-liftings of this kind, the converse is proved precisely as in Theorem 1.3.3.3. Define $\mathscr{R}^*\lambda$ as follows: to each pair (F, H) match a $(\lambda F, S(H, \lambda^\sharp_F))$ obtained via the cartesian lifting of $\lambda^\sharp_F$. Such a definition can be extended to maps using naturality of $\lambda^\sharp$. The natural transformation $\mathscr{R}^*\lambda^\sharp$ at each (F, H) is just the cartesian map lifting $\lambda^\sharp_F$. □

Notation 2.1.0.13 (Substitution)**.** For the time being, and to avoid continuous explicit reference to a given cleavage for each fibration involved, we write $A[\sigma]$

for what was called $S(A, \sigma)$ up to this point. Implications of the existence of such a cleavage were discussed in Section 1.3.2.

2.1.0.14 (Whiskering)**.** As it is quite frequent in 2-category theory (see for instance Section 1.4), one might want to compose 1-cells with 2-cells. As our theory is quite heavily 2-dimensional, it only make sense that we ask that performing such an operation does not bring us out of our logic. We recall the general definition in the 2-category **Cat**, as it is the one we are interested in now. Consider categories, functors, and natural transformations as below.

$$\mathcal{A} \xrightarrow{F} \mathcal{B} \;\underset{G'}{\overset{G}{\Downarrow \alpha}}\; \mathcal{C} \xrightarrow{H} \mathcal{D}$$

One can always define natural transformations $\alpha * F \colon GF \Rightarrow G'F$ and $H * \alpha \colon HG \Rightarrow HG'$ that point-wise act as

$$(\alpha * F)_A = \alpha_{FA} \qquad (H * \alpha)_B = H(\alpha_B).$$

Given classifiers $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$, rules $\lambda, \lambda', \gamma$ and a policy $\lambda^\sharp$, then, we say that the categorized judgemental theory is closed under whiskering in the sense that the red natural transformations are policies too.

$$\mathcal{X} \;\underset{\lambda'}{\overset{\lambda}{\Downarrow \lambda^\sharp}}\; \mathcal{Y} \xrightarrow{\gamma} \mathcal{Z} \qquad\qquad \mathcal{X} \;\underset{\gamma\lambda'}{\overset{\gamma\lambda}{\Downarrow \gamma(\lambda^\sharp)}}\; \mathcal{Z}$$

$$\mathcal{Z} \xrightarrow{\gamma} \mathcal{X} \;\underset{\lambda'}{\overset{\lambda}{\Downarrow \lambda^\sharp}}\; \mathcal{Y} \qquad\qquad \mathcal{Z} \;\underset{\lambda'\gamma}{\overset{\lambda\gamma}{\Downarrow \lambda^\sharp_\gamma}}\; \mathcal{X}$$

Disclaimer 2.1.0.15. We understand that up to this point the reader has been faced with many concepts and strange notations that they have no intuition for. Therefore, before we formally describe what it means to define a calculus based on the blocks that are our categorized judgemental theories, we advise the reader to skip to Section 2.3.3.1 and see what it is that we are trying to achieve.

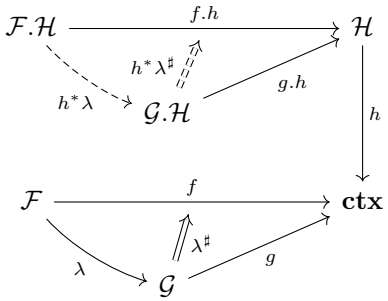
2.1.1 Notions of substitution

Definition 2.1.1.1. A judgement classifier is *(op)substitutional* if it is an (op)fibration. A rule is *(op)cartesian* if it preserves (op)cartesian maps. A policy is *(op)Frobenius* with respect to a given judgement classifier if it has cartesian components.

By extension, we will say that a categorized (pre)judgemental theory is *(op)substitutional* if all judgement classifiers are (op)substitutional, all rules are (op)cartesian, and all policies are (op)Frobenius.

See Section 2.2.6 for what these imply for categorized judgemental theories, for the moment we only prove a couple of technical results.

Lemma 2.1.1.2 (♯-lifting of cartesian functors)**.** Consider a fibration h and a 2-cell $\lambda^\sharp$ as follows, and apply the construction in 2.1.0.12.

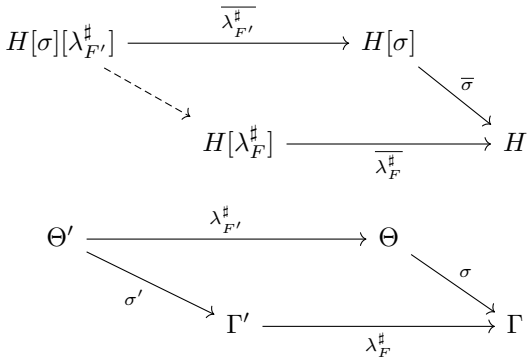


If λ preserves cartesian maps, then so does $h^*\lambda$.

Proof. Consider a morphism $a = (a_1, a_2) \colon (F', H') \to (F, H)$ in $\mathcal{F}.\mathcal{H}$, meaning a pair $a_1 \colon F' \to F$ in $\mathcal{F}$ and $a_2 \colon H' \to H$ in $\mathcal{H}$ such that $f(a_1) = \sigma = h(a_2)$. One can check (see, for example, [Jac93, Proposition 2.6]) that this is cartesian with respect to $h \circ (f.h)$ if and only if both a_1 is f-cartesian and a_2 is h-cartesian. The latter is equivalent to saying that a_2 is of the form $a_2 = \overline{\sigma} \colon H[\sigma] \to H$. Now consider that the functor $h^*\lambda$ acts as follows

$$(a_1, a_2) \colon (F', H') \to (F, H) \quad \mapsto \quad (\lambda a_1, a_2^!) \colon (\lambda F', H'[\lambda^\sharp_{F'}]) \to (\lambda F, H[\lambda^\sharp_F])$$

with $a_2^!$ the unique map induced by naturality of $\lambda^\sharp$ at $h(a_2)$. Assume that a is cartesian, then we end up having



therefore $a_2^! = \overline{\sigma}^!$ is itself cartesian. Hence if λ preserves cartesian maps, then so does $h^*\lambda$. □

Remark 2.1.1.3 (♯-lifting is cartesian)**.** The natural transformation $h^*\lambda^\sharp$ has h-cartesian components.

Proof. This is actually trivial by definition of $h^*\lambda^\sharp$: in fact, it acts as

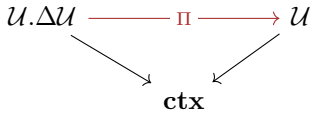
$$(h^*\lambda^\sharp)_{(F,H)} = \overline{\lambda^\sharp_F} \colon H[\lambda^\sharp] \to H\,.$$

□

2.2 Judgement calculi

In the previous section we have introduced categorized judgemental theories, very concrete mathematical objects for which we have presented a suggestive notation referencing some logical intuition. This section is devoted to grounding that intuition and showing that each categorized (pre)judgemental theory is a categorical version of a proof assistant or, more technically, something that supports the categorical semantics for the specification of a type system. We will see how a categorized judgemental theory automatically produces a deductive system via a process of translation. Actually, a judgemental theory is intrinsically a calculus of deduction in a very precise sense.

This section will describe a way to translate the data of a categorized judgemental theory $(\mathbf{ctx}, \mathscr{J}, \mathscr{R}, \mathscr{C})$ into a judgement calculus.



$$(\Pi\mathrm{F})\quad \frac{\Gamma \vdash A\ \mathtt{Type} \qquad \Gamma.A \vdash B\ \mathtt{Type}}{\Gamma \vdash \Pi_A B\ \mathtt{Type}}$$

Of course, such a process of translation requires an almost formal definition of judgement calculus, which must be flexible enough to encode the usual calculi that are used in type theory and in proof theory. For the reasons expressed in the introduction, this is a non-trivial task.

2.2.1 Prolegomena

As we have hinted in the introduction, a very general definition of deductive system or *calculus* is much easier to describe than to actually define. Of course, one can make reference to [Res02], or to [MS84], or to [HHP93], or to several variations of this notion, but there is no unified take we find satisfying. In this subsection we go through a critical analysis of the deductive systems of a dependent type theory and of a proof theory to better motivate the choices of the next subsection.

2.2.1.1 The deductive system of a DTT

We consider the problem of defining the semantics of the underlying signature, judgements and rules defining a formal calculus of a dependent type theory based on Martin-Löf's type theory. There are indeed several approaches in the literature, and the very notion of type theory is somehow (intentionally) fuzzy. We would go as far as to say that a complete agreement on the matter does not exist. Of course, this flexibility is part of the richness of this theory. The informality in the definition of rule and type constructor is one of the reasons for which the topic of (categorical) semantics for dependent type theory is both so popular and so useful in the theoretical research on dependent type theory.Most sources would probably agree that to declare (the calculus of) a dependent type theory means to specify three boxes of data.

(S) Syntax (contexts, types, terms): a theory of dependent types is -informally- a formal system dealing with *types* and *terms* in *context*. From a symbolic point of view, these are a bunch of glyphs that we use as atoms of our

language.

$$\Gamma \qquad A \qquad a : A$$

(J) Judgements (about contexts, types, terms): a judgement is a very simple sentence made up of symbols from the syntax, and whose intention is to somehow bound together pieces of atomic data. The most simple type theories of the sort we refer to present three possible (kinds of) judgements,

$$\vdash \Gamma\ \texttt{ctx} \qquad \Gamma \vdash A\ \texttt{Type} \qquad \Gamma \vdash a : A$$

which are informally interpreted as Γ *is a context*, A *is a type in context* Γ, a *is a term of type* A *in context* Γ.

(R) Rules (to declare new types, terms, and interact with the syntax): Finally, we should be able to interact with and declare a type. For those that are acquainted with a programming language, this need is completely evident. Indeed we might want to introduce a type which is constructed from other types.

$$\text{(DTy)}\ \frac{\Gamma \vdash a : A \qquad \Gamma.A \vdash B\ \texttt{Type}}{\Gamma \vdash B[a]\ \texttt{Type}}$$

Depending on the complexity of the theory, beyond a bunch of basic rules (like type and term dependency (Section 2.3.4)), we find type constructors. Type constructors are packages of rules, labelled by their feature, that allow to construct new types from old ones. Below we list the inescapable labels, for a constructor whose name is - say - Φ.

- ΦF Some *formation rule(s)*, presenting the type. They specify under which circumstances we can assume it to exist (or, from the point of view of programming languages, we can form it). By circumstances, we usually refer to syntactic data.
- ΦI Some *introduction rule(s)*, producing the canonic terms of a such type. Given a set of syntactic data, they tell how to cook up a term of the new type.
- ΦE Some *elimination rule(s)*, specifying the interaction between a term of the new type and the terms of the types that contributed to the formation of the new type.

Additionally, based on the computational semantics associated to the theory one wishes to consider, one needs to describe how introduction and elimination interact with one another, meaning to provide suitable *conversion rules*.

These three packages of data reflect the necessities of a type theory: indeed, type theory emerged as a foundational framework, but from a cultural point of view its history is intertwined with that of programming languages. This deep interaction has shaped several aspects of type theory, we will see this especially in the declarative and interactive nature of the Rules box. Let us stress on the fact that, besides these informal distinctions, there is no formal definition of a type constructor, nor of a rule.

While the three boxes of Syntax, Judgements, and Rules are definitely there in any type theory, the list of rules, judgments, and the sort of symbols that inhabit them is subject to major choices. Even those that we have listed can be seen as somewhat arbitrary. Still, we believe that in any reasonable type theory the data above will be included. In most of the concrete instances, type theories are even richer than what we have listed above:

- *morphisms of contexts* are usually added to Syntax;
- *definitional equality* is usually added to Judgements;
- *β- and η-computation* are almost always added to Rules, and through definitional equality they determine how introduction and elimination interact with one another. Also, we will see to that our type theory has *context formation*, which stands for a set of rules that form fresh contexts from existing types. Finally, if the syntax is enriched with morphisms of contexts, there might be rules regulating their interaction with judgements (that would be *substitution*).

A vast majority of computer scientists and type theorists would probably classify β- and η-computation as an essential feature of a type theory.

2.2.1.2 Natural deduction

As it was said in the introduction, natural deduction has already been shown to be fittingly translatable in the language of types [Mar96b], but we here describe its interpretation separately for multiple reasons:

1. on one hand, natural deduction for first-order logic has had a greater fortune in being studied and *employed* in our schools and universities, and it is the one that we believe is understood best among most people, therefore
2. we believe that its "familiarity" makes it easier for the reader to connect the categorical syntax for the intuition of what the rules should be, moreover
3. such a familiarity allows us the freedom to describe more rules, and the practice of such an encoding is the main aim of this first chapter.

The following presentation is mostly inspired by [Res02, Def 2.18], but it is coherent with the treatment of [NvP08] and [TS00] too.

When specifying a natural deduction calculus we provide three boxes of data:

(S) Syntax (variables, formulae): a natural deduction calculus is -informally- a formal system dealing with *variables*, *lists of formulae* and *formulae*. From a symbolic point of view, these are glyphs that will be the atom of our calculus,

$$x \qquad \Gamma \qquad \phi.$$

Often punctuation symbols as the semicolon *;* are used too to combine the symbols.

(S) Sequents: a sequent is a very simple sentence made up of symbols from the syntax, and whose intention is to specify an entailment relation between the data on the left and the data on the right of the entailment symbol.

$$x; \Gamma \vdash \phi.$$

For example, the sequent above could be read *the list of formulae in* Γ *entails the formula* ϕ*, and they all have (at most) free variables in* x*.*

(R) Rules: in natural deduction rules are used to state atomic consequences, they transform a family of sequents into a (family of) sequent(s).

$$\text{(Cut)} \quad \frac{x; \Gamma \vdash \phi \qquad x; \Gamma, \phi \vdash \psi}{x; \Gamma \vdash \psi}$$

Traditionally, there is a distinction between *structural rules* [Res02, 2.23] and *other rules*. Referring to [Res02, pag. 26], the structural rules influence what we can prove. The more structural rules you have, the more you will be able to prove. The other rules are more in the spirit of type constructors and they account for the behavior of the logical operators, like $\wedge, \vee, \forall, \exists$. In modal logics, they can account for the behavior of modal operators too.

It is pretty intuitive that we can treat a sequent as a form of judgement. This just amounts to a re-tuning of our intuition with respect to the way we are used to read sequents. On the other hand it is not entirely trivial to find a precise correspondence between the rules of proof theory and the constructors of type theory. For example, for the reason that there is not a precise definition of constructor, nor a classification of them.

2.2.1.3 Judgement calculi

Given the discussion above, our challenge is pretty clear: how to accommodate type constructors, connectives, and deduction rules in a conceptually unified and technically coherent framework? Provide that we can see sequents as judgements, how do we formally deal with their manipulation from a semantic point of view? Let us dive into the definitions. For us, to declare a judgement calculus means to specify three boxes of data:

(S) Syntax (contexts and objects);

(J) Judgements (acts of knowledge *bound* to a context and *pertaining to* a (list of) object(s));

(R) Rules (transforming judgements into other judgements).

2.2.2 Syntax

Let $(\mathbf{ctx}, \mathscr{J}, \mathscr{R}, \mathscr{C})$ be a (pre)judgmental theory. Then its corresponding judgement calculus has in its Syntax box letters for each context and each object in a judgement classifier.

Γ	$\Gamma \in \mathbf{ctx}$
F	$F \in \mathcal{F}$

2.2.3 Judgements

Judgemental calculi include two main kinds of judgement for each $\mathcal{F} \in \mathscr{J}$.

- The first kind of judgement acknowledges a $\mathcal{F}$-empirical evidence and clarifies the status of an object. One can see it as a kind of *Tarskian snow* for our approach, meaning something that fulfills Tarski's requirement for something to "characterize unambiguously the class of those words and expressions which are to be considered meaningful" [Tar56]. For $F \in f^{-1}(\Gamma)$, then we find in our set of judgement the writing

$$\Gamma \vdash F\ \mathcal{F}.$$

  This can be understood as *Given* Γ, F *exists* or *Given* Γ, G *is green*, or *Given* Γ, M *is made of marble*. In the case of the same category $\mathcal{F}$ appearing as the domain of two different judgements, we might use $\Gamma \vdash F\ \mathcal{F}(f)$.

- The second kind of judgement is an equality checker for the equality classified by the judgement. We will write

$$\Gamma \vdash F =_{\mathcal{F}} F',$$

  when $F, F' \in f^{-1}(\Gamma)$ and $F = F'$. This could be read as *Given* Γ, F *and* F' *are indistinguishable by existence*[1], or *Given* Γ, G *and* G' *are indistinguishable by green.* Notice that the interpretation of the notion of equality is relative to the choice of the classifier. If we were to look at something classifying types in a type theory, the equality would (and will, in Section 2.3) be indistinguishability up to computations.

In the table below, we find on the left column the judgement and on the right its translation in terms of the categorized judgemental theory.

$\Gamma_1 \vdash F_1\ \mathcal{F} \quad ... \quad \Gamma_n \vdash F_n\ \mathcal{F}$	$(F_1 ... F_n) \in f^{-1}(\Gamma_1) \times ... \times f^{-1}(\Gamma_n)$
$\Gamma \vdash F =_{\mathcal{F}} F'$	$F, F' \in f^{-1}(\Gamma)$ and $F = F'$

It might seem that such simple judgements do not guarantee much in terms of expressiveness. This is in fact far from the truth! Recall that a judgemental theory is closed under finite limits and several constructions, thus we obtain an incredible variety of complex judgements.

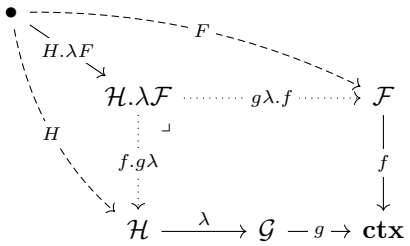
Remark 2.2.3.1 (On notions of equality and the relationship between theory and meta-theory)**.** Notice that all choices made here are to consider as "external", in the sense that they constitute the building blocks of our calculus. They do not prevent from having, say, an identity judgement *in* a judgemental theory, see for example Section 2.3.6.

2.2.3.2 (Nested judgements: pullbacks)**.** Let $(\mathbf{ctx}, \mathscr{J}, \mathscr{R}, \mathscr{C})$ be a categorized judgemental theory and consider a judgement of the form,

$$\Gamma \vdash H.\lambda F \quad \mathcal{H}.\lambda \mathcal{F}.$$

[1] This could be actually read *Are identical.*

We will see that such a judgement classifies a nested family of judgements, depending on the data of $\mathcal{H}, \mathcal{F}$ and λ.



By inspecting the pullback diagram that defines $\mathcal{H}.\lambda\mathcal{F}$ we can see that judgements of this form are in bijection with pairs of judgements of the form,

$$\Gamma \vdash H\ \mathcal{H} \qquad g\lambda H \vdash F\ \mathcal{F}.$$

This means that we are entitled to see the line above, which is composed of two related but separate judgements (in possibly different contexts!), as a single one (in context Γ). We call judgements of this form *nested*. Notice that, depending on what we want to express, we could say that the relation binding H to Γ (hence the judgement classifier with domain $\mathcal{H}$ in the line above) is either $g\lambda$ itself, therefore forcing both H and F to have the same context, or some other h. Still, we chose to present the pullback in its most general form.

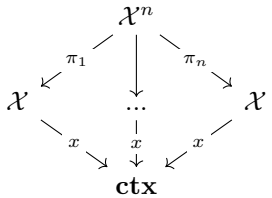
As a string of symbols, notice that a nested judgement is an informal judgement, in the sense that it is not well defined in our framework. Despite this, we will feel free to use notations as that above because they are a bit easier to parse from a human perspective. This means that for the rest of the chapter we will write

$$\cfrac{\Gamma \vdash H.\lambda F\ \mathcal{H}.\lambda\mathcal{F}}{\Gamma \vdash H\ \mathcal{H} \qquad g\lambda H \vdash F\ \mathcal{F}}$$

to intend that the synthetic judgment below is an *alias* for the nested judgement above, which is defined in our context.

Our notation ($\mathcal{H}.\lambda\mathcal{F}$) retains almost all the information needed to predict the kind of nested judgements we classify. This also explains why we write composed judgements the way we do: we think of the component in $\mathcal{H}$ to be free, while that in $\mathcal{F}$ is bounded via the map λ.

Example 2.2.3.3 (Lists)**.** Let $\mathcal{X}$ be a judgement classifier, and consider $\mathcal{X}^n$, which is given by the (wide) pullback below,



then the judgement $\Gamma \vdash X_1.\cdots.X_n\ \mathcal{X}^n$ can be interpreted as a list of judgements, as described below.

$$\frac{\Gamma \vdash X_1.\cdots.X_n\ \mathcal{X}^n}{\Gamma \vdash X_1\ \mathcal{X} \qquad \cdots \qquad \Gamma \vdash X_n\ \mathcal{X}}$$

Notice that the fact that a judgemental theory is by definition closed under finite products implies that these judgements are always available.

Example 2.2.3.4 (Composable arrows)**.** Let $\mathcal{C}$ be a category and consider the following pullback as on the left. The resulting nested judgement, then, reads as on the right and classifies composable arrows in $\mathcal{C}$.

$$\begin{array}{ccc} \mathcal{C}^2\mathrm{dom.cod}\mathcal{C}^2 & \dashrightarrow & \mathcal{C}^2 \\ \downarrow & \lrcorner & \downarrow{\scriptstyle \mathrm{dom}} \\ \mathcal{C}^2 & \xrightarrow[\mathrm{cod}]{} & \mathcal{C} \end{array} \qquad \frac{C \vdash (f,g)\ \mathcal{C}^2\mathrm{dom.cod}\mathcal{C}^2}{C \vdash f\ \mathcal{C}(\mathrm{cod}) \qquad C \vdash g\ \mathcal{C}(\mathrm{dom})}$$

The middle ground is given by the actual *middle* object f and g share. Notice that, though the context (i.e. the object) is the same, their bounds to it are very different (that is, respectively cod and dom).

Example 2.2.3.5 (Toy Martin-Löf type theory)**.** In the categorized judgemental theory generated by that in Example 2.1.0.3, we now have a way of coding, for example, pairs of types in the same context. This is achieved by the pullback $\mathcal{U}u.u\mathcal{U}$.

$$\begin{array}{ccc} \mathcal{U}u.u\mathcal{U} & \dashrightarrow & \mathcal{U} \\ \downarrow & \lrcorner & \downarrow{\scriptstyle u} \\ \mathcal{U} & \xrightarrow[u]{} & \mathbf{ctx} \end{array} \qquad \frac{\Gamma \vdash (A,A')\ \mathcal{U}u.u\mathcal{U}}{\Gamma \vdash A\ \mathcal{U} \qquad \Gamma \vdash A'\ \mathcal{U}}$$

The examples above are not particularly interesting, though we believe they give an intuition of the expressive power of nested judgements. We hope Section 2.3 will be definitive proof.

2.2.3.6 (Nested judgements: equalizers)**.** Similarly to the previous case, equalizers classify nested judgements of the kind below.

$$\mathcal{E}(\lambda,\lambda') \overset{e}{\dashrightarrow} \mathcal{F} \underset{\lambda'}{\overset{\lambda}{\rightrightarrows}} \mathcal{G} \qquad \frac{\Gamma\ \vdash F\ \mathcal{E}(\lambda,\lambda')}{\Gamma \vdash F\ \mathcal{F} \qquad \Gamma \vdash \lambda F =_{\mathcal{G}} \lambda' F}$$

2.2.4 Rules

Let $(\mathbf{ctx}, \mathscr{J}, \mathscr{R}, \mathscr{P})$ be a categorized judgemental theory. Consider judgements and a rule as in the diagram below, for each rule $\lambda : \mathcal{F} \to \mathcal{G}$ and each judgement $\Gamma \vdash F\ \mathcal{F}$, we will write as follows (on the right).

$$\begin{array}{ccc} \mathcal{F} & \xrightarrow{\lambda} & \mathcal{G} \\ \downarrow{\scriptstyle f} & & \downarrow{\scriptstyle g} \\ \mathbf{ctx} & & \mathbf{ctx} \end{array} \qquad (\lambda)\ \frac{\Gamma\ \vdash F\ \mathcal{F}}{g\lambda F \vdash \lambda F\ \mathcal{G}}$$

From a technical level, this is just a compact way to organize the data of the functoriality of λ. Indeed it is true that $\lambda F \in g^{-1}(g\lambda F)$, so that $g\lambda F \vdash \lambda F\ \mathcal{G}$ is actually a judgement in our framework. This is the only kind of rule that we

admit in our judgemental calculi, and in a sense all the rules are the same, there are no intrinsic labels like *structural, introduction, elimination*, and so on. Yet, similarly to the case of judgements, the closure under finite limits guarantees an incredible richness of rules, as we will see for the rest of the subsection.

Example 2.2.4.1 (Toy Martin-Löf type theory)**.** The rule Σ from Example 2.1.0.3 now reads as follows.

$$(\Sigma) \quad \frac{\Gamma \vdash (a, A)\, \dot{\mathcal{U}}}{u\Sigma(a, A) \vdash \Sigma(a, A)\, \mathcal{U}}$$

Now recall that $u\Sigma = \dot{u}$, so the behavior of a policy is implied: see Section 2.3 for more on this. We follow the intuition provided for all the data of the judgemental theory in Example 2.1.0.3 and translate it in the usual type-theoretic strings of symbols. Then it reads as follows

$$(\Sigma) \quad \frac{\Gamma \vdash a : A}{\Gamma \vdash A\ \mathtt{Type}}$$

and depicts the typing rule. We thoroughly detail this process of translation in Section 2.3.3.

Remark 2.2.4.2 (Rules with many outputs)**.** The notion of nested judgement 2.2.3.2 and of our calculus as a whole have one additional very useful feature, and that is allowing for multiple consequents *simultaneously*. In fact, it is very common that one might want to write rules that deduce several judgements from the same (set of) judgement(s), but writing it organically is somewhat frowned upon, so that one usually encounters a proliferation of rules (for example two elimination rules in Section 2.3.6 and in Section 2.4.4). While we mostly follow the tradition with regard to this, the attentive reader will see that in fact they are always the product of the "break-down" of a single nested judgement. We make this explicit once in Remark 2.3.4.6.

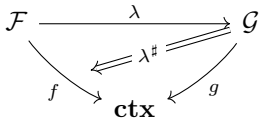
Remark 2.2.4.3 (Soundness and completeness)**.** When one looks at this section as a whole, that is the process of producing a graphical/grammatical bookkeeping of the categorical properties of the judgemental theory, organized in the form of a collection of judgements and deductions, it is natural to raise the question *which deductions are actually produced by a categorized judgemental theory?* There are two possible approaches to this question.

- The first approach is to refine our notion of calculus, and give a precise definition of what we mean by deductive system. Our presentation is not that far from a formalization. Then, one would say that the content of this section provides a kind of soundness/correctness theorem for the categorical syntax, and one could try to provide a completeness result that characterize all the possible judgements and deductions.
- The second approach is to claim that the question contains an implicit bias towards grammatical/syntactic representations of deductions, and that, in a sense, the categorical language already provides the grammar the reader is looking for, while the calculus in this section only represents a way to make it more digestible to the *grammarian*.

Both the approaches are valid, one maybe making a more political statement, and the other being more prone to the classical tradition. Because the author herself does not entirely agree on the path to follow, and because this chapter already contains a lot of material, we choose not to invest more on this question in the present work. We will be greatly interested in developing this more along the line.

2.2.5 Policies

Recall that a policy is a 2-cell as follows

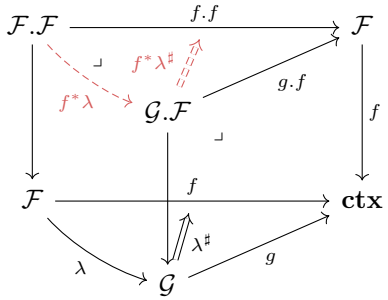


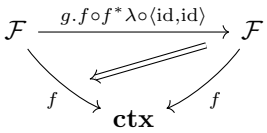
with f, g judgement classifiers and λ a rule. This additional data contains that of the rule λ, hence

$$(\lambda) \quad \frac{\Gamma \vdash F\ \mathcal{F}}{\Theta \vdash \lambda F\ \mathcal{G}}$$

but it also establishes a relation between $\Gamma = fF$ and $\Theta = g\lambda F$, namely $\lambda^\sharp_F : \Theta \to \Gamma$. We wish our judgemental calculus to reflect this.

If f is a fibration by 2.1.0.12 we can $\sharp$-lift $\lambda^\sharp$ along f,





and get a pair $(f^*\lambda, f^*\lambda^\sharp)$ such that on a pair of objects F, F' over the same context

$$f^*\lambda \colon (F, F') \mapsto (\lambda F, F'[\lambda^\sharp_F]),$$

hence we can use the universal property of pullbacks to precompose the policy "on top" with $\langle \mathrm{id}, \mathrm{id} \rangle$ to get the lax triangle on the right. This now reads as

$$(g.f \circ f^*\lambda \circ \langle \mathrm{id}, \mathrm{id} \rangle) \quad \frac{\Gamma\ \vdash F\ \mathcal{F}}{\Theta \vdash F[\lambda^\sharp_F]\ \mathcal{F}}$$

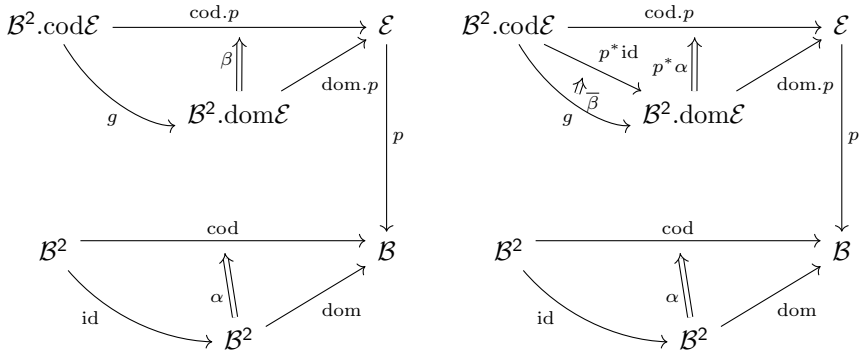
where $\lambda^\sharp_F : \Theta \to \Gamma$.

One could detail a similar argument for covariant policies: we do not do so here because in the present work we will only encounter contravariant ones. Still, the reader can easily see how covariant rules are strictly connected to comonads, and comonads have been proven of special interest in logic (see, for example, the treatment of equality in [DR21]), and this is why we have carried them through all definitions and technical proofs.

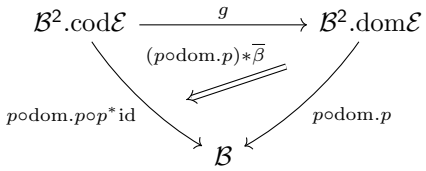
2.2.6 On substitution

Section 2.2.5 is a first instance of application of substitution-like properties in our setting, in it is worth noticing that the additional data of a policy can be only *externalized* in our setting when the target judgement classifier is a fibration. It seems worth it, then, to describe what information - from the point of view of categorized judgemental theories - lies under the assumption that a functor is a fibration. In particular, we break down the universal property described in Theorem 1.3.3.3.

Recall from Theorem 1.3.3.3 that for a fibration $p\colon \mathcal{E} \to \mathcal{B}$ there are $(p^*\mathrm{id}, p^*\alpha)$ such that for any other $\sharp$-lifting (g, β) there is a unique p-vertical $\overline{\beta}$ satisfying $\beta = p^*\alpha * \overline{\beta}$.



Hence the peculiarity of fibrations seems to lie in the existence of a unique vertical $\overline{\beta}$. We break down its meaning in the following policy, resulting from whiskering $\overline{\beta}$ with $p \circ \mathrm{dom}.p$,



and it is easy to see that $p \circ \mathrm{dom}.p \circ p^*\mathrm{id}$ is a fibration, therefore we can apply the discussion in Section 2.2.5 and derive the following rule in our judgemental theory. With $\sigma\colon \Theta \to pA$ and $B(A, \sigma) = (\mathrm{dom}.p \circ g)(A, \sigma)$, given that $(A, \sigma)[\overline{\beta_{A,\sigma}}] = B(A, \sigma)$,

$$\frac{\Theta \vdash (A, \sigma)\ \mathcal{B}^2.\mathrm{cod}\mathcal{E}}{\Theta \vdash (B(A, \sigma), \sigma)\ \mathcal{B}^2.\mathrm{cod}\mathcal{E}}$$

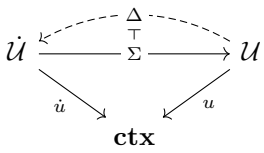
meaning that initiality translates to the fact that any substitution is derivable from the cartesian one.

2.3 Categorized dependent types theories

In this section we show what features must a categorized judgemental theory have in order to support dependent type theory. We show it produces desired rules, and with respect to this provide some evidence of the computational power

of judgemental theories. We then pin-point which rules one needs to add in order to gain typically desirable constructors, for example dependent products and identity. In doing so, we learn something about constructors in general and give a (unifying) definition of type constructor.

Definition 2.3.0.1 (Categorized dependent type theory)**.** A *categorized dependent type theory* is a substitutional categorized judgemental theory generated by the pre-judgemental theory described by the diagram below.



To be precise, we mean that $\mathscr{J} = \{\dot{u}, u\}$ and that those are fibrations, $\mathscr{R} = \{\Sigma, \Delta\}$, $\mathscr{P}$ contains the witness of the commutativity of the solid diagram, and both the unit and the counit (ϵ, η) of the adjunction $\Sigma \dashv \Delta$. Finally, we require that (ϵ, η) are cartesian natural transformations. We call this $cDTT$, for short.

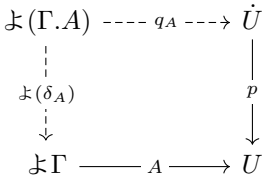
We think of $\mathcal{U}$ as classifying types, $\dot{\mathcal{U}}$ as classifying terms, and the functor Σ as the one performing the typing. Its adjoint Δ will interpret context extension. The choice of the greek letters Σ, Δ is inspired by the notation classically used for polynomials, for example in [GK13, p.7].

Remark 2.3.0.2 (Notational caveats)**.** As we mentioned in 2.1.0.8 our notation, while being very telling, sometimes hides pieces of data. For example one finds that $\dot{\mathcal{U}}.\Sigma\dot{\mathcal{U}} \cong \dot{\mathcal{U}} \times \dot{\mathcal{U}}$. This is an instance of the fact that, depending on the choice of maps along which one performs the pullbacks (and depending on the order in which one does so), one gets a classifier that is either nested, or it is not. In general, the *nesting degree* is subject to change. Such equations, though unpretty, will be interesting from the point of view of the theory. Each time something of this kind happens, we will state it explicitly.

2.3.1 From natural models to (categorical) dtts

Recall that a natural model in the sense of [Awo18] is the data of

1. a category **ctx** with terminal object;
2. an arrow $p : \dot{U} \to U$ in the presheaf category $\mathbf{Psh}(\mathbf{ctx})$;
3. some *representability data*. This means that for all cospans as in the diagram below, we are given an object $\Gamma.A \in \mathcal{C}$, a morphism $\delta_A : \Gamma.A \to \Gamma$ in **ctx** and an arrow $q_A : ⅄(\Gamma.A) \to \dot{U}$, such that the square below is a pullback.



Remark 2.3.1.1 (Use of the Yoneda lemma)**.** When working with natural models, the Yoneda lemma is heavily used and, in particular, for a presheaf X over **ctx** we tend to identify objects that are in a correspondence under the following (natural) bijection.

$$X(\Gamma) \cong \mathbf{NatTr}(\text{ɣ}\Gamma, X)$$

When we want to avoid using such an abuse, we denote with x an element of $X(\Gamma)$ and x^* its corresponding natural transformation and, conversely, for a natural transformation y we call y_* its corresponding element. One of the advantages of dealing with judgemental theories is that such an ambiguity will be avoided entirely.

Theorem 2.3.1.2. A natural model is the same thing as a (judgemental) dependent type theory where the types and terms fibrations are discrete fibrations.

The greatest part of the theorem relies on the following result. Recall that with respect to a discrete fibration, all maps are cartesian (1.3.1.3), hence whatever unit and counit we supply, their component will be, too.

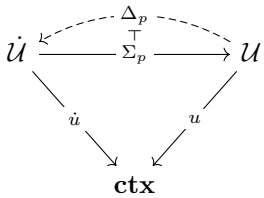
Proposition 2.3.1.3. Let $p : \dot{U} \to U$ a morphism of presheaves over **ctx** and Σ_p its image through the Grothendieck biequivalence 1.5.0.2 restricted to presheaves. The following are equivalent.

1. We are provided with some representability data for p.
2. The functor Σ_p has a right adjoint Δ_p.

Remark 2.3.1.4. It is evident from the discussion between page 245 and 246 of [Awo18] that Awodey was aware of this result, but because he only sketches the proof of the proposition above, we provide it in full.

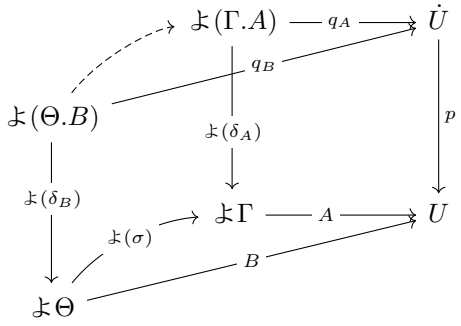
Proof of 2.3.1.3. First of all, let us briefly describe $\Sigma_p : \dot{\mathcal{U}} \to \mathcal{U}$ in terms of $p : \dot{U} \to U$, or at least how it acts on the objects. As detailed in the construction in 1.5, the category $\dot{\mathcal{U}}$ has for objects pairs (Γ, a) where Γ is an object of **ctx** and $a \in \dot{U}(\Gamma)$. Similarly, the category $\mathcal{U}$ has for objects pairs (Γ, A) where Γ is an object of **ctx** and $A \in U(\Gamma)$. The presheaf morphism p induces a function $p_\Gamma : \dot{U}(\Gamma) \to U(\Gamma)$, therefore the (discrete) fibration morphism Σ_p it induces maps a pair (Γ, a) to $(\Gamma, p_\Gamma(a))$. We denote it Σ_p in analogy with Definition 2.3.0.1.

(1 ⇒ 2) We will now construct the functor Δ_p provided that p is representable.



Consider an object $A \in \mathcal{U}$, recall that it corresponds by Remark 2.3.1.1 to an arrow $A^* : \text{ɣ}uA \to U$. Then, we define $\Delta_p : A \mapsto (q_A)_*$, where the

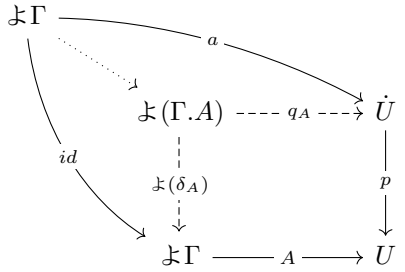
latter is obtained by the representability condition at A^*. On a substitution $\sigma : \Theta \to \Gamma$ we take pullbacks as depicted below.



We now need to show that $\Sigma_p \dashv \Delta_p$. The easiest thing is to provide the unit and the counit.

(ϵ) We want to construct an arrow $\epsilon_A : \Sigma_p \Delta_p A \to A$. We define it to be the cartesian lifting of δ_A at A. Now we need to show that this is a natural transformation, but that follows from the universal property of cartesian lifts. In fact, for each $s : B \to A$, the composition $s \circ u^* \delta_B$ is the cartesian lifting of $\sigma \circ \delta_B$, $u^* \delta_A \circ \Sigma_p \Delta_p s$ that of $\delta_A \circ \Delta_p(\sigma)$, and $\delta_A \circ \Delta_p(\sigma) = \sigma \circ \delta_B$, therefore, by uniqueness (up to iso) of the cartesian lifting, $u^* \delta_A \circ \Sigma_p \Delta_p s = s \circ u^* \delta_B$ too.

(η) We want to construct an arrow $a \to \Delta_p \Sigma_p a$. This is also obtained by cartesian lifting, that of γ_a induced by the dotted arrow in the diagram below.



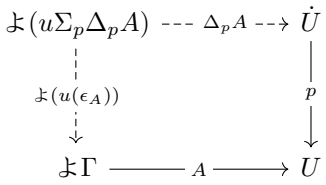
Naturality follows as for ϵ.

Triangle identities of (ϵ, η) lie above commutative diagrams, for Σ_p and Δ_p respectively

$$id_\Gamma = \delta_A \circ \gamma_a \quad \text{and} \quad id_{\Gamma.A} = \delta_{A\delta_A} \circ \gamma_{q_A},$$

therefore they are satisfied again by uniqueness of the cartesian lifting as introduced in 1.3.1.2.

(2) ⇒ (1) The diagram below describes the representability data.



It is a pullback by the universal property of ϵ: for each pair (σ, b) such that $A \circ \sigma = p \circ b$, there is a map

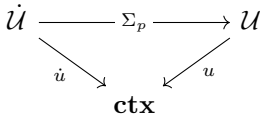
$$s : p \circ b = \Sigma_p(b) \to A$$

induced by precomposition with σ. Therefore there must be a unique ϕ such that $\epsilon \circ \Sigma_p \phi = s$. Now $u(\Sigma_p \phi)$ gives the desired map into $u\Sigma_p\Delta_p A$.

□

Proof of 2.3.1.2. There is a clear correspondence between couples $(\mathbf{ctx}, p : \dot{U} \to U)$ and triangles as below, where $\dot{u}, u$ are discrete fibrations. The correspondence is given by the Grothendieck construction.

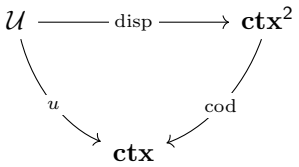
$(\mathbf{ctx}, p : \dot{U} \to U)$



The additional axioms required on both ends are equivalent because of Proposition 2.3.1.3. □

2.3.2 Judgemental dtts vs comprehension categories

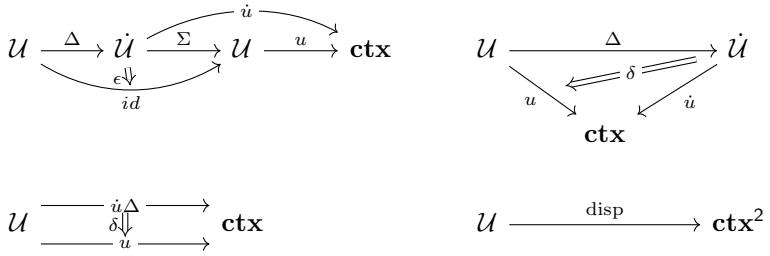
Another categorical approach to dependent type theories which is historically very meaningful was given by Jacobs in [Jac99]. This is the theory of comprehension categories and it is inherently presented in the form of a categorized pre-judgemental theory as below.



Comprehension categories clearly realize some form of context extension, and that is given by display maps.

Construction 2.3.2.1 (From cDTTs to comprehension categories)**.** Each categorized dependent type theory produces a comprehension category as described

by the steps below.



It is enough to follow the picture from left to right (and top to bottom) to see how a categorized dependent type theory in our sense produces a *display* functor, which thus specifies a comprehension category.

Of course it is a legitimate question to ask whether every comprehension category can be realized via a categorized dependent type theory, and indeed something can be said on the topic: they are in fact equivalent, and to prove such a thing is the starting point of Chapter 3.

2.3.3 Dictionary

Dependent type theory has a well established notation, which we switch to in this subsection. The table below declares the dictionary between our framework and the classical notation.

Following the presentation in Section 2.2.1, it will need to take into account Syntax (but there is not much to say there), Judgements, and Rules. What we adopt here is a one-to-one rewriting of (some) components introduced in Section 2.2 in order to make the calculations we will see more transparent. Still, each string of symbols will simply represent its categorical backbone.

2.3.3.1 Dictionary for judgements

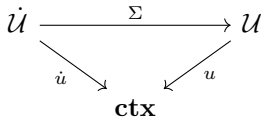
As we mentioned in Definition 2.3.0.1, we think of $\mathcal{U}$ as classifing types, $\dot{\mathcal{U}}$ as classifing terms, and of Σ as performing the typing. We make this clear with the choices in the translation that follow. (Sometimes we might omit the word `Type` for brevity).

$\Gamma \vdash A\ \mathcal{U}$	$\Gamma \vdash A$ `Type`
$\Gamma \vdash a\ \dot{\mathcal{U}} \quad (\Gamma \vdash A\ \mathcal{U} \quad \Gamma \vdash \Sigma a =_{\mathcal{U}} A)$	$\Gamma \vdash a : A$
$\Gamma \vdash A =_{\mathcal{U}} B \quad (\Gamma \vdash A\ \mathcal{U} \quad \Gamma \vdash B\ \mathcal{U})$	$\Gamma \vdash A = B$ `Type`
$\Gamma \vdash a =_{\dot{\mathcal{U}}} b \quad (\Gamma \vdash A\ \mathcal{U} \quad \Gamma \vdash \Sigma a =_{\mathcal{U}} A \quad \Gamma \vdash \Sigma b =_{\mathcal{U}} A)$	$\Gamma \vdash a = b : A$

Remark 2.3.3.1 (How many types to a term?)**.** One might see our choice in the treatment of typing as profoundly Church-like, in the sense that to one term we only assign one type via the functor Σ, and that is far from the practice. The generality of our definition, though, allows for some tweaks, so that if one wishes to have the possibility of assigning different types to the same term (say both $0 : \mathtt{N}$ and $0 : \mathtt{Z}$) one can simply choose $\dot{\mathcal{U}}$ as a subcategory of two categories with, respectively, names for terms and for types (hence code the two above as $(0, \mathtt{N})$ and $(0, \mathtt{Z})$), and make Σ act as a second projection.

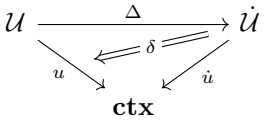
2.3.3.2 Dictionary for rules

We also have a dictionary for rules, which we have (at least) two of. The first is implicitly used in Section 2.3.3.1, and it is the typing.



$$(\Sigma)\ \frac{\Gamma \vdash a\ \dot{\mathcal{U}}}{\Gamma \vdash \Sigma a\ \mathcal{U}} \qquad (\Sigma)\ \frac{\Gamma \vdash a : \Sigma a}{\Gamma \vdash \Sigma a\ \mathtt{Type}}$$

The second is the policy δ from Construction 2.3.2.1, which we here denote as follows.



$$(\delta)\ \frac{\Gamma\ \vdash A\ \mathcal{U}}{\dot{u}\Delta A \vdash \Delta A\ \dot{\mathcal{U}}} \qquad (\delta)\ \frac{\Gamma\ \vdash A\ \mathtt{Type}}{\Gamma.A \vdash q_A : \Sigma\Delta A}$$

Again, such writings are only stand-ins for their categorical counterparts.

2.3.4 Context extension and type dependecy

In this subsection we compute some rules that are automatically deduced by the finite-limit closure of a categorized dependent type theory. As we will see, they correspond to some very well known rules in dependent type theories.

2.3.4.1 Context extension in a DTT, explicitly

Notation 2.3.4.1. For readibility reasons, and in order to highlight the correspondence between the logic and the categories without trivializing it, we denote $A\sigma$ the result of the cartesian lifting of A along σ and $A[\sigma]$ the substitution in the sense of the type theory.

All of the pieces appearing in the dictionary 2.3.3.2 surely do look familiar to the type-theorist reader, all but one, and that is $\Sigma\Delta A$. In fact one might rightfully ask how to compute such an object.

Proposition 2.3.4.2 (On a formal emergence of substitution)**.** Let A be on object in $\mathcal{U}$. Then, $\Sigma\Delta A = A\delta_A$, in the sense of Section 2.2.5.

Proof. We know that there is an arrow $\epsilon_A : \Sigma\Delta A \to A$. By the discussion in Section 2.2.5, the thesis is equivalent to the fact that the cartesian lifting of δ_A along u is precisely ϵ_A. Recall, that δ_A is by definition $u(\epsilon_A)$, therefore it is a lifting. It is cartesian by assumption. □

Notice that this is as trivial as (and in fact it amounts to) proving that the process of computing weakening *can* be simulated in the syntax using substitution, provided that suitable substitution rules do in fact exist. We can re-read the rule hidden in the policy δ as follows.

$$(\delta)\ \frac{\Gamma \vdash A\ \mathcal{U}}{\dot{u}\Delta A \vdash\ \Delta A\dot{\mathcal{U}}} \qquad (\delta)\ \frac{\Gamma \vdash A\ \mathtt{Type}}{\Gamma.A \vdash q_A : A\delta_A}$$

Finally, we observe that the deductive rule on the right is a version of *context extension* in dependent type theory.

2.3.4.2 Type dependency in a DTT, explicitly

Similarly to the case of context extension, in a cDTT as in Definition 2.3.0.1 the most classical instances of type depencency emerge too. Let us produce the following two rules.

$$\text{(DTy)}\ \frac{\Gamma \vdash a : A \qquad \Gamma.A \vdash B\ \mathtt{Type}}{\Gamma \vdash B[a]\ \mathtt{Type}} \qquad \text{(DTm)}\ \frac{\Gamma \vdash a : A \qquad \Gamma.A \vdash b : B}{\Gamma \vdash b[a] : B[a]}$$

In order to do so, we first need (nested) classifiers for the premises. More generally, with an iterated construction we will code composed judgements of the form below.

$$\Gamma \vdash a : A, \Gamma.A \vdash b : B \qquad \Gamma \vdash A, \Gamma.A \vdash b : B$$
$$\Gamma \vdash a : A, \Gamma.A \vdash B \qquad \Gamma \vdash A, \Gamma.A \vdash B$$

This is achieved as follows.

$$\begin{array}{ccccc}
\dot{\mathcal{U}}.\Sigma\Delta\dot{\mathcal{U}} & \longrightarrow & \mathcal{U}.\Delta\dot{\mathcal{U}} & \longrightarrow & \dot{\mathcal{U}} \\
\downarrow & \lrcorner & \downarrow & \lrcorner & \downarrow{\scriptstyle \Sigma} \\
\dot{\mathcal{U}}.\Sigma\Delta\mathcal{U} & \longrightarrow & \mathcal{U}.\Delta\mathcal{U} & \longrightarrow & \mathcal{U} \\
\downarrow & \lrcorner & \downarrow & \lrcorner & \downarrow{\scriptstyle u} \\
\dot{\mathcal{U}} & \xrightarrow[\Sigma]{} & \mathcal{U} & \xrightarrow[\Delta]{} & \dot{\mathcal{U}} \xrightarrow[\dot{u}]{} \mathbf{ctx}
\end{array}$$

For example, the fibration on $\mathcal{U}.\Delta\mathcal{U}$ classifies pairs (A, B) of types such that $u\Sigma\Delta(A) = u(B)$. This is precisely the composed judgement $\Gamma \vdash A, \Gamma.A \vdash B$.

Lemma 2.3.4.3 (Focus on $\mathcal{U}.\Delta\mathcal{U}$)**.** In a judgemental dtt we have the following rules and policy.

$$\mathcal{U}.\Delta\mathcal{U} \xrightarrow{u.\dot{u}\Delta} \mathcal{U} \underset{u}{\overset{\dot{u}\circ\Delta}{\Downarrow \delta}} \mathbf{ctx}$$

$$(\Gamma \vdash A, \Gamma.A \vdash B) \longmapsto (\Gamma \vdash A) \longmapsto (\Gamma.A \to \Gamma)$$

Proof. This is the first detailed instance of two judgement classifiers supported by the same category, since one could perform the two following compositions

$$\begin{array}{ccc}
\mathcal{U}.\Delta\mathcal{U} & \longrightarrow & \mathcal{U} \\
\downarrow & \lrcorner & \downarrow{\scriptstyle u} \\
\mathcal{U} & \xrightarrow{\dot{u}\circ\Delta} & \mathbf{ctx} \\
{\scriptstyle u}\downarrow & \overset{\delta}{\Leftarrow} & \nearrow{\scriptstyle \mathrm{Id}} \\
\mathbf{ctx} & &
\end{array}$$

which are related as discussed in Section 2.3.4.3. Such a policy is the symptom of a shift in perspective: on the upper path, one travels along the pullback diagram above, therefore the context which one lands on is $\Gamma.A$; on the lower, one is concerned with the "original" context of A, therefore getting to Γ. They are related, as we have thoroughly discussed, by δ_A. Notice that the lower path, being a composition of fibrations, is a fibration as well. □

Since the classifier in the lower part of the diagram in 2.3.4.3 will play an important role in a later discussion, we name it,

$$v : \mathcal{U}.\Delta\mathcal{U} \to \mathbf{ctx}.$$

In [Awo18, Prop. 2.2] there is the construction of a presheaf $P(\mathcal{U})$, with P a polynomial functor, classifying the same nested judgement as $\mathcal{U}.\Delta\mathcal{U}$. The polynomial is defined as follows

$$P = P_p \colon \mathbf{Psh}(\mathbf{ctx}) \xrightarrow{\dot{U}^*} \mathbf{Psh}(\mathbf{ctx})_{/\dot{U}} \xrightarrow{\Pi_p} \mathbf{Psh}(\mathbf{ctx})_{/U} \xrightarrow{\Sigma_U} \mathbf{Psh}(\mathbf{ctx})$$

meaning the pullback along the terminal presheaf morphism from $\dot{U}$, followed by the right adjoint to pullback along p, followed by composition with the terminal from U. We apologize for the ambiguous notation ($-^*$, Π, Σ), but we promise this will only be used in the current section.

Lemma 2.3.4.4 (Classifiers à la Awodey)**.** One can show that the fibration $v : \mathcal{U}.\Delta\mathcal{U} \to \mathbf{ctx}$ is precisely the projection $\pi : \text{ɟ}\downarrow P(U) \to \mathbf{ctx}$.

Proof. We sketch the identity fiber-wise. At each Γ, $(\mathcal{U}.\Delta\mathcal{U})_\Gamma$ is comprised of pairs (A, B) with A, B in $\mathcal{U}$ such that $u(B) = u(\Sigma\Delta A)$ and $u(A) = \Gamma$. By Remark 2.3.1.1, such A and B correspond to A^* and B^* fitting in the following diagram,

$$\begin{array}{ccccc} U & \xleftarrow{B^*} & \text{ɟ}\Gamma.A & \longrightarrow & \dot{U} \\ & & \downarrow & \lrcorner & \downarrow{\scriptstyle p} \\ & & \text{ɟ}\Gamma & \xrightarrow[A^*]{} & U \end{array}$$

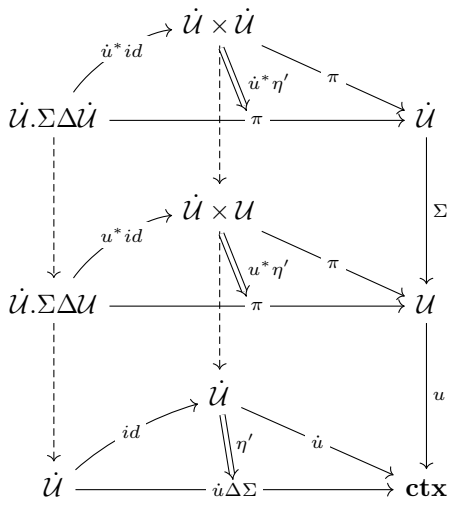
with the central square being a pullback by representability of p. Using a result from [DT87], [Awo18, Prop. 2.2] shows that such diagrams are in a 1-to-1 correspondence with maps of the form

$$\text{ɟ}\Gamma \to P(U)$$

hence with elements of $P(U)(\Gamma)$. □

We have shown that there is a very tight connection between our classifier and Awodey's. We hope that, though almost tautological, this result can convince the reader about the advantages of our construction, as it makes it much easier to predict the correct pullback that constructs the desired classifier (this will be more and more evident in the following sections), while it might not be always easy to find suitable (polynomial) functors to classify complex judgements. Also, we can avoid the complex machinery of polynomial functors (and, in this case, the conflicting notation).

In order to provide the rules (DTm) and (DTy) we build a map out of $\dot{\mathcal{U}}.\Sigma\Delta\mathcal{U}$ (and of $\dot{\mathcal{U}}.\Sigma\Delta\dot{\mathcal{U}}$), and all we have is $\Sigma, \Delta, \eta, \epsilon$, finite limits closure, composition, substitution, whiskering, and ♯-lifting. A few tries lead us to the following choice.



We call $\eta' : \dot{u} \Rightarrow \dot{u}\Delta\Sigma$ the natural transformation induced by η via 2.1.0.14 and apply $\sharp$-lifting (2.1.0.12) as on the left. Write π for projections.

When we compute each lifting, we see that the policy $(\dot{u}^*\eta')$ computes, starting from a pair (a, b) some new term in context Γ, while the policy $(u^*\eta')$ matches to a pair (a, B) a new type in context Γ.

We give each a meaningful name, that is, extensively:

$$\dot{u}^* id(a, b) = (a, b[a]),$$

$$u^* id(a, B) = (a, B[a]).$$

Notice that the typing is appropriate due to the action of the vertical Σ.

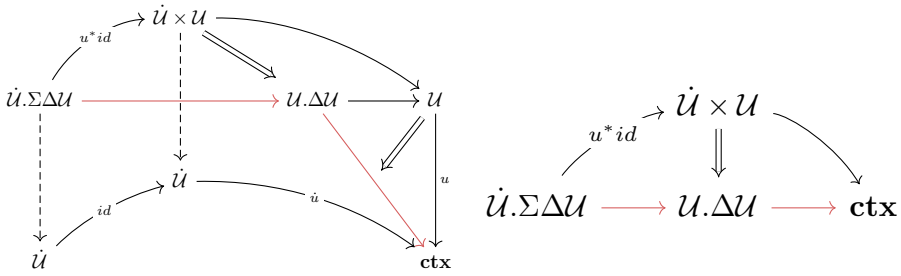
We are now one step away from having (DTy) and (DTm), and in fact the distance between the policies u^*id, $\dot{u}^*id$ and the desired rules is extremely subtle, and one could argue - though the author might disagree - a merely technical one: on the premise of, say, dependent typing, we now have the following nested judgement (which we write in our original notation for judgemental theories, so that we can make the difference evident)

$$\Gamma.A \vdash (a, B)\, \dot{\mathcal{U}}.\Sigma\Delta\mathcal{U}(u \circ \dot{u}\Delta\Sigma.u)$$

while we wish to have the pair stand over Γ. That is achieved by v (that from Lemma 2.3.4.3),

$$\Gamma \vdash (a, B)\, \dot{\mathcal{U}}.\Sigma\Delta\mathcal{U}(v)\,,$$

therefore we need to adjust the two policies accordingly. We can do that by regular 2-categorical manipulations attaching v (the composition of the red arrows below) to the diagram above.



The policy on the right now is (DTy). One could repeat a similar argument for terms, which again have the correct typing because of the action of Σ in the $\sharp$-lifting above.

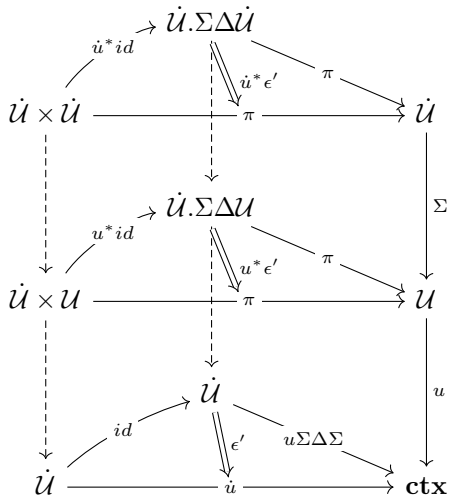
Remark 2.3.4.5 (Similarities between DTy and proof theoretic Cut)**.** In the next section we highlight a remarkable connection between dependent typing and the cut rule from natural deduction: we redirect the reader to Remark 2.4.3.5 for more information.

2.3.4.3 Substitution along display maps

Of course there are (at least) two interesting natural transformations that we know of insisting on

$$\dot{\mathcal{U}} \xrightarrow{\Sigma} \mathcal{U} \xrightarrow{\Delta} \dot{\mathcal{U}} \xrightarrow{\dot{u}} \mathbf{ctx}$$

that is η and ϵ. If η is so interesting, one might wonder what repeating the process discussed in Section 2.3.4.2 with ϵ might bring. We have a hint about its outcome, and that is given by the δ from Construction 2.3.2.1, still we compute it precisely.



We call $\epsilon' : \dot{u}\Delta\Sigma \Rightarrow \dot{u}$. The construction detailed here, when explicitly computed, induces the two following rules involving $\delta_A : \Gamma.A \to \Gamma$,

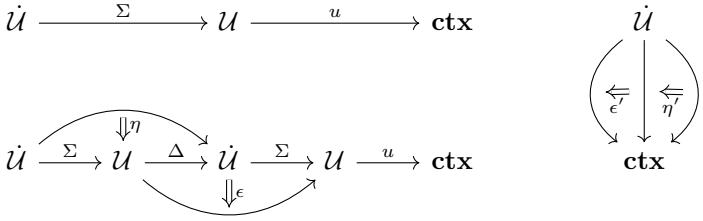
$$\dot{u}^*id(a, a') = (a, a'\delta_A)$$

$$u^*id(a, A') = (a, A'\delta_A)$$

meaning we can transport terms and types along arbitrary display maps, given that they insist on the same context.

$$\frac{\Gamma \vdash a : A \qquad \Gamma \vdash a' : A'}{\Gamma \vdash a : A \qquad \Gamma.A \vdash a'\delta_A : A'\delta_A}$$

Remark 2.3.4.6 (Rules for free)**.** Since we now have rules involving the unit and rules involving the counit of an adjunction, we can exploit their relation to one another and show once again the computational power of categorized judgemental theories. In particular, the (bases of the) constructions in Section 2.3.4.2 and Section 2.3.4.3 are related by the triangle identities:



so that whiskering the two $\sharp$-liftings above to compute ϵ after η yields the functor $\dot{u}$. Then at each level we have the same relation. Therefore

$$(A'\delta_A)[a] = A' \quad \text{and} \quad (a'\delta_A)[a] = a',$$

or, explicitly, we have the following rule

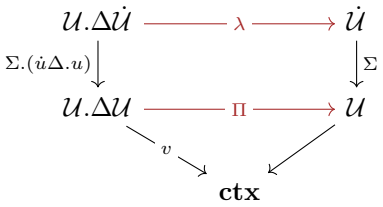
$$\dot{\mathcal{U}} \times \dot{\mathcal{U}} \xrightarrow{\dot{u}^*id} \dot{\mathcal{U}}.\Sigma\Delta\dot{\mathcal{U}} \xrightarrow{\dot{u}^*id} \dot{\mathcal{U}} \times \dot{\mathcal{U}} \qquad \frac{\Gamma \vdash a : A \qquad \Gamma \vdash a' : A'}{\Gamma \vdash a = a : A \qquad \Gamma \vdash (a'\delta_A)[a] = a' : A'}$$

which we did not know before. Such a rule is an instance of the discussion in

Remark 2.2.4.2. Of course we cannot say the same for the opposite composition, but that is telling all in itself.

2.3.5 Dependent type theories with Π-types

Definition 2.3.5.1 (Π-types)**.** A categorized dependent type theory *with* Π-*types* is a cDTT as in Definition 2.3.0.1 having two additional rules Π, λ such that the diagram below is commutative and the upper square is a pullback.



Recall that v is that from Lemma 2.3.4.3. The rest of this subsection is devoted to showing that the proof theory generated by such a categorized judgemental theory actually meets our intuition for having Π-types.

2.3.5.1 À la Martin-Löf

Having Π-types in the sense of [Mar75] means to implement the following rules,

$$(\Pi\mathrm{F})\ \frac{\Gamma \vdash A\ \mathtt{Type} \qquad \Gamma.A \vdash B\ \mathtt{Type}}{\Gamma \vdash \Pi_A B\ \mathtt{Type}} \qquad (\Pi\mathrm{I})\ \frac{\Gamma \vdash A\ \mathtt{Type} \qquad \Gamma.A \vdash b : B}{\Gamma \vdash \lambda_A b : \Pi_A B}$$

$$(\Pi\mathrm{E})\ \frac{\Gamma \vdash f : \Pi_A B \qquad \Gamma \vdash a : A}{\Gamma \vdash f(a) : B[a]} \qquad (\Pi\beta)\ \frac{\Gamma.A \vdash b : B \qquad \Gamma \vdash a : A}{\Gamma \vdash (\lambda_A b)(a) = b[a] : B[a]}$$

plus their congruence with definitional equality.

$$(\Pi\mathrm{F}{=})\ \frac{\Gamma \vdash A = A'\ \mathtt{Type} \qquad \Gamma.A \vdash B = B'\ \mathtt{Type}}{\Gamma \vdash \Pi_A B = \Pi_{A'} B'\ \mathtt{Type}}$$

$$(\Pi\mathrm{I}{=})\ \frac{\Gamma \vdash A = A\ \mathtt{Type} \qquad \Gamma.A \vdash b = b' : B}{\Gamma \vdash \lambda_A b = \lambda_A b' : \Pi_A B}$$

$$(\Pi\mathrm{E}{=})\ \frac{\Gamma \vdash f = f' : \Pi_A B \qquad \Gamma \vdash a = a' : A}{\Gamma \vdash f(a) = f'(a') : B[a]}$$

The first two rules are almost evident in the very definition of dependent type theory with Π-types, while the other rules will be derived by the limit closure of the class of judgements and rules.

(ΠF) Type formation is precisely the rule (Π) in the sense of Section 2.2.4 and Section 2.3.3.1, indeed $\mathcal{U}.\Delta\mathcal{U}$ classifies precisely the premises of (ΠF).

(ΠI) Similarly, the introduction rule is precisely the rule (λ) in the sense of Section 2.2.4 and Section 2.3.3.2, where the commutativity of the diagram ensures the correct typing for the term.

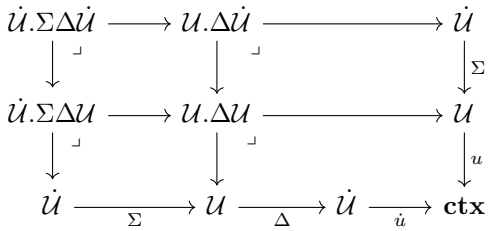
In order to express the elimination rule, we first need to code its premise, that is the nested judgement

$$\Gamma \vdash f : \Pi_A B \quad \Gamma \vdash a : A.$$

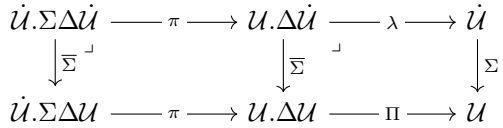
Notice that, because of (ΠF), this is actually silent of two judgements, meaning it should read

$$\Gamma \vdash A \quad \Gamma.A \vdash B \quad \Gamma \vdash f : \Pi_A B \quad \Gamma \vdash a : A,$$

instead, so that this is really the judgement we need to give a classification of. One can check that $\dot{\mathcal{U}}.\Sigma(\mathcal{U}.\Delta\mathcal{U})$ classifies the first, second, and fourth judgement appearing above. Also, we know from Section 2.3.4.2 that $\dot{\mathcal{U}}.\Sigma(\mathcal{U}.\Delta\mathcal{U}) \cong \dot{\mathcal{U}}.\Sigma\Delta\mathcal{U}$. This is an instance of Remark 2.3.0.2, and it just expresses the fact that, whenever we have a term $a : A$, we really have its type in our code already.



To now introduce the term f, we need to perform one more pullback. We attach the diagram above to that in Definition 2.3.5.1. We are entitled to do so because, by hypothesis, the square that Π and λ fit in has the correct map on its left. For brevity, and since it should not cause much trouble, for the remainder of the proof we call all "horizontal" projections π, and all "vertical" ones $\overline{\Sigma}$.

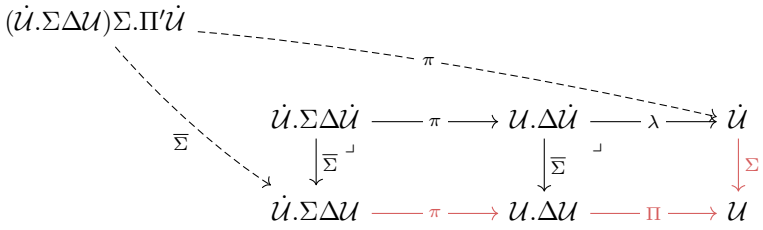


To express the classifier for the whole premise, then, is to compute the pullback against Σ of the composition of Π and π in the lower part of the diagram. Call $\Pi' = \Pi \circ \pi$. The premise of (E) is then classified by $(\dot{\mathcal{U}}.\Sigma\Delta\mathcal{U})\Sigma.\Pi'\dot{\mathcal{U}}$. We can see how it all builds up in the following suggestive writing

$$(\dot{\mathcal{U}}.\Sigma\Delta\mathcal{U})\Sigma.\Pi'\dot{\mathcal{U}} \qquad (a\,.\,(A\,.\,B))\,.\,f$$

which is fibered over Γ: though not all of its components are types or terms specifically in context Γ, every judgement appearing in this nested one is built out of a construction performed entirely in Γ.

From now on, we will write all n-uples as above as traditional n-uples, since all pullbacks are subcategories of a product after all.



We now have two pullbacks insisting on the same cospan, then necessarily it is

$$(\dot{\mathcal{U}}.\Sigma\Delta\mathcal{U})\Sigma.\Pi'\dot{\mathcal{U}} \cong \dot{\mathcal{U}}.\Sigma\Delta\dot{\mathcal{U}}. \tag{2.1}$$

This in *not* an instance of Remark 2.3.0.2, though, and the isomorphism above actually turns out to contain all the information needed to provide rules (E) and (β), and then some.

Clearly there is always a map going from right to left, just consider:

$$(a, b) \mapsto (A, B, \lambda_A b, a),$$

but Eq. (2.1) is adding three more pieces of information, meaning

(i) there is also a map going from left to right, (though we can always *expand* information, only this tells us we can *compact* it);

(ii) starting from the left, going right, and back left again, yields the identity;

(iii) starting from the right, going left, and back right again, yields the identity.

Of these, (i) will induce elimination and (iii) β-computation. The additional piece in (ii) will tell us something about what is generally called the η-rule, which is much more controversial. We will discuss it in detail in Section 2.3.5.3.

Call ζ and θ the inverse maps. A little calculation shows that they act as follows:

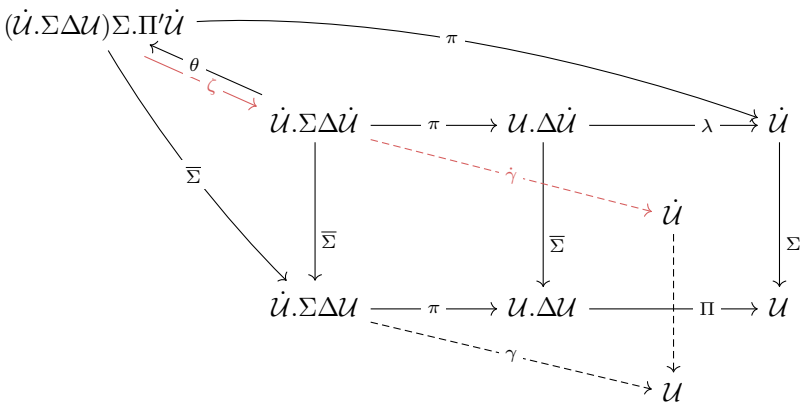
$$\theta : (a, b) \mapsto (A, B, \lambda_A b, a), \quad \zeta : (A, B, f, a) \mapsto (a, f_B),$$

where we write f_B for the term of type B in the second component of ζ. Broadly speaking, θ computes introduction (this is evident by $\lambda \circ \pi = \pi \circ \theta$) and ζ elimination (both because of its typing and because *we say so*).

Before we can provide an explicit representation for the missing rules, we shall be able to account for writings $b[a]$ and $B[a]$. In order to do that, we need to use the diagram in Section 2.3.4.2. We paste it to the previous one as follows, calling

$$\gamma = \pi \circ u^* id \quad \text{and} \quad \dot{\gamma} = \pi \circ \dot{u}^* id.$$

Notice that the map $\overline{\Sigma} : \dot{\mathcal{U}}.\Sigma\Delta\dot{\mathcal{U}} \to \dot{\mathcal{U}}.\Sigma\Delta\mathcal{U}$ is precisely that appearing in Section 2.3.4.2 and Section 2.3.4.3, so that both "rectangles" insist on the same functor. All solid squares are pullbacks, the dashed one is only commutative.



(ΠE) The functor $\dot\gamma\zeta$ is the elimination rule, because to each quadruple it matches a term of the correct type. We call $\dot\gamma(a, f_B) = f_B[a] =: f(a)$.

(Πβ) Computation β amounts to proving that if we apply introduction, followed by elimination, we kind of get to the point we started from. This is a rule with codomain as in 2.2.3.6, therefore we show that identity on $\dot{\mathcal{U}}.\Sigma\Delta\dot{\mathcal{U}}$ equalizes the following pair of arrows,

$$\dot{\mathcal{U}}.\Sigma\Delta\dot{\mathcal{U}} \xrightarrow{\theta} (\dot{\mathcal{U}}.\Sigma\Delta\mathcal{U})\Sigma.\Pi'\dot{\mathcal{U}} \xrightarrow{\zeta} \dot{\mathcal{U}}.\Sigma\Delta\mathcal{U} \xrightarrow{-\dot\gamma\to} \dot{\mathcal{U}}, \qquad \dot{\mathcal{U}}.\Sigma\Delta\dot{\mathcal{U}} \xrightarrow{id} \dot{\mathcal{U}}.\Sigma\Delta\dot{\mathcal{U}}$$

On the upper path is computed $(\lambda_A b)(a)$, on the lower we get $b[a]$. The two paths equalize trivially. The desired rule is then

$$id : \dot{\mathcal{U}}.\Sigma\Delta\dot{\mathcal{U}} \to \dot{\mathcal{U}}.\Sigma\Delta\dot{\mathcal{U}} = \mathcal{E}(\dot\gamma\zeta\theta, \dot\gamma).$$

2.3.5.2 Of congruence rules involving definitional equality.

In our dictionary in 2.3.3 we decided that definitional equality of types and terms should be interpreted as judgemental equality according to u and $\dot u$, respectively, hence as identity of objects in the respective "universe" categories. This guarantees that rules (ΠF=), (ΠI=), (ΠE=) are automatically verified. Rule (ΠI=), also known as the *ξ-rule*, in particular, is not verified by all models, especially those that are more computationally oriented, such as Kleene realizability or game semantics: we are indeed quite extensional in our spirit, but we believe this is more of a choice that we are making than a constraint of categorized judgemental theories, and that it would be interesting to further develop the theory with different, weaker, but still finite-limit stable interpretations of judgemental equality.

2.3.5.3 Of η and elimination.

The η-rule accounts for the need to determine what happens in the case that one wants to apply elimination followed by introduction, and at first it looks exactly as the dual of (ΠβC). While it is clear that β should prescribe equality of two terms, though, there is actually no agreement on the features η should present, so that in the literature we find instances of the resulting computation of η as being a *conversion* (*i.e.* consisting of a definitional equality), interpreted as an *expansion*, or a *reduction* (meaning a non-symmetric relation whose reflexive, symmetric, and transitive closure defines the conversion). The virtue of each process, and each of its 2-categorical delivery, is the topic of [See86].

In our framework, Eq. (2.1) tells us something about which η-rule we should be looking at, and in fact we have

$$(\Pi\eta)\quad \frac{\Gamma \vdash f : \Pi_A B}{\Gamma \vdash f = \lambda_A(f_B) : \Pi_A B}$$

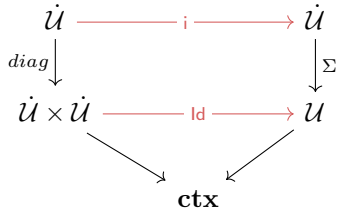
which is precisely what $\theta\zeta = id$ says. This is only one of the possible expressions for η, and it differs from that presented in [Awo18, p.253], which much more swiftly agrees with the tradition of categories with families. This is because, in a sense, we think the notion of elimination presented there, and in Section 2.3.5.1 above, is not the correct one: it really is ζ performing the elimination, and it

really is f_B the term witnessing it. It is not in the computation through $\dot{\gamma}$ that a term of type $\Pi_A B$ turns into a term involving A, B. This argument, together with the possibility of excluding the η rule entirely, will be made much more clear in Section 2.3.7.

2.3.6 Dependent type theories with (extensional) Id-types

For identity types we need to be able to consider pairs of terms of the same type, this is why we begin by pulling back Σ against itself. Call π_1, π_2 the corresponding projections and $diag : \dot{\mathcal{U}} \to \dot{\mathcal{U}} \times \dot{\mathcal{U}}$ the unique map such that $\pi_1 \circ diag = id = \pi_2 \circ diag$.

Definition 2.3.6.1 (Extensional Id-types)**.** A categorized dependent type theory *with extensional* Id*-types* is a cDTT as in Definition 2.3.0.1 having two additional rules Id, i such that the diagram below is commutative and the upper square is a pullback.



Again, the rest of the subsection is dedicated to showing that the proof theory generated by such judgemental theory actually meets our intuition for having Id-types. Classically, having Id-types means to implement the following rules

$$(\mathsf{IdF})\ \frac{\Gamma \vdash A\ \mathsf{Type} \qquad \Gamma \vdash a : A \qquad \Gamma \vdash b : A}{\Gamma \vdash \mathsf{Id}_A(a,b)\ \mathsf{Type}} \qquad (\mathsf{IdI})\ \frac{\Gamma \vdash a : A}{\Gamma \vdash \mathsf{i}(a) : \mathsf{Id}_A(a,a)}$$

$$(\mathsf{IdE})\ \frac{\Gamma \vdash c : \mathsf{Id}_A(a,b)}{\Gamma \vdash a = b : A} \qquad (\mathsf{Id}\eta)\ \frac{\Gamma \vdash c : \mathsf{Id}_A(a,b)}{\Gamma \vdash c = \mathsf{i}(a) : \mathsf{Id}_A(a,a)}$$

As it was for Definition 2.3.5.1, the first two rules are evident in the very definition of dependent type theory with Id-types.

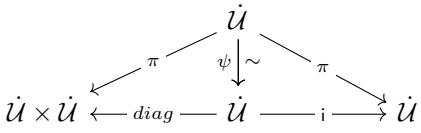
(IdF) Type formation is precisely the rule (Id) in the sense of Section 2.2.4 and Section 2.3.3.1. Clearly $\dot{\mathcal{U}} \times \dot{\mathcal{U}}$ classifies the premises of (IdF).

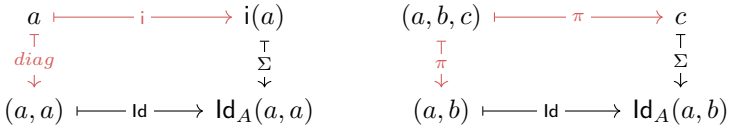
(IdI) Similarly, the introduction rule is the rule (i) in the sense of Section 2.2.4 and Section 2.3.3.2, where the commutativity of the diagram forces the correct typing for the term.

For elimination and conversion we need to pin-point a classifier for judgements of the form

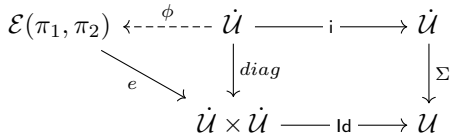
$$\Gamma \vdash c : \mathsf{Id}_A(a,b),$$

but since the square is a pullback insisting on the cospan (Id, Σ), such a feat is achieved by the (upper-left) $\dot{\mathcal{U}}$. Then not only do Id, i compute the appropriate term and type (below on the left), but they also act as projections (below on the right).





The object classifying judgements of the form $\Gamma \vdash a = b : A$, instead, is the equalizer $\mathcal{E}(\pi_1, \pi_2)$. By its universal property there must be a unique ϕ making the following diagram commute.



(IdE) The elimination rule, then, is $\phi : \dot{\mathcal{U}} \to \mathcal{E}(\pi_1, \pi_2)$.

(Idη) The computation rule is computed as

$$\dot{\mathcal{U}} \to \mathcal{E}(\pi, \psi \mathsf{i} \pi_1 e \phi) = \mathcal{E}(\pi, \pi) = \dot{\mathcal{U}}$$

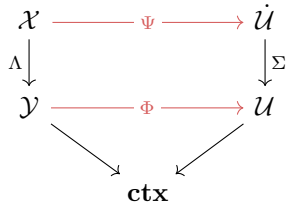
therefore it is the map $id : \dot{\mathcal{U}} \to \dot{\mathcal{U}}$.

There would be a notion of β-computation (in the sense of introduction followed by elimination) here, too, but it is not usually written because it is trivial once one has definitional equality. In fact, it takes the following form.

$$(\mathsf{Id}\beta 1) \quad \frac{\Gamma \vdash a : A}{\Gamma \vdash a = a : A} \qquad (\mathsf{Id}\beta 2) \quad \frac{\Gamma \vdash a : A}{\Gamma \vdash \mathsf{i}(a) = \mathsf{i}(a) : \mathsf{Id}_A(a, a)}$$

2.3.7 A categorical definition of type constructor

Definition 2.3.7.1 (The type constructor Φ)**.** A categorized dependent type theory *with* Φ*-types* is a cDTT as in Definition 2.3.0.1 having two additional rules Φ, Ψ such that the diagram below is commutative and the upper square is a pullback.



Remark 2.3.7.2. Notice that the definition is implicitly assuming that Λ belongs to the closure of the generators under finite limits. Also, it is evident by the previous sections that Π-types and Id-types fall under this definition.

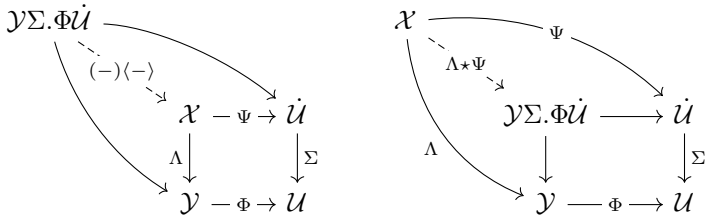
The rest of the subsection is devoted to showing that the proof theory generated by such judgemental theory actually meets our intuition for having Φ-types.

$$(\Phi\mathrm{F}) \quad \frac{\Gamma \vdash Y \; \mathcal{Y}}{\Gamma \vdash \Phi Y \; \mathtt{Type}} \qquad (\Phi\mathrm{I}) \quad \frac{\Gamma \vdash X \; \mathcal{X}}{\Gamma \vdash \Psi X : \Phi \Lambda X}$$

(ΦF) Type formation is precisely the rule (Φ) in the sense of Section 2.2.4 and Section 2.3.3.1.

(ΦI) Similarly, the introduction rule is precisely the rule (Ψ) in the sense of Section 2.2.4 and Section 2.3.3.2, where the commutativity of the diagram forces the correct typing for the term.

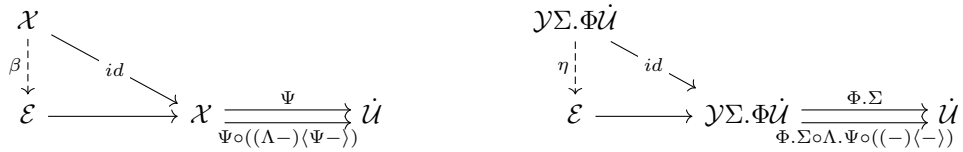
Now, because we have requested that the square in Definition 2.3.7.1 is a pullback, we automatically get the dashed functors below.



(ΦE) The rule associated to functor $(-)\langle-\rangle$ gives us the elimination rule on the right. Indeed the pullback category precisely classifies the premises of (ΦE).

$$(\Phi\text{E}) \quad \frac{\Gamma \vdash a : \Phi Y}{\Gamma \vdash \Phi Y\langle a\rangle\ \mathcal{X}}$$

By essential uniqueness of pullbacks, the compositions $((-)\langle-\rangle) \circ (\Lambda \star \Psi)$ and $(\Lambda \star \Psi) \circ ((-)\langle-\rangle)$ both amount to the identity of the respective object. This observation provided by universal property of the equalizer induces the arrows η and β in the diagram below.



(Φβ) The rule associated to the functor β is our β-computation. Indeed, if we write down the rule explicitly we get the following.

$$(\Phi\beta) \quad \frac{\Gamma \vdash X\ \mathcal{X}}{\Gamma \vdash \Psi X = \Psi((\Lambda X)\langle\Psi X\rangle) : \Phi\Lambda X}$$

(Φη) The rule associated to the functor η is our η-computation. Indeed, if we write down the rule explicitly we get the following.

$$(\Phi\eta) \quad \frac{\Gamma \vdash a : \Phi Y}{\Gamma \vdash a = \Psi(\Phi Y\langle a\rangle) : \Phi Y}$$

Additionally, and as in the case of dependent products in Section 2.3.5.2, we have rules guaranteeing that definitional equality of terms and types is "preserved" through formation, introduction, and elimination. See thereof for a discussion on possible variations.

2.3.7.3 (Weaker notions of type constructors)**.** Our definition of type constructor is very modular: for example, if we request that the square in Definition 2.3.7.1 is a weak pullback (as opposed to a pullback) with a distinguished section, we can still construct the functors $(-)\langle -\rangle$ and $\Lambda \star \Psi$, and one of the two compositions still amounts to the identity. This ensures both elimination and β-computation, while we lose η-computation. This remark generalizes a similar analysis contained in [Awo18, Cor. 2.5].

We believe that Definition 2.3.7.1 is more proof of both the computational and the expressive power of categorized judgemental theories. We now use the construction above to enrich a cDTT with units and dependent sums. We reverse engineer the theory in order to provide the correct definition, and that will be all that we need because of the calculations above. By the end of this chapter, we will have shown that Definition 2.3.7.1 captures dependent products, dependent sums, unit types, extensional identity types. In addition, the construction in [Awo18, §2.4] might suggest that it fits intensional identity, too, but we do not discuss this further here.

Remark 2.3.7.4 (Other type constructors)**.** We are indeed aware that Definition 2.3.7.1 does not capture *all* type constructors used in both the theory and the practice of type theory, for example it does not allow for the description of (co)inductive types, but we believe that our categorical theory of judgement has been proved fruitful in coding syntactic data. Clearly finite limits will capture finite constructions, but 2-category theory is much more than finite, nor it is only about limits, therefore we trust that with some effort this work could be extended to different constructors.

2.3.8 Examples: unit types and Σ-types

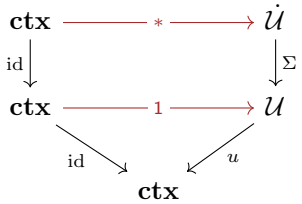
2.3.8.1 Dependent type theories with unit types

Our aim is to describe the premises of introduction and formation, and the relation they are in. Recall that the rules in question are

$$\text{(uI)}\quad \frac{\vdash \Gamma\ \mathtt{ctx}}{\Gamma \vdash 1_\Gamma\ \mathtt{Type}} \qquad \text{(uF)}\quad \frac{\vdash \Gamma\ \mathtt{ctx}}{\Gamma \vdash *_\Gamma : 1_\Gamma}$$

so that both only take in input a context, and the premises are identical, hence our motivation to give the following definition.

Definition 2.3.8.1 (Unit-types)**.** A categorized dependent type theory *with unit-types* is a cDTT having two additional functors 1, $*$ such that the diagram below is commutative and the upper square is a pullback.



We now show that the judgemental theory generated by diagrams in Definition 2.3.8.1 contains codes for formation, introduction, elimination, and computation of unit types. Introduction and formation in fact read as follows

$$(1)\quad \frac{\Gamma \vdash \Gamma\ \mathrm{id}}{\Gamma \vdash 1_\Gamma\ u} \qquad\qquad (*)\quad \frac{\Gamma \vdash \Gamma\ \mathrm{id}}{\Gamma \vdash (*_\Gamma, 1_\Gamma)\ \dot{u}}$$

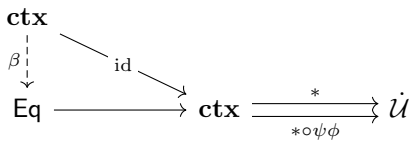
or, in our more familiar writing

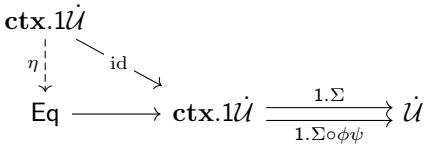
$$(\mathrm{uI})\quad \frac{\vdash \Gamma\ \mathtt{ctx}}{\Gamma \vdash 1_\Gamma\ \mathtt{Type}} \qquad\qquad (\mathrm{uF})\quad \frac{\vdash \Gamma\ \mathtt{ctx}}{\Gamma \vdash *_\Gamma : 1_\Gamma}$$

moreover, the elimination rule is captured by the unique map $\psi\colon \mathbf{ctx}.1\dot{\mathcal{U}} \to \mathbf{ctx}$ and it translates to the syntactic writing on the left, while postcomposed with $*$ it translates as the more familiar rule on the right

$$(\psi)\quad \frac{\Gamma \vdash t : 1_\Gamma}{\vdash \Gamma\ \mathtt{ctx}} \qquad\qquad (* \circ \psi)\quad \frac{\Gamma \vdash t : 1_\Gamma}{\Gamma \vdash *_\Gamma : 1_\Gamma}$$

which is also denoted (uE). Finally, computation β and η can be decoded from the two following diagrams





with ϕ the inverse to ψ, which read, respectively, as follows.

$$(\mathrm{u}\beta)\quad \frac{\vdash \Gamma\ \mathtt{ctx}}{\Gamma \vdash *_\Gamma =_{1_\Gamma} *_\Gamma} \qquad\qquad (\mathrm{u}\eta)\quad \frac{\Gamma \vdash t : 1_\Gamma}{\Gamma \vdash t =_{1_\Gamma} *_\Gamma}$$

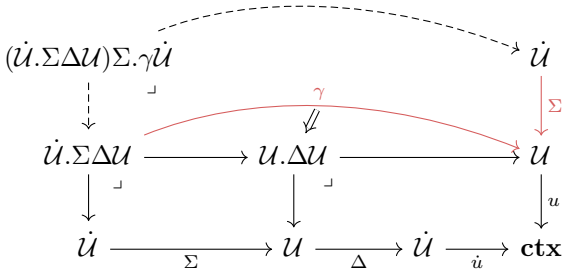
2.3.8.2 Dependent type theories with Σ-types

We hope the reader will forgive us if to avoid confusion we adopt the unusual notation of $\int$ instead of Σ. We then start to look at rules for formation and introduction, which for sum types are usually the following.

$$(\textstyle\int\mathrm{F})\quad \frac{\Gamma \vdash A\ \mathtt{Type} \qquad \Gamma.A \vdash B\ \mathtt{Type}}{\Gamma \vdash \int_A B\ \mathtt{Type}}$$

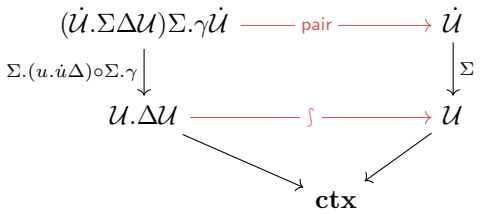
$$(\textstyle\int\mathrm{I})\quad \frac{\Gamma \vdash A\ \mathtt{Type} \qquad \Gamma.A \vdash B\ \mathtt{Type} \qquad \Gamma \vdash a : A \qquad \Gamma \vdash b : B[a]}{\Gamma \vdash \langle a, b\rangle : \int_A B}$$

In order to classify the premise of ($\int$F) we simply use $\mathcal{U}.\Delta\mathcal{U}$ from Section 2.3.4.2. The premise of ($\int$I), instead, can be coded via the following nested judgement classifier



with $\gamma = \pi \circ u^* id$ from Section 2.3.4.2. The desired rule Λ, then, is the functor $(\dot{\mathcal{U}}.\Sigma\Delta\mathcal{U})\Sigma.\gamma\dot{\mathcal{U}} \to \mathcal{U}.\Delta\mathcal{U}$ appearing above.

Definition 2.3.8.2. A categorized dependent type theory *with* $\int$*-types* is a cDTT as in Definition 2.3.0.1 having two additional rules $\int$, pair such that the diagram below is commutative and the upper square is a pullback.



A cDTT with $\int$-types immediately has formation and introduction (with $\langle a, b\rangle = \mathsf{pair}(a, b)$) and, as follows from the content of Definition 2.3.7.1, the three (admittedly hard to look at) rules below. We write Λ for $\Sigma.(u.\dot{u}\Delta) \circ \Sigma.\gamma$.

$$(\textstyle\int\mathrm{E})\quad \dfrac{\Gamma \vdash c : \int_A B}{\Gamma \vdash \big(\int_A B\big)\langle c\rangle\ (\dot{\mathcal{U}}.\Sigma\Delta\mathcal{U})\Sigma.\gamma\dot{\mathcal{U}}}$$

$$(\textstyle\int\eta)\quad \dfrac{\Gamma \vdash c : \int_A B}{\Gamma \vdash c = \mathsf{pair}(\big(\int_A B\big)\langle c\rangle) : \int_A B}$$

$$(\textstyle\int\beta)\quad \dfrac{\Gamma \vdash X\ (\dot{\mathcal{U}}.\Sigma\Delta\mathcal{U})\Sigma.\gamma\dot{\mathcal{U}}}{\Gamma \vdash \mathsf{pair}(X) = \mathsf{pair}(\ (\Lambda\, X)\langle\mathsf{pair}(X)\rangle\ ) : \int\Lambda X}$$

If we break down the job of the classifier $(\dot{\mathcal{U}}.\Sigma\Delta\mathcal{U})\Sigma.\gamma\dot{\mathcal{U}}$ we recover the familiar following ones.

$$(\textstyle\int\mathrm{E}1)\quad \dfrac{\Gamma \vdash c : \int_A B}{\Gamma \vdash \pi_1 c : A} \qquad (\textstyle\int\mathrm{E}2)\quad \dfrac{\Gamma \vdash c : \int_A B}{\Gamma \vdash \pi_2 c : B[\pi_1 c]}$$

$$(\textstyle\int\eta)\quad \dfrac{\Gamma \vdash c : \int_A B}{\Gamma \vdash c = \mathsf{pair}(\pi_1 c, \pi_2 c) : \int_A B}$$

$$(\textstyle\int\beta 1)\quad \dfrac{\Gamma \vdash a : A \quad \Gamma \vdash b : B[a]}{\Gamma \vdash a = \pi_1(\mathsf{pair}(a, b)) : A} \qquad (\textstyle\int\beta 2)\quad \dfrac{\Gamma \vdash a : A \quad \Gamma \vdash b : B[a]}{\Gamma \vdash b = \pi_2(\mathsf{pair}(a, b)) : B[a]}$$

2.3.9 The internal logic of a topos – externally

We will show that the notion of *topos*, in particular, supports a dependent type theory in the sense of Section 2.3. Such a dtt recovers, among other things, the Mitchell-Bénabou language of the topos and nicely interacts with its Kripke-Joyal semantics.

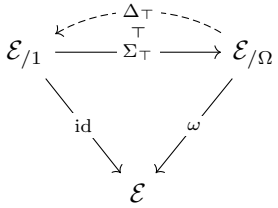
2.3.9.1 (A bit of history)**.** The internal logic of a topos has been discussed by several authors. After [MLM94], this collective humus has been crystallized in the Mitchell-Bénabou language and its *tautological* interpretation, the Kripke-Joyal semantics. These attributions are somewhat symbolic. For what concerns the Mitchell-Bénabou language, the best historical account is given, to our knowledge, by Johnstone [Joh77]. After Mitchell's original contribution [Mit72], Johnstone

refers to the unfindable [Cos72] for Bénabou's contribution, but the paper is actually authored by Coste. [Osi75a] and others were definitely part of the intellectual debate on the topic. For what concerns the Kripke-Joyal semantics the situation is much more cloudy, Osius [Osi75b] tells us that the original ideas from Joyal were never published, while a footprint of Joyal's contribution to the topic only emerges (in French) in [BJ81]. These ideas were later conveyed in several texts with slight variations, like [LS88] and [Bor94]. Both in the case of the language and its semantics, we will refer to the presentation in [MLM94, VI, Sec. 5 and 6] which is in a sense the most informal and essential. Our main objective is to demonstrate that our formalism can reboot the core ideas behind the Mitchell-Bénabou language. We will not discuss in detail Kripke-Joyal semantics, even though the connection could be drawn, as exemplified by the very recent [AGH21]. Finally, if one wanted to strip down which property of a topos corresponds to which portion of the logic, we point the reader to [Mai05].

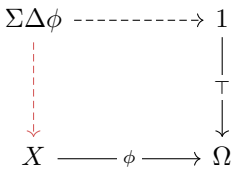
Definition 2.3.9.2 (Topos)**.** A topos $\mathcal{E}$ is category with finite limits, exponentials, and a subobject classifier.

Let us now fix the topos $\mathcal{E}$ and denote with $\top\colon 1 \to \Omega$ its subobject classifier.

Construction 2.3.9.3 (The dtt of an elementary topos)**.** For an an elementary topos $\mathcal{E}$, we can construct a dependent type theory in the sense of Definition 2.3.0.1 as follows.



The map $\Sigma_\top$ is induced (via precomposition) by the map $\top : 1 \to \Omega$ which picks the top-element of Ω. $\Delta_\top$ is given by pullback, and of course the whole discussion fits perfectly with Theorem 2.3.1.2, with the technical advantage that the presheaves in this case are internally represented by objects in the topos, thus there is no need to use the Yoneda embedding.



2.3.9.4 (Comprehension category, display maps, monomorphisms)**.** Of course, at this point the whole content of Section 2.3 applies, and thus we can load a whole judgement calculus for this dependent type theory. For example, the rule

$$(\Delta)\quad \frac{\Gamma \vdash \phi\ \mathcal{E}_{/\Omega}}{\Delta\phi \vdash \Delta\phi\ \mathcal{E}_{/1}}$$

is telling us that to each proposition $\phi : \Gamma \to \Omega$, corresponds an object $\Delta\phi$, which is precisely the object supporting the subobject of X classified by ϕ. Similarly,

following the Construction 2.3.2.1, we obtain a representation of the internal logic of the topos in terms of a comprehension category (3.3.1.1),

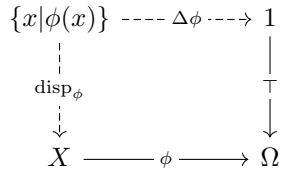
$$\mathrm{disp} : \mathcal{E}_{/\Omega} \to \mathcal{E}^2.$$

Such correspondence maps a formula ϕ to the dashed red arrow in the construction above. It follows that the correspondence maps a proposition to its zero locus, i.e. the monomorphism whose characteristic function is precisely ϕ. Of course, this idea is not novel and it dates back to Taylor's PhD thesis or his more recent [Tay99].

2.3.9.5 (Mitchell-Bénabou reloaded)**.** Following [MLM94, VI Sec 5] we see that there is a canonical dictionary between our judgements classified by $\mathcal{E}_{/\Omega}$ and *formulae*, i.e. terms of type Ω in the sense of [MLM94, pag. 299, right after the bulleted list]. Moreover, and somewhat most importantly, display maps construct subobjects as zero locus of formulae, as explained in [MLM94, pag. 300, right after the bulleted list].

$X \vdash \phi\ \mathcal{E}_{/\Omega}$	$\phi(x)$
$X \vdash \Delta_\top \phi\ \mathcal{E}_{/\top}$	$\{x \mid \phi(x)\}$

Notice the difference between $\Delta\phi$ and disp_ϕ, even though they might seem to be similar things, the first one gives us the support of the subobject, while the second one gives us the subobject itself.

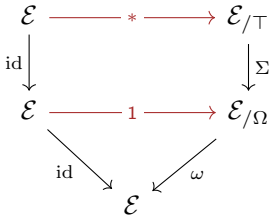


As a result of this discussion, one can use the judgement calculus produced by this dependent type theory to simulate the internal logic of the topos, and the result will be consistent with the Mitchell-Bénabou language of the topos.

Let us give a few examples. Notice that we chose topoi as a very strong theory, but in fact Lemma 2.3.9.6, Lemma 2.3.9.7 show the *modularity* of our approach, in a fashion very much affine to [Mai05].

Lemma 2.3.9.6. The cDTT induced by a topos $\mathcal{E}$ has unit types in the sense of Definition 2.3.8.1.

Proof. It suffices to show that we have functors $*$ and 1 making the following diagram commute and the square a pullback.



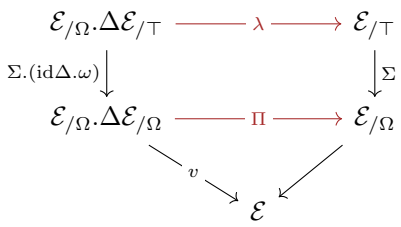
Let us denote $!_X$ the unique map from X to the terminal – for the moment, elsewhere we have and we will use X both for the object and the map to 1. One can easily check that defining $*\colon X \mapsto !_X$, and in the obvious way on morphisms, and $1\colon X \mapsto \top \circ !_X$, and in the obvious way on morphisms, does the job. □

Lemma 2.3.9.7. The cDTT induced by a topos $\mathcal{E}$ has extensional identity types in the sense of Definition 2.3.6.1.

This can be proved in similarly as in Lemma 2.3.9.6, using equalizers. We take a bit of care in proving the following, instead.

Lemma 2.3.9.8. The cDTT induced by a topos $\mathcal{E}$ has dependent product types in the sense of Definition 2.3.5.1.

Proof. It suffices to show that we have functors λ and Π making the following diagram commute and the square a pullback.



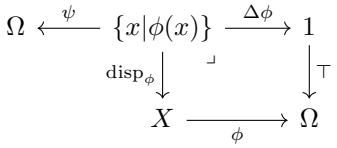
Let us first compute the two categories

$$\mathcal{E}_{/\Omega}.\Delta\mathcal{E}_{/\Omega} \quad \text{and} \quad \mathcal{E}_{/\Omega}.\Delta\mathcal{E}_{/\top}\ .$$

Following the construction in Section 2.3.4.2, we can see that they respectively have objects

$$(\phi, \psi) \quad \text{and} \quad (\phi, \{x|\phi(x)\})$$

with ϕ, ψ as below.



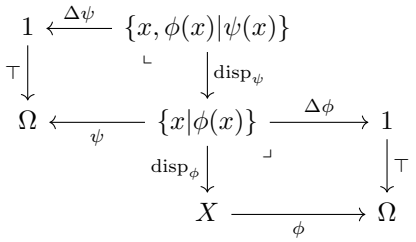
The verical map on the left hand side of the square computes the diagonal of the pullback square above, meaning it acts as $(\phi, \{x|\phi(x)\}) \mapsto (\phi, \phi \circ \mathrm{disp}_\phi)$.

To provide suitable Π, λ we of course look at right adjoints to pullback functors. The fact that they reasonably model dependent products has been widely discussed from the publication of [See84], with distinguished treatments in [CZ21], where an explicit construction is given, and in [Mai05], where it is better framed in the context of the different properties of a topos and their logical counterpart.

One can always show that for a given $\phi\colon X \to \Omega$ (and, in fact, for any $f\colon X \to Y$), we have the following equivalence and adjunction,

$$\mathcal{E}_{/X} \cong (\mathcal{E}_{/\Omega})_{/\phi} \underset{\Pi_\phi}{\overset{\phi^*}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathcal{E}_{/\Omega}$$

see for example [MLM94, IV.7]. Given a pair (ϕ, ψ) in $\mathcal{E}_{/\Omega}.\Delta\mathcal{E}_{/\Omega}$, then, it is natural to compute disp_ψ,



and define $\Pi(\phi, \psi) = \Pi_\phi(\mathrm{disp}_\phi \circ \mathrm{disp}_\psi)$. As for λ, we put $\lambda(\phi, \{x|\phi(x)\}) = \{x|\phi(x)\}$.

The square involving Π, λ commutes because $\{x, \phi(x)|\phi(x)\} = \{x|\phi(x)\}$ hence the composition of displays is mapped to the trivial triangle $\mathrm{disp}_\phi \colon \phi\mathrm{disp}_\phi \to \phi$. The universal property of Π_ϕ is what guarantees that the domain of $\Pi(\phi, \psi)$ is, in fact $\{p|\Pi(\phi, \psi)(p)\}$. From this remark, one can immediately show that the desired square is a pullback. □

This is nothing new, but we believe it provides a different perspective on the internal logic of a topos (or any category, really). This is really close to the following intuition.

> We can then conclude that describing the internal dependent type theory of a category means to capture the type-theoretic properties of the codomain fibration, while describing the internal many-sorted logic of a category – considering the sorts as types – means to capture the properties of the subobject fibration together with the one-dimensional structure of the category under consideration. [Mai05]

In a sense, our work is about extending this process to more than just the codomain fibration.

2.3.9.9 (Other topoi)**.** Of course, Construction 2.3.9.3 also means that whenever we have something of the kind, we can interpret a categorized dependent type theory with characteristics similar to (at least in the structural rules part) the theory of a topos. This could have applications both to topoi in the sense suggested in [MP89, 6.4], and to elementary 2-topoi such as in [Web07], the key example of which is the 2-category of categories. We leave this to future work.

2.4 Categorized first-order logic

In this section we design the categorized judgemental theory that performs the calculus of natural deduction. As for Section 2.3, we introduce the basic judgements and rules and show how they generate the desired structure, then we add more rules to perform additional computations. Though we follow the path of the well-known fibrational approach to first order logic, we spend some time in re-developing it in the context of judgemental theories: this is meant to present the benefits of the judgemental approach, to compare the resulting structure with that of dependent types, and to give a pedagogical example of

how one might want to implement a judgemental theory starting from notions which are known to be fibrational in nature.

Disclaimer 2.4.0.1 (Why we do not start from dependent type theory)**.** As we discussed in Section 2.2.1.2, one could very well follow [Mar75] and use Section 2.3 as a starting point for this analysis by simply restricting it to the proof-irrelevant case. This is what is really happening in Definition 2.4.0.3, but we choose to recover the whole theory from scratch for two reasons: on one hand, we hope that it makes the present work accessible to the non-(type theorist), or to someone who is more familiar with traditional first-order logic; on the other we aim to more swiftly align to the tradition of doctrines [Law70, Pit83, Mak93, MR13].

Again, as explained in 2.2.1.2, our distinction is mathematically artificial, and we will remark that throughout our discussion, see for example 2.4.3.5.

Disclaimer 2.4.0.2 (Why we do not do Gentzen's sequent calculus)**.** On the other hand, we could have chosen to present first-order logic in the formalism of sequent calculus in [Gen35]. Though our framework allows us for it – and in fact many of the categorical constructions in the following section do so, already, starting from Definition 2.4.0.3 – we have chosen to take the perspective of natural deduction because on one hand we believe that, being closer to how logic is used makes it easier to follow what each categorical operation is doing and, secondly, dealing with connective and quantifiers with pairs of introduction/elimination rules, as opposed to right/left introduction rules, helps to keep the connection with dependent types (2.4.0.1) in the back of the reader's mind.

Definition 2.4.0.3 (Categorized natural deduction theory)**.** A *categorized natural deduction theory* is a substitutional (2.1.1.1) categorized judgemental theory ($\mathbf{ctx}, \mathscr{J}, \mathscr{R}, \mathscr{P}$)such that

- $\mathbf{ctx}$ is $\mathbf{Fin}$, the category of finite sets;
- $\mathscr{J}$ can be presented by *one* judgement classifier $p : \mathcal{P} \to \mathbf{ctx}$, which is a faithful fibration, has fibered products and implication, and has fibered initial objects.

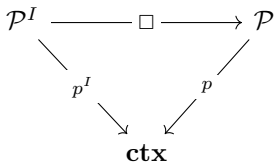
We think of $\mathbf{ctx}$ as the category of *variables and terms* and of $\mathcal{P}$ as the category of *well-formed formulae* fibered over variables. We call this cNDT for short.

Remark 2.4.0.4 (On cardinality)**.** We can define λ-ary theories but we would need to close judgemental theories under λ-small limits, and we would have to replace Fin with the category of λ-small sets.

Construction 2.4.0.5 (From doctrines to classifiers)**.** Let $P : \mathbf{ctx}^{\mathsf{op}} \to \mathbf{Pos}$ be a *doctrine*, intended in the most non-committal sense. Consider $\square : P^I \to P$ any operational property/structure on P, e.g.:

- having (finite) fibered meets $\wedge : P^I \to P$;
- having (finite) fibered joins $\vee : P^I \to P$;
- having a negation operator $\neg : P \to P$.

then, by the Grothendieck construction, we obtain some corresponding diagram of fibrations,



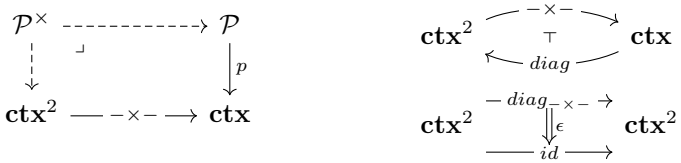
This produces a categorized pre-judgemental theory obtained by $\mathcal{P}$, together with all its structural operators. For example, if P in an Heyting algebra fiber-wise, we have operators $\bot, \top, \wedge, \vee, \Rightarrow$ of the proper arities on $\mathcal{P}$.

Construction 2.4.0.6 (The arrow category)**.** Consider a cNDT. Because it is closed under finite powers (2.1.0.11) we can compute

$$\mathcal{P}^2 \longrightarrow \mathcal{P}$$

and we will show how the operations defined on $\mathcal{P}$ lift to (a suitable subcategory of) $\mathcal{P}^2$ thanks to the closure under finite limits. We will come back to this in Section 2.4.2.

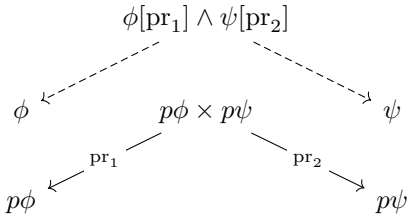
Construction 2.4.0.7 (Weakening)**.** Since p has fibered products we can compute the following nested judgement (on the left)



with an adjunction $diag \dashv - \times -$ whose counit computes projections. These are well known in the literature and perform what is usually called *weakening*:

$$x \times y \vdash x \ \mathbf{ctx}\,.$$

We can now define a span $(p^2, p^\times)$ out of $\mathcal{P}^2$ (as a category, not as the fibration $\mathcal{P} \times \mathcal{P}$) with $p^\times : (\phi, \psi) \mapsto (\phi[\mathrm{pr}_1] \wedge \psi[\mathrm{pr}_2])$ the product of the respective cartesian lifts of ϕ, ψ along $\mathrm{pr}_1, \mathrm{pr}_2$,



and such a span makes the diagram involving $\mathcal{P}^\times$ commute, therefore we have a unique rule

$$w : \mathcal{P}^2 \to \mathcal{P}^\times$$

over $\mathbf{ctx}^2$. If $p(\phi, \psi) = (x, y)$ we might denote $w(\phi, \psi) = w_y\phi \wedge w_x\psi$.

Definition 2.4.0.8 (cNDT with weakening)**.** A cNDT is said to *have weakening* if for each $y \in \mathbf{ctx}$, $- \times y$ is in $\mathscr{J}$.

In this section we will show that, in fact, a cNDT produces the calculus of natural deduction.

2.4.1 Dictionary

As we did in Section 2.3, we declare a local dictionary, both to make the chapter more comprehensible and to account for classical notation.

Notation 2.4.1.1 (Stratified contexts)**.** Notice that already in Construction 2.4.0.7 we follow the intuition and use $x, y, \dots$ to name objects of $\mathbf{ctx}$. In fact, we here want to give a way to present judgements that are traditionally of the form

$$x; \Gamma \vdash \psi$$

so that they read as having two contexts: the free variables in the formulae *and* the formula(e) in the premise of the sequent. In fact, we will "stack up" two fibrations so that the objects living on top ($\Gamma \Rightarrow \psi$) are both fibered on those in the middle (Γ) and those on the bottom (x). We hope to make it all clearer in the table that will follow.

Construction 2.4.1.2 (Entailment)**.** We wish to represent entailment between two formulae in the same context. In order to do that we pick in $\mathcal{P}^2$ all objects belonging to the same fiber. Call $I : \mathbf{ctx} \to \mathbf{ctx}^2$ the functor mapping $\Gamma \mapsto id_\Gamma$ and compute the following (dashed) limit.

$$\begin{array}{ccc} \mathcal{P}^2 I.p^2\mathbf{ctx} & \dashrightarrow & \mathbf{ctx} \\ \downarrow & \lrcorner & \downarrow I \\ \mathcal{P}^2 & \xrightarrow{p^2} & \mathbf{ctx}^2 \\ {\scriptstyle \mathrm{cod}}\downarrow\downarrow{\scriptstyle \mathrm{dom}} & & {\scriptstyle \mathrm{cod}}\downarrow\downarrow{\scriptstyle \mathrm{dom}} \\ \mathcal{P} & \xrightarrow{p} & \mathbf{ctx} \end{array}$$

Both the pullback and all universal arrows belong to the judgemental theory. The classifier we are interested in is the composition $\mathcal{P}^2 I.p^2\mathbf{ctx} \to \mathbf{ctx}$, and will simply denote it with $e : \mathcal{E} \to \mathbf{ctx}$.

Remark 2.4.1.3. The Γ appearing in Notation 2.4.1.1 indicates a finite set of formulae in context x. We can see it as a product in $\mathcal{P}$ and, when we want to do so, we will write ϕ_Γ.

We are finally ready to declare our local dictionary according to the notation of Section 2.2.3, and that is the following.

$x \vdash e\ \mathcal{E} \quad x \vdash (\mathrm{dom} \circ I.p^2)(e) =_{\mathcal{P}} \phi_\Gamma \quad x \vdash (\mathrm{cod} \circ I.p^2)(e) =_{\mathcal{P}} \psi$	$x; \Gamma \vdash \psi$
$x \vdash e\ \mathcal{E} \quad x \vdash (\mathrm{dom} \circ I.p^2)(e) =_{\mathcal{P}} \phi_\Gamma \wedge \phi \quad x \vdash (\mathrm{cod} \circ I.p^2)(e) =_{\mathcal{P}} \psi$	$x; \Gamma, \phi \vdash \psi$

Remark 2.4.1.4. Consider that in the case that ($x; \bot \vdash \phi\ \mathcal{P}$ and $x \vdash \phi \to \psi\ \mathcal{E}$) then $x \vdash \phi \Rightarrow \psi\ \mathcal{P}$, therefore our framework accounts for the classical correspondence for all x, ϕ, ψ,

$$x \vdash \phi \Rightarrow \psi \quad \text{iff} \quad x; \phi \vdash \psi.$$

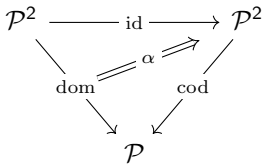
Remark 2.4.1.5. Since each p-fiber is thin, there is at most one $e \in \mathcal{E}$ between each pair of objects $(\phi, \psi) \in \mathcal{P} \times \mathcal{P}$.

Notation 2.4.1.6. In order to make our calculations more readable, we pinpoint a specific notation for $\mathrm{cod}, \mathrm{dom}$ in the case that they follow the inclusion of $\mathcal{E}$ into $\mathcal{P}^2$. We creatively write c and d, respectively.

2.4.2 From properties to rules

Before we begin our analysis of rules of natural deduction, we show how certain properties lift from $\mathcal{P}$ (the category) to $\mathcal{E}$ (the judgement classifier). These will be instrumental in building up rules from p. In a sense, this subsection shows how to turn *internal properties of* p into *external rules about* p, which is precisely what we did for contexts in Example 2.1.0.3.

Remark 2.4.2.1 (The domain-codomain policy)**.** Since we will frequently use either c or d to select the consequent or the antecedent of a sequent, it will be useful to have a way to relate the two. The proof of Lemma 2.4.2.2 is clear evidence in this sense. There is a trivial policy

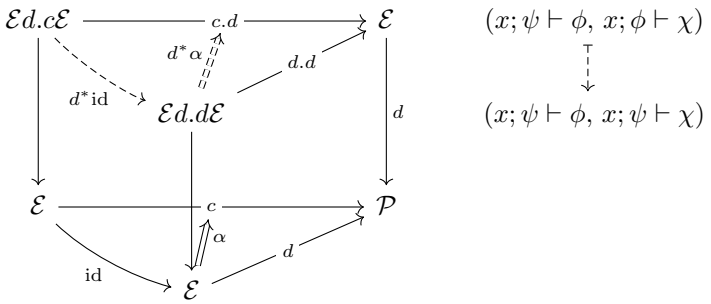


where $\alpha_{\psi\to\phi} = (\psi \to \phi)$. Note that if useful we might bravely invert the direction of id. We call α, too, the obvious whiskering $d \Rightarrow c$.

Lemma 2.4.2.2 (A special instance of cut)**.** The relation captured by $e : \mathcal{E} \to$ **ctx** is transitive in the sense that the rule below is in the cNDT.

$$\text{(T)}\quad \frac{x;\psi \vdash \phi \qquad x;\phi \vdash \chi}{x;\psi \vdash \chi}$$

Proof. Consider the following $\sharp$-lifting of the triangle in Remark 2.4.2.1 along d.



A little computation shows that the upper triangle reads as on the right, producing the desired rule

$$t := d.d \circ d^*\mathrm{id} : \mathcal{E}d.c\mathcal{E} \to \mathcal{E}.$$

□

Lemma 2.4.2.3 (Preservation through product as a rule)**.** Interaction of arrows and products in $\mathcal{P}$ (the category) lifts to $\mathcal{E}$ (the judgement classifier) in the sense that the rule below is in the cNDT.

$$\text{(F)}\quad \frac{x;\psi \vdash \phi \qquad x;\bot \vdash \chi}{x;\psi \wedge \chi \vdash \phi \wedge \chi}$$

Proof. It is coded by a functor $f : \mathcal{E} \times \mathcal{P} \to \mathcal{E}$ which to pairs $(\psi \to \phi, \chi)$ over some x assigns the unique map $\psi \wedge \chi \to \phi \wedge \chi$ defined via the universal property of the product in the fiber over x. □

2.4.3 Formal structural rules

Here we show that an cNDT generates the following formal structural rules.

$$\text{(H)}\ \frac{}{x;\Gamma,\phi\vdash\phi} \qquad \text{(Sw)}\ \frac{x;\Gamma,\Delta\vdash\phi}{x;\Delta,\Gamma\vdash\phi} \qquad \text{(C)}\ \frac{x;\Gamma,\psi,\psi\vdash\phi}{x;\Gamma,\psi\vdash\phi}$$

$$\text{(W)}\ \frac{x;\Gamma\vdash\phi}{x;\Gamma,\psi\vdash\phi} \qquad \text{(Cut)}\ \frac{x;\Gamma\vdash\phi \qquad x;\Gamma,\phi\vdash\psi}{x;\Gamma\vdash\psi}$$

We break the discussion into three parts.

2.4.3.1 Hypothesis and the simple fibration

Clearly for each pair (Γ,ϕ) over the same context, we have that $\mathrm{pr}_2 : \phi_\Gamma\wedge\phi\to\phi$, therefore $\mathcal{E}$ classifies $x;\Gamma,\phi\vdash\phi$.

(H) The Hypothesis rule, then, is coded into the existence of e itself.

We would be content with this already, but it is worth noticing that the association performing the projection pr_2

$$(\Gamma,\phi)\mapsto(\phi_\Gamma\wedge\phi\to\phi)$$

can be described functorially, and it contains some profound information. Such functor, in fact, provides an insight into possible developments of the present work, plus it (almost) allows for a presentation of the *simple fibration* from [Jac99], which has a meaningful logical interpretation: it constitutes the "least informative" type theory one can observe over a category with finite products. Therefore we say a little more about that.

Construction 2.4.3.1 (The simple fibration)**.** Define on the category $\mathcal{E}$ the monad comprised of the following data:

- the functor $S:\mathcal{E}\to\mathcal{E}$ acting as follows

$$\psi\to\phi \quad\mapsto\quad \psi\wedge\phi\to\psi\to\phi;$$

- the 2-cell $\eta : Id\Rightarrow S$ defined via the universal property of products;
- the 2-cell $\mu : S\circ S\Rightarrow S$ acting as $(\mathrm{pr}_1,\mathrm{id})$.

Remark 2.4.3.2. (S,η,μ) is idempotent. This is because μ acts as follows

$$\begin{array}{ccccccc}
(\psi\wedge\phi)\wedge\phi & \longrightarrow & \psi\wedge\phi & \longrightarrow & \psi & \longrightarrow & \phi \\
\downarrow{\scriptstyle \mathrm{pr}_1} & & & & & & \downarrow{\scriptstyle \mathrm{id}} \\
\psi\wedge\phi & & \longrightarrow & & \psi & \longrightarrow & \phi
\end{array}$$

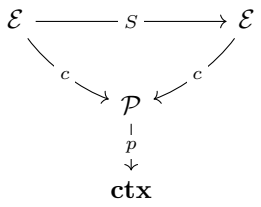
and $(\psi\wedge\phi)\wedge\phi=\psi\wedge\phi$ because p is thin and its products are fibered, and in fact the forgetful functor from algebras over S into $\mathcal{E}$ is fully faithful. All S-algebras are free.

The Kleisli category of S is equivalent to (the total category) of what in [Jac99] is called the *simple fibration* associated to p. That is $\overline{p} : s(\mathcal{P})\to\mathcal{P}$ where $s(\mathcal{P})$ has for objects pairs (ϕ,ϕ') in the same p-fiber and maps $a=(a_1,a_2)$:

$(\phi, \phi') \to (\psi, \psi')$ such that $a_1 : \phi \to \psi$, $a_2 : \phi \wedge \phi' \to \psi'$, and $p(a_1) = p(a_2)$. The functor $\overline{p}$ acts as the first projection. If we call $1 : \mathbf{ctx} \to \mathcal{P}$ the (fibered) terminal object functor, one checks that $\overline{p}.1 \cong p$. Moreover, the functor

$$q : \ (\phi, \phi') \ \mapsto \ (\phi \wedge \phi' \to \phi)$$

induces a comprehension category (as in Construction 2.3.2.1) $s(\mathcal{P}) \to \mathcal{P}^{\to}$. The type theory associated to such a functor is (that equivalent to) untyped lambda calculus.
The functor S induces the following rule.



$$\text{Dictionary in 2.4.1}\ \frac{x;\Gamma \vdash \phi}{x \ \vdash \phi_\Gamma \to \phi \ \mathcal{E}}$$
$$\text{(S)}\ \frac{}{x \vdash S(\phi_\Gamma \to \phi)\ \mathcal{E}}$$
$$\text{Construction 2.4.3.1}\ \frac{}{x \vdash \phi_\Gamma \wedge \phi \to \phi\ \mathcal{E}}$$
$$\text{Dictionary in 2.4.1}\ \frac{}{x;\Gamma, \phi \vdash \phi}$$

2.4.3.2 Swap and Contraction: fibered products everywhere

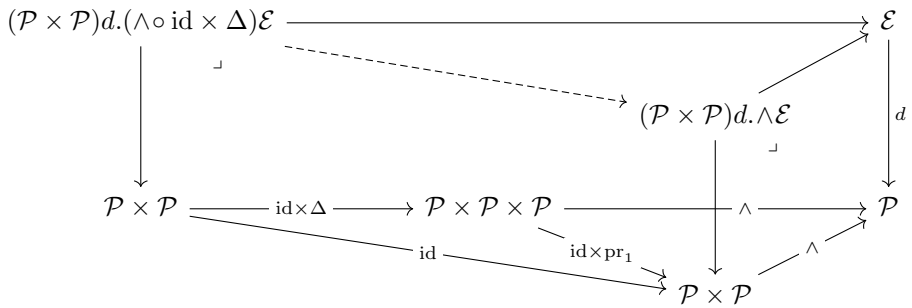
(Sw) The Swap rule holds because the fibered product is symmetric and this too is expressed via a commutative triangle: consider the following composition

$$\mathcal{P} \times \mathcal{P} \xrightarrow{\widetilde{s}} \mathcal{P} \times \mathcal{P} \xrightarrow{\wedge} \mathcal{P}$$

where the map s computes the permutation. The desired rule is computed as the (iso)morphism

$$s.(d.\wedge) : (\mathcal{P} \times \mathcal{P}).\mathcal{E} \to (\mathcal{P} \times \mathcal{P}).\mathcal{E}.$$

(C) Contraction is supported by the following dashed map



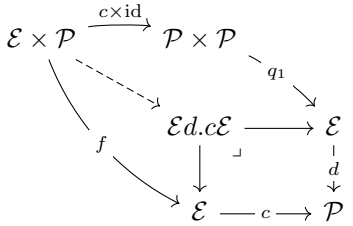
where we write $\wedge$ for the obvious product $\mathcal{P} \times \mathcal{P} \times \mathcal{P} \to \mathcal{P}$. On the bottom we have the triangle on the left commuting, and $\mathrm{id} \times \Delta$ equalizing $\wedge$ and $\mathrm{id} \times \mathrm{pr}_1 \circ \wedge$. The two nested judgements on the top classify, respectively, the antecedent and the consequent of (C), and the dashed map exists by the universal property of the "smaller" pullback.

2.4.3.3 Weakening and Cut: more transitivity

We will see that to provide both Weakening and Cut it is sufficient to apply (T) from Lemma 2.4.2.2 to appropriate triples. Let us start with (W), first: notice that, as it happened in Section 2.3 and is evident from Section 2.4.1, the consequent in (W) is actually silent of (at least) one judgement, that is $x; \bot \vdash \psi$. The procedure we follow for (W) is that of

$$\text{(F)}\ \frac{x;\Gamma\vdash\phi \qquad (\,x;\bot\vdash\psi\,)}{\text{(T)}\ \dfrac{x;\Gamma,\psi\vdash\phi\wedge\psi}{x;\Gamma,\psi\vdash\phi}} \qquad \frac{(x;\Gamma\vdash\phi \qquad x;\bot\vdash\psi\,)}{x;\phi\wedge\psi\vdash\phi}$$

therefore we need to pre-process the premise of t in order to apply it to triples of the form $(\phi_\Gamma \wedge \psi, \phi \wedge \psi, \phi)$. This is achieved via the following diagram



with q_1 the map $(\phi, \psi) \mapsto (\phi \wedge \psi \to \phi)$ acting on pairs in the same p-fiber. Notice how this is related to q in Construction 2.4.3.1.

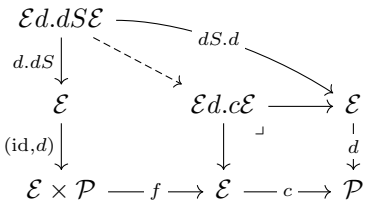
Remark 2.4.3.3 (Cones and branches)**.** Here branches in the tree of a deduction correspond to cones over limit diagrams. We could make this statement more precise, but we hope the following discussion speaks for itself.

(W) Weakening is computed by the dashed arrow above followed by t from Lemma 2.4.2.2.

For (Cut) we again apply Lemma 2.4.2.2, this time to the triple $(\phi_\Gamma, \phi_\Gamma \wedge \phi, \psi)$, that is we will build the diagram corresponding to the following composing rules.

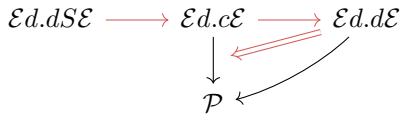
$$\text{(T)}\ \frac{\dfrac{x;\Gamma\vdash\phi \quad x;\Gamma,\phi\vdash\psi}{x;\Gamma\vdash\phi}}{x;\Gamma\vdash\phi_\Gamma\wedge\phi} \qquad \frac{\dfrac{x;\Gamma\vdash\phi \quad x;\Gamma,\phi\vdash\psi}{x;\Gamma,\phi\vdash\psi}}{x;\phi_\Gamma\wedge\phi\vdash\psi}}{x;\Gamma\vdash\psi}$$

Therefore we want a map from $\mathcal{E}d.dS\mathcal{E}$, classifying the premise of Cut, into $\mathcal{E}d.c\mathcal{E}$ so that we then can apply (T). That is achieved as follows.

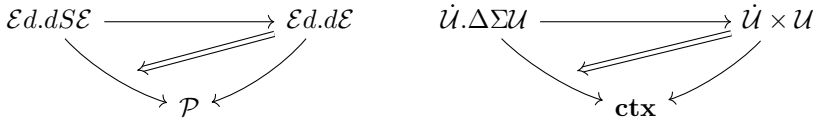


(Cut) Cut is computed by the dashed arrow above followed by t from Lemma 2.4.2.2.

Remark 2.4.3.4 (Cut is a policy)**.** While perhaps not evident, the Cut rule is in fact a policy in the sense of Definition 2.1.0.1: it just preprocesses data going into the policy (T) from Lemma 2.4.2.2.



Remark 2.4.3.5 (Relationship between DTy and Cut)**.** As we have seen, at the very core of Cut sits the policy (T) from Lemma 2.4.2.2. The reader will notice the incredible similarity between the process that constructs (T) and the process that constructs (DTy).



Not only do the diagrams in Lemma 2.4.2.2 and in Section 2.3.4.3 look very similar, but even their ingredients have affine logical meaning. Indeed, in both cases the hypothesis of the policy is a nested judgement where a modality appears: in the case of natural deduction, this is the monad S, in the case of dependent type theory it is the monad $\Delta\Sigma$. Of course, some delicate differences appear too[2]. This kind of thoughts could lead to a general notion of *cut*, a special family of policies, but we leave such a task for a possible future work.

2.4.4 Formal rules for connectives

We here show that the moment we ask that connectives are closed under p-fibers, with p a cNDT, we automatically get the expected rules. Since Definition 2.4.0.3 already contains the requirement that p has fibered products, we here show how to provide in a cNDT rules for $\wedge$, and need to ask nothing more of it. If the reader inspects the constructions below, they will see that such a procedure could be repeated for cNDTs having p *additionally* equipped with $\vee, \neg$.

The ones which are usually required for $\wedge$ are the following.

$$(\wedge\text{I})\ \frac{x;\Gamma \vdash \phi \qquad x;\Gamma \vdash \psi}{x;\Gamma \vdash \phi \wedge \psi} \quad (\wedge\text{E1})\ \frac{x;\Gamma \vdash \phi \wedge \psi}{x;\Gamma \vdash \phi} \quad (\wedge\text{E2})\ \frac{x;\Gamma \vdash \phi \wedge \psi}{x;\Gamma \vdash \psi}$$

($\wedge$I) Introduction is represented by the functor $conj : \mathcal{E}d.d\mathcal{E} \to \mathcal{E}$ induced by the universal property of the fibered product.

($\wedge$E1) In order to represent its domain, we compute the equalizer

$$\mathcal{E}(d.d, conj) \dashrightarrow \mathcal{E}d.d\mathcal{E} \underset{conj}{\overset{d.d}{\rightrightarrows}} \mathcal{E}$$

where by $d.d$ (sadly ambiguous, in this case) we wish to express the first projection of the pullback. The desired rule is then induced by the first product projection and it assumes the following form.

$$\mathcal{E}(d.d, conj) \to \mathcal{E} \to \mathcal{E}$$

[2]For example the *height* at which one performs the action of the monad, that is $\dot{\mathcal{U}}$ and $\mathcal{U}$ are manipulated over contexts while both instances of $\mathcal{E}$ are bounded to formulae.

(∧E2) Dually, we compose the equalizer with the functor induced by the second projection.

Definition 2.4.4.1 (Heyting and Boolean cNDTs)**.** A cNDT is said to

- be *Heyting* if we have operators $\bot, \top, \wedge, \vee, \Rightarrow$ of the proper arities on p;
- be *Boolean* if it is Heyting and, being $\neg := (\text{-}) \Rightarrow \bot$, the morphism of fibrations $\neg\neg$ is equivalent to id.

2.4.5 Substitution

While in Section 2.3 we thought of morphisms of contexts as substitutions, in the setting of proof theory we regard them as terms. When we write a map

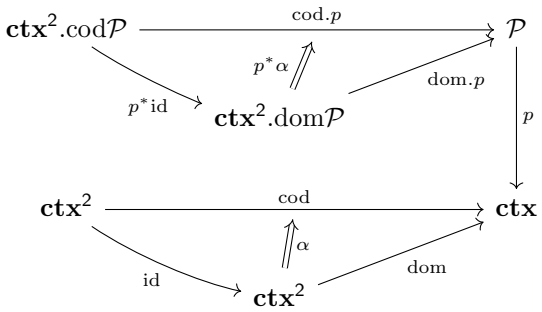
$$y \to x$$

we see it as a list of terms and denote it as such:

$$[t/x] : y \to x.$$

In particular, if $x = x_1 \times \cdots \times x_k$, each term $t_i = \mathrm{pr}_i \circ t$ is a term built up from y and in context x_i, with $i = 1, \ldots, k$. Then we can identify $\mathbf{ctx}_{/x}$ with the classifier collecting *all* terms in context x.

All of this belongs to the intuition and in fact there is nothing more to **ctx** than what described in Definition 2.4.0.3, but it is with this perspective that we now look at how substitution behaves in cNDTs. Recall from Section 2.2.5 that substitutionality allows us to compute rules and policies as the following



$$\frac{x \vdash \phi\ \mathcal{P}}{y \vdash \phi[t/x]\ \mathcal{P}}$$

and here we have only blindly expanded the information contained in the diagram on the left by following the discussion in 2.2.5.

2.4.6 Formal rules for quantifiers

Finally, we wish to give an account of quantifiers, hence we introduce more structure on p and on the categorized judgemental theory it generates.

Definition 2.4.6.1 (First order cNDTs)**.** A cNDT is said to

- be *intuitionistic first order* if it has weakening (Construction 2.4.0.7), is Heyting and w from Construction 2.4.0.7 has left and right adjoints,

$$\exists \dashv w \dashv \forall,$$

where $\exists, \forall$ are morphisms of fibrations and belong to $\mathscr{J}$; we call such theories IcFOTs, for short;

- is *classical first order* if it is intuitionistic first order and also Boolean; we call these cFOTs for short.

We believe that the request of being morphisms of fibrations (i.e. preserve cartesian squares) is related to the more traditional properties required for $\forall, \exists$, namely Frobenius reciprocity and Beck-Chevalley.

Remark 2.4.6.2. Consider a (intuitionistic) first order theory in the traditional sense. Then it induces a (I)cFOT: (the fibration associated to) the hyperdoctrine of Lindenbaum-Tarski algebras of well-formed formulae, which we talked about in Example 1.7.1.2.

We only provide explicit representation of the rules involving $\forall$ in the IcFOT, $\exists$ could be worked out in a similar fashion. First of all, notice that the pair of adjoint functors $w \dashv \forall$ induces (via the hom-set isomorphism) the following rule (on the left)

$$\text{(FA)}\quad \frac{x \times y \vdash w(\phi,\psi) \leq \chi \ \mathcal{P}}{(x,y) \vdash (\phi,\psi) \leq \forall\chi \ \mathcal{P}\times\mathcal{P}} \qquad \frac{x \times y \vdash w_y\phi \wedge w_x\psi \leq \chi \ \mathcal{P}}{x \vdash \phi \leq \forall_y\chi \ \mathcal{P} \qquad y \vdash \psi \leq \forall_x\chi \ \mathcal{P}}$$

which, if we denote $\forall_y = \mathrm{pr}_1 \circ \forall$ and $\forall_x = \mathrm{pr}_2 \circ \forall$, amounts to the rule on the right. The two rules we need to produce are the following.

$$\text{(}\forall\text{I)}\quad \frac{x \times y; w_y\Gamma \vdash \phi}{x;\Gamma \vdash \forall_y\phi} \qquad \text{(}\forall\text{E)}\quad \frac{x;\Gamma \vdash \forall_y\phi}{x;\Gamma \vdash \phi[t/y]}$$

Notice that we included the writing $w_y\Gamma$ (with w_y of the kind described in Construction 2.4.0.7) to express the desired dependency, since in this case we wish to say that there is no y free in Γ. Also, writing $\phi[t/y]$ is a bit improper in the sense that, since $p(\phi) = p(\phi_\Gamma) \times y = x \times y$, each substitution in ϕ should have codomain $x \times y$. It is clear what happens here, but we will go into detail when the time comes.

We begin with Introduction. It does actually pretty much read as the fact that $\forall$ is right adjoint to w "at" the triple $(\,(\phi_\Gamma, \phi_\Gamma),\, \phi\,)$, but if we wish to write a rule in the sense of Definition 2.1.0.1, we shall start computing the premise, which we do via the following pullback

$$\begin{array}{ccc} \mathcal{P}(Pr_1\, p.\times).p\mathcal{P}^\times & \longrightarrow & \mathcal{P}^\times \\ \downarrow & \lrcorner & \downarrow{\scriptstyle p.\times} \\ & & \mathbf{ctx}^2 \\ & & \downarrow{\scriptstyle Pr_1} \\ \mathcal{P} & \xrightarrow{\quad p \quad} & \mathbf{ctx} \end{array}$$

which classifies pairs $(x \vdash \Gamma,\ x \times y \vdash \phi)$. But now we exploit the fact that

$$w_y\Gamma \leq \phi \quad \text{iff} \quad w_y\Gamma \wedge \phi = w_y\Gamma$$

so we ask of the equalizer of the maps

$$\mathcal{P}(Pr_1\, p.\times).p\mathcal{P}^\times \hookrightarrow \mathcal{P}^2 \underset{(Pr_1,Pr_1)}{\overset{\mathrm{id}}{\rightrightarrows}} \mathcal{P}^2 \xrightarrow[p^\times]{} \mathcal{P}$$

with the top one computing $w_y\Gamma \wedge \phi$ and the bottom one $w_y\Gamma$. We denote $\mathcal{E}(\, p^\times, p^\times(Pr_1, Pr_1)\,)$ with $\mathcal{A}$.

(∀I) The introduction rule is the functor $\mathcal{A} \to \mathcal{A}$ which follows from the hom-set isomorphism discussed above. Its inverse implies that actually it is the following.

$$\frac{x \times y; w_y\Gamma \vdash \phi}{x; \Gamma \vdash \forall_y \phi}$$

With Elimination, we (implicitly) use the isomorphism above and write $[t/y]$ for $([x/x], [t/y]) : x \to x \times y$ exploiting $\mathbf{ctx}_{/x\times y} \cong \mathbf{ctx}_{/x} \times \mathbf{ctx}_{/y}$. Using substitution again as in Section 2.2.5, we get

$$\frac{x \times y; w_y\Gamma \vdash \phi}{x; (w_y\Gamma)[t/y] \vdash \phi[t/y]}$$

But recall that $w_y\Gamma = \phi_\Gamma[\mathrm{pr}_1]$, and since

$$[\mathrm{pr}_1][t/y] : x \to x \times y \to x$$

is $\mathrm{id} = [x/x]$, given that also p is faithful, we automatically get $(w_y\Gamma)[t/y] = \Gamma$ concluding the proof.

2.4.7 Cut elimination

In pointing out necessary features of a judgemental analogue of natural deduction, we see that no instance of Cut is (explicitly) mentioned and, instead, in Section 2.4.3 Cut is shown to *automatically* be in the IcFOT generated by $p : \mathcal{P} \to \mathbf{ctx}$. We regard this as an instance of what in sequent calculus is called "cut elimination" (and is shown to be quite hard to prove [Gen64]), or of "normalization" in natural deduction (which, in turn, follows almost instantly from admissibility).

In a very precise sense, such rule is a tool that we already have encoded in the theory the moment we require that it satisfies some properties that we deem fundamental. In fact, curiously, the main reason it works is the existence of the domain-codomain policy (Remark 2.4.2.1) and *not* (only) composition of arrows. More on this peculiarity was discussed in Remark 2.4.3.5.

2.5 Future developments

As we have specified in the introduction to this Chapter, we here only see a couple of possible applications of the framework of judgemental theories, but their versatility suggests many more are possible, for example to modal or linear logic. A taste of the first is already contained in Chapter 3. Moreover, as any other calculus, questions of compactness and normalization arise. We believe trying to answer them would lead to interesting insights into both the logic and the category theory.

Moreover, it feels like our treatment of substitution might intercept some concepts in [MS21], where a calculus of substitution is introduced by means of

composition of certain dinatural transformations. This is a relation that we wish to investigate in future work.

In Remark 2.4.3.5 a well-known link between the cut rule and substitution of terms in expressed in our framework. There we suggested many common features of the two, and a comodality seems to appear. We hope to find more examples of these *cut-like* phenomena, and study their intrinsic properties.

Finally, the attentive reader might have noticed that the choice of fixing a given category for contexts is a mere formality, and it actually makes the definitions less smooth that we wished, see for example the discussion pertaining Definition 2.1.0.4: if anything, this work has convinced us that the notion of *context* in a logical theory is simply a relative one. We believe that this line of thought and work should be explored further. Nevertheless, we decided to keep the exposition closer to classical presentations as not to make an already cryptic theory appear even more strenuous to follow.

Chapter 3

Fibrations for dependent types

> La mia praticità consiste in questo: nel sapere che a battere la testa contro il muro è la testa a rompersi e non il muro.
>
> [Gra10, 19 maggio 1930]

In Section 2.3 we have introduced a new syntactic model for dependent types obtained by generalizing a construction in [Awo18]. In Section 2.3.2 a relation to another categorical model, namely comprehension categories, was suggested. Here we inspect such relation and find that the two notions, though coming from very different perspectives, are actually equivalent: this allows us to position categorized dependent type theories into the wider landscape of fibration-based models of dependent types.

Interestingly, a key concept arises in the form of a particular comonad whose counit models weakening and multiplication represents contraction in our theory. We analyze related constructions to gain further insights into type dependency.

We then apply our results to subtyping and to the study of *inherent* properties of other categorical models.

Contribution

We prove the 2-equivalence of comprehension categories and weakening and contraction comonads outlined in [Jac99, Theorem 9.3.4], then prove that the latters are 2-equivalent to a suitable category of categorized dependent type theories. We infer properties of the corresponding type theories via categorical properties of the structures involved. We extend splitting results for fibrations to the case of weakening and contraction comonads, getting a specialization partly known from [GL23]. We prove the usefulness of non discrete fibrations for the purpose of dependent types by modeling subtyping. Comparing categorized judgemental theories with other models, we prove that some of their properties are inherent to them, for example in the case of CE-systems [AENR21] and dependent sums and units.

3.1 Of judgements and types

Up until this point, we have been using the word "type theory" quite liberally. We now take some time to describe what we actually mean by it: this will not only make the present work more precise, but it will tell us exactly which elements we need in order to give a model of it.

Here we call type theory that described by Per Martin-Löf in [Mar84]. It is a formal system devised to discuss about three kinds of objects - that is *contexts*, *types*, and *terms* - whose mutual relations are described using *judgements*. There are four (plus one) possible kinds of judgement,

$$\Gamma \vdash A\ \texttt{Type} \qquad \Gamma \vdash a : A \qquad \Gamma \vdash A = B \qquad \Gamma \vdash a = b : A \qquad (\vdash \Gamma\ \texttt{ctx})$$

where Γ is a context, A and B are types, a and b are terms. They represent, respectively, that A is a type in context Γ; that a is a term of type A in context Γ; that the types A and B in context Γ are definitionally equal; that the terms a and b of type A in context Γ are definitionally equal. Most of the times one completes the calculus adding a "degenerate" judgement to say that Γ is a context.

These judgements constitute the basic blocks of the calculus, but have no computational power in and of themselves: that is impressed on the judgements by adding *rules* manipulating them. A rule is a process taking in input one or more judgements and producing a new one. See [Hof97] for a comprehensive list of classical rules for dependent types and Section 4.6 for their fuzzy version.

A model of type theory, then, will need to provide an interpretation of contexts, types, and terms, such that

1. they are in the appropriate relations described by the four judgements above, and
2. they can be manipulated following the rules in such a way that they do not fall outside the model.

3.1.1 The category of contexts

Before we begin our discussion, we ought to take a minute to talk about the category of contexts $\mathcal{B}$ we will be working over. Our presentation is heavily focused on the semantics, so we will assume the least possible number of hypotheses on $\mathcal{B}$. Nevertheless, it is convenient to know what we should think of $\mathcal{B}$ as. Needless to say, this discussion will sound a bit circular.

Assume that we are working in a dependent type theory à la Martin-Löf with weakening and substitution. We think of objects of $\mathcal{B}$ as lists of types, $\Gamma = (x_1 : A_1, \dots, x_n : A_n)$ with $\Gamma.A_1.\dots.A_i \vdash A_{i+1}$ for all $i \geq 1$, and of maps

$$\underline{f} \colon \Gamma = (x_1 : A_1, \dots, x_n : A_n) \to (y_1 : B_1, \dots, y_m : B_m)$$

where $\underline{f} = (f_1, \dots, f_m)$ is

$$\begin{cases} \Gamma \vdash f_1 : B_1 \\ \Gamma \vdash f_{j+1} : B[f_1/y_1, \dots, f_j/y_j] \end{cases}$$

for $j \geq 1$. Composition of maps is given by simultaneous substitution, and the identity on a given Γ is clearly $\underline{x}$.

We refer to Example 3.3.6.5 for an explicit construction of the category $\mathcal{B}$, and urge the reader to compare this construction with (the category of contexts of) the doctrine associated to the Lindenbaum-Tarski algebra of a given first order theory (1.7.1.2).

Remark 3.1.1.1 (The empty context)**.** One often asks for a category of contexts to *at least* have an object interpreting the empty context, meaning that $\mathcal{B}$ has (a choice of) a terminal object 1. We do not do this for the moment, but we will come back to the issue in Section 3.4.1.

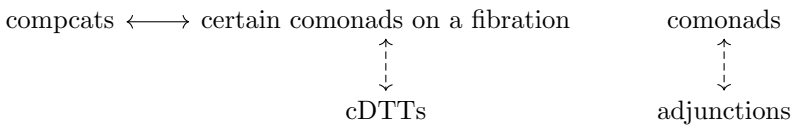
3.1.2 What are we doing?

The first step in our plan is comparing categorized dependent type theories (or cDTTs, Definition 2.3.0.1) with comprehension categories (or compcats, Definition 3.3.1.1): something in this direction was already hinted at in Section 2.3.2, but it is not trivial to prove that they are in fact equivalent, as they were introduced with quite different approaches in mind.

Comprehension categories have a long history and witness the fruitful connection between type theory and category theory, of *reading* the type theory *into* the category (in the spirit of [See84]); moreover, having dependent sums and products is a *property of* the category and not a *structure on* the category. In contrast, categorized dtts are much more computational in flavour, and aim to *use categories* as tools to (externally) describe relations of different entities: for example, terms and types belong to two entirely distinct categories in order to be treated as two entirely distinct entities.

Still, as categorical objects they have a lot in common, so it is entirely natural to wonder whether the two are comparable. We believe it is not only interesting from a mathematical point of view, but philosophically too, since we will show that the two approaches can be actually made into one.

Finally, we apologize for the structure of this exposition, as the following section will seem as it was coming out of the blue. To be honest, we could have done one of the two: either postpone the discussion on comonads and adjunctions to *after* the middle term to compare comprehension categories and cDTTs was introduced, and subsequently describe what categories such structures could be gathered into, or do it this way, and have structures and their categories be defined all at once. If we had picked the first, perhaps the reader would have had some more motivation to survive Section 3.2, but we believe the exposition would have lost in clarity. Choosing the second, we ought to give away our main intuition straight away, to the cost of losing in momentum. So here it is: Section 3.3 will be devoted to proving following relation,



in the sense that comprehension categories will be shown to be equivalent to particular comonads on top of a fibration, so that establishing a connection between comprehension categories and cDTTs will pass through a classical relation between comonads and adjunctions.

3.2 A biadjunction between comonads and adjunctions

Disclaimer 3.2.0.1. The content of this section is well-known to people working in category theory. Nevertheless, we could not find any clear-enough exposition we could refer to — in no small part due to the fact that we are interested in comonads, instead of monads, so one needs to be extremely careful when inverting the appropriate 1- and 2-cells. We recover known results, fix notation, and give definitions that would be of use later, but the expert reader is welcome to skip directly to Section 3.3. A thorough exposition of the 1-dimensional version of the results we are interested in can be found in [MMdPR05, Theorems 39.i and 40.i].

First of all, let us recall some basic definitions and results.

Definition 3.2.0.2 (Comonad)**.** A *comonad* on a category $\mathcal{C}$ is a triple (T, ϵ, ν), with T an endofunctor on $\mathcal{C}$, $\epsilon\colon T \Rightarrow \mathrm{Id}$, called the *counit* of the comonad, and $\nu\colon T \to T \circ T$, called the *comultiplication* of the comonad, such that the following diagrams commute.

$$\begin{array}{ccc} T & \xRightarrow{\nu} & T^2 \\ {\scriptstyle\nu}\Downarrow & & \Downarrow{\scriptstyle \nu T} \\ T^2 & \xRightarrow[T\nu]{} & T^3 \end{array} \qquad \begin{array}{ccccc} T^2 & \xRightarrow{T\epsilon} & T & \xLeftarrow{\epsilon T} & T^2 \\ & \searrow^{\mathrm{Id}} & {\scriptstyle\nu}\Downarrow & \swarrow_{\mathrm{Id}} & \\ & & T^2 & & \end{array}$$

Here we write T^2 for $T \circ T$ and T^3 for $T^2 \circ T$. Often one denotes the comonad simply by its endofunctor T.

As the name might suggest, the notion of comonad has a dual version namely that of *monad*: we refer the reader to [ML78, Chapter VI] for a comprehensive treatment on the topic.

Definition 3.2.0.3 (Coalgebras for a comonad)**.** Let (T, ϵ, ν) a comonad con $\mathcal{C}$. A *coalgebra* for the comonad is a pair (A, a) with A an object of $\mathcal{C}$ and $a\colon A \to TA$ a morphism such that the following diagrams commute.

$$\begin{array}{ccc} A & \xrightarrow{a} & TA \\ & \searrow^{\mathrm{id}} & \downarrow{\scriptstyle \epsilon_A} \\ & & A \end{array} \qquad \begin{array}{ccc} A & \xrightarrow{a} & TA \\ {\scriptstyle a}\downarrow & & \downarrow{\scriptstyle \nu_A} \\ TA & \xrightarrow[Ta]{} & T^2A \end{array}$$

Construction 3.2.0.4 (The co-EM category)**.** Coalgebras for a given comonad T can be assembled into a category $\mathrm{CoAlg}(T)$ – called the *co-Eilenberg-Moore* category of T. Its objects are coalgebras and a coalgebra morphism $(A, a) \to (B, b)$ is a map $f\colon A \to B$ in $\mathcal{C}$ such that the following diagram commutes.

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ {\scriptstyle a}\downarrow & & \downarrow{\scriptstyle b} \\ TA & \xrightarrow[Tf]{} & TB \end{array}$$

Every adjunction $L \dashv R$ determines a comonad on the composite LR (and a monad on RL), as it was first observed in [Hub61]. Conversely, every comonad T

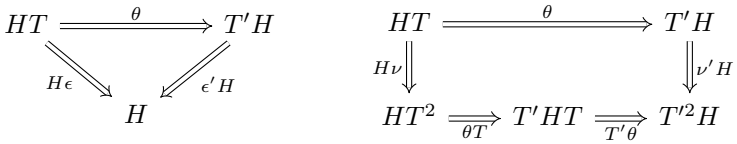
determines an adjunction via the Eilenberg-Moore construction of the category of algebras [EM65]. In fact it determines two adjunctions—the second one being given by the Kleisli construction [Kle65] of the category of free algebras, but we shall only be interested in the former. As it is shown in [Str72], this correspondence between adjunctions and comonads lifts to a 2-adjunction between suitable 2-categories. We now recall the details, which we need to establish an equivalence between categorized dependent type theories and weakening-contraction comonads.

3.2.1 Morphisms of adjunctions and of comonads

Definition 3.2.1.1. The 2-category **Cmd** is defined as follows.

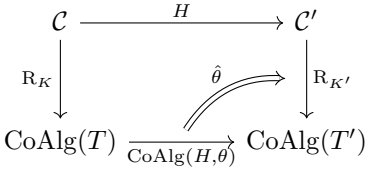
A *0-cell* is a pair of a category $\mathcal{C}$ and a comonad (T, ϵ, ν) on $\mathcal{C}$.

A *1-cell* from $(\mathcal{C}, T, \epsilon, \nu)$ to $(\mathcal{C}', T', \epsilon', \nu')$ is a lax morphism of comonads, that is, a pair (H, θ) of a functor $H\colon \mathcal{C} \to \mathcal{C}'$ and a natural transformation $\theta\colon HT \Rightarrow T'H$ such that the diagrams below commute.



A *2-cell* from (H_1, θ_1) to (H_2, θ_2) is a natural transformation $\phi\colon H_1 \Rightarrow H_2$ such that $(T'\phi)\theta_1 = \theta_2(\phi T)$.

Remark 3.2.1.2. Note that the right-hand diagram in the definition 3.2.1.1 of lax morphism of comonads can be read as saying that, given a lax morphism of coalgebras $(H, \theta)\colon (T, \epsilon, \nu) \to (T', \epsilon', \nu')$, each component θ_E is a morphism of coalgebras $(HTE, \theta_{KE} \circ H\nu_E) \to (T'HE, \nu'_{HE})$. This means that θ lifts to $\hat{\theta}$ below.



We need to consider several kinds of morphisms between adjunctions.

Definition 3.2.1.3. Let (L, R, η, ϵ) and $(L', R', \eta', \epsilon')$ be adjunctions, where $L\colon \mathcal{D} \to \mathcal{C}$ and $L'\colon \mathcal{D}' \to \mathcal{C}'$.

A *left morphism of adjunctions* from (L, R, η, ϵ) to $(L', R', \eta', \epsilon')$ is a pair (F, G) where $F\colon \mathcal{C} \to \mathcal{C}'$, $G\colon \mathcal{D} \to \mathcal{D}'$ are functors such that $L'G = FL$.

A *right morphism of adjunctions* from (L, R, η, ϵ) to $(L', R', \eta', \epsilon')$ is a pair (F, G) where $F\colon \mathcal{C} \to \mathcal{C}'$, $G\colon \mathcal{D} \to \mathcal{D}'$ are functors such that $GR' = RF$.

A *left loose morphism of adjunctions* from (L, R, η, ϵ) to $(L', R', \eta', \epsilon')$ is a triple (F, G, ζ) where $F\colon \mathcal{C} \to \mathcal{C}'$, $G\colon \mathcal{D} \to \mathcal{D}'$ are functors and $\zeta\colon L'G \xrightarrow{\sim} FL$ is a natural iso.

The composite of two left loose morphisms of adjunctions (F_1, G_1, ζ_1) and (F_2, G_2, ζ_2) is $(F_2F_1, G_2G_1, (F_2\zeta_1)(\zeta_2 G_1))$. The straightforward verification that composition is associative is left to the reader.

A *right loose morphism of adjunctions* is defined dually as a triple (F, G, ξ) where $F\colon \mathcal{C} \to \mathcal{C}'$, and $G\colon \mathcal{D} \to \mathcal{D}'$ are functors, and $\xi\colon GR \xRightarrow{\sim} R'F$ is a natural iso.

Definition 3.2.1.4. The 2-category $\mathbf{Adj}_{\mathrm{L}}^{\mathrm{loose}}$ is defined as follows.

A *0-cell* is an adjunction (L, R, η, ϵ).

A *1-cell* is a left loose morphism of adjunctions.

A *2-cell* from (F_1, G_1, ζ_1) to (F_2, G_2, ζ_2) is a pair (ϕ, ψ) of natural transformations $\phi\colon F_1 \Rightarrow F_2$ and $\psi\colon G_1 \Rightarrow G_2$ such that $(\phi L)\zeta_1 = \zeta_2(L'\psi)$.

The 2-category $\mathbf{Adj}_{\mathrm{L}}$ is the wide 2-full sub-2-category of $\mathbf{Adj}_{\mathrm{L}}^{\mathrm{loose}}$ on the 1-cells which are left morphisms of adjunctions.

3.2.2 Biadjunctions between comonads and adjunctions

First of all, we recall the definition of mate of a natural transformation.

Definition 3.2.2.1. Let L and L' be left adjoints and $\zeta\colon L'G \Rightarrow FL$ a natural tranformation. Its *mate* $\zeta^{\#}$ is defined as in (3.1).

$$\begin{array}{ccc} GR & \xRightarrow{\zeta^{\#}} & R'F \\ {\scriptstyle \eta' GR}\Downarrow & \ddots & \Uparrow{\scriptstyle R'F\epsilon} \\ R'L'GR & \xRightarrow[R'\zeta R]{} & R'FLR \end{array} \tag{3.1}$$

Remark 3.2.2.2. Using naturality of the arrows involved and the triangular identities, it is straightforward to verify the following facts.

1. Let (F_1, G_1, ζ_1) and (F_2, G_2, ζ_2) be two composable left loose morphisms of adjunctions. Then
$$(F_2\zeta_1 \circ \zeta_2 G_1)^{\#} = \zeta_2^{\#} F_1 \circ G_2 \zeta_1^{\#}.$$

2. Let (F, G, ζ) be a left loose morphism of adjunctions. Then the two squares below commute.
$$\begin{array}{ccc} G & \xRightarrow{\eta' G} & R'L'G \\ {\scriptstyle G\eta}\Downarrow & & \Downarrow{\scriptstyle R'\zeta} \\ GRL & \xRightarrow[\zeta^{\#}L]{} & R'FL \end{array} \qquad \begin{array}{ccc} L'GR & \xRightarrow{\zeta R} & FLR \\ {\scriptstyle L'\zeta^{\#}}\Downarrow & & \Downarrow{\scriptstyle F\epsilon} \\ L'R'F & \xRightarrow[\epsilon' F]{} & F \end{array}$$

3. Consider two left loose morphisms of adjunctions (F_1, G_1, ζ_1) to (F_2, G_2, ζ_2) and a pair (ϕ, ψ) of natural transformations $\phi\colon F_1 \Rightarrow F_2$ and $\psi\colon G_1 \Rightarrow G_2$. Then the left-hand square below commutes if and only if the right-hand one does.
$$\begin{array}{ccc} L'G_1 & \xRightarrow{\zeta_1} & F_1L \\ {\scriptstyle L'\psi}\Downarrow & & \Downarrow{\scriptstyle \phi L} \\ L'G_2 & \xRightarrow[\zeta_2]{} & F_2L \end{array} \qquad \begin{array}{ccc} G_1R & \xRightarrow{\zeta_1^{\#}} & R'F_1 \\ {\scriptstyle \psi R}\Downarrow & & \Downarrow{\scriptstyle R'\phi} \\ G_2R & \xRightarrow[\zeta_2^{\#}]{} & R'F_2 \end{array}$$

Remark 3.2.2.3. Given a 2-category $\mathbf{C}$, we write $\mathbf{C}^{\mathrm{op}}$ for the 2-category with the 1-cells reversed, and $\mathbf{C}^{\mathrm{co}}$ for the 2-category with the 2-cells reversed. The 2-category where both 1-cells and 2-cells are reversed is denoted $\mathbf{C}^{\mathrm{coop}}$.

Let $(-)^{\mathrm{o}} \colon \mathbf{Cat} \to \mathbf{Cat}^{\mathrm{co}}$ denote the 2-functor that maps a category to its opposite and a natural transformation $\phi \colon F \to G \colon \mathcal{C} \to \mathcal{D}$ to $\phi^{\mathrm{o}} \colon G^{\mathrm{o}} \to F^{\mathrm{o}} \colon \mathcal{C}^{\mathrm{o}} \to \mathcal{D}^{\mathrm{o}}$. This 2-functor gives rise to the isomorphisms of 2-categories

$$\mathbf{Cmd} \cong \mathbf{Mnd}^{\mathrm{co}} \qquad \mathbf{Adj}_{\mathrm{R}}^{\mathrm{co}} \cong \mathbf{Adj}_{\mathrm{L}} \tag{3.2}$$

as we describe below.

A lax morphism of comonads $(H, \theta) \colon (\mathcal{C}, T, \epsilon, \nu) \to (\mathcal{C}', T', \epsilon', \nu')$ induces a morphism of monads $(H^{\mathrm{o}}, \theta^{\mathrm{o}}) \colon (\mathcal{C}^{\mathrm{o}}, T^{\mathrm{o}}, \epsilon'^{\mathrm{o}}, \nu'^{\mathrm{o}}) \to (\mathcal{C}'^{\mathrm{o}}, T'^{\mathrm{o}}, \epsilon^{\mathrm{o}}, \nu^{\mathrm{o}})$ as defined in [LHTVM19, Section 3]. A 2-cell $\phi \colon (H_1, \theta_1) \to (H_2, \theta_2)$ in $\mathbf{Cmd}$ induces a transformation of monads $(H_2^{\mathrm{o}}, \theta_2^{\mathrm{o}}) \to (H_1^{\mathrm{o}}, \theta_1^{\mathrm{o}})$. As both correspondences are bijective, it follows that $\mathbf{Cmd}$ is 2-isomorphic to $\mathbf{Mnd}^{\mathrm{co}}$.

A left morphism of adjunctions $(F, G) \colon (L, R, \eta, \epsilon) \to (L', R', \eta', \epsilon')$ clearly induces a right morphism of adjunctions $(G^{\mathrm{o}}, F^{\mathrm{o}}) \colon (R^{\mathrm{o}}, L^{\mathrm{o}}, \epsilon^{\mathrm{o}}, \eta^{\mathrm{o}}) \to (R'^{\mathrm{o}}, L'^{\mathrm{o}}, \epsilon'^{\mathrm{o}}, \eta'^{\mathrm{o}})$. A 2-cell $(\phi, \psi) \colon (F_1, G_1) \to (F_2, G_2)$ in $\mathbf{Adj}_{\mathrm{L}}$ is such that $\phi L = L' \psi$, and it induces a pair $(\psi^{\mathrm{o}}, \phi^{\mathrm{o}}) \colon (G_2^{\mathrm{o}}, F_2^{\mathrm{o}}) \to (G_1^{\mathrm{o}}, F_1^{\mathrm{o}})$ such that $\phi^{\mathrm{o}} L^{\mathrm{o}} = L'^{\mathrm{o}} \psi^{\mathrm{o}}$. Again, being bijective, these correspondences induce a 2-isomorphism between $\mathbf{Adj}_{\mathrm{L}}$ and $\mathbf{Adj}_{\mathrm{R}}^{\mathrm{co}}$ as defined in [LHTVM19, Section 3].

Theorem 3.1 in [LHTVM19] (see also [CVST10]) proves that there is a 2-adjunction:

$$\mathbf{Mnd} \underset{\mathbf{EM}}{\overset{\mathbf{M}}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathbf{Adj}_{\mathrm{R}} \tag{3.3}$$

where $\mathbf{M}$ is the monad induced by an adjunction, and $\mathbf{EM}$ is the free-forgetful adjunction given by the Eilenberg–Moore category of algebras. The counit of $\mathbf{EM} \dashv \mathbf{M}$ is the identity $\mathbf{M} \circ \mathbf{EM} = \mathbf{Id}$. It follows that $\mathbf{Mnd}$ is a 2-reflective sub-2-category of $\mathbf{Adj}_{\mathrm{R}}$.

Composing the 2-adjunction in (3.3) with the 2-isomorphisms in (3.2)

$$\begin{aligned} \mathbf{Cmd}(\mathbf{C}(L, R, \eta, \epsilon), (T, \epsilon, \nu)) &\cong \mathbf{Mnd}\,(\mathbf{M}(R^{\mathrm{o}}, L^{\mathrm{o}}, \epsilon^{\mathrm{o}}, \eta^{\mathrm{o}}), (T^{\mathrm{o}}, \epsilon^{\mathrm{o}}, \nu^{\mathrm{o}}))^{\mathrm{op}} \\ &\cong \mathbf{Adj}_{\mathrm{R}}\,((R^{\mathrm{o}}, L^{\mathrm{o}}, \epsilon^{\mathrm{o}}, \eta^{\mathrm{o}}), \mathbf{EM}(T^{\mathrm{o}}, \epsilon^{\mathrm{o}}, \nu^{\mathrm{o}}))^{\mathrm{op}} \\ &\cong \mathbf{Adj}_{\mathrm{L}}\,((L, R, \eta, \epsilon), \mathbf{EM}(T, \epsilon, \nu)) \end{aligned}$$

we see that the 2-category $\mathbf{Cmd}$ is a 2-reflective sub-2-category of $\mathbf{Adj}_{\mathrm{L}}$. We record this fact in the theorem below.

Theorem 3.2.2.4. There is a 2-adjunction

$$\mathbf{Cmd} \underset{\mathbf{EM}}{\overset{\mathbf{C}}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathbf{Adj}_{\mathrm{L}}$$

such that the counit is the identity $\mathbf{C} \circ \mathbf{EM} = \mathbf{Id}_{\mathbf{Cmd}}$. In particular, the right adjoint $\mathbf{EM}$ is injective on objects and fully faithful.

In fact, the 2-adjunction in Theorem 3.2.2.4 can be extended to a biadjunction involving the 2-category whose 1-cells are left loose morphisms of adjunctions. This is not hard to see, but we first need to recall some facts from [LHTVM19, Section 3], translated along the 2-isomorphisms in 3.2.

The right adjoint **EM** maps a comonad (T, ϵ, ν) to the Eilenberg-Moore adjunction:

$$\mathrm{CoAlg}(T) \underset{\mathrm{U}_T}{\overset{\mathrm{R}_T}{\underset{\longrightarrow}{\overset{\longleftarrow}{\top}}}} \mathcal{C}$$

whose counit is $\epsilon\colon \mathrm{U}_T\mathrm{R}_T = T \Rightarrow \mathrm{Id}_{\mathcal{C}}$ and whose unit $\eta^T\colon \mathrm{Id}_{\mathrm{CoAlg}(T)} \Rightarrow \mathrm{R}_T\mathrm{U}_T$ has component at a coalgebra $(A, a\colon A \to TA)$ the arrow a itself seen as a morphism of coalgebras $(A, a) \to (TA, \nu_A)$. A lax morphism of comonads $(H, \theta)\colon T \to T'$ induces a functor $\mathrm{CoAlg}(H, \theta)\colon \mathrm{CoAlg}(T) \to \mathrm{CoAlg}(T')$ which maps a T-coalgebra (A, a) to the T'-coalgebra $(HA, \theta_A \circ Ha)$. Clearly, $\mathrm{U}_{T'}\mathrm{CoAlg}(H, \theta) = H\mathrm{U}_T$. Therefore the pair $(H, \mathrm{CoAlg}(H, \theta))$ is a left morphism of adjunctions, which gives the action of the 2-functor **EM** on 1-cells. Finally, it is easy to see that every 2-cell ϕ in **Cmd** lifts to a natural transformation $\mathrm{CoAlg}(\phi)\colon \mathrm{CoAlg}(H_1, \theta_1) \Rightarrow \mathrm{CoAlg}(H_2, \theta_2)$ whose component at (A, a) is ϕ_A itself. Therefore $(\phi, \mathrm{CoAlg}(\phi))$ is a 2-cell in $\mathbf{Adj}_{\mathrm{L}}$, which gives the action of **EM** on 2-cells.

The left adjoint **C** maps an adjunction (L, R, η, ϵ) to the comonad $(LR, \epsilon, L\eta R)$. A left morphism of adjunctions (F, G) induces a lax morphism of comonads $\mathbf{C}(F, G) = (F, L'\mathrm{id}^{\#})$, where $\mathrm{id}^{\#} = (R'F\epsilon)(\eta' GR)\colon GR \Rightarrow R'F$ is the mate of $\mathrm{id}\colon L'G \Rightarrow FL$. In fact, in Lemma 3.2.3.1.1 we will prove that a left loose morphism of adjunctions induces a lax morphism of comonads as well. A 2-cell (ϕ, ψ) in $\mathbf{Adj}_{\mathrm{L}}$ is simply mapped to ϕ. It is straightforward to verify that ϕ is a 2-cell $(F_1, L'\mathrm{id}^{\#}) \to (F_2, L'\mathrm{id}^{\#})$ in **Cmd**.

Every adjunction (L, R, η, ϵ) gives rise to a canonical comparison functor $\mathrm{K}_{L,R}$ making the diagram below commute.

$$\begin{array}{ccc} \mathcal{D} & \xrightarrow{\mathrm{K}_{L,R}} & \mathrm{CoAlg}(LR) \\ & {\scriptstyle L}\searrow \quad \swarrow{\scriptstyle \mathrm{U}_{LR}} & \\ & \mathcal{C} & \end{array} \tag{3.4}$$

Recall that $\mathrm{K}_{L,R}$ maps an object A to the coalgebra $L\eta_A R\colon LRA \to (LR)^2 A$, and an arrow $f\colon A \to B$ to $LRf\colon LRA \to LRB$.

The unit $\boldsymbol{\eta}$ of the 2-adjunction $\mathbf{C} \dashv \mathbf{EM}$ at (L, R, η, ϵ) is defined as the left morphism of adjunctions $(\mathrm{Id}, \mathrm{K}_{L,R})\colon (L, R, \eta, \epsilon) \to (\mathrm{U}_{LR}, \mathrm{R}_{LR}, \eta^{LR}, \epsilon)$. This family is natural in (L, R, η, ϵ) since, for every left morphism of adjunctions (F, G),

$$\begin{aligned} L'\mathrm{id}^{\#}_{LA} \circ FL\eta_A &= L'R'F\epsilon_{LA} \circ L'\eta'_{GRLA} \circ L'G\eta_A \\ &= L'R'F\epsilon_{LA} \circ L'R'L'G\eta_A \circ L'\eta'_{GA} \\ &= L'\eta'_{GA} \end{aligned}$$

and $FLf = L'Gf$ for A and f in $\mathcal{D}$, imply $\mathrm{CoAlg}(\mathbf{C}(F, G, \zeta)) \circ \mathrm{K}_{L,R} = \mathrm{K}_{L',R'} \circ G$.

On the other hand, for every comonad (T, ϵ, ν) we have $\mathbf{C} \circ \mathbf{EM}(T, \epsilon, \nu) = (T, \epsilon, \nu)$, since $\mathrm{U}_K\mathrm{R}_K = K$ and $\mathrm{U}_K\eta\mathrm{R}_K = \nu$. The counit of $\mathbf{C} \dashv \mathbf{EM}$ at (T, ϵ, ν) is simply the identity. Naturality amounts to $\mathbf{C} \circ \mathbf{EM}(H, \theta) = (H, \theta)$, for every lax morphism of comonads $(H, \theta)\colon (K, \epsilon, \nu) \to (K', \epsilon', \nu')$. To prove that equation, recall that $\eta'_{(A,a)} = a$ and use the two diagrams in Definition 3.2.1.1 to show that $\mathrm{U}_{K'}\mathrm{id}^{\#}_E = K'H\epsilon_E \circ \mathrm{U}_{K'}\eta'_{(HKE, \theta_{KE} \circ H\nu_E)}$ equals θ_E.

This can be verified with a simple calculation: consider $\mathrm{id}^{\#}\colon \mathrm{CoAlg}(H,\theta)\mathrm{R}_K \Rightarrow \mathrm{R}_{K'}H$, then for E in $\mathcal{C}$,

$$\begin{aligned}\mathrm{U}_{K'}\mathrm{id}^{\#}_E &= K'H\epsilon_E \circ \mathrm{U}_{K'}\eta'_{(HKE,\theta_{KE}\circ H\nu_E)}\\ &= K'\epsilon'_{HE}\circ K'\theta_E\circ\theta_{KE}\circ H\nu_E\\ &= K'\epsilon'_{HE}\circ\nu'_{HE}\circ\theta_E\\ &= \theta_E\end{aligned}$$

using the definition of the unit η' of $\mathrm{U}_{K'} \dashv \mathrm{R}_{K'}$ after Theorem 3.2.2.4, the definition of lax morphism of comonads in Definition 3.2.1.1, and an identity of the comonad on K'. As the equality is trivial on 2-cells, we have also shown that

$$\mathbf{C}\circ\mathbf{EM} = \mathbf{Id}.$$

Finally, the triangular identities follow from the equations in (3.5).

$$\mathbf{C}\boldsymbol{\eta} = \mathbf{id}_{\mathbf{C}} \qquad\qquad \boldsymbol{\eta}\mathbf{EM} = \mathbf{id}_{\mathbf{EM}} \tag{3.5}$$

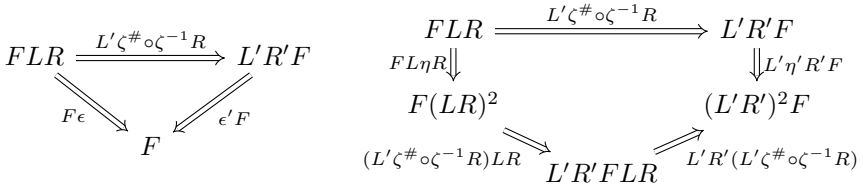
The left-hand one holds since the mate $\mathrm{id}^{\#}\colon \mathrm{K}_{L,R}R \Rightarrow \mathrm{R}_{LR}$ of $\mathrm{id}\colon \mathrm{U}_{LR}\mathrm{K}_{L,R} \Rightarrow L$ is itself an identity. The right-hand one holds since $\mathrm{K}_{\mathrm{U}_T,\mathrm{R}_T} = \mathrm{Id}_{\mathrm{CoAlg}(T)}$

Now we turn to the case of left loose morphisms of adjunctions.

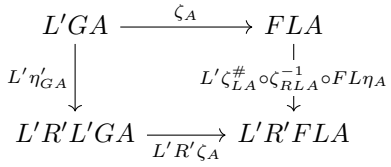
3.2.3 The biadjunction in the loose case

Lemma 3.2.3.1. Let $(F,G,\zeta)\colon (L,R,\eta,\epsilon)\to(L',R',\eta',\epsilon')$ be a left loose morphism of adjunctions. Then the following facts hold.

1. The two diagrams below commute.



2. For every object A of $\mathcal{D}$, the square below commutes.



Proof. Both claims are easily verified using naturality of the arrows involved and the triangular identities of the adjunctions.

1.

$$\begin{aligned}\epsilon'F\circ L'\zeta^{\#}\circ\zeta^{-1}R &= \epsilon'F\circ L'R'F\epsilon\circ L'R'\zeta R\circ L'\eta'GR\circ\zeta^{-1}R\\ &= F\epsilon\circ\zeta R\circ\epsilon'L'GR\circ L'\eta'GR\circ\zeta^{-1}R\\ &= F\epsilon\end{aligned}$$

$$
\begin{aligned}
&L'R'(L'\zeta^{\#} \circ \zeta^{-1}R) \circ L'\zeta^{\#}LR \circ \zeta^{-1}RLR \circ FL\eta R = \\
&= L'R'(L'\zeta^{\#} \circ \zeta^{-1}R) \circ L'R'F\epsilon LR \circ L'R'\zeta RLR \circ L'\eta' GRLR \circ \zeta^{-1}RLR \circ FL\eta R \\
&= L'R'L'(R'F\epsilon \circ R'\zeta R \circ \eta' GR) \circ L'\eta' GR \circ \zeta^{-1}R \\
&= L'\eta' R'F \circ L'\zeta^{\#} \circ \zeta^{-1}R
\end{aligned}
$$

2.

$$
\begin{aligned}
L'\zeta^{\#}_{LA} \circ \zeta^{-1}_{RLA} \circ FL\eta_A \circ \zeta_A &= L'R'(F\epsilon_{LA} \circ \zeta_{RLA}) \circ L'\eta'_{GRLA} \circ L'G\eta_A \\
&= L'R'(F\epsilon_{LA} \circ \zeta_{RLA} \circ L'G\eta_A) \circ L'\eta'_{GA} \\
&= L'R'\zeta_A \circ L'\eta'_{GA}.
\end{aligned}
$$

□

Corollary 3.2.3.2. The 2-functor $\mathbf{C}$ extends along $\mathbf{Adj}_{\mathrm{L}} \hookrightarrow \mathbf{Adj}_{\mathrm{L}}^{\mathrm{loose}}$ to a 2-functor $\mathbf{C_l} \colon \mathbf{Adj}_{\mathrm{L}}^{\mathrm{loose}} \to \mathbf{Cmd}$.

Proof. We only need to consider 1-cells. Lemma 3.2.3.1.1 ensures that $(F, L'\zeta^{\#} \circ \zeta^{-1}R)$ is a lax morphism of comonads whenever (F, G, ζ) is a left loose morphism of adjunctions. Functoriality follows from (1). □

Remark 3.2.3.3. Consider a left loose morphism of adjunctions $(F, G, \zeta) \colon (L, R, \eta, \epsilon) \to (L', R', \eta', \epsilon')$. Then Lemma 3.2.3.1.2 entails that the natural iso $\zeta \colon L'G \xRightarrow{\sim} FL$ lifts to a natural iso

$$\hat{\zeta} \colon \mathrm{K}_{L',R'} \circ G \xRightarrow{\sim} \mathrm{CoAlg}(\mathbf{C_l}(F, G, \zeta)) \circ \mathrm{K}_{L,R}$$

meaning that $\mathrm{U}_{L'R'}\hat{\zeta} = \zeta$.

Theorem 3.2.3.4. The 2-adjunction from Theorem 3.2.2.4 extends along $\mathbf{Adj}_{\mathrm{L}} \hookrightarrow \mathbf{Adj}_{\mathrm{L}}^{\mathrm{loose}}$ to a biadjunction

$$\mathbf{Cmd} \underset{\mathbf{EM_l}}{\overset{\mathbf{C_l}}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathbf{Adj}_{\mathrm{L}}^{\mathrm{loose}}$$

such that the counit is the identity $\mathbf{C_l} \circ \mathbf{EM_l} = \mathbf{Id_{Cmd}}$. In particular, the right adjoint $\mathbf{EM_l}$ is injective on objects and fully faithful.

Proof. It only remains to show that the unit $\boldsymbol{\eta} \colon \mathbf{Id} \Rightarrow \mathbf{EM_l} \circ \mathbf{C_l}$ lifts to a pseudo-natural transformation $\boldsymbol{\eta} \colon \mathbf{Id} \Rightarrow \mathbf{EM_l} \circ \mathbf{C_l}$. This amounts to give, for every left loose morphism of adjunctions $(F, G, \zeta) \colon (L, R, \eta, \epsilon) \to (L', R', \eta', \epsilon')$, an invertible 2-cell $(F, \mathrm{K}_{L',R'} \circ G, \zeta) \to (F, \mathrm{CoAlg}(\mathbf{C_l}(F, G, \zeta)) \circ \mathrm{K}_{L,R}, \mathrm{id})$ in $\mathbf{Adj}_{\mathrm{L}}^{\mathrm{loose}}$. For this 2-cell we can take $(\mathrm{id}_F, \hat{\zeta})$, where $\hat{\zeta}$ is the natural iso from Remark 3.2.3.3. □

3.3 Comparing type-theoretic comprehensions

3.3.1 Comprehension categories

Definition 3.3.1.1 ([Jac93], Def. 4.1)**.** A comprehension category consists of a Grothendieck fibration $p \colon \mathcal{E} \to \mathcal{B}$ together with a functor $\chi \colon \mathcal{E} \to \mathcal{B}^2$ such that

1. $p = \mathrm{cod}\chi$, and

2. χ preserves cartesian arrows.

Sometimes we might refer to p as the *type* fibration.

Intuitively, the category $\mathcal{E}$ collects all types and these are fibered over contexts *i.e.* objects in $\mathcal{B}$. The functor χ maps each type to its context extension projection,

$$A \quad \mapsto \quad \Gamma.A \to \Gamma\,.$$

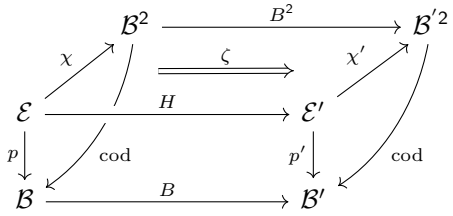
Condition 2 guarantees that substitution of types along morphisms behaves as a pullback of the corresponding map. Terms, though not appearing explicitly, are interpreted to sections of the projection maps.

Construction 3.3.1.2. Consider a functor $F\colon \mathcal{C} \to \mathcal{D}$. It induces a functor $\mathcal{C}^2 \to \mathcal{D}^2$, we call it F^2. Similarly, we call α^2 the natural transformation $F^2 \Rightarrow G^2$ induced by $\alpha\colon F \Rightarrow G$.

Definition 3.3.1.3. Let (p, χ) and (p', χ') be comprehension categories. A *lax morphism of comprehension categories* from (p, χ) to (p', χ') is a triple (H, B, ζ) as in the diagram below, where B^2 is the functor induced by B, such that

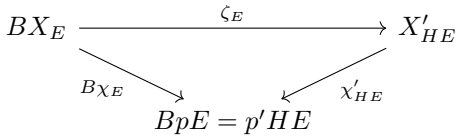
1. (H, B) is a 1-cell in **Fib**, and

2. $\mathrm{cod}\zeta = \mathrm{Id}_{Bp}$.



A lax morphism of comprehension categories (H, B, ζ) is a *pseudo* (respectively, *strict*) *morphism of comprehension categories* if ζ is invertible (respectively, the identity).

Remark 3.3.1.4. Unfolding the definition of lax morphism of comprehension categories in Definition 3.3.1.1, one sees that to have a natural transformation $\zeta\colon B^2\chi \Rightarrow \chi' H$ such that $\mathrm{cod}\zeta = \mathrm{Id}_{Bp}$ amounts to have a natural transformation $B\mathrm{dom}\chi \Rightarrow \mathrm{dom}\chi' H$, which we also denote ζ, such that, for each E in $\mathcal{E}$, the triangle below commutes.



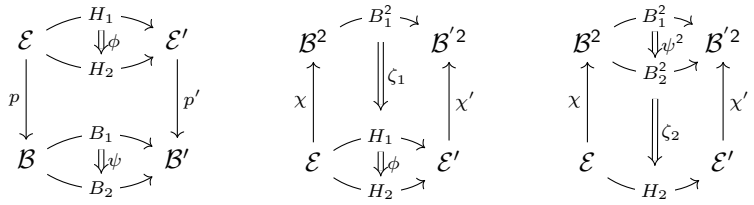
It follows that a lax morphism (H, B, ζ) is pseudo (respectively, strict) if and only if, for every E in $\mathcal{E}$, the arrow ζ_E above is invertible (respectively, the identity).

Definition 3.3.1.5. The *2-category* **CompCat** *of comprehension categories* is defined as follows.

A *0-cell* is a comprehension category (p, χ).

A *1-cell* $(p, \chi) \to (p', \chi')$ is a lax morphism of comprehension categories from (p, χ) to (p', χ').

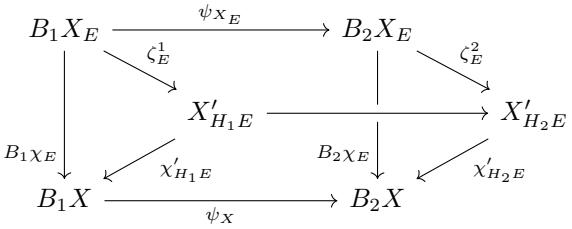
A *2-cell* $(H_1, B_1, \zeta_1) \Rightarrow (H_2, B_2, \zeta_2)$ is a 2-cell $(\phi, \psi) \colon (H_1, B_1) \Rightarrow (H_2, B_2)$ in **Fib** as in the left-hand diagram below, such that $\phi * \zeta_1 = \zeta_2 * \psi^2$.



We write $\mathbf{CompCat_{ps}}$ and $\mathbf{CompCat_{str}}$ for the 2-full 2-subcategory of **CompCat** with the same 0-cells and only those 1-cells which are pseudo (respectively, strict) morphisms of comprehension categories.

Remark 3.3.1.6. What in the literature is usually called **CompCat** here is $\mathbf{CompCat_{str}}$.

Remark 3.3.1.7. Let $(H_1, B_1, \zeta^1), (H_2, B_2, \zeta^2) \colon (p, \chi) \to (p', \chi')$ be lax morphisms of comprehension categories. Using Remark 3.3.1.4 one sees that a 2-cell $(\phi, \psi) \colon (H_1, B_1) \Rightarrow (H_2, B_2)$ in **Fib** is a 2-cell in **CompCat** if and only if, for every E in $\mathcal{E}$ over X, the top square in the diagram below commutes,



where the front square is the image under χ' of $\phi_E \colon H_1E \to H_2E$, the back square is naturality of ψ, and the side triangles are those from Remark 3.3.1.4.

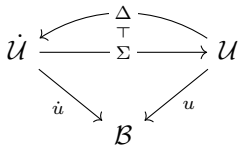
If (H_1, B_1, ζ^1) and (H_2, B_2, ζ^2) are strict morphisms, the top square above commutes if and only if its horizontal arrows coincide. Therefore (ϕ, ψ) is a 2-cell between strict morphisms if and only if $\mathrm{dom}\chi'\phi = \psi\mathrm{dom}\chi$, if and only if $\chi'\phi = \psi^2\chi$.

3.3.2 Categorized dependent type theories

Remark 3.3.2.1 (On the definition of cDTT)**.** If we treat a cDTT as a categorized judgemental system, and use it to do calculations, we are interested in a 2-subcategory of **Cat** containing rules and policies described in Definition 3.3.2.2 and such that is closed under finite limits and ♯-lifting as described in Chapter 2. In fact, that presented in 3.3.2.2 is called *pre*-categorized dependent type theory, the cDTT is said subcategory.

From a traditional use of categorical models, instead, a cDTT is simply the data of two fibrations and a pair of adjoint functors connecting them. For the first part of this work, this is the only thing we need to consider. See Section 2.3 for a use of the full deductive power of a cDTT.

Definition 3.3.2.2 (Categorized dependent type theory)**.** A *categorized dependent type theory* is the data of



where $\dot{u}, u$ are fibrations, $\Sigma\colon \dot{u} \to u$ in **Fib**, Δ is right adjoint to Σ with cartesian unit and counit. We call this *cDTT* for short.

Cartesianity of the unit and counit is meant in the following sense.

Definition 3.3.2.3 (Cartesian natural transformation)**.** Let $p : \mathcal{E} \to \mathcal{B}$ a fibration and $F, G : \mathcal{E}' \to \mathcal{E}$ two functors. A natural transformation $\alpha : F \Rightarrow G$ is said to be *cartesian* if each of its components is p-cartesian.

Terms and types are interpreted to two different categories, respectively $\dot{\mathcal{U}}$ and $\mathcal{U}$, both fibered over contexts. The functor Σ performs the typing, while Δ details context extension,

$$A \quad \mapsto \quad x_A\,(:A^+)$$

where A^+ is the type A extended to the context $\Gamma.A$ (*i.e.* the result of substituting A along the projection map, here represented by $u\epsilon_A$) and x_A is a fresh variable of type A^+.

Remark 3.3.2.4. Each component of the unit in a cDTT is a monic arrow. Indeed, let $f, g\colon a \to b$ in $\dot{\mathcal{U}}$ be such that $\eta_b f = \eta_b g$. It follows that

$$\dot{u}f = (u\epsilon_{\Sigma b})(\dot{u}\eta_b)(\dot{u}f) = (u\epsilon_{\Sigma b})(\dot{u}\eta_b)(\dot{u}g) = \dot{u}g$$

and, in turn, that $f = g$ since η_b is cartesian.

The left adjoint Σ is then faithful. In fact, it is easy to see that Σ induces a bijection

$$\dot{\mathcal{U}}(a,b) \xrightarrow{\sim} \{f \in \mathcal{U}(\Sigma a, \Sigma b) \mid (\Sigma\eta_b)f = (\Sigma\Delta f)(\Sigma\eta_a)\}. \tag{3.6}$$

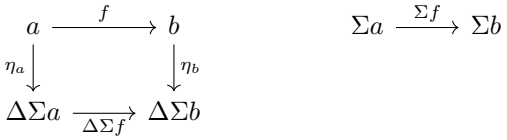
Indeed, since η_b is cartesian, the counter-image of f is the only arrow g in $\dot{\mathcal{U}}(a,b)$ over uf such that $\eta_b g = (\Delta f)\eta_a$.

Lemma 3.3.2.5. Let $(u, \dot{u}, \Sigma \dashv \Delta)$ a cDTT. Then we have that

(i) Δ preserves cartesian maps iff Σ reflects cartesian maps;

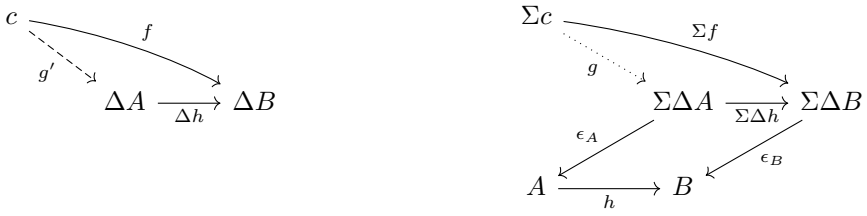
(ii) Δ preserves cartesian maps.

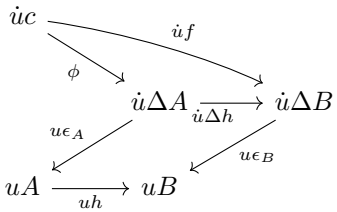
Proof. Let us start with (i): let $f\colon a \to b$ in $\dot{\mathcal{U}}$ such that Σf is cartesian, then $\Delta\Sigma f$ is cartesian, and we have the following



with cartesian units, hence $\eta_b f$ is cartesian with η_b cartesian. By Remark 3.3.3.3, f is cartesian too. The converse can be worked out exactly the same way using counits.

As for (ii), let $h\colon A \to B$ in $\mathcal{U}$ cartesian and consider $f\colon c \to \Delta B$ and $\phi\colon \dot{u}c \to \dot{u}\Delta A$ such that $\dot{u}f = \dot{u}\Delta h \circ \phi$. In diagrams, as follows.

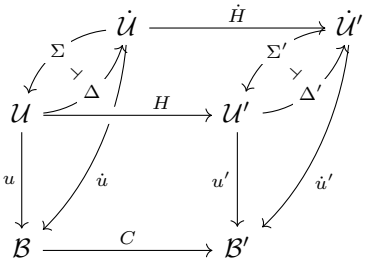




Recall that $\dot{u} = \Sigma u$ and that $\Sigma\Delta h$ is cartesian by Remark 3.3.3.3 because its postcomposition with ϵ_B is. Now, Σf is u-over $\dot{u}f$ and $\Sigma\Delta h$ is u-over $\dot{u}\Delta h$, so that there exists a unique dotted map $g\colon \Sigma c \to \Sigma\Delta A$ making that diagram commute. Therefore we have a unique map $\epsilon_A g\colon \Sigma c \to A$. Our claim is that its correspondent map under $\Sigma \dashv \Delta$, $g'\colon c \to \Delta A$, makes the desired triangle commute: this follows from $\Sigma(g') = g$ and Σ being faithful by Remark 3.3.2.4. □

Definition 3.3.2.6. Let $\mathbb{J}$ and $\mathbb{J}'$ be cDTT's. A *(lax) cDTT morphism* from $\mathbb{J}$ to $\mathbb{J}'$ is a triple of functors $(C, H, \dot{H})$ as in the diagram below, such that

1. $(C, H)\colon u \to u'$ is a 1-cell in **Fib**,
2. $(C, \dot{H})\colon \dot{u} \to \dot{u}'$ is a 1-cell in **Fib**, and
3. $(H, \dot{H})$ is a left morphism of adjunctions, *i.e.* $\Sigma'\dot{H} = H\Sigma$.



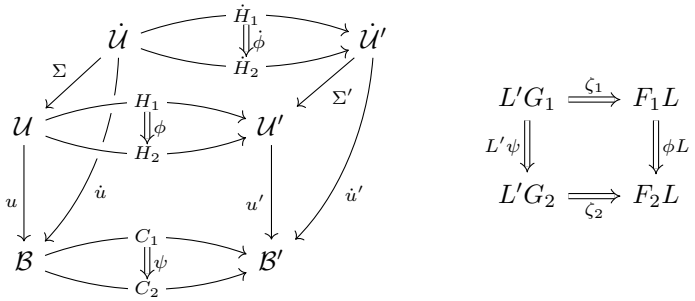
Similarly, a *(lax) loose cDTT morphism* from $\mathbb{J}$ to $\mathbb{J}'$ is a quadruple $(C, H, \dot{H}, \zeta)$ such that

1. $(C, H)\colon u \to u'$ is a 1-cell in **Fib**,
2. $(C, \dot{H})\colon \dot{u} \to \dot{u}'$ is a 1-cell in **Fib**, and
3. $(H, \dot{H}, \zeta)$ is a left loose morphism of adjunctions, *i.e.* $\Sigma'\dot{H} \xRightarrow{\sim} H\Sigma$ (see Definition 3.2.1.3).

Given a loose cDTT morphism $(C, H, \dot{H}, \zeta)$, consider the mate $\zeta^{\#}\colon \dot{H}\Delta \Rightarrow \Delta' H$ as in (3.1). A (loose) cDTT morphism $(H, \dot{H}, C)$ is a *pseudo (loose) cDTT morphism* if the mate $\zeta^{\#}$ is invertible. It is a *strict (loose) cDTT morphism* if $\zeta^{\#}$ is the identity. In particular, in this case $\dot{H}\Delta = \Delta' H$.

Definition 3.3.2.7. The *2-category* $\mathbf{cDTT}^{\text{loose}}$ *of cDTT's and loose cDTT morphisms* has these as objects and arrows, and a 2-cell $(C_1, H_1, \dot{H}_1, \zeta_1) \to (C_2, H_2, \dot{H}_2, \zeta_2)$ is a triple $(\phi, \dot{\phi}, \psi)$ of natural transformations as in the diagram below, such that

1. (ϕ, ψ) is a 2-cell in **Fib**,
2. $(\dot{\phi}, \psi)$ is a 2-cell in **Fib**, and
3. the right-hand diagram below commutes.



The *2-category* **cDTT** *of cDTT's and cDTT morphisms* is defined as the wide 2-full sub-2-category of $\mathbf{cDTT}^{\text{loose}}$ on the cDTT morphisms.

We write $\mathbf{cDTT}_{\mathbf{ps}}$ and $\mathbf{cDTT}_{\mathbf{str}}$ for the 2-full 2-subcategory of **cDTT** with the same 0-cells, and only those 1-cells which are pseudo (respectively, strict) cDTT morphisms.

3.3.3 Weakening and contraction comonads

Turns out that we can compare the two using an intermediate notion introduced in [Jac99] to equivalently present the data of a comprehension category.

Definition 3.3.3.1 ([Jac99], Def. 9.3.1)**.** Let $p\colon \mathcal{E} \to \mathcal{B}$ a fibration. A *weakening and contraction comonad on* p is a comonad (K, ϵ, ν) on $\mathcal{E}$ such that

1. the counit ϵ is p-cartesian and,

2. for every cartesian arrow $f\colon A \to B$ in $\mathcal{E}$ the image in $\mathcal{B}$ under p

$$\begin{array}{ccc} pKA & \xrightarrow{p\epsilon_A} & pA \\ {\scriptstyle pKf}\downarrow & & \downarrow{\scriptstyle pf} \\ pKB & \xrightarrow{p\epsilon_B} & pB \end{array} \tag{3.7}$$

of the naturality square is a pullback square.

We call these *wc-comonads* for short, and usually write $pA.A$ for pKA. We often use the word to indicate the pair of the comonad and the fibration it insists on.

As in Definition 3.3.1.1, $\mathcal{E}$ models types which are fibered over contexts via p. The counit of the comonad models weakening, the comultiplication contraction, and the comonad equations witness the fact that performing contraction after weakening yields the identity (up to α-equivalence).

Remark 3.3.3.2. Given a wc-comonad (K, ϵ, ν) on a fibration p, the naturality square of the counit ϵ for a cartesian arrow is a pullback too. This follows from the fact that, in a fibration, a square in $\mathcal{E}$ is a pullback if it has two parallel cartesian sides and it is sitting over a pullback in $\mathcal{B}$.

In particular, the naturality square for ϵ_A is itself is a pullback. It follows that the comultiplication is canonically determined by the counit ϵ via the two counitality axioms. Thus one could equivalently define a wc-comonad to be a copointed endofunctor which enjoys conditions 1 and 2 in Definition 3.3.3.1. See also [Jac99, p.536].

Remark 3.3.3.3.

1. Cartesian maps enjoy the following property: if gf is cartesian and g is locally monic, then f is cartesian too. An arrow $g\colon B \to C$ is locally monic if, for every pair of arrows h, h' such that $ph = ph'$, then $h = h'$ whenever $gh = gh'$.
2. If (E, e) is a coalgebra for a wc-comonad, counitality implies that e is cartesian.
3. If gf is cartesian and g is cartesian, then f is cartesian too.
4. It follows from Remark 3.3.3.2 that the functor K preserves cartesian arrows.

Definition 3.3.3.4. The *2-category* **wcCmd** *of wc-comonads* is defined as follows.

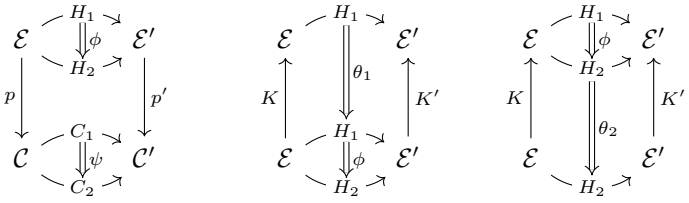
A *0-cell* is a pair $\mathbb{K} = (p, K)$ with p a fibration and K a wc-comonad with respect to p.

A *1-cell* between wc-comonads $\mathbb{K}$ and $\mathbb{K}'$ is a triple (H, C, θ) as in the diagram below, such that

1. $(H, C)\colon p \to p'$ is a 1-cell in **Fib**
2. (H, θ) is a lax morphism of comonads as in Definition 3.2.1.1.

$$\begin{array}{ccc} \mathcal{E} & \xrightarrow{H} & \mathcal{E}' \\ {\scriptstyle K}\uparrow & \Downarrow{\scriptstyle \theta} & \uparrow{\scriptstyle K'} \\ \mathcal{E} & \xrightarrow{H} & \mathcal{E}' \\ {\scriptstyle p}\downarrow & & \downarrow{\scriptstyle p'} \\ \mathcal{C} & \xrightarrow{C} & \mathcal{C}' \end{array}$$

A *2-cell* between 1-cells (H_1, C_1, θ_1) and (H_2, C_2, θ_2) is a 2-cell $(\phi, \psi)\colon (H_1, C_1) \to (H_2, C_2)$ in **Fib** as in the left-hand diagram below, such that $\phi * \theta_1 = \theta_2 * \phi$.



We write $\mathbf{wcCmd_{ps}}$ and $\mathbf{wcCmd_{str}}$ for the 2-full 2-subcategories of **wcCmd** with the same 0-cells, and only those 1-cells (H, C, θ) such that (H, θ) is a pseudo (respectively, strict) morphism of comonads.

3.3.4 wcComonads v comprehension categories

First of all, we prove the 2-equivalence suggested in [Jac99, 9.3.4]. For the following result we need to assume that the underlying fibrations of comprehension categories and wc-comonads are cloven, see Remark 1.5.0.1 for a discussion on what this entails. Morphisms, however, are not required to preserve cleavages (but they will in Proposition 3.4.2.1).

Lemma 3.3.4.1. There is a 2-functor $\mathbf{X}\colon \mathbf{CompCat} \to \mathbf{wcCmd}$.

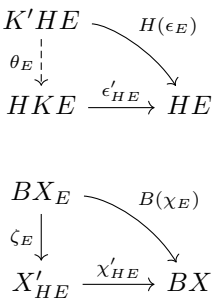
Proof. Let $(p, \chi)\colon \mathcal{E} \to \mathcal{B}^2$ be a comprehension category and S a cleavage for p. For each E in $\mathcal{E}$, consider the reindexing of E along its comprehension χ_E as below.

$$\begin{array}{ccc} S(E, \chi_E) & \xrightarrow{\overline{\chi_E}} & E \\ \\ X_E & \xrightarrow{\chi_E} & pE \end{array} \qquad \begin{array}{c} \mathcal{E} \\ \downarrow{\scriptstyle p} \\ \mathcal{B} \end{array}$$

since cartesian lifts are defined by a universal property, the assignment $K_\chi E = S(E, \chi_E)$ induces an endofunctor K_χ on $\mathcal{E}$. Moreover, K_χ is copointed because the transformation $\epsilon_E := \overline{\chi_E}$ is natural by the very definition of K_χ on arrows. It satisfies condition 1 from Definition 3.3.3.1 by construction and condition 2 because χ preserves cartesian arrows. By Remark 3.3.3.2, this is sufficient to define a wc-comonad.

A lax morphism of comprehension categories $(H, C, \zeta)\colon (p, \chi) \to (p', \chi')$ induces a 1-cell $(H, C)\colon p \to p'$ in **Fib** by its very definition. To obtain a lax morphism of wc-comonads $(H, C, \theta)\colon K_\chi \to K_{\chi'}$, it only remains to provide $\theta\colon HK_\chi \Rightarrow K_{\chi'}H$ that makes (H, θ) into a lax morphisms of comonads. For E over X, the component θ_E can be obtained, using the fact that ϵ'_{HE} is cartesian,

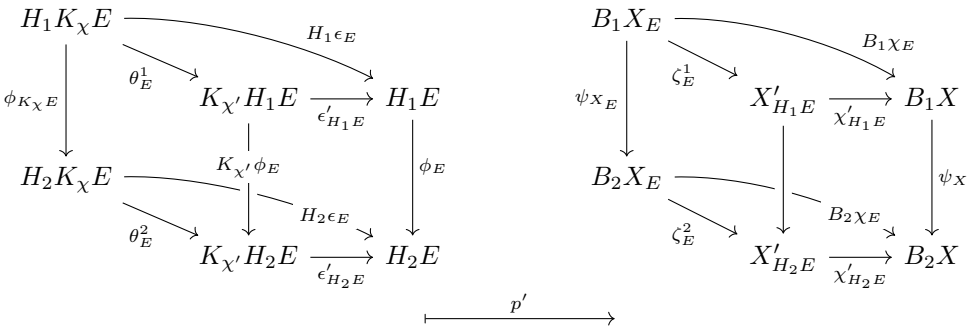
as the universal arrow induced by $H\epsilon_E$ as in the diagram below.



(3.8)

Naturality of θ follows from that of ϵ and ϵ' using again the fact that the components of ϵ' are cartesian. Finally, θ commutes with the counits by definition, and it does so with the comultiplications since these are canonically determined by counits as in Remark 3.3.3.2. This action is clearly functorial in H and C, and it is so in ζ since θ is defined by a universal property.

To conclude the definition of the desired 2-functor, we show that a 2-cell $(\phi, \psi)\colon (H_1, B_1, \zeta^1) \Rightarrow (H_2, B_2, \zeta^2)$ in **CompCat** is also a 2-cell $(\phi, \psi)\colon (H_1, B_1, \theta^1) \Rightarrow (H_2, B_2, \theta^2)$ in **wcCmd**. As $(\phi, \psi)\colon (H_1, B_1) \Rightarrow (H_2, B_2)$ is a 2-cell in **Fib**, it only remains to check that $\phi * \theta^1 = \theta^2 * \phi$. This amounts to verifying that, for every E over X, the left-hand square in the left-hand diagram below commutes.



But this follows from the fact that ϵ'_{H_2E} is cartesian once we show that the other faces and the right-hand diagram commute. The right-hand diagram commutes by Remark 3.3.1.7, the two triangles commute by definition of θ (3.8), and the back and front squares by naturality of ϕ and ϵ', respectively. Functoriality is trivial. □

Lemma 3.3.4.2. There is a 2-functor $\mathbf{Y}\colon \mathbf{wcCmd} \to \mathbf{CompCat}$.

Proof. On objects, it suffices to map a pair $(p\colon \mathcal{E} \to \mathcal{B}, K)$ to $\chi\colon \mathcal{E} \to \mathcal{B}^2$, $\chi(E) := p\epsilon_E$.

To define its action on a 1-cell (H, C, θ), we use θ to induce a suitable

$\zeta : C^2\chi \Rightarrow \chi' H$ as follows:

$$\begin{array}{ccc} CpKE & \xrightarrow{Cp\epsilon_E} & CpE \\ {\scriptstyle \mathrm{id}_{KE}}\downarrow & & \\ p'HKE & & \downarrow{\scriptstyle \mathrm{id}_E} \\ {\scriptstyle p'\theta_E}\downarrow & & \\ p'K'HE & \xrightarrow{p'\epsilon'_{HE}} & p'HE \end{array}$$

on the top row we read $C^2\chi$, on the bottom $\chi' H$, and the square commutes because by hypothesis $Cp = p'H$. With this definition, proving that such functor maps 2-cells to 2-cells is straightforward. □

Theorem 3.3.4.3 (Based on [Jac99], Thm 9.3.4)**.** The two 2-functors $\mathbf{Y} : \mathbf{wcCmd} \leftrightarrows \mathbf{CompCat} : \mathbf{X}$ give rise to an adjoint biequivalence, meaning that there are natural isos $\zeta \colon \mathbf{YX} \xRightarrow{\sim} \mathrm{Id}$ and $\xi \colon \mathbf{XY} \xRightarrow{\sim} \mathrm{Id}$ such that

$$\zeta\mathbf{Y} = \mathbf{Y}\xi \quad \text{and} \quad \xi\mathbf{X} = \mathbf{X}\zeta.$$

Proof. We just need to show that they are 2-inverses. One composition is precisely the identity

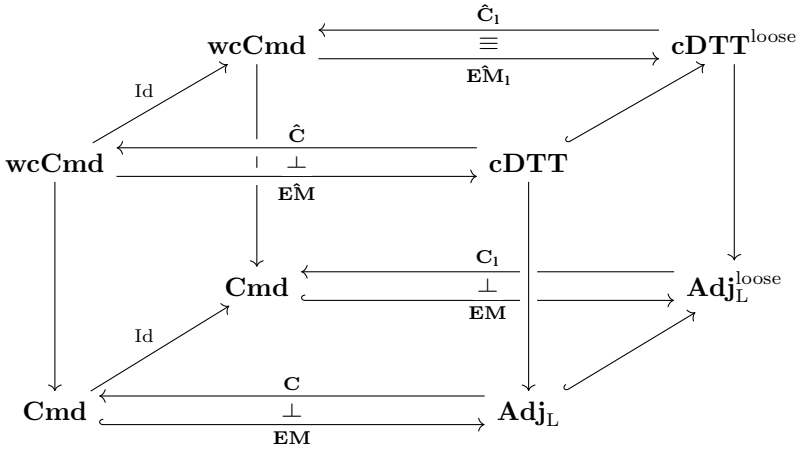
$$(\mathbf{YX}\chi)_E = p(\overline{\chi_E}) = \chi_E,$$

so that we can define $\zeta = \mathrm{id}$. The other leads to compare (cartesian) ϵ_E to the cartesian lift of $p\epsilon_E$: they clearly induce a vertical isomorphism that extends to a suitable 2-isomorphism $\xi \colon \mathbf{XY} \cong \mathrm{Id}$.

Provided that ζ is the identity and ξ is vertical, the desired equations follow from the fact that both $\mathbf{YXY} = \mathrm{Id}$ and $\mathbf{XYX} = \mathrm{Id}$. □

3.3.5 wcComonads v cDTTs

Theorem 3.3.5.1. The adjunction in Theorem 3.2.2.4 lifts to a 2-adjunction between **wcCmd** and **cDTT**, and to an adjoint biequivalence between **wcCmd** and $\mathbf{cDTT}^{\mathrm{loose}}$ as shown in the diagram below,



where the vertical arrows are the obvious functors that forget the fibration from the objects of the 2-categories in the upper square, and act similarly on 1-cells and 2-cells.

The rest of the section is devoted to the proof of Theorem 3.3.5.1. We begin with two lemmas ensuring that the 2-functors **EM** and **C** lift to 2-functors between **cDTT** and **wcCmd**.

Lemma 3.3.5.2.

1. If (p, K, ϵ, ν) is a wc-comonad, then $p\mathrm{U}_K \colon \mathrm{CoAlg}(K) \to \mathcal{B}$ is a fibration.
2. If $(B, H, \theta) \colon (p, K, \epsilon, \nu) \to (p', K', \epsilon', \nu')$ is a lax morphism of wc-comonads, then $(B, \mathrm{CoAlg}(H, \theta)) \colon p\mathrm{U}_K \to p'\mathrm{U}_{K'}$ is a morphism of fibrations.

Proof. 1. Consider the Eilenberg-Moore adjunction associated to (K, ϵ, ν)

$$\mathrm{CoAlg}(K) \underset{\mathrm{U}_K}{\overset{\mathrm{R}_K}{\underset{\longrightarrow}{\overset{\longleftarrow}{\top}}}} \mathcal{E}$$

and let Γ in $\mathcal{B}$, $e \colon E \to KE$ a coalgebra, and $\sigma \colon \Gamma \to p\mathrm{U}_K(E, e) = pE$. A $p\mathrm{U}_K$-cartesian lifting is a pullback square insisting on e. First of all, we can lift σ to a p-cartesian $s \colon E\sigma \to E$ arrow in $\mathcal{E}$. Now, since K preserves cartesian maps, if there is any map $e\sigma$ making the left-hand diagram in (3.9) commute

$$\begin{array}{ccc} E\sigma & \xrightarrow[s]{} & E \\ {\scriptstyle e\sigma}\downarrow & & \downarrow{\scriptstyle e} \\ K(E\sigma) & \xrightarrow[K(s)]{} & K(E) \end{array} \qquad\qquad \begin{array}{ccc} K(E\sigma) & \xrightarrow[K(s)]{} & K(E) \\ {\scriptstyle \epsilon_{E\sigma}}\downarrow & & \downarrow{\scriptstyle \epsilon_E} \\ E\sigma & \xrightarrow[s]{} & E \end{array} \tag{3.9}$$

then such square is immediately a pullback, hence a cartesian lifting of σ. Therefore it is sufficient to equip $E\sigma$ with such a coalgebra structure map $e\sigma$.

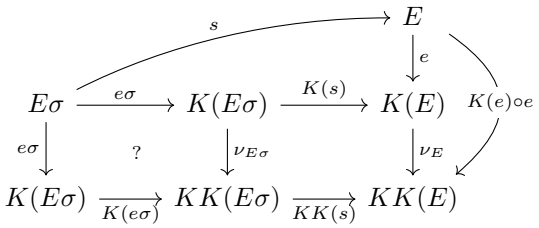
The right-hand square in (3.9) is a pullback by Remark 3.3.3.2, therefore the span

$$E\sigma \xleftarrow{id} E\sigma \xrightarrow{e \circ s} E$$

induces a map $e\sigma$ such that

$$\epsilon_{E\sigma} \circ e\sigma = \mathrm{id}_{E\sigma} \qquad \text{and} \qquad K(s) \circ e\sigma = e \circ \sigma. \tag{3.10}$$

The left-hand equation in (3.10) is counitality for $e\sigma$. To show comultiplication consider the following diagram:



Each square but the desired one commutes because of (3.10), naturality of ν, and the fact that (E, e) is a coalgebra for K. Then the inner left-hand square commutes since $KK(s)$ is cartesian.

2. Clearly, $p'\mathrm{U}_{K'}\mathrm{CoAlg}(H, \theta) = Bp\mathrm{U}_K$ by definition of $\mathrm{CoAlg}(H, \theta)$ after Theorem 3.2.2.4 and since (B, H) is a morphism of fibrations. It only remains to show that $\mathrm{CoAlg}(H, \theta) \colon \mathrm{CoAlg}(K) \to \mathrm{CoAlg}(K')$ preserves cartesian arrows. But cartesian arrows between coalgebras are pullback squares, and these are clearly preserved by $\mathrm{CoAlg}(H, \theta)$. □

Corollary 3.3.5.3. The 2-functor $\mathbf{EM}\colon \mathbf{Cmd} \to \mathbf{Adj}_{\mathrm{L}}^{\mathrm{loose}}$ lifts to a 2-functor

$$\begin{array}{ccc} \mathbf{wcCmd} & \xrightarrow{\hat{\mathbf{EM}}_{\mathbf{l}}} & \mathbf{cDTT}^{\mathrm{loose}} \\ (p,K,\epsilon,\nu) & & (p, p\mathrm{U}_K, \mathbf{EM_l}(K,\epsilon,\nu)) \\ {\scriptstyle (C,H,\theta)}\downarrow & \longmapsto & \downarrow{\scriptstyle (C,H,\mathrm{CoAlg}(H,\theta))} \\ (p',K',\epsilon',\nu') & & (p', p'\mathrm{U}_{K'}, \mathbf{EM_l}(K',\epsilon',\nu')) \end{array}$$

Proof. First, we need to verify that $(p, p\mathrm{U}_K, \mathbf{EM_l}(K,\epsilon,\nu))$ is a cDTT. Thanks to Lemma 3.3.5.2.1, it only remains to show that the components of the unit and counit of $\mathrm{U}_K \dashv \mathrm{R}_K$ are cartesian arrows. For the counit this holds by assumption, and the component of the unit at a coalgebra is the coalgebra structure map, which is cartesian by Remark 3.3.3.3.

Given a lax morphism of wc-comonads $(C,H,\theta)\colon (p,K,\epsilon,\nu) \to (p',K',\epsilon',\nu')$, we have that (C,H) is a morphism of fibrations by assumption, $(C, \mathrm{CoAlg}(H,\theta))$ is a morphism of fibrations by Lemma 3.3.5.2-2, and $(H, \mathrm{CoAlg}(H,\theta)) = \mathbf{EM}(H,\theta)$ is a left morphism of adjunctions by Theorem 3.2.2.4. This proves that $(C,H,\mathrm{CoAlg}(H,\theta))$ is a cDTT morphism.

Given a 2-cell $(\gamma,\phi)\colon (C_1,H_1,\theta_1) \Rightarrow (C_2,H_2,\theta_2)$ in $\mathbf{wcCmd}$, the pair $(\phi, \mathrm{CoAlg}(\phi))$ is a 2-cell in $\mathbf{Fib}$, and thus $(\gamma,\phi,\mathrm{CoAlg}(\phi))$ is a 2-cell in $\mathbf{cDTT}^{\mathrm{loose}}$. □

We now turn to the 2-functor from adjunctions to comonads.

Lemma 3.3.5.4. If $(u,\dot{u},\Sigma,\Delta)$ is a cDTT, then for every cartesian arrow $f\colon A \to B$ in $\mathcal{U}$ the square

$$\begin{array}{ccc} X.A & \xrightarrow{u\epsilon_A} & X \\ {\scriptstyle \dot{u}\Delta f}\downarrow & & \downarrow{\scriptstyle uf} \\ Y.B & \xrightarrow{u\epsilon_B} & Y \end{array}$$

is a pullback in $\mathcal{B}$.

Proof. Let $k\colon Z \to X$ and $h\colon Z \to pKB$ be such that $(u\epsilon_B)h = (uf)k$ and consider a cartesian arrow $b\colon M \to \Delta B$ in $\dot{\mathcal{U}}$ over h. The arrow induced by h and k will be the image under $\dot{u}$ of a (cartesian) arrow $d\colon M \to \Delta A$ in $\dot{\mathcal{U}}$ such that $(\Delta f)d = b$. Note first that, since f is cartesian, there is a unique arrow $a\colon \Sigma M \to A$ in $\mathcal{U}$ over k such that the left-hand diagram in (3.11) commutes. In particular, a is cartesian since f and $\epsilon_B(\Sigma b)\colon \Sigma M \to B$ are.

$$\begin{array}{ccc} \Sigma M & \xrightarrow{a} & A \\ {\scriptstyle \Sigma b}\downarrow & & \downarrow{\scriptstyle f} \\ \Sigma\Delta B & \xrightarrow[\epsilon_B]{} & B \end{array} \qquad \begin{array}{ccc} M & \xrightarrow{a^\#} & \Delta A \\ {\scriptstyle b}\downarrow & & \downarrow{\scriptstyle \Delta f} \\ \Delta B & \xrightarrow[\mathrm{id}_{\Delta B}]{} & \Delta B \end{array} \qquad \begin{array}{ccc} \Sigma M & \xrightarrow{a} & A \\ {\scriptstyle \Sigma a^\#}\downarrow & & \downarrow{\scriptstyle \mathrm{id}_A} \\ \Sigma\Delta A & \xrightarrow[\epsilon_A]{} & A \end{array} \tag{3.11}$$

Transposing the left-hand square in (3.11) yields the central one, while transposing a trivial square involving $a^\#$ yields the right-hand one. It follows that all three squares in (3.11) commute.

Define

$$d := a^\# \colon M \to \Delta A,$$

which is cartesian because $d = \Delta a \circ \eta_M$, the unit is cartesian and Δ preserves cartesian maps by Lemma 3.3.2.5. Commutativity of the central and right-hand square in (3.11) entails that $(\dot{u}\Delta f)(\dot{u}d) = h$ and $(u\epsilon_A)(\dot{u}d) = k$, respectively. We are left to prove that $\dot{u}d$ is the unique such.

Let $l\colon Z \to X.A$ be such that $(\dot{u}\Delta f)l = h$ and $(u\epsilon_A)l = k$. Since Δf is cartesian, there is a unique arrow $l'\colon M \to \Delta A$ over l such that $(\Delta f)l' = b$. Transposing as above yields $fl'^{\#} = \epsilon_B(\Sigma b) = fa$. As $u(l'^{\#}) = u(\epsilon_A(\Sigma l')) = k$, it follows that $l'^{\#} = a$, and thus $l' = d$. □

Corollary 3.3.5.5. The 2-functor $\mathbf{C}\colon \mathbf{Adj}_{\mathrm{L}}^{\mathrm{loose}} \to \mathbf{Cmd}$ lifts to a 2-functor

$$\begin{array}{ccc} \mathbf{cDTT}^{\mathrm{loose}} & \xrightarrow{\quad\hat{\mathbf{C}}_{\mathbf{l}}\quad} & \mathbf{wcCmd} \\ (u, \dot{u}, \Sigma, \Delta) & & (u, \mathbf{C}_{\mathbf{l}}(\Sigma, \Delta)) \\ {\scriptstyle (C,F,G,\zeta)}\downarrow & \longmapsto & \downarrow{\scriptstyle (C,\mathbf{C}_{\mathbf{l}}(F,G,\zeta))} \\ (u', \dot{u}', \Sigma', \Delta') & & (u', \mathbf{C}_{\mathbf{l}}(\Sigma', \Delta')) \end{array}$$

Proof. To verify that, when $(u, \dot{u}, \Sigma, \Delta)$ is a cDTT, the comonad $\mathbf{C}(\Sigma, \Delta)$ is a wc-comonad with fibration u. Condition 1 in Definition 3.3.3.1 is satisfied since the counit is cartesian by assumption. Condition 2 in Definition 3.3.3.1 is in Lemma 3.3.5.4.

To verify that $(C, \hat{\mathbf{C}}_{\mathbf{l}}(F, G, \zeta))$ is a morphism of wc-comonads whenever (C, F, G, ζ) is a 1-cell in $\mathbf{cDTT}^{\mathrm{loose}}$ note that the functor component of $\hat{\mathbf{C}}_{\mathbf{l}}(F, G, \zeta)$ is F. Thus (C, F) is a morphism of fibrations by assumption and $\hat{\mathbf{C}}_{\mathbf{l}}(F, G, \zeta)$ is a lax morphism of comonads by definition.

It remains to prove that $(\gamma, \phi)\colon (C_1, \hat{\mathbf{C}}_{\mathbf{l}}(F_1, G_1, \zeta_1)) \to (C_2, \hat{\mathbf{C}}_{\mathbf{l}}(F_2, G_2, \zeta_2))$ is a 2-cell in $\mathbf{wcCmd}$ whenever $(\gamma, \phi, \psi)\colon (C_1, F_1, G_1, \zeta_1) \to (C_2, F_2, G_2, \zeta_2)$ is a 2-cell in $\mathbf{cDTT}^{\mathrm{loose}}$. This follows from condition 3 in Definition 3.3.2.7 as below.

$$\Sigma'\Delta'\phi \circ \Sigma'\zeta_1^{\#} = \Sigma'\zeta_2^{\#} \circ \Sigma'\psi\Delta = \Sigma'\zeta_2^{\#} \circ \phi\Sigma\Delta$$

When $\zeta_i\colon \Sigma' G_i \xRightarrow{\sim} F_i\Sigma$, and $(\phi\Sigma)\zeta_1 = \zeta_2(\Sigma'\psi)$, it follows that $\Delta'\phi \circ \zeta_1^{\#} = \zeta_2^{\#} \circ \psi\Delta$ and

$$\Sigma'\Delta'\phi \circ \Sigma'\zeta_1^{\#} \circ \zeta_1^{-1}\Delta = \Sigma'\zeta_2^{\#} \circ \Sigma'\psi\Delta \circ \zeta_1^{-1}\Delta = \Sigma'\zeta_2^{\#} \circ \zeta_2^{-1}\Delta \circ \phi\Sigma\Delta. \qquad \square$$

Proof of Theorem 3.3.5.1. From Corollary 3.3.5.3 and Corollary 3.3.5.5 we have 2-functors $\mathbf{E}\hat{\mathbf{M}}_{\mathbf{l}}\colon \mathbf{wcCmd} \leftrightarrows \mathbf{cDTT}^{\mathrm{loose}} : \hat{\mathbf{C}}$. We begin by showing that the 2-adjunction from Theorem 3.2.2.4 and the biadjunction from Theorem 3.2.3.4 lift as well.

As in Theorem 3.2.2.4, the composite $\hat{\mathbf{C}}_{\mathbf{l}} \circ \mathbf{E}\hat{\mathbf{M}}_{\mathbf{l}}$ is the identity on $\mathbf{wcCmd}$. Next, we show that the unit $\boldsymbol{\eta}$ of $\mathbf{EM} \dashv \mathbf{C}$ lifts to a pseudo-natural transformation $\hat{\boldsymbol{\eta}}\colon \mathbf{Id}_{\mathbf{Cmd}} \to \hat{\mathbf{C}}_{\mathbf{l}} \circ \mathbf{E}\hat{\mathbf{M}}_{\mathbf{l}}$. The component of the unit at an adjunction (L, R, η, ϵ) is the left morphism of adjunctions $(\mathrm{Id}, \mathrm{K}_{L,R})$, where $\mathrm{K}_{L,R}$ is the canonical comparison functor (3.4). Given a cDTT $(u, \dot{u}, \Sigma, \Delta)$, the functor $\mathrm{K}_{\Sigma,\Delta}$ preserves cartesian arrows since Σ does and coalgebra structure maps are cartesian by Remark 3.3.3.3. It follows that the triple $(\mathrm{Id}_{\mathcal{C}}, \mathrm{Id}_{\mathcal{U}}, \mathrm{K}_{\Sigma,\Delta})$ is a cDTT morphism. To see that this choice is pseudo-natural in $(u, \dot{u}, \Sigma, \Delta)$, let $(C, F, G, \zeta)\colon (u, \dot{u}, \Sigma, \Delta) \to (u', \dot{u}', \Sigma', \Delta')$ be a loose cDTT morphism. The required invertible 2-cell is $(\mathrm{id}_C, \mathrm{id}_F, \hat{\zeta})\colon (C, F, \mathrm{K}_{\Sigma',\Delta'}G, \zeta) \to (C, F, \mathrm{CoAlg}(\mathbf{C}_{\mathbf{l}}(F, G, \zeta))\mathrm{K}_{\Sigma,\Delta}, \mathrm{id})$,

where $\hat{\zeta}$ is the natural iso from Remark 3.2.3.3. Clearly, if $\zeta = \mathrm{id}$ so is the invertible 2-cell, meaning that $\hat{\boldsymbol{\eta}}$ is natural with respect to cDTT morphisms.

The triangular identities follow immediately from those of $\mathbf{C} \dashv \mathbf{EM}$ in (3.5):

$$\hat{\mathbf{C}}_{\mathbf{l}}\hat{\boldsymbol{\eta}}_{u,\dot{u},\Sigma,\Delta} = (\mathrm{Id}_{\mathcal{C}}, \mathbf{C}\boldsymbol{\eta}_{\Sigma,\Delta}) = \mathbf{id}_{\hat{\mathbf{C}}_{\mathbf{l}}(u,\dot{u},\Sigma,\Delta)}$$
$$\hat{\boldsymbol{\eta}}_{\mathbf{E}\hat{\mathbf{M}}_{\mathbf{l}}(K,\epsilon,\nu)} = (\mathrm{Id}_{\mathcal{B}}, \boldsymbol{\eta}_{\mathbf{EM}(K,\epsilon,\nu)}) = \mathbf{id}_{\mathbf{E}\hat{\mathbf{M}}_{\mathbf{l}}(K,\epsilon,\nu)}$$

It remains to show that the biadjunction $\hat{\mathbf{C}}_{\mathbf{l}} \dashv \mathbf{E}\hat{\mathbf{M}}_{\mathbf{l}}$ is in fact a biequivalence. This amounts to show that the unit $\hat{\boldsymbol{\eta}}$ is a pseudo-natural equivalence. Lemma 3.3.5.7 entails that each component of $\hat{\boldsymbol{\eta}}$ is an equivalence in $\mathbf{cDTT}^{\mathrm{loose}}$, with weak inverse given by the loose cDTT morphism $(\mathrm{Id}, \mathrm{Id}_{\mathcal{U}}, \mathrm{J}_{\Sigma,\Delta}, \mathrm{U}_{\Sigma\Delta}\xi)$ and invertible two cells given by

$$(\mathrm{Id}_{\mathcal{C}}, \mathrm{Id}_{\mathcal{U}}, \mathrm{KJ}, \mathrm{U}\xi) \xrightarrow{(\mathrm{id},\mathrm{id},\xi)} (\mathrm{Id}_{\mathcal{C}}, \mathrm{Id}_{\mathcal{U}}, \mathrm{Id}_{\mathrm{CoAlg}(\Sigma\Delta)})$$

$$(\mathrm{Id}_{\mathcal{C}}, \mathrm{Id}_{\mathcal{U}}, \mathrm{JK}, \mathrm{U}\xi\mathrm{K}) \xrightarrow{(\mathrm{id},\mathrm{id},\zeta)} (\mathrm{Id}_{\mathcal{C}}, \mathrm{Id}_{\mathcal{U}}, \mathrm{Id}_{\dot{\mathcal{U}}})$$

As $\hat{\boldsymbol{\eta}}$ is pseudo-natural, so is the family of its weak inverses. This concludes the proof. □

Remark 3.3.5.6. One can retrace the proof of Theorem 3.3.5.1 and see that it suitably restricts to the lax case or, equivalently, that the adjunction $\mathbf{C} \dashv \mathbf{EM}$ in Theorem 3.2.2.4 lifts to cDTTs with lax morphisms, too.

We now wish to show that the adjunction in Theorem 3.3.5.1 actually produces an equivalence. Notice that what follows requires that the term fibration $\dot{u}$ is cloven, and since Σ preserves cartesian maps u is automatically cloven, too. Recall that a similar condition appears in Section 3.3.4.

Lemma 3.3.5.7. There are a functor $\mathrm{J}_{\Sigma,\Delta} \colon \mathrm{CoAlg}(\Sigma\Delta) \to \dot{\mathcal{U}}$ and natural isos $\zeta \colon \mathrm{J}_{\Sigma,\Delta}\mathrm{K}_{\Sigma,\Delta} \xRightarrow{\sim} \mathrm{Id}_{\dot{\mathcal{U}}}$ and $\xi \colon \mathrm{K}_{\Sigma,\Delta}\mathrm{J}_{\Sigma,\Delta} \xRightarrow{\sim} \mathrm{Id}_{\mathrm{CoAlg}(\Sigma\Delta)}$ making $\mathrm{K}_{\Sigma,\Delta}$ and $\mathrm{J}_{\Sigma,\Delta}$ into an adjoint equivalence of categories, meaning that also

$$\mathrm{K}_{\Sigma,\Delta}\zeta = \xi\mathrm{K}_{\Sigma,\Delta} \qquad \text{and} \qquad \mathrm{J}_{\Sigma,\Delta}\xi = \zeta\mathrm{J}_{\Sigma,\Delta}.$$

Proof. The functor $\mathrm{J}_{\Sigma,\Delta}$ is defined on a coalgebra $h \colon A \to \Sigma\Delta A$ as the reindexing of ΔA along uh. The action on a morphism of coalgebra f is induced accordingly using the cartesian lifting $\overline{uh}$ that defines $\mathrm{J}_{\Sigma,\Delta}h$ as depicted below, where both squares above, in $\mathcal{U}$ and $\dot{\mathcal{U}}$ respectively, sit on the square below.

$$\begin{array}{ccc} B & \xrightarrow{k} & \Sigma\Delta B \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle \Sigma\Delta f} \\ A & \xrightarrow{h} & \Sigma\Delta A \end{array} \qquad\qquad \begin{array}{ccc} \mathrm{J}_{\Sigma,\Delta}k & \xrightarrow{\overline{uk}} & \Delta B \\ {\scriptstyle \mathrm{J}_{\Sigma,\Delta}f}\downarrow & & \downarrow{\scriptstyle \Delta f} \\ \mathrm{J}_{\Sigma,\Delta}h & \xrightarrow{\overline{uh}} & \Delta A \end{array}$$

$$\begin{array}{ccc} Y & \xrightarrow{uk} & Y.B \\ {\scriptstyle uf}\downarrow & & \downarrow{\scriptstyle \dot{u}\Delta f} \\ X & \xrightarrow{uh} & X.A \end{array}$$

As the action on arrows is defined by a universal property, functoriality of $\mathrm{J}_{\Sigma,\Delta}$ is straightforward.

Recall that $\mathrm{K}_{\Sigma,\Delta}a = \Sigma\eta_a$. Therefore $\mathrm{J}_{\Sigma,\Delta} \circ \mathrm{K}_{\Sigma,\Delta}a$ is defined as the domain of a cartesian lifting of $\dot{u}\eta_a$ in $\dot{\mathcal{U}}$. But the component η_a of the unit at an object a of $\dot{\mathcal{U}}$ is also cartesian. Therefore there is a unique vertical iso $\zeta_a \colon \mathrm{J}_{\Sigma,\Delta}\Sigma\eta_a \to a$ such that $\eta_a\zeta_a = \overline{\dot{u}\eta_a}$. Naturality of ζ_a in a is straightforward. It follows that $\zeta \colon \mathrm{J}_{\Sigma,\Delta}\mathrm{K}_{\Sigma,\Delta} \xRightarrow{\sim} \mathrm{Id}_{\dot{\mathcal{U}}}$.

On the other hand, since Σ preserves cartesian arrows, both h and $\Sigma\overline{uh}$ are cartesian over uh in $\mathcal{U}$. It follows that there is a unique vertical iso $\xi_h \colon \Sigma\mathrm{J}_{\Sigma,\Delta}h \to A$ such that $h\xi_h = \Sigma\overline{uh}$. Again, naturality of ξ_h in h is straightforward and it follows that $\xi \colon \Sigma\mathrm{J}_{\Sigma,\Delta} \xRightarrow{\sim} \mathrm{U}_{\Sigma\Delta}$. To obtain a natural iso $\mathrm{K}_{\Sigma,\Delta}\mathrm{J}_{\Sigma,\Delta} \xRightarrow{\sim} \mathrm{Id}_{\mathrm{CoAlg}(\Sigma\Delta)}$, it is enough to show that ξ_h is in fact a morphism, and thus an iso, of coalgebras from $\Sigma\eta_{\mathrm{J}_{\Sigma,\Delta}h}$ to h. This amounts to the commutativity of the square below.

$$\begin{array}{ccc} \Sigma\mathrm{J}_{\Sigma,\Delta}h & \xrightarrow{\xi_h} & A \\ {\scriptstyle \Sigma\eta_{\mathrm{J}h}}\downarrow & \searrow^{\Sigma\overline{uh}} & \downarrow{\scriptstyle h} \\ \Sigma\Delta\Sigma\mathrm{J}_{\Sigma,\Delta}h & \xrightarrow[\Sigma\Delta\xi_h]{} & \Sigma\Delta A \end{array}$$

The upper-right triangle commutes by definition of ξ_h. The lower-left triangle is the image under Σ of the left-hand square below, which is the transpose under $\Sigma \dashv \Delta$ of the right-hand square.

$$\begin{array}{ccc} \mathrm{J}_{\Sigma,\Delta}h & \xrightarrow{\eta_{\mathrm{J}h}} & \Delta\Sigma\mathrm{J}_{\Sigma,\Delta}h \\ {\scriptstyle \overline{uh}}\downarrow & & \downarrow{\scriptstyle \Delta\xi_h} \\ \Delta A & \xrightarrow[\mathrm{id}_A]{} & \Delta A \end{array} \qquad \begin{array}{ccc} \Sigma\mathrm{J}_{\Sigma,\Delta}h & \xrightarrow{\mathrm{id}_{\mathrm{J}h}} & \Sigma\mathrm{J}_{\Sigma,\Delta}h \\ {\scriptstyle \Sigma\overline{uh}}\downarrow & & \downarrow{\scriptstyle \xi_h} \\ \Sigma\Delta A & \xrightarrow[\epsilon_A]{} & A \end{array}$$

The right-hand square commutes since $\Sigma\overline{uh} = h\xi_h$ and $\epsilon_A h = \mathrm{id}_A$. It follows that the other two squares commute as well.

To see that $\mathrm{K}_{\Sigma,\Delta}\zeta = \xi\mathrm{K}_{\Sigma,\Delta}$ note that, for every $a \in \dot{\mathcal{U}}$, $\Sigma\eta_a \circ \Sigma\zeta_a = \Sigma\overline{\dot{u}\eta_a}$ by definition of ζ_a. It follows that $\Sigma\zeta_a = \xi_{\Sigma\eta_a}$ as required. The other equation $\mathrm{J}_{\Sigma,\Delta}\xi = \zeta\mathrm{J}_{\Sigma,\Delta}$ follows from the fact that $\overline{uh}$ is cartesian and commutativity of the left-hand square above:

$$\overline{uh}\mathrm{J}_{\Sigma,\Delta} \circ \xi_h = \Delta\xi_h \circ \overline{\dot{u}\eta_{\mathrm{J}h}} = \Delta\xi_h \circ \eta_{\mathrm{J}h} \circ \zeta_{\mathrm{J}h} = \overline{uh} \circ \zeta_{\mathrm{J}h}. \qquad \square$$

3.3.6 Examples

3.3.6.1 Other categorical models

Example 3.3.6.1 (Categories with attributes, [Car86, Mog91])**.** A *category with attributes* is the data of

- a category $\mathcal{C}$ with terminal object $\top$;
- a functor $\mathsf{Ty} \colon \mathcal{C}^{\mathsf{op}} \to \mathbf{Set}$;
- for each A in $\mathsf{Ty}(\Gamma)$ an object $\Gamma.A$ and a morphism $\mathsf{p}_A \colon \Gamma.A \to \Gamma$;

- for each $\sigma\colon \Theta \to \Gamma$ and A in $\mathsf{Ty}(\Gamma)$ a pullback diagram

$$\begin{array}{ccc} \Theta.(\mathsf{Ty}\,\sigma)(A) & \xrightarrow{\mathsf{q}_{\sigma,A}} & \Gamma.A \\ {\scriptstyle \mathsf{p}_{(\mathsf{Ty}\,\sigma)(A)}}\downarrow & & \downarrow{\scriptstyle \mathsf{p}_A} \\ \Theta & \xrightarrow[\sigma]{} & \Gamma \end{array}$$

  such that $\mathsf{q}(\mathrm{id}_\Gamma, A) = \mathrm{id}_{\Gamma.A}$ and $\mathsf{q}(\tau \circ \sigma, A) = \mathsf{q}(\tau, (\mathsf{Ty}\,\sigma)(A)) \circ \mathsf{q}(\sigma, A)$.

The attentive reader might have recognized the equations in 3.3.6.1 to be that of the splitting of a fibration, in this case that of types. In fact, one can show [Jac93, Example 4.10] that categories with attributes are equivalent to full split comprehension categories, or to comprehension categories with discrete type fibration.

In an attempt to obtain a model closer to the syntax, and explicitly describing terms, one turns to categories with families. Recall that **Fam** is the category of families of sets, meaning that having for objects pairs (I, B) where I is a set and $B = (B_i)_{i\in I}$ is a family of I-indexed sets and for maps reindexing functions.

Example 3.3.6.2 (Categories with families, [Dyb96])**.** A *category with families* is the data of

- a category $\mathcal{C}$ with terminal object $\top$;
- a functor $F = (\mathsf{Ty}, \mathsf{Tm})\colon \mathcal{C}^{\mathsf{op}} \to \mathbf{Fam}$;
- for each Γ in $\mathcal{C}$ and A in $\mathsf{Ty}(\Gamma)$ an object $\Gamma.A$ in $\mathcal{C}$, together with two projections $\mathsf{p}_A\colon \Gamma.A \to \Gamma$ and $\mathsf{v}_A \in \mathsf{Tm}(\Gamma.A, \mathsf{Ty}\,\mathsf{p}_A(A))$ such that for each $\sigma\colon \Theta \to \Gamma$ and $a \in \mathsf{Tm}(\mathsf{Ty}\sigma(A))$ there exists a unique morphism $\Theta \to \Gamma.A$ making the obvious triangles commute.

In particular,

$$F(\Gamma) = (\ \mathsf{Ty}(\Gamma), (\mathsf{Tm}(\Gamma, A))_{A\in\mathsf{Ty}(\Gamma)}\ )$$

so that types are fibered over contexts and terms are fibered over types. One can show [Hof97, §3] that these are equivalent to categories with attributes.

Getting closer to the syntax allows us to gain in clarity, but makes our definition much heavier. A more categorical perspective allows for a different, perhaps more natural definition.

Example 3.3.6.3 (Natural models, [Awo18])**.** A *natural model* on a small category $\mathcal{C}$ is a representable map of presheaves $p\colon \dot{u} \to u$, meaning a presheaf morphism such that for every Γ in $\mathcal{C}$ and $A \in u(\Gamma)$ there is a $\Gamma.A$ in $\mathcal{C}$, a $\pi_A\colon : \Gamma.A \to \Gamma$, and a $q_A \in \dot{u}(\Gamma.A)$ such that the following square is a pullback.

$$\begin{array}{ccc} \text{ﾖ}\Gamma.A & \xrightarrow{q_A} & \dot{u} \\ {\scriptstyle \text{ﾖ}\pi_A}\downarrow & & \downarrow{\scriptstyle p} \\ \text{ﾖ}\Gamma & \xrightarrow[A]{} & u \end{array}$$

Notice that this all heavily relies on Yoneda lemma applied to (discrete) fibrations (1.6.1.1), so that there is a one-to-one correspondence

$$A \in u(\Gamma) \quad \leftrightarrow \quad A\colon \text{ﾖ}\Gamma \to u\,.$$

These are themselves equivalent [Awo18, Prop. 1.2] to categories with families.

Putting together 3.3.6.1, 3.3.6.2, and 3.3.6.3 it is clear the direction of the increasing level of abstraction and, at the same time, the effort of clearly expressing the data needed to work with dependent types.

We wish to include cDTTs in this midst: their being equivalent to comprehension categories establishes that they are a suitably coherent generalization (away from discrete type fibrations) of categories with families, with attributes, and natural models, but we believe their use of computation makes them still decisively intelligible. The motivation of our care in avoiding discrete types will be clear in Section 3.5.

We conclude this brief exposition by presenting a last strategy for modeling dependent types using categories, based on the idea of display maps, in some sense isolating that of comprehensions for a comprehension category.

Example 3.3.6.4 (Display-map categories, [Tay99, HP87])**.** A *display-map category* is a pair $(\mathcal{C}, \mathcal{D})$ with $\mathcal{C}$ a category and $\mathcal{D} = \{p_A : \Gamma.A \to \Gamma\}$ a class of morphisms in $\mathcal{C}$ called *displays* or *projections* such that:

1. for each $p_A : \Gamma.A \to \Gamma$ in $\mathcal{D}$ and $\sigma : \Theta \to \Gamma$ in $\mathcal{C}$, there exists a choice of a pullback of p_A along σ and it is again in $\mathcal{D}$,

$$\begin{array}{ccc} \Theta.A[s] & \xrightarrow{\overline{\sigma}} & \Gamma.A \\ {\scriptstyle p_{A[\sigma]}}\downarrow & & \downarrow{\scriptstyle p_A} \\ \Theta & \xrightarrow[\sigma]{} & \Gamma \end{array}$$

2. $\mathcal{D}$ is closed under pre and post-composition with isomorphisms.

One can see that these are equivalent to comprehension categories having the comprehension functor be a full inclusion.

In a sense, display map categories forget about all the infrastructure and just focus on the relations one might need: given a type in context, the first thing we need to be able to do is extending that context with such type, the second is substitution, and there is nothing more than that. We will see display-map categories again in Chapter 4.

3.3.6.2 In "nature"

We have many examples of comprehension categories, or wc-comonads, or cDTTs, and their restrictions.

Example 3.3.6.5 (Syntactic model)**.** Given an extensional calculus of dependent types, for example [Mar84, Tro87], one can build a full comprehension category as follows. Let $\mathcal{B}$ the category having for objects (α-)equivalence classes of contexts, denoted $[\Gamma] = x_1 : A_1, \dots, x_n : A_n$, $[\Theta] = y_1 : B_1, \dots, y_m : B_m$ and so on, and such that a morphism

$$t \colon [\Theta] \to [\Gamma] \quad \text{is} \quad \text{(equivalence classes of) terms } [t_1], \dots, [t_n]$$

such that $\Theta \vdash t_i : A_i[t_1/y_1, \dots, t_{i-1}/y_{i-1}]$ for each $i = 1 : n$. Identity of maps is induced by definitional equality. The category $\mathcal{E}$ is defined to have objects $[\Gamma \vdash A]$ (equivalence classes of) typing judgements and arrows $(t, s) \colon [\Theta \vdash B] \to [\Gamma \vdash A]$

with $t\colon [\Theta] \to [\Gamma]$ and $\Theta, y : B \vdash s : A[t/x]$. The functor $\chi\colon \mathcal{E} \to \mathcal{B}^2$ acts as the projection

$$[\Gamma \vdash A] \quad \mapsto \quad [\Gamma, x : A] \to [\Gamma].$$

If the latter example seems little informative, it is because really all validity results are: it just means comprehension categories do a good job at doing what they were designed to do.

Example 3.3.6.6 (Simple fibration)**.** Let $\mathcal{B}$ be a category with products, and consider the category $s(\mathcal{B})$ whose

- objects are pairs (I, X) of objects in $\mathcal{B}$;
- arrows $(J, Y) \to (I, B)$ are pairs (u, f) of maps in $\mathcal{B}$, with $u\colon J \to I$ and $f\colon J \times Y \to X$.

Composition can be defined using the universal property of products, and identities are of the form $(\mathrm{id}, \mathrm{pr}_2)$. The obvious projection functor $s(\mathcal{B}) \to \mathcal{B}$ sending

$$(I, X) \mapsto I \quad \text{and} \quad (u, f) \mapsto u$$

is a fibration. Its fibers are also denoted $s(\mathcal{B})_I = \mathcal{B}_{//I}$, they have objects X in $\mathcal{B}$ and maps $X' \to X$ are $f\colon I \times X' \to X$ in $\mathcal{B}$: one can think of these as I-indexed families $f_i\colon X' \to X$ with fixed domain and codomain.

Such a construction can be reformulated to describe a comprehension category, meaning there is a functor

$$s(\mathcal{B}) \to \mathcal{B}^2, \quad (I, X) \mapsto (\mathrm{pr}_1 : I \times X \to I)$$

satisfying the axioms of a comprehension. It models simple type theory.

Example 3.3.6.7 (Elementary topos)**.** Let $\mathcal{B}$ be a topos with subobject classifier $\top\colon 1 \to \Omega$. Then we can map each "predicate" ϕ to its comprehension monomorphism, meaning

$$\mathcal{B}_{/\Omega} \to \mathcal{B}^2, \quad (\phi\colon A \to \Omega) \mapsto (\{\phi\} \rightarrowtail A)$$

where

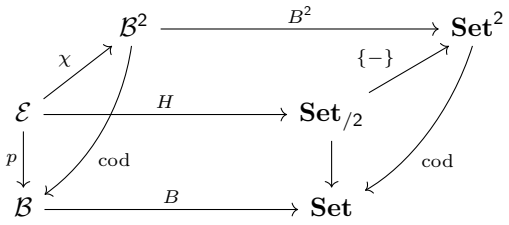
$$\begin{array}{ccc} \{\phi\} & \longrightarrow & 1 \\ \big\downarrow & \lrcorner & \big\downarrow \\ A & \xrightarrow{\phi} & \Omega \end{array}$$

and such an assigment produces a comprehension category.

Remark 3.3.6.8 (Models as morphisms of comprehension categories)**.** As it is customary in the case of doctrines Section 1.7, and since we now have all the necessary instruments, we can see how models (in the traditional, set-based sense) can be related to special morphisms of structures encoding what we are interested in.

In particular, we think of models of a given dependent type theory as strict morphisms from its syntactic comprehension category (3.3.6.5) to that on the

elementary topos of sets (3.3.6.7).



In particular, each (equivalence class of) context(s) is sent to a set, each typing judgement to a subset so that for $[\Gamma \vdash A]$ we have that $H[\Gamma \vdash A] = \phi_A$ satisfies

$$\{\phi_A\} \subseteq B[\Gamma],$$

meaning that the extension of the typing judgement is contained in the extension of the context: we can think of this as saying that the interpretation of A consists in picking up elements in the extension of its context. Moreover, from the top square, we get

$$\{\phi_A\} = B[\Gamma.A],$$

so that the extension of a given A consists precisely in that of its extended context. If we consider lax morphisms, instead, this latter condition turns to a function $B[\Gamma.A] \to \{\phi_A\}$.

Finally, we describe a few particular cDTTs - each time trivializing one of their main elements - as to better understand our model. They will produce some interesting type theories.

Example 3.3.6.9 (Types in no context)**.** Consider $\mathcal{B} = 1$ and both $u, \dot{u}$ the unique terminal functors. They are fibrations where the only cartesian maps are the vertical isomorphisms, hence η, ϵ are themselves isos and Σ, Δ form an equivalence. Each type is closed and inhabited. This is in initial in cDTTs whose $\mathcal{B}$ has a terminal object and morphisms preserve them.

Example 3.3.6.10 (Non-dependent types)**.** If $\mathcal{B} = \mathcal{U}$ and $u = \mathrm{id}$, it follows that $\Sigma = \dot{u}$ hence a fibration with a right adjoint functor Δ, so that $\dot{\mathcal{U}}$ is non-trivially fibered over $\mathcal{U}$ via Σ. The counit is of course cartesian, so we only ask that η is. Each type has its own context (itself) and can be indefinitely extended with itself, but we can no longer describe judgements of the form

$$\Gamma.A \vdash B,$$

so that there is no real type dependency.

3.3.7 Main theorem

To make our presentation smoother, for this section we assume that all fibrations are cloven. If not, the following result holds in a weaker, but informatively equivalent, form.

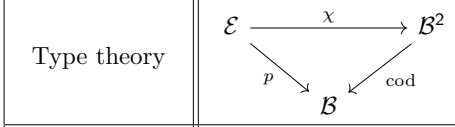
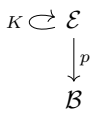
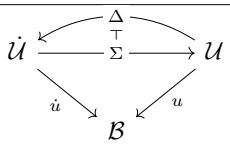
Theorem 3.3.7.1. There is a biequivalence $\mathbf{CompCat} \equiv \mathbf{cDTT}^{\mathrm{loose}}$.

Proof. Trivial from Theorem 3.3.4.3 and Theorem 3.3.5.1. □

We could interpret such a result as the following statement:

all terms are sections,

in the sense that even when one decides to treat terms and types as particular objects of different statuses, such as belonging to different categories, the relation that exists between them forces them to be sections and projections. Provided that many models (*e.g.* categories with families and natural models) do not deal with terms using sections of the context projection maps *explicitly*, this might be slightly surprising, but in fact one can use the universal properties of pullbacks involved in the respective definitions to recover sections from terms, see for example [CD11, §3]. We conclude our discussion by fixing some notation for the (equivalent) models.

Type theory			
contexts	$\mathsf{Ob}(\mathcal{B})$	$\mathsf{Ob}(\mathcal{B})$	$\mathsf{Ob}(\mathcal{B})$
types	$\mathsf{Ob}(\mathcal{E})$	$\mathsf{Ob}(\mathcal{E})$	$\mathsf{Ob}(\mathcal{U})$
$\Gamma \vdash A$	$pA = \Gamma$	$pA = \Gamma$	$uA = \Gamma$
$\Gamma.A \to \Gamma$	χ_A	$p\epsilon_A$	$u\epsilon_A$
$\Gamma.A$	$\mathrm{dom}(\chi_A)$	pKA	$u\Sigma\Delta A$
A^+ (A in $\Gamma.A$)	$\chi_A^*(A)$	KA	$\Sigma\Delta A$
terms	sections	sections	$\mathsf{Ob}(\dot{\mathcal{U}})$
$\Gamma \vdash a : A$	section of χ_A	section of ϵ_A	$\Sigma a = A$

3.4 Properties of comprehension categories

We now explore what happens to Theorem 3.3.7.1, first when we add more restrictions to the respective 1-cells, then when we ask specific properties of the types and terms fibrations.

3.4.1 The empty context

As we have mentioned in Remark 3.1.1.1, one often wants to be able to interpret the empty context, hence it makes sense to ask that the categories of contexts have a terminal object, and that all fibration morphisms involved in **cDTT**, **wcCmd**, **CompCat** preserve them. Given that Theorem 3.3.7.1 is the identity on the type fibration, restricting the equivalence to such 0- and 1-cells produces an equivalence, still: the only non trivial part is that the induced functor of coalgebras does, but that too is immediate by the definition in 3.2.1.2.

3.4.2 Lax, pseudo, strict

As for Section 3.3.4, assume that our type fibrations are cloven.

Proposition 3.4.2.1. Theorem 3.3.4.3 restricts to:

- the respective pseudo 2-subcategories, and

- the respective strict 2-subcategories with cleavage-preserving fibration morphisms.

Proof. Let $(H, C, \zeta)\colon (p, \chi) \to (p', \chi')$. If ζ is invertible, then the induced θ

$$\begin{array}{ccc} K'HE & & \\ {\scriptstyle\theta_E}\downarrow & \searrow^{H(\epsilon_E)} & \\ HKE & \xrightarrow{\epsilon'_{HE}} & HE \end{array} \qquad \begin{array}{ccc} BX_E & & \\ {\scriptstyle\zeta_E}\downarrow & \searrow^{B(\chi_E)} & \\ X'_{HE} & \xrightarrow{\chi'_{HE}} & BX \end{array}$$

is invertible because both ϵ'_{HE} and $H(\epsilon_E)$ are cartesian and p' is cloven. When $\zeta_E = \mathrm{id}$, the iso θ_E is an identity because $K'HE = HKE$ since H preserves cleavages. □

Proposition 3.4.2.2. The equivalence in Theorem 3.3.5.1 restricts to the respective pseudo and strict 2-subcategories.

Proof. The case of going from cDTTs to wc-comonads is trivial because $\theta = \Sigma'\gamma$. For the opposite, consider $\theta\colon HK \Rightarrow K'H$ iso. We need to show that the canonical γ described in Definition 3.3.2.7

$$\gamma_A\colon \hat{H}CA \xrightarrow{\eta'_{\hat{H}CA}} C'U'\hat{H}CA \xrightarrow{\mathrm{id}} C'HUCA \xrightarrow{C'(H\epsilon_A)} C'HA$$

is an isomorphism. We will prove that it isomorphic to a (triangular) identity. Clearly θ iso implies that $H\epsilon \cong \epsilon'_H$, hence $C'H\epsilon_A \cong C'\epsilon'_{HA}$. Not only that, but also $\hat{H}C \cong C'H$, because the following diagram commutes by definition of comonad morphism.

$$\begin{array}{ccc} HKA & \xrightarrow[\theta_A]{\sim} & K'HA \\ {\scriptstyle H\nu_A}\downarrow & & \\ HKKA & & \downarrow{\scriptstyle \nu'_{HA}=C'H(A)} \\ {\scriptstyle\theta_{KA}}\downarrow & & \\ K'HKA & \xrightarrow[K'(\theta_{KA})]{\sim} & K'K'HA \end{array} \qquad \hat{H}C(A)$$

Therefore we have $\gamma_A \cong C'(\epsilon'_{HA}) \circ \eta'_{C'HA} = \mathrm{id}_{HA}$. If θ is the identity, we have precisely $\gamma_A = C'(\epsilon'_{HA}) \circ \eta'_{C'HA}$. □

Corollary 3.4.2.3. The biequivalence $\mathbf{CompCat} \equiv \mathbf{cDTT}^{\mathrm{loose}}$ restricts to:

- the respective pseudo 2-subcategories, and
- the respective strict 2-subcategories with cleavage-preserving fibration morphisms.

3.4.3 Properties of categorical models of dependent types

3.4.3.1 Discrete

A big portion of the literature on categorical models actually considers the fibration of types to be discrete, as it appears clear reading Section 3.3.6.1.

Remark 3.4.3.1. The biequivalence described in Theorem 3.3.7.1 is the identity on the type fibration.

Hence, while the type fibration remains the same, one could wonder whether discreteness can be induced on the term fibration, as well. We have a partial result, that is the following.

Lemma 3.4.3.2 ([Str22, Lemma 2.1])**.** Let $(u, \dot{u}, \Sigma \dashv \Delta)$ a cDTT with u discrete. Then Σ is a fibration.

Unfortunately, we can in no way prove that, if u is discrete, $\dot{u}$ is itself discrete. This is unpleasant but, after all, discreteness is not stable under equivalence of categories. The "closest" categorical notion we can describe is that of *essentially discrete fibration*, see Definition 1.3.1.5, and we can in fact show that if u is essentially discrete, then so is $\dot{u}$.

Definition 3.4.3.3. Call $\mathbf{CompCat}^{pos}$ the category of comprehension categories where the type fibration is faithful. Call $\mathbf{CompCat}^{gpd}$ the category of comprehension categories where the type fibration reflects isomorphisms.

Lemma 3.4.3.4. $\mathbf{CompCat_{ps}}^{pos} = \mathbf{CompCat_{str}}^{pos}$.

Lemma 3.4.3.5. $\mathbf{CompCat}^{gpd} = \mathbf{CompCat_{ps}}^{gpd}$.

Corollary 3.4.3.6. For a class of fibrations x,

$$\mathbf{CompCat}^{x} = \mathbf{CompCat_{ps}}^{x} = \mathbf{CompCat_{str}}^{x}$$

if and only if x is the class of essentially discrete fibrations.

We now show that, in fact, essential discreteness can be induced from the type to the term fibration. In other terms, we can show the following.

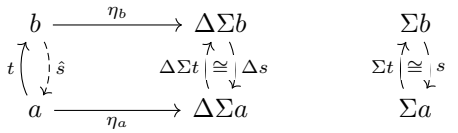
Proposition 3.4.3.7. Theorem 3.3.7.1 restricts to the respective 2-subcategories with essentially discrete terms and types fibrations.

Proof. They are both 1- and 2-full, so it suffices to show what happens to 0-cells. The result follows from Proposition 3.4.3.8 and Remark 3.4.3.1. □

Proposition 3.4.3.8. In every cDTT, if the fibration u is faithful, respectively groupoidal, then so is the fibration $\dot{u}$.

Proof. That $\dot{u}$ is faithful whenever u is follows from the fact that Σ is faithful by Remark 3.3.2.4.

Suppose now that u is groupoidal, and consider a vertical $t\colon a \to b$ in $\dot{\mathcal{U}}$. Then its (still vertical) image through Σ has an inverse $s : \Sigma b \to \Sigma a$ by hypothesis on u. Our claim is that this induces a suitable $\hat{s}$ inverse to t. Out of Σt and s we get the following commutative diagrams and isomorphisms.



Since s is vertical, so is Δs, so that $\Delta s \eta_b$ and η_a lie above the same map. We define $\hat{s}$ to be the unique map induced by cartesianity of η_a. Now,

$$\eta_a \hat{s} t = (\Delta s)(\Delta \Sigma t)\eta_a = \eta_a$$

and η_a is monic by Remark 3.3.2.4, so that $\hat{s}t = \mathrm{id}$. One can similarly show that $t\hat{s} = \mathrm{id}$ and conclude the proof. □

3.4.3.2 Split and full split

Definition 3.4.3.9. Call $\mathbf{CompCat}^{split}$ of comprehension categories where the type fibration is split and morphisms preserve the split cleavage.

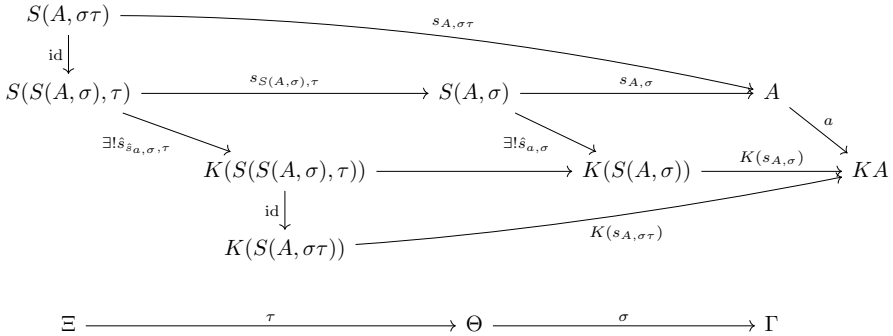
Call $\mathbf{CompCat}^{full\ split}$ that where additionally χ is fully faithful.

We show a result similar to that of the previous subsection.

Lemma 3.4.3.10. Let (p, K) a wc-comonad with p split. Then the induced term fibration pU is itself split and U (trivially) preserves the cleavages.

Proof. Let A an object in $\mathcal{E}$ and $\sigma : \Theta \to \Gamma$ a map in $\mathcal{B}$ with $pA = \Gamma$. We denote by $s_{A,\sigma} : S(A, \sigma) \to A$ our choice of cartesian lifting of σ at A. We can define a cleavage on pU by steps as follows: compute $s_{A,\sigma}$, since this is cartesian and K preserves cartesian maps by Remark 3.3.3.3, $K(s_{A,\sigma})$ is cartesian, then pick the obvious induced unique map $S(A, \sigma) \to K(S(A, \sigma))$ (see the diagram below). We now show that such cleavage is in fact split.

Consider a morphism $\tau : \Xi \to \Theta$, and repeat the procedure above.



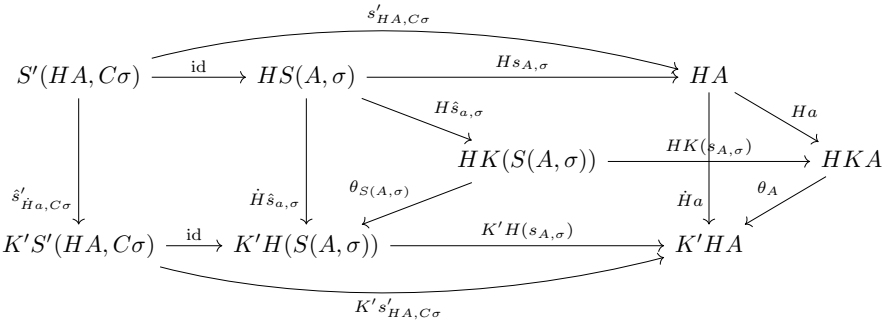
Since the starting cleavage is split both triangles commute, and by uniqueness of the map $S(A, \sigma\tau) \to K(S(A, \sigma\tau))$ we deduce that our cleavage is split. □

Proposition 3.4.3.11. Theorem 3.3.7.1 restricts to the respective 2-subcategories with split terms and types fibrations.

Proof. We begin with 0-cells. Consider a cDTT with split terms and types fibration and Σ preserving cleavages, then clearly the induced comprehension category has split type fibration. Conversely, consider a split comprehension category, then the induced term fibration is split: this is the content of Lemma 3.4.3.10.

We denote the choice of cartesian lifting of a map $\sigma : \Theta \to \Gamma = pA$ at a coalgebra $a : A \to KA$ by $\hat{s}_{a,\sigma}$.

Now to 1-cells. A lax split cDTT morphism is, in particular, a morphism of fibrations preserving cleavages of the type fibrations, hence it induces a lax morphism of split comprehension categories. Conversely, we wish to prove that any cleavage preserving $(H, C) \colon p \to p'$ lifts to a cleavage preserving $(\dot{H}, C) \colon pU \to p'U'$. We start from a coalgebra $a : A \to KA$ and its cartesian choice of lifting $\hat{s}_{a,\sigma}$. Recall that each lax wc-comonad morphism (H, C, θ) induces $\dot{H} : \mathrm{CoAlg}(K) \to \mathrm{CoAlg}(K')$ defined by $\dot{H}a = \theta_A \circ Ha$, hence we can fill the right-hand side of the diagram below, which is commutative by construction of $\hat{s}$, by naturality of θ, and by definition of $\dot{H}$.



$$C\Theta \xrightarrow{\quad C\sigma \quad} C\Gamma$$

Now, by hypothesis we have that (H, C) is a cleavage preserving morphism, so that both the top and the bottom "triangles" commute as well. The map $\hat{s}'_{Ha,C\sigma}$ is defined to be that making the square involving $s'_{HA,C\sigma}$ and $\dot{H}a$ commute, so that the left-hand square commutes as well, proving $\dot{H}\hat{s}_{a,\sigma} = \hat{s}'_{Ha,C\sigma}$ as desired.

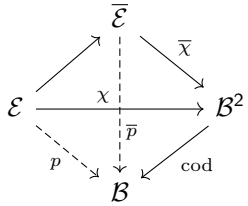
The case for 2-cells is trivial because our 2-subcategories are 2-full. □

3.4.4 Discrete vs full split

There is some result relating discrete and full split comprehension categories.

Lemma 3.4.4.1. [Jac93, Example 4.10] $\mathbf{CompCat}^{disc} \equiv \mathbf{CompCat}_{\mathbf{str}}{}^{full\ split}$.

Sketch of the proof. Starting from a discrete comprehension category $\chi \colon \mathcal{E} \to \mathcal{B}^2$, we can define a new category $\overline{\mathcal{E}}$ having the same objects that $\mathcal{E}$ has, and $\overline{\mathcal{E}}(A, B) = \mathcal{B}^2(\chi_A, \chi_B)$. There is an obvious identity-on-objects functor $\mathcal{E} \to \overline{\mathcal{E}}$ making the following diagram commute.



Now $\overline{\chi}$ is trivially full, one can show that discreteness of p implies $\overline{p}$ split. On the opposite direction, one simply forgets all maps in the fibers. □

Though equivalent, they are very different as subcategories of **CompCat**: the first is full, while the second is not. Perhaps a look at it in our setting can pin-point the difference and provide a logical interpretation for the mismatch. We leave this for a future work.

3.4.5 Splitting a wc-comonad

It is well-known that there are two canonical ways to split a fibration, see for example our discussion in Section 1.6. We want to extend such constructions to wc-comonads. Notice that results similar to Theorem 3.4.5.2 and Theorem 3.4.5.5 appear in [LW15], where comprehension categories are used instead of wc-comonads.

Definition 3.4.5.1. Call $\mathbf{wcCmd}^{split}$ the category of wc-comonads with split type fibration and lax morphisms preserving the split cleavage.

Theorem 3.4.5.2 (Fibered Yoneda for wc-comonads)**.**

$$\mathbf{wcCmd}^{split}(\mathcal{B}) \underset{U}{\overset{S}{\underset{\longrightarrow}{\overset{\longleftarrow}{\top}}}} \mathbf{wcCmd}(\mathcal{B})$$

Proof. We define the 2-functor S on objects as follows,

$$S\,(p, K) \;:=\; (\int Spr(p), \tilde{K}\,)$$

where $\int Spr(p)$ is the Grothendieck construction associated to

$$Spr(p)\colon \mathcal{B}^{op} \to \mathbf{Cat}, \quad \Gamma \mapsto \mathbf{Fib}(\underline{\Gamma}, p),$$

where $\underline{\Gamma}$ is $\mathrm{dom}\colon \mathcal{B}_{/\Gamma} \to \mathcal{B}$, and which produces a split fibration as discussed [Str22, §3]. On this we define a wc-comonad $\tilde{K}$

$$\begin{array}{ccc} \mathcal{B}_{/\Gamma} & \xrightarrow{\;h\;} & \mathcal{E} \\ & {\scriptstyle \underline{\Gamma}}\searrow \quad \swarrow{\scriptstyle p} & \\ & \mathcal{B} & \end{array} \qquad\qquad h^+\colon \begin{array}{ccccc} \mathcal{B}_{/\Gamma.H} & \xrightarrow{p\epsilon_H \circ -} & \mathcal{B}_{/\Gamma} & \xrightarrow{\;h\;} & \mathcal{E} \\ & {\scriptstyle \underline{\Gamma.H}}\searrow & \downarrow{\scriptstyle \underline{\Gamma}} & \swarrow{\scriptstyle p} & \\ & & \mathcal{B} & & \end{array}$$

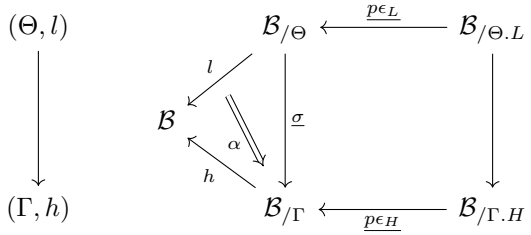
$$(\Gamma, h) \quad\longmapsto\quad (\Gamma.H, h^+)$$

where $H := h(\mathrm{id}_\Gamma)$ and $\Gamma.H = pKH = pH^+$ with the usual notation. Clearly $\tilde{K}(h)$ is a morphism of fibrations because it is the composition of such. Let us write $\underline{\sigma}$, with $\sigma : \Theta \to \Gamma$ a map in $\mathcal{B}$, the corresponding postcomposition functor $\mathcal{B}_{/\Theta} \to \mathcal{B}_{/\Gamma}$. Recall that a morphism in $\int Spr(p)$ is $(\sigma, \alpha) : (\Theta, l) \to (\Gamma, h)$ where $\sigma : \Theta \to \Gamma$ in $\mathcal{B}$ and $\alpha : l \Rightarrow h \circ \underline{\sigma}$ in $\mathbf{Fib}(\underline{\Theta}, p)$, and that it is cartesian iff $\alpha = \mathrm{id}$.

On a pair (σ, α) we can define $\tilde{K}$ as follows. Consider the composition

$$L = l(\mathrm{id}_\Theta) \xrightarrow{\alpha_{\mathrm{id}_\Theta}} h \circ \underline{\sigma}(\mathrm{id}_\Theta) = h(\sigma) \xrightarrow{h(!)} h(\mathrm{id}_\Gamma) = H$$

and call this $s : L \to H$. The counit natural square induces the commutative square below,



(3.12)

which, in turn, induces a suitable $(pKs, \alpha * \mathrm{id}) : (\Theta.L, l^+) \to (\Gamma.H, h^+)$.

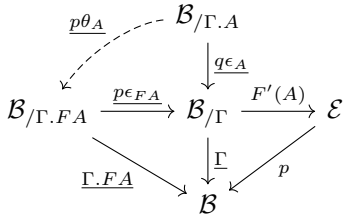
We now show how this functor supports a wc-comonad structure induced by that of $K = (K, \epsilon, \nu)$ on p. We denote this $(\tilde{K}, \tilde{\epsilon}, \tilde{\nu})$. By definition of $\tilde{K}$, $\tilde{\epsilon}$ can be trivially defined as $(\underline{p\epsilon_H}, \mathrm{id})$ and is thus $\int Spr(p)$-cartesian. Similarly, a cartesian $(\sigma, \mathrm{id}) : (\Theta, l) \to (\Gamma, h)$ induces $s = h(!) : h(\sigma) \to h(\mathrm{id}_\Gamma)$ which is cartesian because ! is and h preserves cartesian maps. Therefore the square in 3.12 is (forgetfully) over a pullback by hypothesis on K.

Additionally, one can check that S maps lax morphisms of comonads to lax morphims of comonads: again, it all boils down to commuting squares in the context category.

Finally, we prove that there is an appropriate natural isomorphism.

$$\mathbf{wcCmd}^{split}(\mathcal{B})\ (\,(q, C), S(p, K)\,) \cong \mathbf{wcCmd}(\mathcal{B})\ (\,U(q, C), (p, K)\,)$$

Each lax morphism (F, θ) in $\mathbf{wcCmd}(\mathcal{B})$ induces one between the split counterparts: define (F', θ') with $F'(A) = F(-^* A)$, which is well defined because q is split. Then $F'(A)(\mathrm{id}_\Gamma) = FA$ therefore we have the following commutative diagram

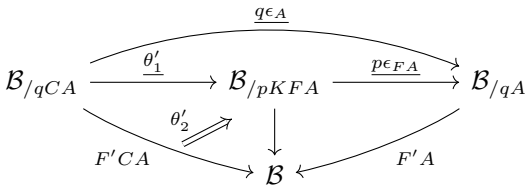


(pointwise) describing a lax comonad morphism $C \to \tilde{K}$.

One can check that, for the same reasons, the discussion above produces a functor in the opposite direction, $(F', \theta') \to (F, \theta)$, where

$$F = F'(-)(\mathrm{id}_{q(-)})$$

and θ can be calculated out of the following diagram where $(\theta'_1, \theta'_2) = \theta'$.



It is clear that both constructions are one inverse to the other. □

Lemma 3.4.5.3. $US \cong \mathrm{Id}$.

Remark 3.4.5.4 (On the meta-theory)**.** Theorem 3.4.5.2, as its version in Theorem 1.6.1.2, uses the axiom of choice for classes to prove that the counit is an equivalence, but that $U \dashv S$ is independent from it.

Theorem 3.4.5.5 (Left 2-adjoint to splitting for wc-comonads)**.**

$$\mathbf{wcCmd}^{split}(\mathcal{B}) \underset{F}{\overset{U}{\underset{\longleftarrow}{\overset{\longrightarrow}{\top}}}} \mathbf{wcCmd}(\mathcal{B})$$

Proof. Again, we follow [Str22, §3]: consider a normalized cleavage on p and define F on objects as

$$F(p, K) := (\int Spl(p), \overline{K})$$

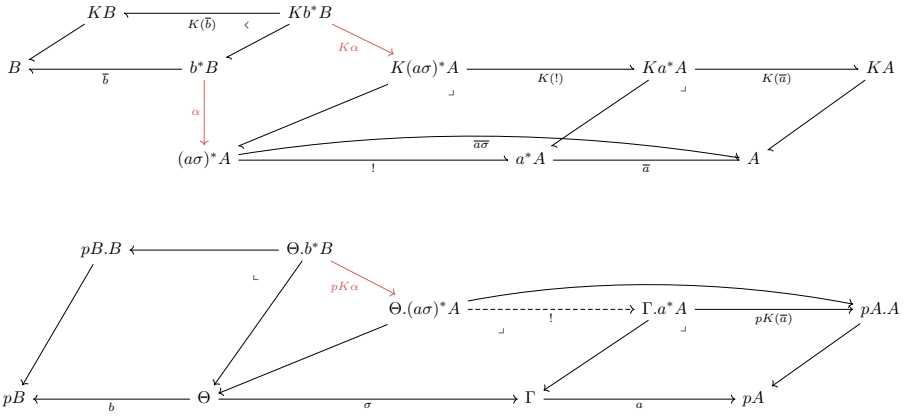
where $\int Spl(p)$ is the Grothendieck contruction associated to

$$Spl(p) : \mathcal{B}^{op} \to \mathbf{Cat}$$

such that objects in $Spl(p)(\Gamma)$ are pairs (A, a) with $a : \Gamma \to pA$ and morphisms $\alpha : (B, b) \to (A, a)$ are vertical maps $b^*B \to a^*A$. On the total category of $\int Spl(p)$ we can define a wc-comonad insisting on the endofunctor acting on objects as

$$\overline{K} : \ (\,\Gamma,\ (A, a)\,) \mapsto (\,\Gamma.a^*A,\ (KA, pK(\overline{a}))\,)$$

and on a map $(\,\Theta,\ (B, b)\,) \to (\,\Gamma,\ (A, a)\,)$, which is the data of a pair (σ, α) with $\sigma : \Theta \to \Gamma$ and $\alpha : (B, b) \to Spl(p)(\sigma)(A, a)$, as in the following diagram. We mark with a $\rightharpoonup$ those maps that are p-cartesian.



Filling in the diagram above out of (σ, α) is only a matter of patience and heavily relies on (1), (2), and Remark 3.3.3.2.

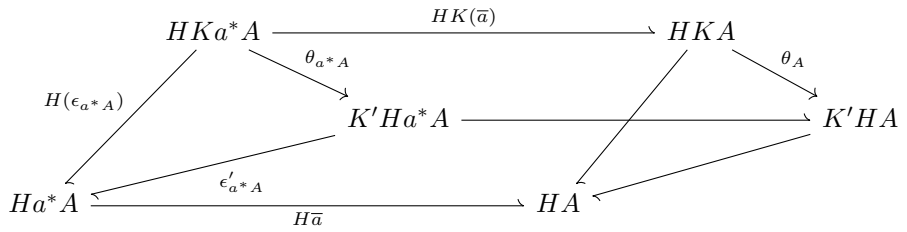
One can check that $\overline{K}$ is copointed, as well, because it is possible to define a natural $\overline{\epsilon} : \overline{K} \Rightarrow \mathrm{id}$ with components

$$(p\epsilon_{a^*A}, \alpha) : \overline{K}(\,\Gamma,\ (A, a)\,) \to (\,\Gamma,\ (A, a)\,).$$

Since $(KA, pK(\overline{a})) = (a \circ p\epsilon_{a^*A})^*A$ we can define α to be the identity, therefore $\overline{\epsilon}$ has cartesian components. Moreover, for a cartesian $(\sigma, \mathrm{id}) : (\,\Theta,\ (B, b)\,) \to$

$(\Gamma,\ (A,a)\,)$ we have that the underlying square is a pullback: this we can read again in the diagram above, where $\alpha = \mathrm{id}$, therefore $pK\alpha = \mathrm{id}$ itself and (what becomes) the middle square in the second row is the desired one.

Consider a pair $(H,\theta) : (p,K) \to (p',K')$: we can define $F(H,\theta) = (\overline{H},\overline{\theta})$ with $\overline{H}(\,\Gamma,\ (A,a)\,) := (\,\Gamma, (HA,a)\,)$ and the following diagram



can be used to show that we have a suitable $\overline{\theta} : \overline{H}\,\overline{K} \Rightarrow \overline{K'}\,\overline{H}$ as well.

We now need to show that such a functor produces a left adjoint to U,

$$\mathbf{wcCmd}^{split}(\mathcal{B})\ (\,F(p,K),(q,C)\,) \cong \mathbf{wcCmd}(\mathcal{B})\ (\,(p,K),U(q,C)\,).$$

For each lax morphism $(H,\theta) : (p,K) \to U(q,C)$ we can define

$$H' : \int Spl(p) \to \mathcal{F}, \quad (\,\Gamma,\ (A,a)\,) \mapsto H(a^*A)$$

and this can suitably extended to morphisms in $\int Spl(p)$. But then $H'K(\Gamma,A,a) = HK(a^*A)$ and $CH'(\Gamma,A,a) = CH(a^*A)$, so that we can define θ' precisely as θ on each component. Conversely, to a lax morphism (H',θ') we can match (H,θ) with

$$H(A) := H'(\,pA,\ (A,\mathrm{id})\,)$$

and with this we can rightfully define $\theta_A = \theta'_{(\,pA,\,(A,\mathrm{id})\,)}$. It is clear that these constructions are one inverse to the other. □

Remark 3.4.5.6. The unit of $F \dashv U$ is not an isomorphism.

3.5 Subtyping

Most models use a discrete type fibration but we have been taking a lot of care in avoiding this request. One of the reasons for this, is that we believe that vertical maps suitably model subtyping, so that

$$\Gamma \vdash A' \leq_f A \quad \text{corresponds to} \quad f\colon A' \to A \text{ } u\text{-vertical.}$$

If we wish to model something of the sort of

$$\text{(Sbsm)}\ \ \frac{\Gamma \vdash a : A' \qquad \Gamma \vdash A' \leq_f A}{\Gamma \vdash a : A}$$

we need to manipulate our starting (non-discrete) cDTT to allow for a term to have "multiple" types. Intuitively, we want to be able to speak of triples

$$(a, A, f) \text{ where } f\colon \Sigma a \to A \text{ is vertical.}$$

The categorical notion we use, then, is that of comma object.

3.5.1 Comma objects induced by a cDTT

We aim to pick vertical maps only, therefore we need our comma category to be *fibration sensitive*, hence we do not compute it in **Cat**, but in **Fib**($\mathcal{B}$): this is the 2-category of fibrations, cartesian functors, and natural transformations with vertical components Definition 1.4.0.2.

Lemma 3.5.1.1. For each cDTT $(u, \dot{u}, \Sigma \dashv \Delta)$ over $\mathcal{B}$ the comma object (Σ/Id) in **Fib**($\mathcal{B}$) exists and is the desired one.

Proof. We need to provide a fibration over $\mathcal{B}$, two fibration morphisms p_1 and p_2 and a 2-cell α as below, such that they are universal with respect to such property.

$$\begin{array}{ccc} (\Sigma/\mathrm{Id}) & \xrightarrow{p_1} & \dot{u} \\ {\scriptstyle p_2}\downarrow & \overset{\alpha}{\swarrow\!\!\swarrow} & \downarrow{\scriptstyle \Sigma} \\ u & \xrightarrow[\mathrm{Id}]{} & u \end{array}$$

Our claim is that $(\Sigma/\mathrm{Id})\colon \Sigma{\downarrow^v}\mathcal{U} \to \mathcal{B}$ has domain the subcategory of $\Sigma{\downarrow}\mathcal{U}$ with only vertical structure maps, p_1, p_2 the obvious morphisms, and $\alpha_{(a,A,f)} = f$ for each triple (a, A, f). It maps (a, A, f) to $\dot{u}a$, which is equal to uA because f is vertical.

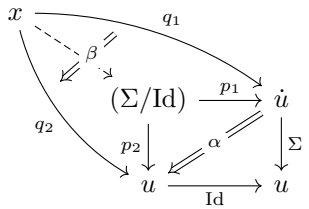
One can check that a map in $\Sigma{\downarrow^v}\mathcal{U}$ is cartesian if and only if its components are, respectively, $\dot{u}$- and u-cartesian. This implies that p_1 and p_2 are actually cartesian functors. Moreover, each $\sigma\colon \Theta \to \Gamma$ in $\mathcal{B}$ can be lifted at triple (a, A, f) over Γ as follows,

$$a[\sigma] \xrightarrow[\overline{\sigma}]{} a \qquad\qquad \begin{array}{ccc} \Sigma a[\sigma] & \xrightarrow{\Sigma\overline{\sigma}} & \Sigma a \\ {\scriptstyle f[\sigma]}\vdots\downarrow & & \downarrow{\scriptstyle f} \\ A[\sigma] & \xrightarrow[\overline{\overline{\sigma}}]{} & A \end{array}$$

$$\Theta \xrightarrow[\sigma]{} \Gamma$$

with $\overline{\sigma}$ a $\dot{u}$-cartesian lifting of σ at a and $\overline{\overline{\sigma}}$ a u-cartesian lifting of σ at A. Since f is vertical, there exists a unique vertical $f[\sigma]$, plus Σ preserves cartesian maps so that the morphism $(\overline{\sigma}, \overline{\overline{\sigma}})$ is cartesian.

It has the desired universal property because another $x : \mathcal{X} \to \mathcal{B}$, q_1, q_2, and β as below,



there exists a unique cartesian functor $F : x \to (\Sigma/\mathrm{Id})$ such that $\beta = F * \alpha$ and $p_i F \cong q_i$ for $i = 1, 2$. In fact, it is induced by the functor obtained from the underlying comma category: since $\Sigma\downarrow\mathcal{U}$ is a comma object in **Cat** and, in particular, $\mathcal{X}$, q_1, q_2, β satisfy appropriate hypotheses, there is a unique

$F\colon \mathcal{X} \to \Sigma\downarrow\mathcal{U}$, namely that acting on objects as $X \mapsto (q_1X, q_2X, \beta_X)$. Now, this restricts to $\Sigma\downarrow^v\mathcal{U}$ because β has vertical components since it is a 2-cell in $\mathbf{Fib}(\mathcal{B})$. One can check that it is in fact cartesian and it satisfies all the right equations. □

Given that we are working in the context of categorized judgemental theories, we need to relate such a construction - plus many that will follow - to what is computationally allowed for a cDTT.

Remark 3.5.1.2. Any comma object can be constructed by means of pullbacks and cotensors. In fact, given 1-cells $F\colon \mathcal{A} \to \mathcal{C}$ and $G\colon \mathcal{B} \to \mathcal{C}$, one can compute

$$\begin{array}{ccccc}
(F/G) & \longrightarrow & \mathcal{P} & \longrightarrow & \mathcal{A} \\
\downarrow & \lrcorner & \downarrow & \lrcorner & \downarrow{\scriptstyle F} \\
\mathcal{Q} & \longrightarrow & \mathcal{C}^2 & \xrightarrow[\mathrm{dom}]{} & \mathcal{C} \\
\downarrow & \lrcorner & \downarrow{\scriptstyle \mathrm{cod}} & & \\
\mathcal{B} & \xrightarrow[G]{} & \mathcal{C} & &
\end{array}$$

to get (F/G).

Remark 3.5.1.3. Given a cospan of fibrations $p \to r \leftarrow q$ in $\mathbf{Fib}(\mathcal{B})$, it is equivalent to provide a comma object for p, r, q in $\mathbf{Fib}(\mathcal{B})$ or a subcategory of their comma object in $\mathbf{Cat}$ only considering r-vertical maps.

Because of Remark 3.5.1.2, a categorized judgemental theory is always closed with respect to comma objects, and to compute (Σ/Id) we need to either expand our definition to include finite limits in $\mathbf{Fib}(\mathcal{B})$, or to allow us to get subcategories as described in Remark 3.5.1.3. Either way, we believe our request to add such a possibility is both very natural and not at all burdensome.

We now have a notion of typing that is *labelled*, meaning that if we denote, in the language of cDTTs,

$$\frac{\Gamma \vdash (a, A, f)\ (\Sigma/\mathrm{Id})}{\Gamma \vdash a :_f A}$$

we can express judgements of the form "in context Γ, the term a is of type A as witnessed by f" or, in the style of [LSX13], "in context Γ, f *coerces* a to be of type A".

Additionally, we need a fibration classifying all subtyping judgements, meaning of the form

$$\Gamma \vdash A' \leq_f A\,,$$

which we read as "in context Γ, A' is a subtype of A by the *coercion* f". In an entirely similar fashion as Lemma 3.5.1.1, one can show that $(\mathrm{Id}/\mathrm{Id})$ on u exists and it captures all vertical maps in $\mathcal{U}$.

Lemma 3.5.1.4. For each cDTT $(u, \dot{u}, \Sigma \dashv \Delta)$ over $\mathcal{B}$ the comma object $(\mathrm{Id}/\mathrm{Id})$ on u in $\mathbf{Fib}(\mathcal{B})$ exists.

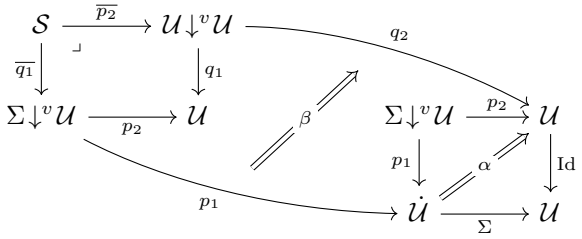
Putting together the two notions, we can finally prove the following.

Proposition 3.5.1.5. The rule

$$\frac{\Gamma \vdash a :_g A' \qquad \Gamma \vdash A' \leq_f A}{\Gamma \vdash a :_{fg} A}$$

is valid in the judgemental theory induced by a cDTT.

Proof. The premise of the rule is computed by the pullback $\mathcal{S}$,



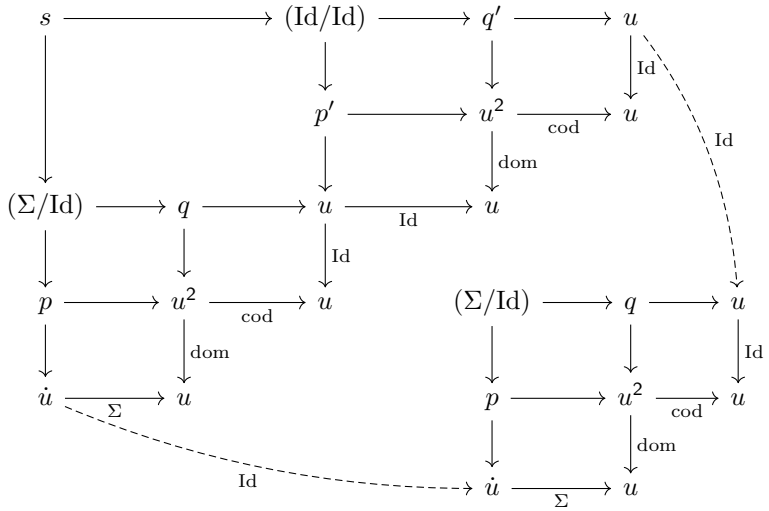
then we can map $\mathcal{S}$ to the cospan (Σ, Id) via p_1 and q_2. Now, there is a natural transformation β with

$$\beta_{(\,g\colon \Sigma a \to A'\,,\, f\colon A' \to A\,)} = fg\,.$$

The universal property of the comma category $\Sigma\downarrow\mathcal{U}$ induces a unique functor, which restricts to $\Sigma\downarrow^v\mathcal{U}$ and is that witnessing the desired rule. □

Of course we should show that β in the definition above is itself in the cDTT. We provide a proof just this once by means of pullbacks as detailed in Remark 3.5.1.2. Every other 2-cell appearing in this section is in the cDTT for similar reasons.

Consider the following diagram



computing $\Sigma \downarrow \mathcal{U}$, $\mathcal{U} \downarrow \mathcal{U}$ and then $\mathcal{S}$. Then the dashed maps in the outer diagram identify β. The unique map $s \to (\Sigma/\mathrm{Id})$ produces the desired functor representing the rule in Proposition 3.5.1.5.

Now that in our system we have at our disposal classifiers for both

$$\Gamma \vdash a :_f A \quad \text{and} \quad \Gamma \vdash A' \leq_g A,$$

encoded, respectively, in

$$(\Sigma/\mathrm{Id}) \quad \text{and} \quad (\mathrm{Id}/\mathrm{Id}),$$

we show that they end up representing a variation of a well-known kind of subtyping.

3.5.2 Rules for subtyping

Section 3.5.1 was devoted to showing that a cDTT contains certain pieces of information that can be interpreted as a form of subtyping. We here make that intuition precise, and describe rules and properties of such subtyping.

As we have mentioned, our intuition closely follows that of *coercive subtyping*: this is the product of thinking about subtyping as an abbreviation mechanism, meaning that we say that a given type A' is a subtype of A if there is a unique *coercion* from A' to A. Whenever we need a term of type A, then, it suffices to have a term of type A', which we can "plug-in" into A. Coercive subtyping has many computational properties, and throughout the years it has been made to behave very well with other common structures in dependent type theory [Luo99, Che03, LSX13].

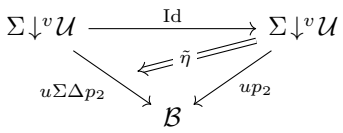
We here list a few meaningful rules that can be found in a cDTT.

Proposition 3.5.2.1 (Structural subtyping rules)**.** A cDTT verifies the following rules.

$$(\text{Trans}) \quad \frac{\Gamma \vdash A' \leq_f A \qquad \Gamma \vdash A'' \leq_g A'}{\Gamma \vdash A'' \leq_{fg} A}$$

$$(\text{Sbst}) \quad \frac{\Gamma.A \vdash B' \leq_f B \qquad \Gamma \vdash a :_g A}{\Gamma \vdash B'[a] \leq_{\dot{u}(\Delta g \eta_a)^* f} B[a]} \qquad (\text{Wkn}) \quad \frac{\Gamma \vdash A' \leq_f A \qquad \Gamma \vdash B}{\Gamma.B \vdash A' \leq_{(u\epsilon_B)^* f} A}$$

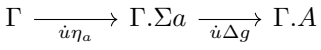
Proof. As for transitivity, one follows a path similar to that in Proposition 3.5.1.5, simply replacing $\Sigma\downarrow^v\mathcal{U}$ with $\mathcal{U}\downarrow^v\mathcal{U}$. Substitution is achieved by the same means of regular substitution in a cDTT Section 2.3.4, meaning using some variation of the unit of $\Sigma \dashv \Delta$. Consider the following 2-cell,



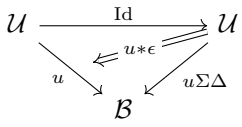
where the first leg computes $\Gamma.A$, the second simply Γ, and

$$\tilde{\eta}_{(g\colon \Sigma a \to A)} = \dot{u}(\Delta g \circ \eta_a) \colon \Gamma \to \Gamma.\Sigma a \to \Gamma.A\,.$$

See that $\sharp$-lifting Theorem 1.3.3.3 this along $\mathcal{U}\downarrow^v\mathcal{U} \to \mathcal{B}$ performs the cartesian lifting of f along $\tilde{\eta}_g$, meaning



then the induced $\dot{u}(\Delta g \eta_a)^* f$ is vertical and gives the coercion in the output. Weakening needs no "intermediate" step, because $u(f) = \mathrm{id}$ immediately: simply compute the $\sharp$-lifting of the following 2-cell



along $\mathcal{U}\downarrow^v \mathcal{U} \to \mathcal{B}$. Notice that A, A' appearing in the consequent of the rule should really read A^+, A'^+ for the respective extended types, but we follow the tradition on notation here. □

3.5.2.2 (Comparison with coercive subtyping)**.** We here refer to coercive subtyping in its formulation appearing in [LSX13, 2.2]. There, the starting point is a type theory T specified in a logical framework and C a set of subtyping judgements. It is additionally asked that C is "coherent", meaning that

- if a judgement $\Gamma \vdash A <_c B$ is in C, then $\Gamma \vdash A$, $\Gamma \vdash B$, $\Gamma \vdash c : A \to B$;
- the relation is not reflexive, meaning that $\Gamma \vdash A <_c A$ is not in C for any Γ, A, c;
- if both $\Gamma \vdash A <_{c_1} B$ and $\Gamma \vdash <_{c_2} B$ are in C, then $\Gamma \vdash c_1 = c_2 : A \to B$.

Theorem 3.5.2.3 ([LSX13, Theorems 3.9(2), Theorem 3.10(3)])**.** Let T a MLTT and C a set of subtyping judgements. If C is coherent, we can extend T with C adding appropriate judgements and rules, we call this $T[\mathrm{C}]$, and on it a calculus $T[\mathrm{C}]^*$ by extending terms to include a new application. Then

(i) $T[\mathrm{C}]^*$ is a conservative extension of T, and

(ii) $T[\mathrm{C}]$ and $T[\mathrm{C}]^*$ are equivalent.

The subtyping of a cDTT does not share all properties required by coercive subtyping, in the sense that, provided that our intuition is to identify C with the collection of all u-vertical arrows,

1. for each judgement in $\Gamma \vdash A \leq_c B$, c is a term of the dependent product, while for us it is a vertical map, and
2. such a collection does indeed contain elements witnessing reflexivity, as identities in the total categories are, in particular, vertical.

Nonetheless, we feel that these two problems are somewhat artificial when *employing* subtyiping, because when choosing the set C, 2 can be avoided by simply taking out all identities, and the dependent product condition in 1 is due to the fact that one artificially extends the theory to include such function symbols *but* they are not inherently included in T, so perhaps when working semantically, that can be avoided, while categorical computations remain available. Uniqueness of coercions is addressed below.

Remark 3.5.2.4. For every structural rule of subtyping (3.5.2.1), the coercions appearing in the consequences are uniquely determined by those appearing in the premises.

This guarantees that, starting from a given coercion, there is a unique coercion that can be produced out of it by means of structural calculations. On the other hand, of course we have the following.

Remark 3.5.2.5. If the type fibration is faithful (see Proposition 3.4.3.8), then the subtyping here described is coherent in the sense of [LSX13].

This does not mean that we can say that coercive subtyping is modeled by subtyping in a cDTT, but the spirit in which both work is very much akin. We leave it to future work – and to a few more conversations with experts on the topic – to further develop the study of the relation between the two.

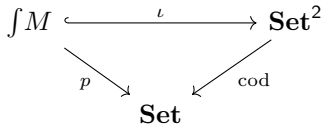
3.5.3 An example: semantic subtyping

In this section we provide an interpretation of *semantic subtyping* as discussed in [BCF08]. Consider the following pseudofunctor

$$M\colon \mathbf{Set}^{\mathsf{op}} \to \mathbf{Cat}$$

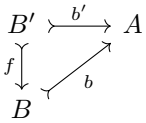
mapping each set A to the full subcategory of $\mathbf{Set}_{/A}$ containing only monos. When one performs the Grothendieck construction associated to M, one gets a subfibration of $\mathrm{cod}\colon \mathbf{Set}^2 \to \mathbf{Set}$.

Remark 3.5.3.1. The embedding of $\int M$ into cod is part of the data of a comprehesion category.



The corresponding wc-comonad K on p maps each mono to its kernel pair and has counit the pullback square computing it. Coalgebras are sections of monos. This construction will be discussed in some detail in Fact 3.6.2.2.

Breaking up the discussion in Section 3.5.1 and Section 3.5.2 we get the following interpretation.

classical syntax	judgemental syntax	semantics
$A \vdash B$	$A \vdash (A,b)\ \int M$	$b\colon B \rightarrowtail A$
$A \vdash y : B$	$A \vdash (\,(A,b),y\,)\ \mathrm{CoAlg}(K)$	$y : A \to B$ s.t. $by = \mathrm{id}$
$A \vdash B' \leq_f B$	$A \vdash (b',b,f)\ (\mathrm{Id}/\mathrm{Id})$	

Remark 3.5.3.2. More precisely, the subtyping presented in [BCF08] coincides with subtyping in our sense where the type fibration is obtained by poset reflection of $M(A)$'s. Then the typing relation becomes unlabelled because of what was observed in Remark 3.5.2.5.

3.6 Case study: CE-systems

The general framework established in the present work allows for a swift and comprehensive treatment of different categorical models of dependent type

theory. Moreover, the possibility of linking them to a cDTT allows for a precise description of the behaviour of type constructors in each. To exemplify such a procedure, in this section we study the case of CE-systems, which were introduced in [AENR21] and can be thought of as a non-stratified version of Cartmell's contextual categories [Car86] or Voevodsky's C-systems [Voe16, Lecture 5].

Definition 3.6.0.1 (CE-system)**.** A *CE-system* consists of two strict category structures $\mathcal{F}, \mathcal{B}$ on the same set of objects $\mathsf{Ob}(\mathcal{F}) = \mathsf{Ob}(\mathcal{B})$ and an identity-on-objects functor $I \colon \mathcal{F} \to \mathcal{B}$, together with

1. a chosen object 1 which is terminal in $\mathcal{F}$;
2. for any $\sigma \colon \Theta \to \Gamma$ in $\mathcal{B}$ and any A in $\mathcal{F}_{/\Gamma}$, there is a functorial choice of a $\sigma^* A$ in $\mathcal{F}_{/\Theta}$ such the following

$$\begin{array}{ccc} \Theta.\sigma^* A & \xrightarrow{\sigma^+} & \Gamma.A \\ {\scriptstyle I(\sigma^* A)}\downarrow & \lrcorner & \downarrow{\scriptstyle I(A)} \\ \Theta & \xrightarrow[\sigma]{} & \Gamma \end{array}$$

is a pullback square in $\mathcal{B}$.

Definition 3.6.0.2 (Morphism of CE-systems)**.** Given two CE-systems $I \colon \mathcal{F} \to \mathcal{B}$ and $I' \colon \mathcal{F} \to \mathcal{B}$, a *CE-system morphism* consists of a pair of functors F, B such that

$$\begin{array}{ccc} \mathcal{F} & \xrightarrow{F} & \mathcal{F}' \\ {\scriptstyle I}\downarrow & & \downarrow{\scriptstyle I'} \\ \mathcal{B} & \xrightarrow[C]{} & \mathcal{B}' \end{array}$$

commutes, $F(1_{\mathcal{F}}) = 1_{\mathcal{F}'}$, and

$$F(\sigma^* A) = (C\sigma)^*(FA) \quad \text{and} \quad C(\sigma^+) = (C\sigma)^+ .$$

Definition 3.6.0.3 (The category of CE-systems)**.** We write **CEsys** for the category of CE-systems and CE-system morphisms.

We interpret the objects underlying both $\mathcal{F}$ and $\mathcal{B}$ to contexts, and use I to pick out in $\mathcal{B}$ only those maps corresponding to context extension/projection: we identify types with their projection, and terms of a given type are again sections.

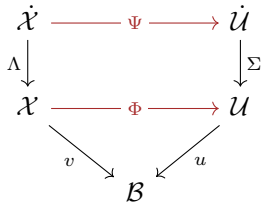
3.6.1 Type constructors in a cDTT

Our plan is to show that CE-systems *inherently* contain the structure of dependent sums and units, but before we do that we need to recall the way in which we deal with type constructors in the context of cDTTs.

As it was briefly discussed in Remark 3.3.2.1, cDTTs were introduced in the more general context of categorized judgemental theories (Chapter 2), which aim to describe the calculus one encounters when starting reading functors (mostly, fibrations) as judgements and (lax) commutative triangles as rules. In principle, its approach is pretty far from that of a categorical model such as a comprehension category. In Section 3.3 we have proved that the two are closely related and, in fact, can be seen as different presentations of the same thing.

Still, the fact that cDTTs make sense in a calculus-like world allows for much more freedom the moment one wants to add structure to the theory. In particular, the treatment of type constructors simply amounts to asking the existence of two functors, one into $\mathcal{U}$ performing formation, one into $\dot{\mathcal{U}}$ performing introduction, satisfying some additional property. On the contrary, though some models (*e.g.* [Awo18], partially [Jac99]) do, giving an account of type constructors is in general not an easy feat. In fact, the definition of what a type constructor is, though widely clear, could not be formalized.

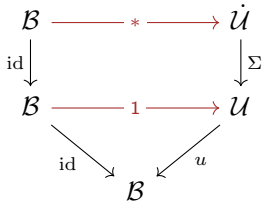
Definition 3.6.1.1 (The type constructor Φ)**.** A categorized dependent type theory *with* Φ*-types* is a cDTT having two additional functors Φ, Ψ such that the diagram below is commutative and the upper square is a pullback.



Intuitively, the category $\mathcal{X}$ is that coding - via universal properties of finite limits - the premise of the formation rule, the category $\dot{\mathcal{X}}$ that coding the one of the introduction rule, Λ is some elaborated typing functor, sending each term in $\dot{\mathcal{X}}$ to its type. The way in which such a diagram describes a constructor is detailed in Chapter 2, but to the benefit of the reader we recall what happens for units in Proposition 3.6.1.3.

3.6.1.1 Units

Definition 3.6.1.2 (Unit-types)**.** A categorized dependent type theory *with unit-types* is a cDTT having two additional functors 1, $*$ such that the diagram below is commutative and the upper square is a pullback.



Proposition 3.6.1.3 (The definition is correct)**.** The categorized judgemental theory generated by diagrams in Definition 3.3.2.2 and Definition 2.3.8.1 contains codes for formation, introduction, elimination, and computation of unit types.

Proof. Introduction and formation read as follows

$$(1)\quad \frac{\Gamma \vdash \Gamma\ \mathrm{id}}{\Gamma \vdash 1_\Gamma\ u} \qquad (*)\quad \frac{\Gamma \vdash \Gamma\ \mathrm{id}}{\Gamma \vdash (*_\Gamma, 1_\Gamma)\ \dot{u}}$$

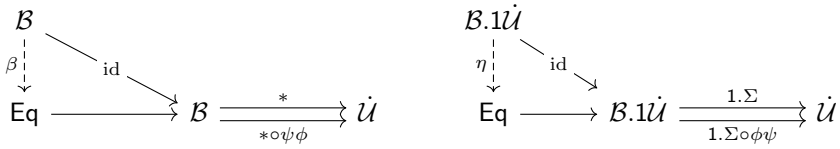
or, in a more familiar writing

$$(\mathrm{uI})\quad \frac{\vdash \Gamma\ \mathtt{ctx}}{\Gamma \vdash 1_\Gamma\ \mathtt{Type}} \qquad (\mathrm{uF})\quad \frac{\vdash \Gamma\ \mathtt{ctx}}{\Gamma \vdash *_\Gamma : 1_\Gamma}$$

moreover, the elimination rule is captured by the unique map $\psi\colon \mathcal{B}.1\dot{\mathcal{U}} \to \mathcal{B}$ and it translates to the syntactic writing on the left, while postcomposed with $*$ it translates as the more familiar rule on the right

$$(\psi)\quad \frac{\Gamma \vdash t : 1_\Gamma}{\vdash \Gamma\ \mathtt{ctx}} \qquad\qquad (* \circ \psi)\quad \frac{\Gamma \vdash t : 1_\Gamma}{\Gamma \vdash *_\Gamma : 1_\Gamma}$$

which is also denoted (uE). Finally, computation β and η can be decoded from the two following diagrams



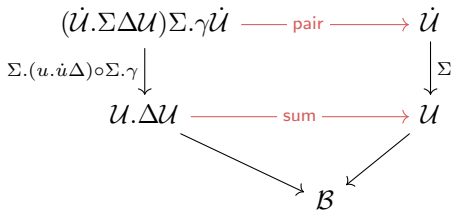
with ϕ the inverse to ψ, which read, respectively, as follows.

$$(\mathrm{u}\beta\mathrm{C})\quad \frac{\vdash \Gamma\ \mathtt{ctx}}{\Gamma \vdash *_\Gamma =_{1_\Gamma} *_\Gamma} \qquad\qquad (\mathrm{u}\eta\mathrm{C})\quad \frac{\Gamma \vdash t : 1_\Gamma}{\Gamma \vdash t =_{1_\Gamma} *_\Gamma}$$

□

3.6.1.2 Dependent sums

Definition 3.6.1.4. A categorized dependent type theory *with* sum*-types* is a cDTT as in Definition 3.3.2.2 having two additional rules sum, pair such that the diagram below is commutative and the upper square is a pullback.



The premise of the formation rule encodes pairs (A, B) of types such that the context of ΔA is the same as that of B. The premise of introduction is computed by iterated pullbacks of $u\Sigma\Delta\Sigma$ first along u and then along Σ. A 2-cell γ is involved, see Section 2.3.8.2 for more details and Lemma 3.6.3.1 for an explicit calculation.

3.6.2 Comparing CE-systems to cDTTs

We wish to relate CE-systems to either one of the following biequivalent categories,

$$\mathbf{CompCat} \equiv \mathbf{wcCmd} \equiv \mathbf{cDTT}^{\mathrm{loose}}$$

or their respective subcategories. Since terms are interpreted to sections of $I(A)$, we might find it easier to aim at the first two.

Proposition 3.6.2.1. Each CE-system $I\colon \mathcal{F} \to \mathcal{B}$ induces a wc-comonad on a split fibration.

Proof. First of all, let us define a fibration with base $\mathcal{B}$ as follows:

$$p := \mathrm{cod} \circ I^2 \colon \mathcal{F}^2 \to \mathcal{B}^2 \to \mathcal{B},$$

with $\mathcal{F}^2$ the "I-full" subcategory or $\mathcal{B}^2$, meaning that its objects are morphisms in $\mathcal{F}$ and its morphisms

$$\begin{array}{ccc} B \\ \downarrow \\ A \end{array} \quad \text{is} \quad \begin{array}{ccc} \Theta.B & \xrightarrow{I(B)} & \Theta \\ \downarrow & & \downarrow \\ \Gamma.A & \xrightarrow[I(A)]{} & \Gamma \end{array}$$

are commutative squares in $\mathcal{B}$.

It really is a fibration because to each morphism of contexts $\sigma \colon \Theta \to \Gamma$ and $A \in \mathcal{F}_{/\Gamma}$ it corresponds a functorial choice of an object $\sigma^* A \in \mathcal{F}_{/\Theta}$ such that

$$\begin{array}{ccc} \Theta.\sigma^* A & \longrightarrow & \Gamma.A \\ {\scriptstyle I(\sigma^* A)}\downarrow & \lrcorner & \downarrow{\scriptstyle I(A)} \\ \Theta & \xrightarrow[\sigma]{} & \Gamma \end{array}$$

is a pullback in $\mathcal{B}$ *i.e.* a cartesian lifting of σ in $\mathcal{F}^2$. Notice that the functoriality appearing in Definition 3.6.0.1 implies that p is split.

Let us now define a suitable comonad $(K, \epsilon, \nu) : \mathcal{F}^2 \to \mathcal{F}^2$. Intuitively, K is intended to map types to their image under substitution along the projection map, in this case $I(A)$. Hence we can define

$$K(A) := I(A)^* A = A^+$$

which induces a comonad on $\mathcal{F}^2$. This is an instance of the more general Fact 3.6.2.2.

Finally, we need to show that its counit is cartesian, which is trivial because it is a pullback in the first place, and that each cartesian map in $\mathcal{F}^2$ induces a pullback as described in Definition 3.3.3.1, which is again true because a cartesian arrow in $\mathcal{F}^2 = \mathcal{E}$ *is* the required square, hence a pullback by definition. □

Fact 3.6.2.2 (Kernel-pair comonad)**.** Let $\mathcal{B}$ be a category such that we can always compute kernel pairs. Let $K \colon \mathcal{B}^2 \to \mathcal{B}^2$ the functor mapping to each σ its kernel pair: that is, the following iterated pullbacks, together with the use of their universal property,

$$\begin{array}{ccccccc} k_\sigma & \overset{!}{\dashrightarrow} & k_{\sigma.\sigma} & \longrightarrow & k_\sigma & \longrightarrow & \Theta \\ \downarrow{\scriptstyle \sigma.\sigma} & & \downarrow{\scriptstyle (\sigma.\sigma).\sigma} & & \downarrow{\scriptstyle \sigma.\sigma} & & \downarrow{\scriptstyle \sigma} \\ \Theta & \xrightarrow[!]{} & k_\sigma & \xrightarrow{\sigma.\sigma} & \Theta & \xrightarrow[\sigma]{} & \Gamma \end{array} \tag{3.13}$$

where all red compositions are identities. Such a K induces a comonad on $\mathcal{B}$ with counit ϵ_σ the last square on the right and comulitplication $\nu\sigma$ the one on the left. One can check that the (K, ϵ, ν)-coalgebras with structure map σ are exactly the sections of σ.

Proposition 3.6.2.3. Each wc-comonad $(p \colon \mathcal{E} \to \mathcal{B}, K)$ with p split induces a CE-system.

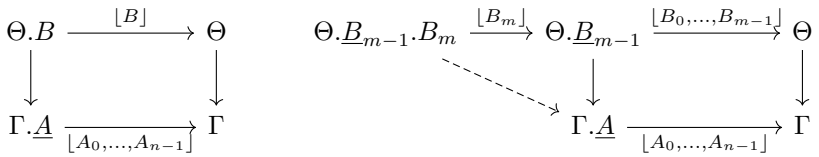
Proof. We only need to define a category $\mathcal{F}_K$ and an appropriate identity-on-objects map. We then consider the following directed graph G_K having

- as vertices the same objects as $\mathcal{B}$;
- as edges objects in $\mathcal{E}$, so that an edge A has source pKA and target pA.

Call $\mathcal{C}_K$ the free category on the graph G_K, and define $\mathcal{F}_K$ to be the category having the same objects as $\mathcal{C}_K$ and morphisms

$$\mathcal{F}_K(\lfloor B_0, \dots, B_{m-1} \rfloor, \lfloor A_0, \dots, A_{n-1} \rfloor)$$

by induction on m.

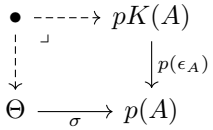


If $m = 0$, a morphism $\lfloor B \rfloor \to \lfloor A_0, \dots, A_{n-1} \rfloor$ is a commutative square in $\mathcal{B}$ as on the left; if $m \geq 1$, a morphism $\lfloor B_0, \dots, B_{m+1-1} \rfloor \to \lfloor A_0, \dots, A_{n-1} \rfloor$ is defined as the composition of $\lfloor B_0, \dots, B_{m-1} \rfloor \to \lfloor A_0, \dots, A_{n-1} \rfloor$ with $\lfloor B_m \rfloor$.

Define $I_K \colon \mathcal{F}_K \to \mathcal{B}$ as

$$I_K(\Gamma) = \Gamma, \qquad I_K(A) = p(\epsilon_A)$$

and let us show that it satisfies the characterizing property of CE-systems, that is that, for each choice of σ in $\mathcal{B}$ and $p(\epsilon_A)$, a functorial choice of an object in $\mathcal{F}_{K\,/\Theta}$ such that it completes the cospan below to a pullback.



Notice that the choice need not to be unique, and in fact it is far from being so since the same map $p(\epsilon_A)$ could be realized as the counit at different A's. Since p is split, we can compute the cartesian lifting of σ at A, and find a map $s \colon A\sigma \to A$ in $\mathcal{E}$, hence a counit square $\epsilon_{A\sigma} \to \epsilon_A$. Because it is a pullback, one can check (the image through p of) such square provides the desired pullback above. □

Theorem 3.6.2.4 (Adjunction between CE-systems and discrete wc-comonads)**.**

The constructions in Proposition 3.6.2.1 and Proposition 3.6.2.3 can be promoted to a pair of adjoint functors

$$\mathbf{CEsys} \underset{R}{\overset{L}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathbf{wcCmd_{str}}^{split}$$

where $\mathbf{wcCmd_{str}}^{split}$ is seen as a 1-category.

Proof. First of all, we need to show that both can be extended to functors. A strict morphism of wc-comonads (see Definition 3.3.3.4) is in particular a 1-cell (H, C) in **Fib** such that $HK = K'H$. We claim that we can define a functor $\overline{H}$ such that the following square

$$\begin{array}{ccc} \mathcal{F}_K & \overset{\overline{H}}{\dashrightarrow} & \mathcal{F}_{K'} \\ {\scriptstyle I}\downarrow & & \downarrow{\scriptstyle I'} \\ \mathcal{B} & \xrightarrow[C]{} & \mathcal{B}' \end{array}$$

commutes and the appropriate equations are satisfied. By the free-forgetful adjunction of graphs and categories, defining such a functor is equivalent to providing a directed graph morphism $G_K \to U\mathcal{F}_{K'}$. We can define it as

$$\overline{H}(\Gamma) = C(\Gamma), \quad \overline{H}(A) = H(A)\,.$$

The object HA in $\mathcal{E}'$ has source $p'K'HA = C(pKA)$ and target $p'HA = C(pA)$, therefore it is an actual graph morphism, and its corresponding functor commutes with C and I's. The additional conditions that $\overline{H}(\sigma^* A) = (C\sigma)^*(\overline{H}A)$ and $C(\sigma^+) = (C\sigma)^+$ follow directly from the fact that p is split.

Conversely, a morphism (F, C) of CE-systems induces a strict morphism between the associated wc-comonads: the required 1-cell in **Fib** is witnessed by the following commutative diagram

$$\begin{array}{ccc} \mathcal{F}^2 & \xrightarrow{F^2} & \mathcal{F}'^2 \\ {\scriptstyle I^2}\downarrow & & \downarrow{\scriptstyle I'^2} \\ \mathcal{B}^2 & \xrightarrow[C^2]{} & \mathcal{B}'^2 \\ {\scriptstyle \mathrm{cod}}\downarrow & & \downarrow{\scriptstyle \mathrm{cod}} \\ \mathcal{B} & \xrightarrow[C]{} & \mathcal{B}' \end{array}$$

and F^2 is cartesian because it maps pullback diagrams to pullback diagrams. We only need to show that $F^2K = K'F^2$, where K, K' are the respective kernel-pair comonads. Consider that $K'F^2(A)$ is computed as follows

$$\begin{array}{ccc} \bullet & \longrightarrow & C(\Gamma.A) \\ {\scriptstyle I'(K'FA)}\downarrow & \lrcorner & \downarrow{\scriptstyle I'(FA)} \\ C(\Gamma.A) & \xrightarrow[I'(FA)]{} & C(\Gamma) \end{array}$$

but since $I'F = CI$ by definition of morphism of CE-systems, by uniqueness of the choice of the pullback we have that $F^2K(A) = K'F^2(A)$.

Call L the functor induced by Proposition 3.6.2.3 and R that induced by Proposition 3.6.2.1, we show that $L \dashv R$ with unit ζ and counit θ. We start by describing the counit $\theta\colon LR \to \mathrm{id}$. Let $I\colon \mathcal{F} \to \mathcal{B}$ a CE-system, then $LR(I)\colon \mathcal{F}_{LR} \to \mathcal{B}$ is the CE-system having:

- the free category $\mathcal{F}_{LR}$ generated by the graph G_{LR} having the same objects as $\mathcal{B}$ and morphisms those A in $\mathcal{F}^2$;

- $LR(I)(\Gamma) = \Gamma$, $LR(I)(A) = p(\epsilon_A) = I(A)$.

Since I is a functor, and $\mathcal{B}$ has the same objects as $\mathcal{F}$, G_{LR} is itself a category and $\mathcal{F}_{LR} = \mathcal{F}$. We can take θ to be the identity in **CEsys**.

Conversely, let $(p\colon \mathcal{E} \to \mathcal{B}, K)$ a wc-comonad with p split, and compute RL, which yields the following:

- the free category $\mathcal{F}_K$ generated by the graph with vertices objects in $\mathcal{B}$ and edges objects in $\mathcal{E}$ with appropriate sources and targets;
- the fibration acting as $\mathrm{cod} \circ I_K(A) = \mathrm{cod}(p\epsilon_A) = pA$;
- the comonad K' mapping each A to the cartesian lifting of A at $p(\epsilon_A)$.

Since p is split, $K'A = KA$ by definition of wc-comonad. The unit $\zeta_{(p,K)}$, then can be defined as follows

$$\begin{array}{ccc} \mathcal{E} & \xrightarrow{H} & \mathcal{F}_K^2 \\ {\scriptstyle K}\downarrow & & \downarrow{\scriptstyle K'} \\ \mathcal{E} & \xrightarrow{H} & \mathcal{F}_K^2 \\ {\scriptstyle p}\downarrow & & \downarrow{\scriptstyle \mathrm{cod}\circ I_K^2} \\ \mathcal{B} & \xrightarrow{C} & \mathcal{B} \end{array}$$

with $C = \mathrm{id}$ and H the functor acting on objects as $A \mapsto A$, so that the upper square trivially commutes. On morphisms, we can define it as

$$s\colon B \to A \quad \mapsto \quad (pKs, ps)\colon B \to A$$

because the latter fits an appropriate square with sides B, A. Moreover, if s is cartesian, such a square is a pullback by condition 2 in Definition 3.3.3.1, hence it is $(\mathrm{cod} \circ I_K^2)$-cartesian, so that (H, C) truly is 1-cell in **Fib**.

Finally, we need to show that (L, R, θ, ζ) satisfies the triangle identities, but since $\theta = \mathrm{id}$ this amounts to proving that $L\zeta_{(p,K)} = \mathrm{id}_{(p,K)}$ and $\zeta_{R(I)} = \mathrm{id}_{R(I)}$, and that is readily checked. □

Corollary 3.6.2.5. R is fully faithful.

Construction 3.6.2.6. Combining the results in Theorem 3.6.2.4 and Section 3.3.5,

$$\mathbf{CEsys} \underset{R}{\overset{L}{\underset{\longrightarrow}{\overset{\longleftarrow}{\bot}}}} \mathbf{wcCmd}_{\mathbf{str}}{}^{split} \xrightarrow{\equiv} \mathbf{cDTT}_{\mathbf{str}}{}^{\mathrm{split}}$$

plus observations in Section 3.4, allows us to describe the cDTT associated to a CE-system:

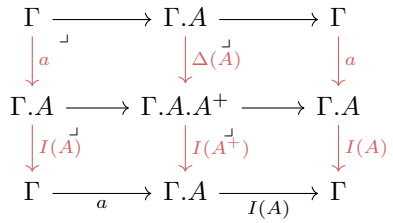
$$\begin{array}{ccc} \mathrm{Sec}_I & \xrightarrow{\Sigma} & \mathcal{F}^2 \\ \downarrow & \lrcorner & \downarrow{\scriptstyle I^2} \\ \mathrm{CoAlg}(K) & \xrightarrow[U]{} & \mathcal{B}^2 \\ & \searrow^{} & \swarrow{\scriptstyle \mathrm{cod}} \\ & \mathcal{B} & \end{array}$$

where the external diagram describes our choices for terms and types fibrations. The right adjoint Δ to the typing functor is defined on objects as follows,

$$\Delta(A) := \nu I(A)$$

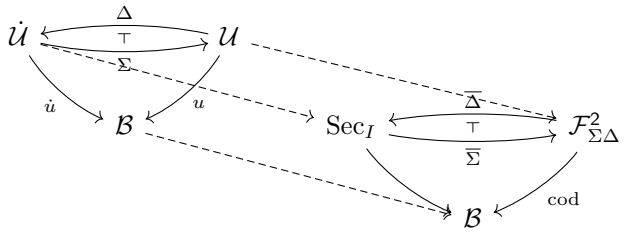
and on maps in the obvious way.

The counit at A is precisely the composition of diagrams appearing in (3.13), and that is clearly a pullback. The unit at a section a is provided by unitality, since we have the following diagram,



where all red compositions are identities.

Because it will be of use later (see Theorem 3.6.3.5), we describe the image of the unit of $L \dashv R$ in $\mathbf{cDTT_{str}}^{\text{split}}$: consider



with

- the category $\mathcal{F}^2_{\Sigma\Delta}$ having for objects finite lists $\lfloor A_0, \dots, A_{n-1} \rfloor$ of objects in $\mathcal{U}$ such that $u(\Sigma\Delta(A_i)) = u(A_{i+1})$ for all $0 \leq i \leq n-2$;
- the category Sec_I of sections of $u(\epsilon_{A_0}) \circ \cdots \circ u(\epsilon_{A_{n-1}})$.

The triple

$$\dot{u}\eta_{(-)} : \dot{\mathcal{U}} \to \mathrm{Sec}_I, \qquad \lfloor - \rfloor : \mathcal{U} \to \mathcal{F}^2_{\Sigma\Delta} \qquad \mathrm{id} : \mathcal{B} \to \mathcal{B},$$

with $\lfloor - \rfloor$ the functor mapping an object A to the list $\lfloor A \rfloor$ of length one (and each morphism to the desired square), is the unit of (the image through the equivalence) of $L \dashv R$.

3.6.3 Constructors for free

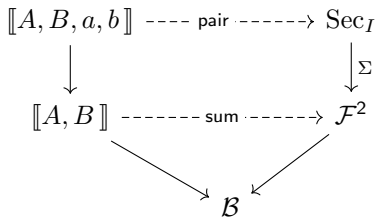
3.6.3.1 CE-systems have dependent sums and units

Lemma 3.6.3.1. Each cDTT obtained from a CE-system following Construction 3.6.2.6 has dependent sums in the sense of Section 3.6.1.2.

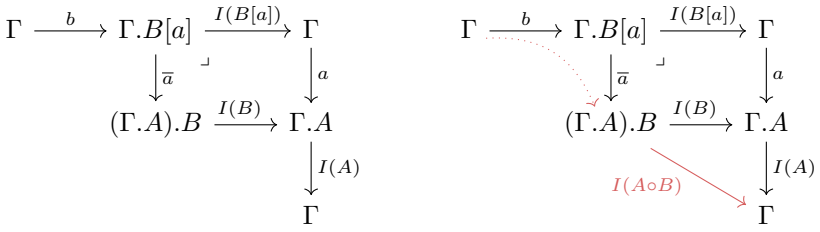
Proof. It has dependent sums if, given the categories encoding the two nested judgements of the premises from, respectively, the formation and the introduction rule,

$$\Gamma \vdash A \quad \Gamma \vdash B \quad (\,\Gamma \vdash a : A \quad \Gamma \vdash b : B[a]\,)$$

and the typing functor connecting them, there is a pair of functors as below



making the triangle commute and the square a pullback. Computing the appropriate categories $[\![A, B, a, b]\!]$ and $[\![A, B]\!]$, one finds that objects of the first are diagrams as on the left



for $A, B, B[a]$ in $\mathcal{F}$ and a, b sections in $\mathcal{B}$. The category $[\![A, B]\!]$ only keeps track of the lower part of the diagram, and the desired vertical functor "forgets" the upper part. Now, since I is functorial, $I(A) \circ I(B) = I(A \circ B)$ *i.e.* there is a type $A \circ B$ and $(\Gamma.A).B = \Gamma.(A \circ B)$. Then we can define on objects

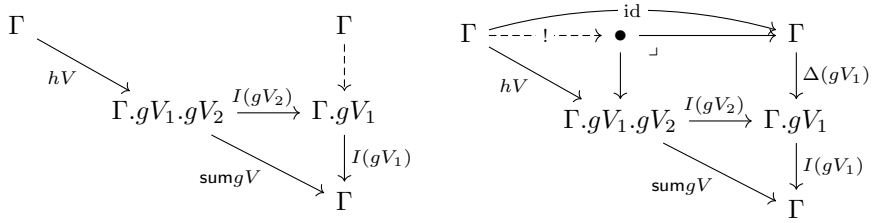
$$\mathsf{sum} : (A, B) \mapsto A \circ B$$

and that will make the triangle commute. It is clear that this extends to a functor $[\![A, B]\!] \to \mathcal{U}$. To define pair, we need to provide a section of $I(A \circ B)$, but now it is sufficient to consider the composition

$$\mathsf{pair} : (A, B, a, b) \mapsto \overline{a} \circ b$$

as in the diagram on the right, and this is again functorial. Clearly the square involving sum, pair commutes, let us now check that it is a pullback.

Let $\mathcal{V}$ a judgement classifier and g, h rules such that $\mathsf{sum}\, g = \Sigma\, h$. This means that gV is a pair of maps gV_1, gV_2 in $\mathcal{F}$ such that we have the following diagram on the left.



There is always a term of type gV_1, that is $\Delta(gV_1)$, which is the dashed arrow in the first diagram above. We complete the pullback and find the unique arrow that induces in the diagram on the right. Then we have described the desired functor $\mathcal{V} \to [\![A, B, a, b]\!]$. □

Corollary 3.6.3.2 (Free dependent sums)**.** One can always freely add dependent sums to a CE-system.

Lemma 3.6.3.3. Each cDTT obtained from a CE-system following Construction 3.6.2.6 has unit types in the sense of Definition 2.3.8.1.

Proof. We can easily provide $\mathsf{1}$, $*$ that do the job: just match to each $\Gamma \in \mathcal{B}$ its identity in both cases. □

Corollary 3.6.3.4 (Free units)**.** One can always freely add unit types to a CE-system.

3.6.3.2 cDTTs with sums and units are CE-systems

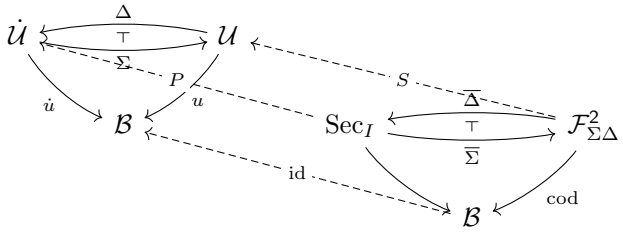
Not only CE-systems have sums and units, but they capture all discrete models of dependent types that do, that is: the following holds.

Theorem 3.6.3.5. The counit of 3.6.2.6 is an isomorphism on all split cDTTs with dependent sums and units.

Proof. Since we already proved Lemma 3.6.3.1 and Lemma 3.6.3.3, combining Theorem 3.6.2.4 and the content of Section 3.3.5 we get the diagram above, but since the counit θ is the identity on all CE-systems, we only need to show that for each split $\mathbb{J}$ with dependent sums and unit types, the unit $\zeta_{\mathbb{J}}$ is itself an iso. Recall from Construction 3.6.2.6 that its components are

$$\dot{u}\eta_{(-)} : \dot{\mathcal{U}} \to \mathrm{Sec}_I, \qquad \lfloor - \rfloor : \mathcal{U} \to \mathcal{F}^2_{\Sigma\Delta} \qquad \mathrm{id} : \mathcal{B} \to \mathcal{B},$$

with $\lfloor - \rfloor$ the functor mapping an object A to the list $\lfloor A \rfloor$ of length one. We define (P, S, id) a strict morphism of split cDTTs $RL\mathbb{J} \to \mathbb{J}$ and show that it is an inverse to $\zeta_{\mathbb{J}}$.



Our strategy is to implicitly use the structure on $L\mathbb{J}$ to *break up* pieces of data, which can be then canonically sent to $\mathbb{J}$, and then *built back up* with the existing structure there.

Let us start from types: we will take advantage of sum (on the original cDTT!) to describe a sort of concatenation on types, then use it to define the action of S inductively on the length of lists in $\mathcal{F}_{\Sigma\Delta}$. Let $\lfloor A, B \rfloor$ in $\mathcal{F}^2_{\Sigma\Delta}$, we then have the following

$$u(A) \xleftarrow{A} u(\Sigma\Delta(A)) = u(B) \xleftarrow{B} u(\Sigma\Delta(B))\,.$$

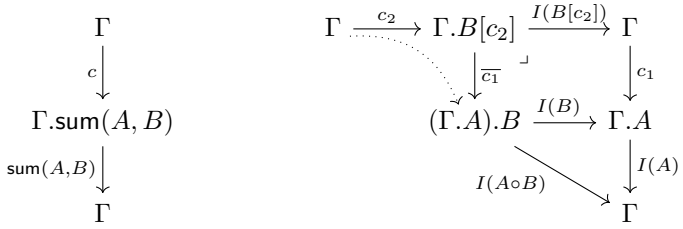
The middle condition is precisely stating that (A, B), as a pair of objects in $\mathcal{U}$, is in $[\![A, B]\!]$. Then, since $\mathbb{J}$ has dependent sums, we can compute its image through sum. We define the functor S on objects as follows, then:

- $S\lfloor\ \rfloor = 1_\Gamma$ for Γ the base-point of the loop;
- $S\lfloor A\rfloor = A$ for A in $\mathcal{U}$;
- $S\lfloor A_0, \dots, A_{n-1}\rfloor = \mathsf{sum}(S\lfloor A_0, \dots, A_{n-2}\rfloor, S\lfloor A_{n-1}\rfloor)$.

It is clear that this produces a fibration morphism, given that sum is one. On one hand, $S \circ \lfloor - \rfloor = \mathrm{id}$ by construction of S, on the other we only need to check that

$$\lfloor \mathsf{sum}(A, B)\rfloor = A \circ B,$$

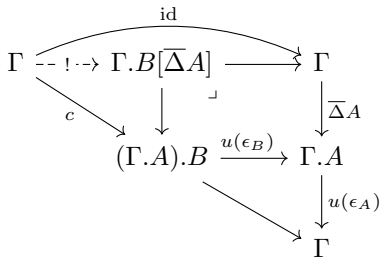
and the rest follows by induction on the length of lists. Since we have dependent sums in the sense of Section 3.6.1.2, by pullback we have an isomorphism of categories matching to each section c of $\mathsf{sum}(A, B)$ a pair (c_1, c_2) as follows.



(3.14)

The computation η, in particular, yields $\mathsf{pair}(A, B, c_1, c_2) = c$, so that $I(A \circ B) = \mathsf{sum}(A, B)$ and $\overline{c_1} \circ c_2 = c$.

We now define a compatible functor $P\colon \mathrm{Sec}_I \to \dot{\mathcal{U}}$, again on the length of the type of the section c. Notice that if c is a section of $I(A \circ B) = u(\epsilon_A) \circ u(\epsilon_B)$, then there is a pair of sections as below



which we write (c_1, c_2), hence from a type of length 2 we have canonically produced two types of length 1: we can compute P one step down the desired induction

$$P(c, \lfloor A, B\rfloor) = \mathsf{pair}(\, P(c_1, \lfloor A\rfloor)\,,\, P(c_2, \lfloor B[\overline{\Delta}A]\rfloor)\,),$$

meaning

- if c is a section t of a type of length 0 with base-point Γ, define $P(t, \lfloor\ \rfloor)$ to be the base-point loop $*_\Gamma$;

- if c is a section a of $I(A) = u(\epsilon_A)$, we exploit the fact that $\dot{u}$ is a fibration,

$$\begin{array}{ccc} a^*\Delta A & \xrightarrow[a']{} & \Delta A \\ \Gamma & \xrightarrow[a]{} & \dot{u}(\Delta A) = u(\Sigma\Delta A) \xrightarrow[u(\epsilon_A)]{} u(A) \end{array} \qquad \begin{array}{c} \dot{\mathcal{U}} \\ \downarrow \\ \mathcal{B} \end{array}$$

  and claim that $\eta_{a^*\Delta A} = a'$, hence we can define $P(a, \lfloor A \rfloor) = a^*\Delta A$;

- finally, define $P(c, \lfloor A_0, \dots, A_{n-1} \rfloor)$ as follows

$$\mathsf{pair}(\, \lfloor P(c_1, A_0, \dots, A_{n-2}) \rfloor \,,\, \lfloor P\,(\, c_2 \,,\, A_{n-1}[\overline{\Delta}\lfloor A_0, \dots, A_{n-2} \rfloor]\,)\rfloor\,)$$

  with (c_1, c_2) constructed as detailed above.

It is a fibration morphism because of Remark 3.3.3.3 and because pair is. We only need to show that it actually is $\eta_{a^*\Delta A} = a'$, but this follows from $\dot{u}$ split, while $P \circ \dot{u}\eta_{\_} = \mathrm{id}$ is trivial by construction.

Finally, for it to be part of a morphism of cDTTs, we need commutativity of certain squares: we only prove it for terms with type of length 1. Everything else follows immediately from the fact that the original cDTT has dependent sums and, in particular, for what was observed in (3.14). Let us first check that $S\overline{\Sigma} = \Sigma P$: a given section a of $\lfloor A \rfloor : uA \to u\Sigma\Delta A$ is sent to the cartesian lifting $a^*\Delta A$ whose image through Σ is A

$$\begin{array}{ccccc} a^*\Delta A & \xrightarrow[a']{} & \Delta A & & \\ \Sigma(a^*\Delta A) & \xrightarrow[\Sigma(a')]{} & \Sigma\Delta A & \xrightarrow{\epsilon_A} & A \\ \Gamma & \xrightarrow[a]{} & \Gamma.A & \xrightarrow[\lfloor A \rfloor]{} & \Gamma \end{array}$$

(with id : $\Sigma(a^*\Delta A) \to A$)

because $\Sigma(a')$ is both cartesian and a section of $u\epsilon_A$. Moreover, it is also $P\overline{\Delta} = \Delta S$ because at a given $\lfloor A \rfloor$ we compute the following

$$\begin{array}{ccccc} \overline{\Delta}\lfloor A \rfloor^*(\Delta A^+) & \xrightarrow[\overline{\Delta}\lfloor A \rfloor']{} & \Delta A^+ = \Delta\Sigma\Delta A & \xrightarrow[\Delta\epsilon_A]{} & \Delta A \\ \Gamma.A & \xrightarrow[\overline{\Delta}\lfloor A \rfloor]{} & \Gamma.A.A^+ & \xrightarrow[\lfloor A^+ \rfloor]{} & \Gamma.A \\ & & \downarrow{\scriptstyle \lfloor A^+ \rfloor} & & \downarrow{\scriptstyle \lfloor A \rfloor} \\ & & \Gamma.A & \xrightarrow[\lfloor A \rfloor]{} & \Gamma \end{array}$$

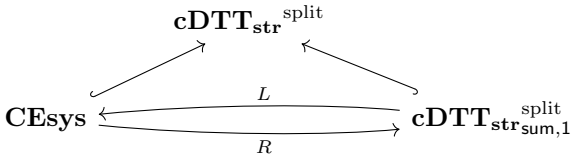
(with id : $\Gamma.A \to \Gamma.A$ and id : $\Gamma.A \to \Gamma.A.A^+$)

and since u is split, the cartesian lifting of $\mathrm{id} : \Gamma.A \to \Gamma.A.A^+ \to \Gamma.A$ is itself the identity, so that, since both maps in $\dot{\mathcal{U}}$ are cartesian by construction, because ϵ is, and by Lemma 3.3.2.5, then $\overline{\Delta}\lfloor A \rfloor^*(\Delta A^+) = \Delta A = \Delta S\lfloor A \rfloor$. □

Corollary 3.6.3.6. CE-systems are in a 1-to-1 correspondence with split cDTTs with dependent sums and units.

A more precise and desirable result would be the following.

Theorem 3.6.3.7. The adjunction $L \dashv R$ restricts to an equivalence of categories



where $\mathbf{cDTT_{str}}^{\text{split}}_{\mathsf{sum},1}$ is the subcategory of $\mathbf{cDTT_{str}}^{\text{split}}$ of cDTTs with dependent sums and units.

The problem is that that would require us to describe what happens to 1-cells in $L \dashv R$, and what 1-cells in $\mathbf{cDTT_{str}}^{\text{split}}_{\mathsf{sum},1}$ should even be. We leave this problem for a later time, nevertheless we hope that this result appropriately witnesses the use in the judgemental approach to the analysis of categorical models dependent types.

3.7 Future developments

We have framed cDTTs in the crowded world of categorical models of dependent types, and have provided at least two good reasons as to why that could be of use (3.5, 3.6).

Many more seem within reach: more structure on the collection of types over a given context provides more tools to relate them, for example one could be interested in mimicking what happens with doctrines (2.4.0.5) with connectives and quantifiers. Another point to make would be comparing compcats and cDTTs not only with respect to structural rules, but with type constructors, as well.

Chapter 4

Fuzzy dependent types

> "What giants?" asked Sancho Panza. "Those thou seest there," answered his master, "with the long arms, and some have them nearly two leagues long." "Look, your worship," said Sancho, "what we see there are not giants but windmills, and what seem to be their arms are the sails that turned by the wind make the millstone go." "It is easy to see," replied Don Quijote, "that thou art not used to this business of adventures."
>
> [CS10, Chapter 8]

The long-distance scope of this project is applying categorical tools usually found in opinion dynamics, namely cellular sheaves [GH20, GR22], to a fuzzy environment. The first step in the process is that of producing a suitable logic to express judgements with variable degrees of confidence, and we do that combining notions of enriched category theory with a classical model for dependent types.

Contribution

We begin the study of a theory of fuzzy types from their structural rules. We prove soundness and completeness for their calculus.

4.1 Basic enriched category theory

Enriched category theory stems from the observation that sometimes in math the collection of maps between two given objects has itself some structure other than that of a set, for example, given a field k, the set of k-linear maps between k-vector spaces is itself a k-vector space. Indeed it is clear how the very own definition of category (Section 1.1.1) and functor (Section 1.1.2) heavily relies

on the notion of set and, more technically, on the category **Set**, and this fails to capture, for example, the case of vector spaces. Sets seem unefficient even when one wants to express the concept of morphisms between functors, and this is just needed to get us to 2-categories (1.1.3). But as it has been hinted in Section 1.1.4, putting sets on a pedestal is not at all necessary: on one hand, we could use different foundations (see for example [Bén85]), on the other, we could acknowledge the problem and build a new theory on top of the existing one with the purpose of fixing this glitch. We follow this second route in the steps of [Kel82].

Notice that, for size reasons, our discussion only takes into account locally small categories, meaning categories whose collections of morphisms for any given pair of objects is a set. If we wished to be completely general, we would need to introduce a category of big sets, probably calling that **SET** or **Class**, and so on. We hope the reader sees that that does not change the discussion that follows.

4.1.1 Monoidal categories

First of all, we ought to understand what it is about sets that makes them good at collecting morphisms, at composing them, at picking identities, and so on. Recall from Definition 1.1.1.1 that a category is

- a collection of objects, and
- for each pair of objects A, B a set of morphisms from A to B,

moreover

1. for any object A there is a designated identity morphism $\mathrm{id}_A \colon A \to A$;
2. for any composable pair of morphisms f, g there is a morphism gf computing their composition;

and these must satisfy the following equations: for $f \colon A \to B$, $g \colon B \to C$, $h \colon C \to D$,

$$f\mathrm{id}_A = f = \mathrm{id}_B f, \quad h(gf) = (hg)f.$$

Denoting $\mathcal{C}(A, B)$ the set of maps from A to B, we can reformulate the conditions above in term of structures in **Set** as follows: a category is

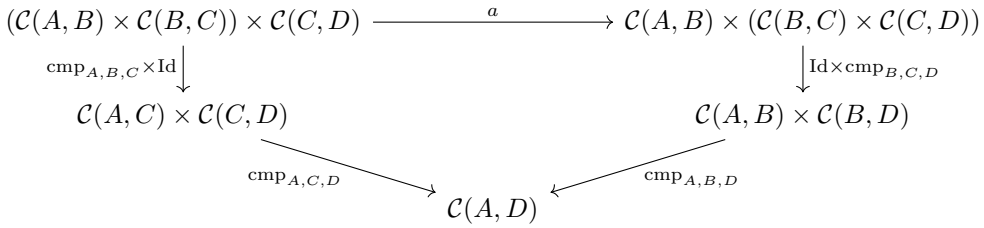
- a collection of objects, and
- for each pair of objects A, B an *hom-set* $\mathcal{C}(A, B)$ in **Set**

moreover

1. for any A there is a function $\mathrm{id}_A \colon 1 \to \mathcal{C}(A, A)$;
2. for any A, B, C there is a function $\mathrm{cmp}_{A,B,C} \colon \mathcal{C}(A, B) \times \mathcal{C}(B, C) \to \mathcal{C}(A, C)$;

such that the following diagrams commute.

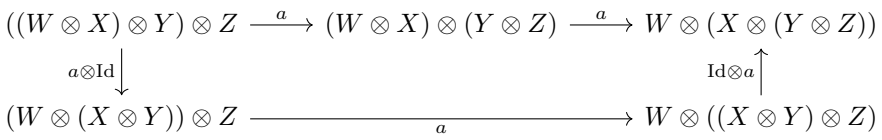
$$\begin{array}{ccccc}
1 \times \mathcal{C}(A,B) & \xrightarrow{l} & \mathcal{C}(A,B) & \xleftarrow{r} & \mathcal{C}(A,B) \times 1 \\
{\scriptstyle \mathrm{id}_A \times \mathrm{Id}}\downarrow & \nearrow_{\mathrm{cmp}_{A,A,B}} & & \nwarrow_{\mathrm{cmp}_{A,B,B}} & \downarrow{\scriptstyle \mathrm{Id} \times \mathrm{id}_B} \\
\mathcal{C}(A,A) \times \mathcal{C}(A,B) & & & & \mathcal{C}(A,B) \times \mathcal{C}(B,B)
\end{array}$$

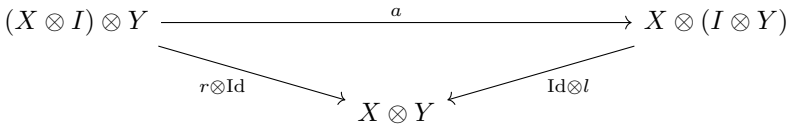


Here $1 = \{*\}$, functions l, r project on the $\mathcal{C}(A,B)$ component, and a performs association.

The *intrinsic* structure of **Set** needed is the object 1, the cartesian product $\times$ - although we do not use its universal property, just that it allows for functions to be applied "in parallel" - and the existence of l, r, a. On top of this intrinsic structure on set, $\mathcal{C}$ is a category if suitable $\mathrm{id}, \mathrm{cmp}$ exist: generalizing this will be the aim of Section 4.1.2.

Definition 4.1.1.1 (Monoidal category)**.** A *monoidal category* $\mathcal{V} = (\mathcal{V}_0, \otimes, I, a, l, r)$ consists of a category $\mathcal{V}_0$, a functor $\otimes\colon \mathcal{V}_0 \otimes \mathcal{V}_0 \to \mathcal{V}_0$, a designated object I of $\mathcal{V}_0$, and natural isomorphisms $a_{X,Y,Z}\colon (X\otimes Y)\otimes Z \to X\otimes(Y\otimes Z)$, $l_X\colon I\otimes X \to X$, $r_X\colon X\otimes I \to X$ such that the two following diagrams, representing what are called the *coherence axioms*, commute.





Often we identify $\mathcal{V}_0$ with $\mathcal{V}$, but it is important to notice that there may be different monoidal structures on the same category: this will be the case for ref, for example. We conclude this discussion by listing a few properties of monoidal categories.

Definition 4.1.1.2 (Symmetric monoidal category)**.** A monoidal category is said to be *symmetric* if there is a natural isomorphism $c_{X,Y}\colon X\otimes Y \to Y\otimes X$ satisfying suitable coherence axioms [Kel82, § 1.4].

Definition 4.1.1.3 (Cartesian monoidal category)**.** A monoidal category is said to be *cartesian* if $\otimes$ is a product, I the terminal object, l, r are projections, and a is the natural isomorphism induced by the universal property of products.

Definition 4.1.1.4 (Closed monoidal category)**.** A monoidal category is said to be *closed* if each functor $-\otimes Y\colon \mathcal{V}_0 \to \mathcal{V}_0$ has a right adjoint $[Y, -]$. The object $[Y, Z]$ is called the *internal hom* of Y and Z.

Remark 4.1.1.5 (Internal homs)**.** Consider the bijection induced by $- \otimes Y \dashv [Y, -]$ at I

$$\mathcal{V}_0(I \otimes Y, Z) \cong \mathcal{V}_0(I, [Y, Z]),$$

then, given $I \otimes Y \cong Y$, we get

$$\mathcal{V}_0(Y, Z) \cong \mathcal{V}_0(I, [Y, Z])$$

so that we can relate homomorphisms between Y and Z to their internal hom.

4.1.2 Enriched categories

Now that we have the appropriate structure on a category $\mathcal{V}$, we can define what it means for a category to be $\mathcal{V}$-enriched. We fix a monoidal category $\mathcal{V}$ for the rest of this discussion.

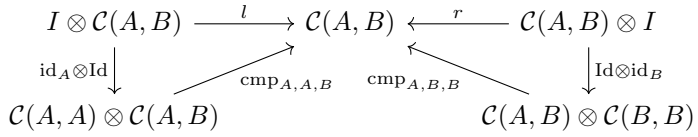
Definition 4.1.2.1 ($\mathcal{V}$-enriched category)**.** A *$\mathcal{V}$-category* $\mathcal{C}$ is the data of

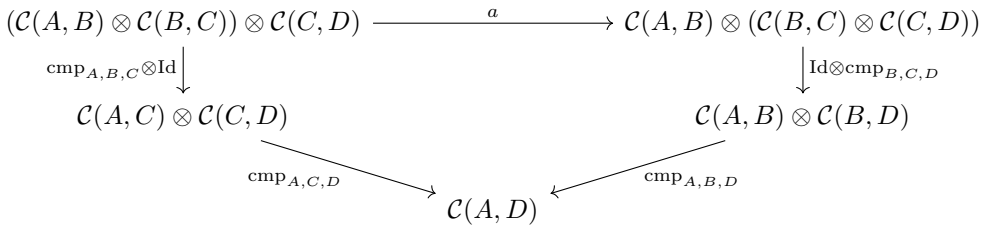
- a collection of objects,
- for each pair of objects A, B an *hom-object* $\mathcal{C}(A, B)$ in $\mathcal{V}_0$

moreover

1. for any A there is a $\mathcal{V}$-morphism $\mathrm{id}_A \colon I \to \mathcal{C}(A, A)$;
2. for any A, B, C there is a $\mathcal{V}$-morphism $\mathrm{cmp}_{A,B,C} \colon \mathcal{C}(A, B) \otimes \mathcal{C}(B, C) \to \mathcal{C}(A, C)$;

such that the following diagrams commute.





Remark 4.1.2.2 (The definition is meaningful)**.** Considering each time $\mathcal{V}$ to be (the obvious monoidal structures on) $\mathbf{Set}, \mathbf{Cat}, \mathbf{2}, \mathbf{Ab}, \mathbf{DG}\text{-}R\text{-}\mathbf{Mod}, \overline{\mathbb{R}}_+$ we respectively recover the notions of locally small category, 2-category, pre-ordered set, additive category, differential graded category, and (generalized) metric space.

4.1.3 Enriched functors and natural transformations

Similarly, we can think of a functor $F\colon \mathcal{C} \to \mathcal{D}$ between regular categories as function

$$\mathcal{C}(A,B) \to \mathcal{D}(FA, FB)$$

for each pair of objects A, B.

Definition 4.1.3.1 ($\mathcal{V}$-functor)**.** For $\mathcal{V}$-categories $\mathcal{C}, \mathcal{D}$, a *$\mathcal{V}$-functor* $F\colon \mathcal{C} \to \mathcal{D}$ consists of

- for each object A in $\mathcal{A}$, an object FA in $\mathcal{D}$, and
- for each pair of objects A, B, a $\mathcal{V}$-morphism $F_{AB}\colon \mathcal{C}(A,B) \to \mathcal{D}(FA, FB)$,

subject to the compatibility with composition and with identities, meaning that the following diagrams must commute.

$$\begin{array}{ccc} \mathcal{C}(A,B) \otimes \mathcal{C}(B,C) & \xrightarrow{\mathrm{cmp}^{\mathcal{C}}} & \mathcal{C}(A,C) \\ {\scriptstyle F\otimes F}\downarrow & & \downarrow{\scriptstyle F} \\ \mathcal{D}(FA,FB) \otimes \mathcal{D}(FB,FC) & \xrightarrow[\mathrm{cmp}^{\mathcal{D}}]{} & \mathcal{D}(FA,FC) \end{array} \qquad \begin{array}{ccc} I & \xrightarrow{\mathrm{id}^{\mathcal{C}}} & \mathcal{C}(A,A) \\ & {\scriptstyle \mathrm{id}^{\mathcal{D}}}\searrow & \downarrow{\scriptstyle F} \\ & & \mathcal{D}(FA,FA) \end{array}$$

Similarly one can define a notion of *enriched natural transformation*.

Definition 4.1.3.2 ($\mathcal{V}$-natural transformation)**.** Let $F, G\colon \mathcal{C} \to \mathcal{D}$ $\mathcal{V}$-functors, a *$\mathcal{V}$-natural transformation* $\alpha\colon F \Rightarrow G$ is a family of maps $\alpha_A\colon I \to \mathcal{D}(FA, GA)$ such that the following diagram commutes.

$$\begin{array}{ccccc} & & I \otimes \mathcal{C}(A,B) & \xrightarrow{\alpha_A \otimes G} & \mathcal{D}(FA,GA) \otimes \mathcal{D}(GA,GB) \\ & {\scriptstyle l^{-1}}\nearrow & & & \searrow{\scriptstyle \mathrm{cmp}} \\ \mathcal{C}(A,B) & & & & & \mathcal{D}(FA,GB) \\ & {\scriptstyle r^{-1}}\searrow & & & \nearrow{\scriptstyle \mathrm{cmp}} \\ & & \mathcal{C}(A,B) \otimes I & \xrightarrow[F\otimes \alpha_B]{} & \mathcal{D}(FA,FB) \otimes \mathcal{D}(FB,GB) \end{array}$$

In the case of natural transformations of mixed variance, some work is being done in the direction of a calculus of substitution [MS21].

4.1.4 Forgetful into CAT

$\mathcal{V}$-categories, $\mathcal{V}$-functors (and $\mathcal{V}$-natural transformations) assemble into a (2-)category.

Definition 4.1.4.1 (The category of $\mathcal{V}$-enriched categories)**.** We call $\mathcal{V}$-**CAT** the (2-)category of $\mathcal{V}$-categories, $\mathcal{V}$-functors (and $\mathcal{V}$-natural transformations).

Definition 4.1.4.2 (The forgetful V)**.** If $\mathcal{V}$ has $\mathcal{V}_0$ locally small, we have a functor

$$V = \mathcal{V}_0(I, -)\colon \mathcal{V}_0 \to \mathbf{Set}$$

which computes what is called the *underlying set*.

The functor V often has some strong properties, for example when $\mathcal{V} = \mathbf{Set}, \mathbf{Ord}, \mathbf{Top}, \mathbf{Ab}, R\text{-}\mathbf{Mod}$ it is faithful and acts as the usual forgetful functor, but this is not always the case.

Definition 4.1.4.3 (The forgetful $(-)_0$)**.** Call $\mathcal{I}$ the $\mathcal{V}$-category with a single object $*$ and $\mathcal{I}(*,*) = I$. Then the 2-functor

$$(-)_0 = \mathcal{V}\text{-}\mathbf{CAT}(\mathcal{I}, -) \colon \mathcal{V}\text{-}\mathbf{CAT} \to \mathbf{CAT}$$

computes what is called the *underlying category*.

Remark 4.1.4.4. Unfolding the definition of $(-)_0$ shows that the category $\mathcal{C}_0$ has the same objects as $\mathcal{C}$, and a map $f \colon A \to B$ in $\mathcal{C}_0$ is in particular a choice of a map in $\mathcal{C}(A, B)$.

4.1.5 Representable $\mathcal{V}$-functors

For the rest of this section we assume $\mathcal{V}$ symmetric (4.1.1.3) monoidal (4.1.1.4) with $\mathcal{V}_0$ locally small.

Remark 4.1.5.1. The internal hom described in Remark 4.1.1.5 makes $\mathcal{V}$ a $\mathcal{V}$-category: its objects are those of $\mathcal{V}_0$, and for each pair X, Y we say that $\mathcal{V}(X, Y)$ is $[X, Y]$.

Remark 4.1.5.2. For each $\mathcal{V}$-category $\mathcal{C}$ and object A in $\mathcal{C}$, we can define the *(covariant) representable functor*

$$\mathcal{C}(A, -) \colon \mathcal{C} \to \mathcal{V}$$

sending B to $\mathcal{C}(A, B)$ and on pairs A, B acting as follows

$$\mathcal{C}(A, -)_{BC} \colon \mathcal{C}(B, C) \to \mathcal{V}(\mathcal{V}(A, B), \mathcal{C}(A, C)) = [\mathcal{C}(A, B), \mathcal{C}(A, C)]$$

corresponding under the adjunction in Definition 4.1.1.4 to

$$\mathrm{cmp}_{A,B,C} \colon \mathcal{C}(A, B) \otimes \mathcal{C}(B, C) \to \mathcal{C}(A, C).$$

Similarly one can define a *contravariant representable functor*.

Combining both the contravariant and the covariant representable functors, one can define a functor

$$\mathcal{C}^{\mathsf{op}} \otimes \mathcal{C} \to \mathcal{V}.$$

4.1.6 Let us end with $\mathcal{V}$

We conclude this extremely brief introduction with a look at ends in this enriched context: they are a technical tool that will be of fundamental importance when we compute weighted limits, which in turn will be crucial in modeling substitution.

For this discussion we will assume that $\mathcal{V}$ is symmetric monoidal closed, with $\mathcal{V}_0$ locally small, and additionally that $\mathcal{V}_0$ is complete. These hypothesis will all be satisfied by our categories of interest.

Definition 4.1.6.1 (End)**.** Consider a $\mathcal{V}$-functor $T \colon \mathcal{C}^{\mathsf{op}} \otimes \mathcal{C} \to \mathcal{V}$. If there exists a universal $\mathcal{V}$-natural family $\lambda_A \colon K \to T(A, A)$, we call (K, λ) the *end* of T. We write $\int_{A \in \mathcal{C}} T(A, A)$ for the object K and call λ_A the *counit* of the end.

Notice that in the naturality condition in Definition 4.1.3.2 in the case where $\mathcal{D} = \mathcal{V}$ translates because of $\mathcal{V}_0(X, [Y, Z]) \cong \mathcal{V}_0(X \otimes Y, Z) \cong \mathcal{V}_0(Y, [X, Z])$ to commutativity of squares of the following kind,

$$\begin{array}{ccc} K & \xrightarrow{\lambda_A} & T(A,A) \\ {\scriptstyle \lambda_B}\downarrow & & \downarrow{\scriptstyle \rho_{AB}} \\ T(B,B) & \xrightarrow[\sigma_{AB}]{} & [\mathcal{C}(A,B), T(A,B)] \end{array}$$

where σ and ρ are the images through the isomorphism above of, respectively, $T(-,B)_{BA}$ and $T(A,-)_{AB}$.

Notice also that the universal property describing ends, meaning that for any other $\mathcal{V}$-natural transformation $\mu_A \colon X \to T(A,A)$ there is a unique $x \colon X \to K$ such that $\mu_A = \lambda_A x$, is in such terms expressed as a bijection of sets, but can be lifted to an isomorphism in $\mathcal{V}_0$. In fact, consider

$$[\mathrm{id}_X, \lambda_A] \colon [X, \int_A T(A,A)] \to [X, T(A,A)],$$

one can see that

$$\int_A [X, T(A,A)] \cong [X, \int_A T(A,A)]$$

and it is so precisely by counit $[\mathrm{id}_X, \lambda_A]$. For more on this topic we refer to [Kel82, §2], while for an in-depth treatment of ends we point the reader to [Lor21]. We conclude this discussion providing two examples in the familiar case where $\mathcal{V} = \mathbf{Set}$ with the cartesian monoidal structure.

Example 4.1.6.2 (Natural transformations)**.** Natural transformations can be written as an end. Letting $F, G : \mathcal{C} \to \mathcal{D}$ be functors, natural transformations between F and G form a subset of $\prod_{X:\mathcal{C}} \mathcal{D}(F(X), G(X))$.

Then, taking the end of $\mathcal{D}(F(-), G(-)) : \mathcal{C}^{\mathsf{op}} \times \mathcal{C} \to \mathbf{Set}$, we have by the commutativity requirement that the following diagram commutes:

$$\begin{array}{ccccccc} \tau & \in & \int_X \mathcal{D}(F(X), G(X)) & \xrightarrow{\pi_Y} & \mathcal{D}(F(Y), G(Y)) & \ni & \tau_Y \\ & & {\scriptstyle \pi_X}\downarrow & & \downarrow{\scriptstyle F(f)^*} & & \\ \tau_X & \in & \mathcal{D}(F(X), G(X)) & \xrightarrow[G(f)_*]{} & \mathcal{D}(F(X), G(Y)) & \ni & G(f)_* \tau_X = F(f)^* \tau_Y \end{array}$$

and this actually is exactly the condition that τ forms a natural transformation from F to G, as this diagram commuting implies that the following diagram commutes.

$$\begin{array}{ccc} F(X) & \xrightarrow{F(f)} & F(Y) \\ {\scriptstyle \tau_X}\downarrow & & \downarrow{\scriptstyle \tau_Y} \\ G(X) & \xrightarrow[G(f)]{} & G(Y) \end{array}$$

Example 4.1.6.3 (Limits)**.** Recall that for a diagram $D : \mathcal{D} \to \mathcal{C}$, the limit of D is the object $\lim D$ in $\mathcal{C}$ such that, for all X in $\mathcal{C}$,

$$\mathcal{C}(X, \lim D) \cong (\Delta_X \Rightarrow D)$$

That is, the natural transformations from the constant functor at X to D is in bijection with the morphisms from X to $\lim D$.

Then, we can represent $\Delta_X \Rightarrow D$ by the end $\int_{Y:\mathcal{D}} \mathcal{C}(X, D(Y))$, so that we get that the the limit of a diagram $D : \mathcal{D} \to \mathcal{C}$ can be equivalently stated as the object $\lim D$ in $\mathcal{C}$ such that, for all X in $\mathcal{C}$,

$$\mathcal{C}(X, \lim D) \cong \int_Y \mathcal{C}(X, D(Y)). \tag{4.1}$$

4.2 Propositions and types (and opinions)

Our aim is to – very roughly – model opinions, so we should first start by saying what it is that we consider to be an opinion. On this, we follow the path suggested by the correspondence that is mostly known as Curry-Howard: since there is a one-to-one correspondence between logics and programming languages, which kind of looks like the following

proofs	executions (terms)
formula	specification of progr. (type)

what we do is simply add a leg to it:

proofs	programs (terms)	motivations
formula	specification of progr. (type)	belief

In our setting, programs are beliefs and executions are thoughts that lead to holding such beliefs. This is of course reductive of the human mind. Nevertheless, we might be able to gain some insight out of it.

Now, given a belief and a motivation for it, I might consider it good or bad. For example, I could motivate the belief "bees should be protected" either with "they carry pollen between plants" or "I like honey", but perhaps they are not influential *to the same extent*. This is where fuzziness comes into play.

In the end, our task is the following: we need to model types (read: opinions) in a way such that their terms are fuzzy in some sense. Our solution entails considering notion of type to its formulation by Per Martin-Löf [Mar75] and restricting it to the interpretation of types to sets and terms to their elements. This context is inherently binary, so that looking for a notion of fuzzy type amounts to finding a filler in the following table.

	binary	*fuzzy*
propositions	$\{0, 1\}$	$[0, 1]$
types	**Set**	?

Our claim is that such a candidate should be $\Sigma_{S:\mathbf{Set}} S \to [0, 1]$. It is also the starting point of our journey.

4.3 The category of fuzzy sets

4.3.1 Measuring fuzziness

The minimum requirement we are willing to start from is that of a set with an ordered relation, so that we compare confidence, have it increase and so on, and a monoidal structure, so that we can enrich in, and that these interact properly. This leads to the notion of commutative ordered monoid.

Definition 4.3.1.1 (Commutative ordered monoid)**.** We say that $\mathbb{M} = (M, \cdot, 1, \leq)$ is a *commutative ordered monoid* if $(M, \leq)$ is a partial order, $(M, \cdot, 1)$ is a monoid, and for all $m, n, x \in M$,

$$\text{if } m \leq n \text{ then } x \cdot m \leq x \cdot n.$$

We say that $\mathbb{M}$ is

1. *integral* when the unit of the monoid is the top element of the order,
2. *idempotent* when $x \cdot x = x$, for all $x \in \mathbb{M}$.

Remark 4.3.1.2. A commutative monoid $(M, \cdot, 1)$ is in fact a small thin symmetric monoidal category.

A particular class of commutative ordered monoids is that of quantales.

Definition 4.3.1.3 (Quantale)**.** A *quantale* $\mathbb{Q} = (Q, \cdot, 1, \leq)$ is a complete lattice with the structure of a commutative monoid that satisfies the following distributive laws: for all $a \in Q$ and $\{b_i\}_{i \in I} \subseteq Q$

$$a \cdot (\bigvee_{i \in I} b_i) = \bigvee_{i \in I} (a \cdot b_i)\,.$$

Often the definition of a quantale only requires the structure of a (non-necessarily abelian) semigroup in $(Q, \cdot)$, but in this work we will always consider quantales which are commutative and with units.

Remark 4.3.1.4. A quantale supports (at least) *two* monoidal structures, where one is that in Remark 4.3.1.2, and one is the cartesian one. This will produce two monoidal structures on our category of fuzzy sets of choice, see Proposition 4.3.2.4-2 and Proposition 4.3.2.9.

Proposition 4.3.1.5 (Characterizing quantales in complete lattices)**.** A complete lattice $\mathbb{Q}$ with the structure of a commutative monoid is a quantale iff it admits an operation $\to\colon \mathbb{Q}^{op} \times \mathbb{Q} \to \mathbb{Q}$ satisfying

$$a \cdot b \leq c \quad \text{iff} \quad a \leq (b \to c)$$

for all $a, b, c \in \mathbb{Q}$.

Proof. Define $b \to c = \bigvee\{a \in \mathbb{Q} \,:\, a \cdot b \leq c\}$. Suppose $\mathbb{Q}$ is a quantale. If $a \cdot b \leq c$, then $a \leq b \to c$, since $\to$ is the supremum. On the other hand, if $a \leq b \to c$ then

$$\begin{aligned} a \cdot b &\leq (b \to c) \cdot b \\ &= \bigvee\{a \in \mathbb{Q} \,:\, a \cdot b \leq c\} \cdot b \\ &= \bigvee\{a \cdot b \,:\, a \cdot b \leq c\} \\ &\leq c. \end{aligned}$$

Now, suppose $\mathbb{Q}$ is a complete lattice satisfying $a \cdot b \leq c$ iff $a \leq (b \to c)$ for all $a, b, c \in \mathbb{Q}$. Such condition means that the functor $(-) \cdot b$ has a right adjoint $b \to (-)$. So $(-) \cdot b$ preserves colimits. In particular, it preserves joins. □

We now give some examples of $\mathbb{M} = (M, \cdot, 1, \leq)$ commutative ordered monoids that are integral and complete, most of them are integral quantales.

Example 4.3.1.6 (Commutative ordered monoids)**.**

1. $(\{0,1\}, \cdot, 1, \leq)$ where the product and the order are the usual, i.e., inherited from the real numbers.

2. $(\mathcal{O}(X), \cap, X, \subseteq)$ where $\mathcal{O}(X)$ is the set of open subsets of a topological space X, the product is the intersection and the order is given by the inclusion of sets.

3. $([0,\infty],\ \max\ , 0, \geq)$ where the product is the maximum and we consider the reverse order.

4. $([0,\infty], +, 0, \geq)$ where the product is the sum of non-negative real numbers and we consider the reverse order.

5. $\mathbb{I} = ([0,1], \cdot, 1, \leq)$ where the product and the order are the usual.

6. $([0,1], T, 1, \leq)$ where T is a t-norm. A t-norm is a binary operator $T : [0,1]^2 \to [0,1]$ such that for all $a, b, c \in [0,1]$, $T(a,b) = T(b,a), T(a, T(b,c)) = T(T(a,b), c), T(a,1) = a$, and $T(a,b) \leq T(c,d)$, if $a \leq c$ and $b \leq d$.

7. $(\mathcal{I}(R), \cdot, R, \subseteq)$ where $\mathcal{I}(R)$ is the set of ideals of a commutative ring R with unit, the product is the product of ideals and the order is given by the inclusion.

8. If $\mathbb{M}$ is a commutative ordered monoid and X a fixed set, then the set $\mathbb{M}^X$ of all functions from X to $\mathbb{M}$ is a commutative ordered monoid where we the order of two functions is given by $f \leq g$ iff $f(x) \leq g(x), \forall x \in X$, the product is pointwise, and the unity is the constant function whose value is the unity of $\mathbb{M}$.

9. The set $\triangle = \{f : [0,\infty] \to [0,1] \ : \ f \text{ is monotone and } f(x) = \bigvee_{y<x} f(y)\}$ with point-wise order and product for $f, g \in \triangle$ given by $(f \otimes g)(t) = \bigvee_{r+s \leq t} f(r) \cdot g(s)$. See [HR13] for details.

Proposition 4.3.1.7. Let $\mathbb{Q}$ be a commutative integral quantale. Then we can only have one of the two:

1. the product $\cdot$ is idempotent and $\cdot = \wedge$, $\mathbb{Q}$ is a frame, or

2. the product $\cdot$ in *not* idempotent.

Proof. Assume $\cdot$ idempotent. If we have $a \leq b$ and $a \leq c$, then $a = a \cdot a \leq b \cdot c$. Conversely we have $b \cdot c \leq b \cdot 1 = b$ and, similarly, $b \cdot c \leq c$, hence both $a \leq b$ and $a \leq c$, meaning that we have

$$a \leq b \cdot c \quad \text{iff} \quad a \leq b \text{ and } a \leq c.$$

□

4.3.2 Fuzzy sets

Definition 4.3.2.1 (Category of $\mathbb{M}$-fuzzy sets)**.** Call $\mathbf{Set}(\mathbb{M})$ the category having

- for objects $X = (X^0, |\ -\ |_X)$ where X^0 is a set and $|\ -\ |_X$ is a function $X^0 \to M$;
- morphisms $f : X \to Y$ are functions $f : X^0 \to Y^0$ such that

$$|\, x \,|_X \leq |\, f(x) \,|_Y$$

for all $x \in X^0$.

Example 4.3.2.2 (Fuzzy sets)**.**

1. (X, χ_A) where X is a set and χ_A is the characteristic function of a subset A of X and $\mathbb{M} = (\{0,1\}, \cdot, 1, \leq)$. Since every function $f : X \to Y$ satisfies that $f(A) \subseteq Y$, we always have $\chi_A \leq \chi_{f(A)}$.
2. For $\mathbb{M} = ([0,1], \cdot, 1, \leq)$, take (X, μ_A) where X is a set and μ_A is a membership function of a subset A of X that measures in which degree an element $x \in X$ is in A

$$\mu_A(x) = \begin{cases} 0, \text{ if } x \text{ is not a member} \\ \mu_A(x) \in (0,1), \text{ if } x \text{ is a fuzzy member} \\ 1, \text{ if } x \text{ is a full member.} \end{cases}$$

3. For $\mathbb{M} = ([0,\infty], +, 0, \geq)$, consider (X, δ_z) where X is a pseudo-metric space and $\delta_z(x) = \delta(x,z)$ is the distance between $x \in X$ and a fixed point $z \in X$. Then any non-expansive map $f : X \to Y$ is an example of a morphism of such $\mathbb{M}$-sets since $\delta_X(x,z) \geq \delta_Y(f(x), f(z))$.

Remark 4.3.2.3. There are many more different choices for morphisms - hence, categories - of fuzzy sets. See for example [Zad65, Wyl91, HWW14].

We describe a few properties of the category $\mathbf{Set}(\mathbb{M})$.

Proposition 4.3.2.4 (Properties of $\mathbf{Set}(\mathbb{M})$)**.**

1. $\mathbf{Set}(\mathbb{M})$ has a terminal object $\mathbf{1}_1 = (\{*\}, * \mapsto 1)$. Actually, the set $\{*\}$ supports different $\mathbb{M}$-set structures, for an α in $\mathbb{M}$ we write $\mathbf{1}_\alpha$ for the pair $(\{*\}, * \mapsto \alpha)$.
2. Given X and Y objects in $\mathbf{Set}(\mathbb{M})$, their product is $X \times Y = (X^0 \times Y^0, |\ -\ |_{X\times Y})$, where $|\, (x,y) \,|_{X\times Y} = |\, x \,|_X \wedge |\, y \,|_Y$.
3. For maps $f\colon X \to Z$ and $g\colon Y \to Z$ we can compute their pullback in $\mathbf{Set}(\mathbb{M})$ and show that it is the $\mathbb{M}$-set with underlying set the pullback in sets

$$(X \times_Z Y)^0 = \{(x,y) \,|\, f(x) = g(y)\}$$

and $|\, (x,y) \,|_{X\times_Z Y} = |\, x \,|_X \wedge |\, y \,|_Y$.

Remark 4.3.2.5. In $\mathbf{Set}(\mathbb{M})$:

1. monomorphisms are injective functions in $\mathbf{Set}(\mathbb{M})$,

2. epimorphisms are surjective functions in $\mathbf{Set}(\mathbb{M})$,

3. isomorphisms are bijective functions in $\mathbf{Set}(\mathbb{M})$; in particular, if $f : X \to Y$ is an isomorphism in $\mathbf{Set}(\mathbb{M})$, then $|\, x \,|_X = |\, f(x) \,|_Y$.

Proposition 4.3.2.6. There is a pair of adjoint functors as below.

$$\mathbf{Set}(\mathbb{M}) \underset{F}{\overset{U_1}{\underset{\longleftarrow}{\overset{\longrightarrow}{\top}}}} \mathbf{Set}$$

Proof. The functor F maps a set S to the pair $(S, const_1)$. Conversely, U_1 acts as

$$X = (X^0, |\, -\, |_X) \quad \mapsto \quad \{x \in X^0 \text{ s.t. } |\, x \,|_X = 1\}.$$

□

Construction 4.3.2.7. For each $\alpha \in M$ we can define a functor $U_\alpha : \mathbf{Set}(\mathbb{M}) \to \mathbf{Set}$

$$X = (X^0, |\, -\, |_X) \quad \mapsto \quad \{x \in X^0 \text{ s.t. } |\, x \,|_X \geq \alpha\}.$$

If there is an initial object 0, $U_0(X) = X^0$.

Proposition 4.3.2.8. If $\alpha \leq \beta$ there is a natural transformation $\lambda \colon U_\alpha \Rightarrow U_\beta$.

Proof. λ is simply defined as set inclusion at each component. □

Proposition 4.3.2.9. $\mathbf{Set}(\mathbb{M})$ inherits the monoidal structure from $\mathbb{M}$.

Proof. Note that $\mathbb{M}$ has two monoidal structures: the multiplication $\cdot$ and the meet $\wedge$. Since the meet is a particular (idempotent) case of the multiplication, we will prove the statement only for the most general operation.

Given X and Y objects in $\mathbf{Set}(\mathbb{M})$, their monoidal product is $X \otimes Y = (X^0 \times Y^0, |\, -\, |_{X\otimes Y})$, where $|\, (x,y) \,|_{X\otimes Y} = |\, x \,|_X \cdot |\, y \,|_Y$.

The tensor unit is given by $I = (I^0, |\, -\, |_I)$ where $|\, i \,|_I = 1_{\mathbb{M}}$.

There are associators because

$$\begin{aligned} |\, ((x,y),z) \,|_{(X\otimes Y)\otimes Z} &= |\, (x,y) \,|_{X\otimes Y} \cdot |\, z \,|_Z \\ &= |\, x \,|_X \cdot |\, y \,|_Y \cdot |\, z \,|_Z \\ &= |\, x \,|_X \cdot |\, (y,z) \,|_{Y\otimes Z} \\ &= |\, (x,(y,z)) \,|_{X\otimes(Y\otimes Z)} \end{aligned}$$

There are left unitors since

$$|\, (i,x) \,|_{I\otimes X} = |\, i \,|_I \cdot |\, x \,|_X = 1 \cdot |\, x \,|_X = |\, x \,|_X$$

The existence of right unitors follows analogously. The triangle identity holds since

$$\begin{aligned} |\, ((x,i),z) \,|_{(X\otimes I)\otimes Z} &= |\, (x,(i,z)) \,|_{X\otimes(I\otimes Z)} \\ &= |\, x \,|_X \cdot |\, (i,z) \,|_{I\otimes Z} \\ &= |\, x \,|_X \cdot 1 \cdot |\, z \,|_Z \\ &= |\, x \,|_X \cdot |\, z \,|_Z = |\, (x,z) \,|_{X\otimes Z} \end{aligned}$$

The pentagon identity follows by the same reasoning.

□

Remark 4.3.2.10. Defining $X \otimes Y = (X^0 \otimes Y^0, |-|_{X\times Y})$, where $|(x,y)|_{X\times Y} = |x|_X \cdot |y|_Y$ it is not monoidal cartesian. Recall that the cartesian product of two sets is equipped with projections π_1 and π_2 such that for any pair of morphism $f : Z \to X$ and $g : Z \to Y$ there is a unique morphism $h : Z \to X \times Y$ such that $\pi_1 \circ h = f$ and $\pi_2 \circ h = g$. Moreover, know that $h(z) = (f(z), g(z))$, for any $z \in Z$. However, such h may not be a morphism in $\mathbf{Set}(\mathbb{M})$. Take $\mathbb{M} = ([0,1], \cdot, 1, \leq)$, then $|h(z)|_{X\times Y} = |(f(z), g(z))|_{X\times Y} = |f(z)|_X \cdot |g(z)|_Y \geq |z|_Z \cdot |z|_Z \leq |z|_Z$.

Of course, when $\cdot = \wedge$, then $|(f(z), g(z))|_{X\times Y} = |f(z)|_X \wedge |g(z)|_Y \geq |z|_Z \wedge |z|_Z = |z|_Z$. This arguments works because $\wedge$ is an idempotent operation.

It is known that $\mathbf{Set}(\mathbb{M})$ is cartesian closed category if and only if $\mathbb{M}$ is a complete Heyting Algebra, see [Wyl91, Proposition 71.4]. Using an analogous argument we now prove that $\mathbf{Set}(\mathbb{M})$ is monoidal closed if and only if $\mathbb{M}$ is a quantale.

Proposition 4.3.2.11. Let $\mathbb{M}$ be a integral commutative ordered monoid, then $\mathbf{Set}(\mathbb{M})$ is monoidal closed if and only if $\mathbb{M}$ is a quantale.

Proof. Given $\mathbb{M}$-fuzzy sets Y and Z, define a $\mathbb{M}$-fuzzy set $Z^Y = (\{h : Y \to Z\}, |-|_{Z^Y})$ where $|h|_{Z^Y} = \bigwedge_{y\in Y} (|y|_Y \to |h(y)|_Z)$.

It is clear that $\mathbf{Set}(X, Z^Y) \cong \mathbf{Set}(X \times Y, Z)$, since $\mathbf{Set}$ with the cartesian product is closed, but we want to show that $\mathbf{Set}(\mathbb{M})(X, Z^Y) \cong \mathbf{Set}(\mathbb{M})(X \times Y, Z)$. A morphism in $\mathbf{Set}(\mathbb{M})(X \times Y, Z)$ is a function $f : X \times Y \to Z$ satisfying

$$|x|_X \cdot |y|_Y \leq |f(x,y)|_Z$$

for all $(x, y) \in X \times Y$.

Since $\mathbb{M}$ is a quantale, this is equivalent to

$$|x|_X \leq |y|_Y \to |f(x,y)|_Z,$$

which happens if and only if

$$|x|_X \leq |y|_Y \to |\tilde{f}(x)(y)|_Z,$$

where $\tilde{f}$ is the exponentially adjoint to f in $\mathbf{Set}$.

Observe that $|\tilde{f}(x)|_{Y^Z} = \bigwedge_{y\in Y} (|y|_Y \to |\tilde{f}(x)(y)|_Z)$. Thus, by definition of infimum, we have a last equivalence:

$$|x|_X \leq |\tilde{f}(x)|_{Y^Z}$$

Therefore, $\tilde{f} : X \to Z^Y$ is the morphism in $\mathbf{Set}(\mathbb{M})$ that testifies the desired isomorphism.

Conversely, suppose that $\mathbf{Set}(\mathbb{M})$ is monoidal closed. Consider the $\mathbb{M}$-fuzzy set $(\{*\}, m)$ where $\{*\}$ denotes the singleton set and m denotes the constant function $|*|_{\{*\}} = m$, for each $m \in \mathbb{M}$. We have $m_i \cdot m_j \leq m_i$ for all $m_i, m_j \in \mathbb{M}$. So we have morphisms $id_* : (\{*\}, m_i \cdot m_j) \to (\{*\}, m_i)$, for all $i \in I$. A diagram of such morphisms has a colimit cone of morphisms $id_{\{*\}} : (\{*\}, m_i) \to (\{*\}, \bigvee_{i\in I} m_i)$.

$\mathbf{Set}(\mathbb{M})$ is monoidal closed, so the tensor product is a left adjoint functor and thus it has to preserve colimits. Take $\mathbb{M}$-fuzzy set $(\{*\}, n)$. Observe that $(\{*\}, n) \otimes (-)$ preserves the above colimit iff $n \cdot (\bigvee_{i\in I} m_i) = \bigvee_{i\in I} (n \cdot m_i)$. Therefore, $\mathbb{M}$ is a quantale. □

We will see that we will need for $\mathbf{Set}(\mathbb{M})$ to be monoidal closed, the reason for this is contained in the discussion in 4.1.5, hence we always assume that we are in the hypotheses of Proposition 4.3.2.11.

4.3.3 Enriching over fuzzy sets

Let $\mathbb{M}$ be an integral commutative monoid which is also a quantale, and consider its monoidal structure based on "$\cdot$". We break down Definition 4.1.2.1 and see that a $\mathbf{Set}(\mathbb{M})$-category $\mathcal{C}$ is the data of

- a collection of objects,
- for each pair of objects X, Y an $\mathcal{C}(X,Y)$ in $\mathbf{Set}(\mathbb{M})$, such that
- for each X there is an element $\mathrm{id}_X \in \mathcal{C}(X,X)^0$ such that

$$|\,\mathrm{id}_X\,|_{\mathcal{C}(X,X)} = 1,$$

- for each X, Y, Z and $f \in \mathcal{C}(X,Y)^0$, $g \in \mathcal{C}(Y,Z)^0$ there is an element $g \circ f \in \mathcal{C}(X,Z)^0$ such that

$$|\,f\,|_{\mathcal{C}(X,Y)} \cdot |\,g\,|_{\mathcal{C}(Y,Z)} \leq |\,g \circ f\,|_{\mathcal{C}(X,Z)},$$

- and all the appropriate diagrams commute.

In order to make the text more readable, we might omit writing the $\mathbb{M}$-set in the valuation. We call $\mathbf{Set}(\mathbb{M})$-**Cat** the category of $\mathbf{Set}(\mathbb{M})$-categories and $\mathbf{Set}(\mathbb{M})$-functors.

4.4 Substitution in the enriched setting

As we have extensively seen in Chapter 3, and will see again in Section 4.5.1 and Section 4.5.2, the categorical tool describing substitution is that of *pullback* in a category. But now that we are moving in a fuzzy setting, it might be the case that the regular definition of pullback is not the one we are really interested in.

Informally, the universal property of the pullback in a regular set-based category states that, provided a cospan $A \to C \leftarrow B$, there are three maps from the same object, one into A, one into B, and one into C such that the triangles they form commute, and they are universal with respect to this property. Now that we are in a setting where maps come equipped with certain values, what do we do with them? How do we formally ask that such maps into A, B, C have precise values? Can we? What universal property do they need to have? In order to answer these questions precisely, we use the notion of *weighted pullback*, specialize it to the enriched setting, and - considering the logic we want to interpret - describe what we need. Weighted pullbacks are a specialization of weighted limits, so we begin with them.

Recall from Definition 4.1.6.1 that we can express limits as ends, so with 4.1 in mind we want to extend it all to the enriched setting.

Definition 4.4.0.1. (Weighted limits) Let $\mathcal{V}$ be a closed symmetric monoidal category and consider two $\mathcal{V}$-enriched categories $\mathcal{C}$ and $\mathcal{D}$. Then the weighted

limit of the diagram $D : \mathcal{D} \to \mathcal{C}$ with weights $W : \mathcal{D} \to \mathcal{V}$ is the object $\lim^W D$ in $\mathcal{C}$ given by the following universal property:

$$\mathcal{C}(H, \overset{W}{\lim} D) \cong \int_{\mathcal{D}} [W-, \mathcal{C}(H, D-)]$$

for all objects H in $\mathcal{C}$. Here $[\cdot, \cdot]$ represents the internal hom in $\mathcal{V}$.

The weight functor can be thought of as prescribing the weight of an arrow to each of the objects in the diagram. We are interested in the case where $\mathcal{V}$ is the category of fuzzy sets, and D is a cospan diagram, but in order to compute the functor $\mathcal{C}(X, D-)$ we first need to make a little technical observation.

Remark 4.4.0.2 (Postcomposition in an enriched category)**.** Computing the wedge described in Definition 4.4.0.1 requires us to be very careful: in the classical, set-based context, for each map $f\colon A \to B$ in a category $\mathcal{C}$ one can define for each object H in $\mathcal{C}$ a *function*

$$\mathcal{C}(H, A) \to \mathcal{C}(H, B), \quad g \mapsto f \circ g \tag{4.2}$$

between the hom-sets. This in turn allows us to define for any X in $\mathcal{C}$ the map

$$[X, \mathcal{C}(H, A)] \to [X, \mathcal{C}(H, B)],$$

which is the one that is required by the definition of the wedge in 4.4.0.1. This is not always the case in a $\mathcal{V}$-enriched category, because the map described in 4.2, while being a morphism in **Set**, might not be a morphism in $\mathcal{V}$ - in fact $\mathcal{V}$ might not even be concrete. What we can do is unfold the behavior underlying 4.2 using the monoidal structure, because that is the structure producing composition: in fact, what one can do is compute the following

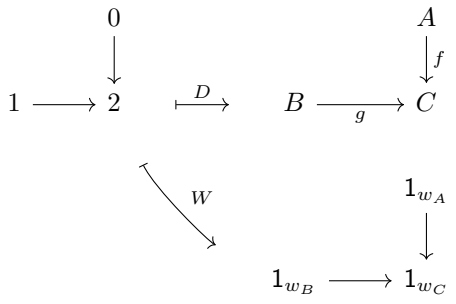
$$\mathcal{C}(H, A) \otimes I \to \mathcal{C}(H, A) \otimes \mathcal{C}(A, B) \to \mathcal{C}(H, B),$$

in which the first maps picks out f in $\mathcal{C}(A, B)$ and the second computes composition. We only consider the case we are interested in, meaning when $\mathcal{V} = \mathbf{Set}(\mathbb{M})$. In order for this to work, though, we cannot have $I = 1_1$, meaning the singleton with constant value 1 - or we would only be allowed to catch those fs that have value 1 - but $1_{|f|}$, so that

$$|g|_{\mathcal{C}(H,A)} \cdot |*|_{1_{|f|}} \leq |g|_{\mathcal{C}(H,A)} \cdot |f|_{\mathcal{C}(A,B)} \leq |f \circ g|_{\mathcal{C}(H,B)}.$$

This will change the nodes in the wedge we compute for the weighted pullback.

Construction 4.4.0.3 (Weighted pullbacks in categories enriched in fuzzy sets)**.** Let $\mathcal{C}$ a $\mathbf{Set}(\mathbb{M})$-enriched category. We are only concerned with studying pullbacks, so we pick $\mathcal{D}$ to be the cospan $0 \to 2 \leftarrow 1$.

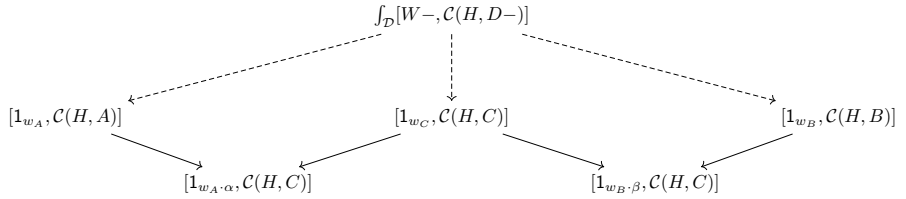


Our aim is to mimic pullbacks in an enriched category, so while D does what it usually does in **Set**-enriched categories, we pick the weights to be singletons with constant value possibly less than 1:

- singletons because to each vertex A, B, C we want to get *one* arrow,
- with value possibly less than 1 because if it was only 1 we would recover only maps with value 1 and thus lose all the *fuzzy-ness* of substitution.

Notice that our choice requires $w_A \leq w_C$ and $w_B \leq w_C$ because the weights diagrams need to live in $\mathbf{Set}(\mathbb{M})$.

Now say $|\, f \,| = \alpha$ and $|\, g \,| = \beta$, then by Definition 4.4.0.1 for any H in $\mathcal{C}$ we have that $\mathcal{C}(H, \lim^W D)$ is isomorphic to the limit below.



(4.3)

Let us first of all comment the mixed term $[1_{w_A \cdot \alpha}, \mathcal{C}(H, C)]$: this is precisely an instance of Remark 4.4.0.2 because $1_{w_A \cdot \alpha} \cong 1_{w_A} \otimes 1_\alpha$. An $\mathbb{M}$-set of the form

$$[1_w, \mathcal{C}(H, X)]$$

is the set of underlying functions $x \colon (1_w)^0 = \{*\} \to \mathcal{C}(H, X)^0$ with valuation

$$|\, x \,|_{[1_w, \mathcal{C}(H,X)]} = \bigwedge_{y \in (1_w)^0} |\, y \,|_{1_w} \to |\, x(y) \,|_{\mathcal{C}(H,X)} = w \to |\, x(*) \,|_{\mathcal{C}(H,X)},$$

so that it basically picks up $\mathbb{M}$-set morphisms $x \colon H \to X$ and it assigns them a new value depending on both their original one and on w.

Let us show that both

1. $[1_{w_A}, \mathcal{C}(H, A)] \to [1_{w_A \cdot \alpha}, \mathcal{C}(H, C)]$, and
2. $[1_{w_C}, \mathcal{C}(H, C)] \to [1_{w_A \cdot \alpha}, \mathcal{C}(H, C)]$

are well defined. Clearly 2 acts as a sort of inclusion, because provided a map c in $\mathcal{C}(H, C)_0$ we have

$$w_C \to |\, c \,| \leq w_A \cdot \alpha \to |\, c \,|$$

if $w_C \geq w_A \cdot \alpha$, but this is true because $w_C \geq w_A \geq w_A \cdot \alpha$. As for 1, we want to perform postcomposition: to a map $a \in \mathcal{C}(H, A)^0$ we want to assign $f \circ a$, which produces a morphism in $\mathbf{Set}(\mathbb{M})$ if

$$w_A \to |\, a \,| \leq w_A \cdot \alpha \to |\, f \circ a \,|$$

but by definition of composition in $\mathcal{C}$ we have

$$|\, f \,| \cdot |\, a \,| \leq |\, f \circ a \,| \quad \text{or, equivalently,} \quad |\, a \,| \leq |\, f \,| \to |\, f \circ a \,| = \alpha \to |\, f \circ a \,|$$

hence

$$w_A \to |\, a \,| \leq w_A \to (\alpha \to |\, f \circ a \,|) = w_A \cdot \alpha \to |\, f \circ a \,|$$

by currying, yielding the desired result.

Now, one can show that the desired limit can be computed by (two) iterated pullbacks in $\mathbf{Set}(\mathbb{M})$, which we know how to compute from Proposition 4.3.2.4, 3. It follows that $\int_{\mathcal{D}}[W-, \mathcal{C}(H, D-)]$ has underlying set

$$\{(a, c, b) \mid f \circ a = c = g \circ b\} \subseteq \mathcal{C}(H, A)^0 \times \mathcal{C}(H, C)^0 \times \mathcal{C}(H, B)^0$$

and valuation $|\,(a, c, b)\,| = w_A \to |\,a\,| \wedge w_C \to |\,c\,| \wedge w_B \to |\,b\,|$.

In particular, we can use this to describe the universal maps by checking what happens to the image of $\mathrm{id}_{\lim^W D}$ through

$$\mathcal{C}(\overset{W}{\lim} D, \overset{W}{\lim} D) \cong \int_{\mathcal{D}} [W-, \mathcal{C}(\overset{W}{\lim} D, D-)]\,.$$

Remember that the identity has value 1, hence maps a, c, b from $\lim^W D$ to, respectively, A, C, B must satisfy

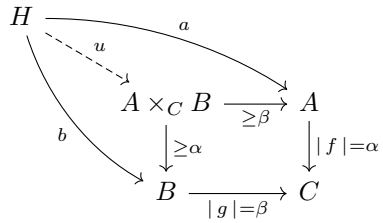
$$1 \leq |\,(a, c, b)\,| = w_A \to |\,a\,| \wedge w_C \to |\,c\,| \wedge w_B \to |\,b\,|$$

so that $w_A \to |\,a\,| = 1$, hence $w_A \leq |\,a\,|$, and so on for each term.

Following the discussion above, we make a first choice for the weights: we chose w_A, w_B to be as informative as possible, and $w_C = 1$ so that the square commutes with certainty 1.

Definition 4.4.0.4 (Having weighted pullbacks)**.** A $\mathbf{Set}(\mathbb{M})$-category *has weighted pullbacks* if for each cospan (f, g) it has $(1_{|\,f\,|}, 1_1, 1_{|\,g\,|})$-weighted pullbacks.

Remark 4.4.0.5. If $\mathcal{C}$ has weighted pullbacks, then for each f, g



and a, b such that $f \circ a = g \circ b$ there is a unique $u\colon H \to A \times_C B$ with value $|\,u\,| = \beta \to |\,a\,| \wedge \alpha \to |\,b\,|$.

Finally, we show that weighted pullbacks compose appropriately.

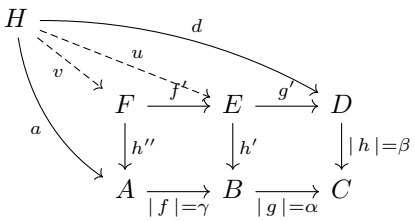
Lemma 4.4.0.6 (Pullback pasting lemma)**.** Consider a commutative diagram in $\mathbf{Set}(\mathbb{M})$-enriched category $\mathcal{C}$ as follows

$$\begin{array}{ccccc} F & \xrightarrow{f'} & E & \xrightarrow{g'} & D \\ \downarrow{\scriptstyle h''} & & \downarrow{\scriptstyle h'} & & \downarrow{\scriptstyle |\,h\,|=\beta} \\ A & \xrightarrow[|\,f\,|=\gamma]{} & B & \xrightarrow[|\,g\,|=\alpha]{} & C \end{array}$$

such that $|\,h'\,| \leq |\,h''\,|$. If the right square is a weighted pullback, then:

- the outer rectangle is a weighted pullback if the left square is a weighted pullback;
- the left square is a weighted pullback if the outer rectangle is a weighted pullback.

Proof. Suppose the left square is a pullback. Let H in $\mathcal{C}$ with maps a and d such that $g \circ f \circ a = h \circ d$. Then we have $f \circ a$ and d insisting on the cospan (g, h). Since the right square is a pullback we have a unique $u \colon H \to E$

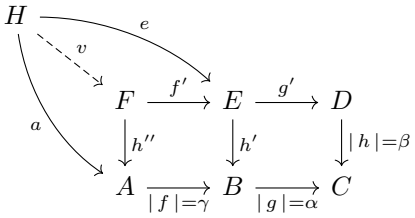


which in turn induces a unique $v \colon H \to F$ with value

$$\begin{aligned}
|\, v \,| &= \gamma \to |\, u \,| \wedge |\, h' \,| \to |\, a \,| \\
&= \gamma \to (\alpha \to |\, d \,| \wedge \beta \to |\, a \,|) \wedge |\, h' \,| \to |\, a \,| \\
&\geq \gamma \to (\alpha \to |\, d \,|) \wedge \gamma \to (\beta \to |\, a \,|) \wedge |\, h' \,| \to |\, a \,| \\
&= \gamma \cdot \alpha \to |\, d \,| \wedge \gamma \cdot \beta \to |\, a \,| \wedge |\, h' \,| \to |\, a \,| \\
&\geq |\, gf \,| \to |\, d \,| \wedge \beta \to |\, a \,|
\end{aligned}$$

because $|\, gf \,| \geq \gamma \cdot \alpha$, $|\, h' \,| = \beta \geq \gamma \cdot \beta$.

Conversely, we wish to show that if the outer rectangle is weighted pullback, then the left square is, too. Consider a pair a, e as below,



then by the universal propery of the rectangle we have a unique $v \colon H \to F$ with value

$$|\, v \,| = |\, gf \,| \to |\, g'e \,| \wedge \beta \to |\, a \,|,$$

which we wish to show to be an appropriate universal map for the left square: commutativity is trivial, so we only need to show that it has the appropriate value. Given that the right square is a pullback, we trivially have

$$|\, e \,| = \alpha \to |\, g'e \,| \wedge \beta \to |\, h'e \,|,$$

therefore

$$\begin{aligned}
\gamma \to |\, e \,| \wedge \beta \to |\, a \,| &= \gamma \to (\alpha \to |\, g'e \,| \wedge \beta \to |\, h'e \,|) \wedge \beta \to |\, a \,| \\
&\geq \gamma \to (\alpha \to |\, g'e \,|) \wedge \gamma \to (\beta \to |\, h'e \,|) \wedge \beta \to |\, a \,| \\
&= \gamma \cdot \alpha \to |\, g'e \,| \wedge \gamma \cdot \beta \to |\, h'e \,| \wedge \beta \to |\, a \,| \\
&\geq |\, fg \,| \to |\, g'e \,| \wedge \gamma \cdot \beta \to |\, h'e \,| \wedge \beta \to |\, a \,| \\
&= |\, fg \,| \to |\, g'e \,| \wedge \beta \to |\, a \,|
\end{aligned}$$

because $\beta \leq |\, h'' \,|$ by hypothesis, so that

$$\gamma \cdot \beta \to |\, fa \,| \geq \gamma \cdot |\, h'' \,| \to |\, fa \,| \geq |\, fh'' \,| \to |\, fa \,| = |\, h'f' \,| \to |\, h'e \,|.$$

□

4.5 From syntax to semantics and back

We can now finally describe the necessary categorical structure needed in order to interpret fuzzy dependent types. We refer to Section 3.1 for a review of all the elements needed, and to Section 3.3.6 for a comparison with alternative models.

4.5.1 Reading type theory into a category

Definition 4.5.1.1 (Display-map category [Tay99, HP87])**.** A *display-map category* is a pair $(\mathcal{C}, \mathcal{D})$ with $\mathcal{C}$ a category and $\mathcal{D} = \{p_A : \Gamma.A \to \Gamma\}$ a class of morphisms in $\mathcal{C}$ called *displays* or *projections* such that:

1. for each $p_A : \Gamma.A \to \Gamma$ in $\mathcal{D}$ and $s : \Delta \to \Gamma$ in $\mathcal{C}$, there exists a choice of a pullback of p_A along s and it is again in $\mathcal{D}$,

$$\begin{array}{ccc} \Delta.A[s] & \xrightarrow{\overline{s}} & \Gamma.A \\ {\scriptstyle p_{A[s]}}\downarrow & & \downarrow{\scriptstyle p_A} \\ \Delta & \xrightarrow[s]{} & \Gamma \end{array}$$

2. $\mathcal{D}$ is closed under pre and post-composition with isomorphisms;

3. $\mathcal{C}$ has a terminal object $\mathbb{1}$.

As in Chapter 3, we have used a suggestive notation to describe all of the elements introduced in Definition 4.5.1.1 - and we will readily explain it - but the reader should not forget that it is nothing more than that: *notation*. When it comes to it, a display-map category is a category with essentially a choice of a class of maps such that it is closed under pullback along maps in the category.

Now to the interpretation. If p_A is a projection, we write Γ and $\Gamma.A$ for its codomain and domain, respectively, and this because we think of projections as types, and of objects of $\mathcal{C}$ as contexts: given a type p_A (or, simply, A), we can always recover both its context and the one obtained extending it with A itself.

Pullback is meant to represent substitution of a type A along any morphism of contexts s, and we denote the resulting type with $A[s]$. Asking for a choice of pullback allows for coherence issues to be dealt with, while $\mathbb{1}$ is the empty context. In this setting, terms of a given type A are represented by sections of p_A. See Theorem 3.3.7.1 for why this is a reasonable assumption.

4.5.2 Reading fuzzy type theory into a category

Definition 4.5.2.1 (Fuzzy display-map category)**.** A *fuzzy display-map category* is a pair $(\mathcal{C}, \mathcal{D})$ with $\mathcal{C}$ a $\mathbf{Set}(\mathbb{M})$-category and $\mathcal{D} = \{p_A : \Gamma.A \to \Gamma\}$ a class of morphisms in $\mathcal{C}$ called *fuzzy displays* or *fuzzy projections* such that:

1. for each $p_A : \Gamma.A \to \Gamma$ in $\mathcal{D}$ and $s : \Delta \to \Gamma$ in $\mathcal{C}$, there exists a choice of a weighted pullback (in the sense of Definition 4.4.0.4) of p_A along s and its underlying map is again in $\mathcal{D}$,

$$\begin{array}{ccc} \Delta.A[s] & \xrightarrow{\overline{s}} & \Gamma.A \\ {\scriptstyle p_{A[s]}}\downarrow & & \downarrow{\scriptstyle p_A} \\ \Delta & \xrightarrow[s]{} & \Gamma \end{array}$$

2. $\mathcal{D}$ is closed under pre and post-composition with isomorphisms;

3. $\mathcal{C}$ has a terminal object $\mathbb{1}$;

4. for all A, $|\, p_A \,|_{\mathcal{C}(\Gamma.A,\Gamma)} = 1$.

Again, we think of objects in $\mathcal{C}$ as contexts, projections as types, (weighted) pullback as substitution, $\mathbb{1}$ as the empty context. Notice that we additionally ask for projections to always have the maximum possible value, so that types themselves are not fuzzy.

Remark 4.5.2.2. Our types are not fuzzy. One could easily change that by removing 4. See Section 4.7.3 for further discussion on this.

We now want to look at sections of projections, and say that they describe terms in our new setting. We need to be able to *fully* exploit our enrichment, and have sections come equipped with a given $\mathbb{M}$-value, which we interpret to be the desired "extent" discussed in Section 4.2. This is the motivation behind the following definition. Recall 4.3.2.7 for the subtleties in the notation below.

Definition 4.5.2.3. (α-sections) Let p_A a projection. An *α-section of p_A* is a morphism s in $\mathcal{C}(\Gamma, \Gamma.A)$ such that

- $p_A \circ s = \mathrm{id}$, and
- $|\, s \,| \geq \alpha$.

$$\Gamma \xrightarrow[s]{} \Gamma.A \xrightarrow[p_A]{} \Gamma$$

This means that we can finally describe our interpretation to its full extent: we understand objects of $\mathcal{C}$ as contexts, projections as types, terms (of confidence *at least* α) as (α-)sections. If s is a term of type A in context Γ with confidence α we write

$$\Gamma \vdash s :_\alpha A.$$

Classical judgements involving types are written in the usual manner. Notice that variables which are written in contexts only do not possess a confidence of their own.

Remark 4.5.2.4 (Confidence is preserved)**.** For all $\beta \leq \alpha$, the following holds.

$$\frac{\Gamma \vdash s :_\alpha A}{\Gamma \vdash s :_\beta A}$$

While in Section 4.2 we have discussed the classical version of the *propositions-as-types* interpretation, now that we have both fuzziness and our technical machinery, it is worth spending a few words on how they shape such an interpretation.

Perhaps it is most useful considering it in its Curry-Howard form: this is where we were at.

proofs	programs (terms)	motivations
formula	specification of progr. (type)	belief

Now, on top of this, we have a categorical model for executions and programs, so that our correspondence looks like the following.

proofs	programs (terms)	motivations	sections
formula	specification of progr. (type)	belief	projection

4.6 Rules for fuzzy type theory

Now that we have all the basic elements of our theory, we need to express *what* we can do with them. This section of the chapter flirts with moving swiftly between the logical side of things and the categorical one, so we better spend some time explaining each, and how the two are related.

Type theory as described in [Hof97] is a logical system in which one gives an account of judgements pertaining terms and types in context. Out of these basic blocks, one builds up a system imposing rules that dictate the behaviour of all of these different pieces together: for example, out of a type in context, one can produce a new context "pasting" the type to its context. Each rule has a label that is meant to be descriptive of its meaning, for example the rule we just described is usually denoted (C-Ext) for context extension.

On its categorical side, the rules represent operations one can perform in the category, given the axioms one starts from, so that they are usually built in the original definition of the categorical structure one chooses. In the case of context extension, for example, the action of computing the context obtained by extending a context with a type amounts to computing the domain of the corresponding projection. If this sounds tautological at all, it is because the categorical structure one considers is meant to precisely mimic the logic. When our effort is successful, we say that the structure *verifies* the rules.

Remark 4.6.0.1 (Our strategy)**.** Here, we sort of work the other way round: we have defined a structure that is directly built on top of the "regular" definition, now we repeat the same constructions, and *read* into the category the correct formulation of the rules. Nonetheless, Theorem 4.6.0.2 and Theorem 4.6.0.4 are expressed in the traditional form. Each is followed by a detailed explanation of the rules from the point of view of the logic.

Theorem 4.6.0.2. (Soundness I) Let $(\mathcal{C}, \mathcal{D})$ a fuzzy display-map category. Then it verifies the following rules for fuzzy type theory.

$$\frac{}{\vdash \diamond \ \mathtt{ctx}}(\text{C-Emp}) \qquad \frac{\Gamma \vdash A \ \mathtt{Type}}{\vdash \Gamma, x : A \ \mathtt{ctx}}(\text{C-Ext})$$

$$\frac{\vdash \Gamma, x : A, \Delta \ \mathtt{ctx}}{\Gamma, x : A, \Delta \vdash x :_1 A}(\text{Var})$$

Proof. We interpret contexts to objects in $\mathcal{C}$, types to fuzzy projections. To compute the context of a type one needs to read the codomain of the associated projection. Terms with confidence α are α-sections.

The terminal object in $\mathcal{C}$ provides the empty context. Given a type A in context Γ, hence a projection

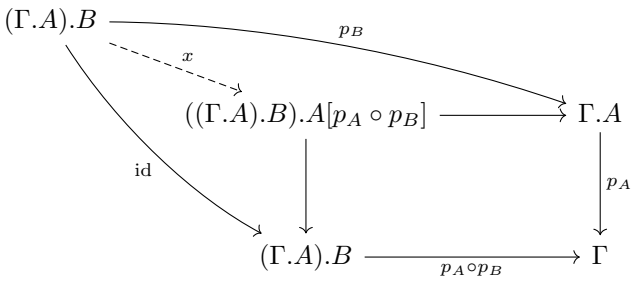
$$p_A : \bullet \to \Gamma$$

we define the extended context of Γ with A as $\mathrm{dom}(p_A)$. (In fact we have been writing this as "$\Gamma.A$" this entire time.)

We only prove the variable rule in the case that $\Delta = y : B$ a single type, as it will be clear that the general case can be proved in an entirely similar way. Since $\Gamma, x : A, y : B$ is a context we have that:

- Γ is a context;
- A is a type in context Γ, hence there is a projection $p_A : \Gamma.A \to \Gamma$;
- B is a type in context $\Gamma, x : A$, hence there is a projection $p_B : (\Gamma.A).B \to \Gamma.A$.

Therefore we can consider the following weighted pullback.



By hypothesis 1 on fuzzy display-map categories, there is a functorial choice of a type $A[p_A \circ p_B]$ making the square a weighted pullback, with all weights 1. Moreover, there is a unique x' and its value is $(1 \to 1) \wedge (1 \to 1) = 1$. This concludes our proof, see Remark 4.6.0.3 for a discussion on why is that so. □

Let us now more carefully describe the meaning of each of the rules above.

(C-Emp) There is at least one context, that is the empty one.

(C-Ext) When one has a type in context, it is always possible to extend the initial context with the type. In other logical systems, this would be called an "assumption rule".

(Var) The Variable rule tells us that whenever we assume an $x : A$, we can always provide an x in A with confidence 1. It describes a tautology, but it also provides us with a way of extracting out of a given type terms with confidence 1, namely the trivial ones.

Remark 4.6.0.3 (Is $A[p_A \circ p_B]$ equal to A?)**.** [Hof97], which we follow for classical rules of type theory, and many others write the variable rule as in Theorem 4.6.0.2. The main problem with doing so arises in our proof above and is somewhat philosophical: the same type *cannot* have two different contexts, as it seems to be the case in (Var). In fact, it seems that A is supposed to be both in context Γ *and* in context $\Gamma, x : A, y : B$. Same goes for x. What actually happens is that we substitute A with its correspondent $A[p_A \circ p_B]$ in the extended context, without adding any essentially new information *but* the new context, and, similarly, if we can pick out a variable x of type A, there is a way for us to pick out a term in this new type $A[p_A \circ p_B]$, that is x', which is that adding precisely no information. This is also the cause of the confusion of x having no confidence itself and suddenly having it be 1.

Mind that this is in no part due to our working with fuzzy terms, and is in fact a feature of much of the literature on types and their categorical models, we just point it out because we need to be particularly careful when dealing with this extra level of complexity. Still, so that we are not *too* pedantic, we try to stick to the classical notation as much as possible, and write A for A[composition of p's].

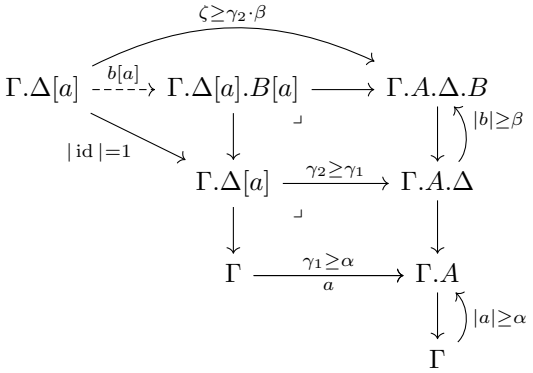
Notice that we had a similar discussion for natural deduction calculus in Section 2.4.

Theorem 4.6.0.4. (Soundness II) Let $(\mathcal{C}, \mathcal{D})$ a fuzzy display-map category. Then it verifies the following rules for fuzzy type theory.

$$\frac{\Gamma, \Delta \vdash B\,\mathtt{Type} \quad \Gamma \vdash A\,\mathtt{Type}}{\Gamma, x : A, \Delta \vdash B\,\mathtt{Type}}(\mathrm{Weak}_{ty}) \qquad \frac{\Gamma, \Delta \vdash b :_\beta B \quad \Gamma \vdash A\,\mathtt{Type}}{\Gamma, x : A, \Delta \vdash b :_\beta B}(\mathrm{Weak}_{tm})$$

$$\frac{\Gamma, x : A, \Delta \vdash B\,\mathtt{Type} \quad \Gamma \vdash a :_\alpha A}{\Gamma, \Delta[a/x] \vdash B[a/x]\,\mathtt{Type}}(\mathrm{Subst}_{ty}) \qquad \frac{\Gamma, x : A, \Delta \vdash b :_\beta B \quad \Gamma \vdash a :_\alpha A}{\Gamma, \Delta[a/x] \vdash b[a/x] :_\beta B[a/x]}(\mathrm{Subst}_{tm})$$

Proof. We begin with Substitution: unwinding it for types and terms relies heavily on (1) in Definition 4.5.2.1. The type B in context Γ, A, Δ amounts to a display map $p_B : \Gamma.A.\Delta.B \to \Gamma.A.\Delta$ built up iteratively, while a is a section of p_A. We compute the iterated pullback below, and obtain first $p_{\Delta[a]}$, then $p_{B[a]}$. One can see that it all type-checks. As for terms we additionally have a b section of the aftermentioned p_B. The universal property of weighted pullbacks guarantees that we have a section of $p_B[a]$. We call this $b[a]$.

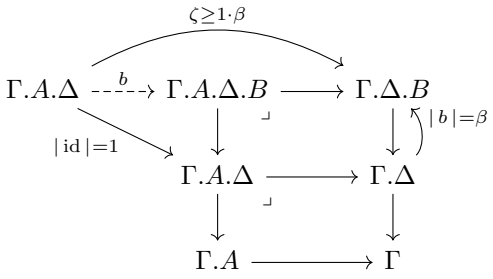


We can easily estimate a lower bound for the confidence of $b[a]$:

$$|\, b[a] \,| = \gamma_2 \to \zeta \wedge 1 \to 1 \geq \gamma_2 \to (\gamma_2 \cdot \beta) \geq \beta,$$

which concludes our proof.

Weakening works in a similar way, but in some sense in the opposite direction. We leave the diagram below to be interpreted by the reader.



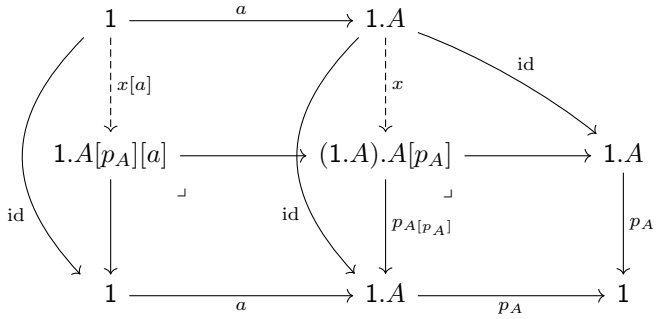
Recall the discussion in Remark 4.6.0.3 for how we might denote types and their correspondent in an extended context. □

Again, we describe the meaning of the rules above.

(Weak) The rules describing Weakening tell us that if one can have a type or a term in a given context (b, B in context Γ, Δ), it is always possible to express it in a context having more assumptions (Γ, A, Δ), provided that that is coherent (A is in context Γ). Moreover, the confidence of terms is preserved.

(Subst) The rules describing Substitution explain how substituting a given term a in a type or term (B or b) in a suitable context (depending on the type of a) provides a new type or term. Once more, confindence is preserved, but one should remember that, though it is not apparent, the α-confidence of the term one is substituting still weighs on the result of the rule. See the following Example 4.6.0.5 for further discussion on it.

Example 4.6.0.5 (Confidence *on* types)**.** One might be tempted to compose the two constructions in Theorem 4.6.0.4. We do it in a simple case: consider a term $\diamond \vdash a :_\alpha A$ and compute the following diagram.



As it was discussed in Theorem 4.6.0.2, $|\, x \,| = 1$, and by hypothesis $|\, a \,| = \alpha$ therefore $|\, x[a] \,| = (\alpha \to \alpha) \wedge (1 \to 1) = 1$ itself. We have proved the following rule,

$$\frac{\diamond \vdash a :_\alpha A}{\diamond \vdash x[a] :_1 A[a]} \tag{4.4}$$

which looks strange at first, because it seems to guarantee that out of any term of any confidence we can build up a term of confidence 1. The key to understanding this apparent incongruence, again, stems from the substitution axiom (1), because the type A and the type $A[a]$ (or, more precisely, $A[p_A][a]$, since $A[a]$ itself is not well-typed) are *not* the same. In fact, $A[p_A][a]$ is obtained first by extending the context of A with a "copy" of A itself, meaning $(x : A)$, and then substituting a for x. We can interpret it as follows:

if I can prove A by a with confidence α,
then I can prove "I can prove A by a with confidence α" with confidence 1.

Such an example should warn us against foregoing writing substitutions explicitly in all cases but along projections.

4.7 On definitional equality

We have yet to discuss definitional equality, and depending on your perspective this is the point where categories for type theories either become incredibly convincing, or fail. Per Martin-Löf writes:

> Definitional equality is intensional equality, or equality of meaning (synonymy). [...] Definitional equality $\equiv$ is a relation between linguistic expressions; it should not be confused with equality between objects (sets, elements of a set etc.), [...] [it] is the equivalence relation generated by abbreviatory definitions, changes of bound variables and the principle of substituting equals for equals. Therefore it is decidable, but not in the sense that $a \equiv b \vee \neg(a \equiv b)$ holds, simply because $a \equiv b$ is not a proposition in the sense of the present theory. [MS84, Definitional equality]

From this we gather that:

1. it should be an equivalence relation, hence
2. for each pair of things in the same class, there should be (at least) one process turning one into the other;
3. it should be part of the theory, and, as such,
4. it should be decidable.

In fact, when one builds a category out of a given type theory, to avoid any confusion one turns to equivalence classes.

Example 4.7.0.1 (Syntactic model)**.** Recall from Example 3.3.6.5 that we can build $\mathcal{C}$ the category having for objects (α-)equivalence classes of contexts, denoted $[\Gamma] = x_1 : A_1, \ldots, x_n : A_n$, $[\Theta] = y_1 : B_1, \ldots, y_m : B_m$ and so on, and such that a morphism

$$t \colon [\Theta] \to [\Gamma] \quad \text{is} \quad \text{(equivalence classes of) terms } [t_1], \ldots, [t_n]$$

such that $\Theta \vdash t_i : A_i[t_1/y_1, \ldots, t_{i-1}/y_{i-1}]$ for each $i = 1 : n$. One can define projections

$$p_A \colon [\Gamma, x : A] \to [\Gamma]$$

and these can be collected into a display-map category.

This has the effect of collapsing definitional equality to identity in the category, which is itself a metatheoretical notion, and a very elusive one. But we wish to regard categories as a logical *tool*, more than merely a model, then we argue that the right notion to consider is that of isomorphism of objects. In fact:

1. it is an equivalence relation, and
2. it says that for each pair of things in the same class, there is (at least) one process turning one into the other, namely (one of) the morphism(s) involved;
3. if categories are our theory, it is in fact part of the theory, and

4. in that it is decidable.

It should be noticed that, in general, in the context of enriched category theory it is not immediate to choose what an isomorphism should be: *a priori*, in fact, we cannot pick elements in the hom object, hence we cannot say something like *a map f such that there is a map g...*. Here we enrich with a concrete category, namely $\mathbf{Set}(\mathbb{M})$, so perhaps using its relation to **Set** might be helpful, but there are still two reasonable choices:

- consider isomorphisms those $f \in \mathcal{C}(A,B)^0$ such that there is a $g \in \mathcal{C}(B,A)^0$ such that $gf = \mathrm{id}$, $fg = \mathrm{id}$, or
- consider isomorphisms those elements f of $\mathcal{C}(A,B)$, meaning $I \to \mathcal{C}(A,B)$ such that there is a an element $g\colon I \to \mathcal{C}(B,A)$ such that $gf = \mathrm{id}$, $fg = \mathrm{id}$, so that all isomorphisms are of value 1.

This brings us to three possible solutions to the problem of interpreting definitional equality. We analyze them in detail, then compare them.

4.7.1 Certain equality

We might decide to say that two contexts are equal if they are isomorphic with value 1 in the category, meaning that there are inverse maps, each with value one, between them. Then two types in context Γ are equal if they are isomorphic with value 1 in the slice over Γ, two terms are equal if they are isomorphic with value 1 in the co-slice over Γ.

We can interpret this as saying that we only identify objects that we are sure to be equal with confidence 1. This works well, to a certain extent.

Proposition 4.7.1.1 (Soundness III)**.** Let $(\mathcal{C},\mathcal{D})$ a fuzzy display-map category. Then it verifies the following rules for fuzzy type theory,

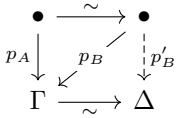
$$\frac{\vdash \Gamma \equiv \Delta \qquad \Gamma \vdash A \equiv B}{\vdash \Gamma.A \equiv \Delta.B} \quad \text{(C-Ext-Eq)}$$

$$\frac{\vdash \Gamma \equiv \Delta \qquad \Gamma \vdash B}{\Delta \vdash B} \quad \text{(Ty-Conv)}$$

in addition to (C-R), (C-S), (C-T), (Ty-R), (Ty-S), (Ty-T), meaning those expressing reflexivity, symmetry, and transitivity of $\equiv$ for contexts and types.

Proof. Proof of rules (C-R), (C-S), (C-T), (Ty-R), (Ty-S), (Ty-T) is trivial. Now recall that a type in context $\Gamma \vdash A$ is identified by a projection $p_A\colon \Gamma.A \to \Gamma$. As in Theorem 4.6.0.2, we omit writing their domains to avoid misleading interpretations.

Consider the following diagram portraying the premises of (C-Ext-Eq),

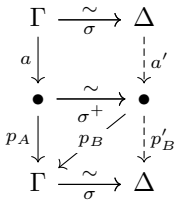


then $\Delta \vdash B$ is portrayed by the induced projection p'_B, which is a display by 2 in Definition 4.5.2.1. In particular, such a composition proves (Ty-Conv). The top isomorphism proves $\Gamma.A \equiv \Delta.B$. □

Remark 4.7.1.2 (The issue with typing)**.** We would also wish to interpret the following rule,

$$\frac{\Gamma \vdash a :_\alpha A \qquad \vdash \Gamma \equiv \Delta \qquad \Gamma \vdash A \equiv B}{\Delta \vdash a :_\alpha B} \quad \text{(Tm-Conv)}$$

which does have some sort of match in our category: consider the diagram



where $a' = \sigma^+ a \sigma^{-1}$. This produces a section of p'_B because

$$p'_B a' = \sigma p_B \sigma^+ a \sigma^{-1} = \sigma p_A a \sigma^{-1} = \mathrm{id},$$

and $|\, a' \,| \geq |\, \sigma^+ \,| \cdot |\, a \,| \cdot |\, \sigma^{-1} \,| = |\, a \,|$.

Hence there is a term of the necessary type, but that is not *precisely* a, it is only isomorphic to it. It if was allowed, we would write something like

$$\frac{\Gamma \vdash a :_\alpha A \qquad \vdash \Gamma \equiv \Delta \qquad \Gamma \vdash A \equiv B}{a \equiv a' \qquad \Delta \vdash a' :_\alpha B} \quad \text{(Tm-Conv}')$$

but judgement of mixed context, such as $a \equiv a'$ would be, are not allowed in our system. Of course something of this kind could be expressed in the language of judgemental theories Chapter 2, instead.

4.7.2 Skeletons

Our proposed solution matches the approach in Example 4.7.0.1 and the categorical tool "hiding" (such) isomorphisms is that of *skeleton*. We adjust the classical definition to the enriched context to only include isomorphisms of value 1.

Definition 4.7.2.1 (Integral skeleton)**.** Let $\mathcal{C}$ a $\mathbf{Set}(\mathbb{M})$-enriched category. Its *integral skeleton* $\mathsf{sk}(\mathcal{C})$ is a full $\mathbf{Set}(\mathbb{M})$-subcategory of $\mathcal{C}$ where

- the inclusion $I \colon \mathsf{sk}(\mathcal{C}) \hookrightarrow \mathcal{C}$ is *integrally essentially surjective*, meaning that for any A in $\mathcal{C}$ there is an X such that IX is iso to A with value 1, and
- $\mathsf{sk}(\mathcal{C})$ is *integrally skeletal*, meaning that no two distinct object in $\mathsf{sk}(\mathcal{C})$ are isomorphic with value 1.

Remark 4.7.2.2 (On the existence of the integral skeleton)**.** If we assume a form of choice for classes, then every $\mathbf{Set}(\mathbb{M})$-category has a skeleton. Such a hypothesis can be definitely weakened, for example to choice for sets and small $\mathbf{Set}(\mathbb{M})$-categories, or to even finer settings. We refer to the literature on skeletons in classical category theory (see for example [AHS90]) for a deeper treatment of the problem.

Proposition 4.7.2.3 (Properties of the integral skeleton)**.**

1. Any two integral skeletons of a given $\mathbf{Set}(\mathbb{M})$-category are isomorphic.
2. Any integral skeleton of a given $\mathcal{C}$ is equivalent to $\mathcal{C}$.

Proof. As for 1, let $\mathsf{sk}(\mathcal{C})$ and $\mathsf{sk}(\mathcal{C})'$ be two skeletons of $\mathcal{C}$. Since they are both integrally essentially surjective in $\mathcal{C}$, by transitivity each X in $\mathsf{sk}(\mathcal{C})$ is iso with value 1 to an X' in $\mathsf{sk}(\mathcal{C})'$, say by an f_X in $\mathcal{C}$. We can define a $\mathbf{Set}(\mathbb{M})$-functor $F\colon \mathsf{sk}(\mathcal{C}) \to \mathsf{sk}(\mathcal{C})'$ mapping X to X' and on maps

$$F(h\colon X \to Y) = f_Y \circ h \circ f_X^{-1}.$$

It is a $\mathbf{Set}(\mathbb{M})$-functor because $|\, f_Y \circ h \circ f_X^{-1} \,| \geq |\, h \,|$ for all h.

As for 2, the inclusion is a $\mathbf{Set}(\mathbb{M})$-equivalence with inverse the $\mathbf{Set}(\mathbb{M})$-functor sending each A to its corresponding X. This is a functor because we only consider isos with value 1. □

Lemma 4.7.2.4 (Skeleton preserves display maps)**.** Let $(\mathcal{C}, \mathcal{D})$ a fuzzy display-map category. Call $\mathsf{sk}(\mathcal{D})$ the restriction of $\mathcal{D}$ to $\mathsf{sk}(\mathcal{C})$. Then $(\mathsf{sk}(\mathcal{C}), \mathsf{sk}(\mathcal{D}))$ is a fuzzy display-map category.

Proof. Trivial. □

Corollary 4.7.2.5. It follows that $(\mathsf{sk}(\mathcal{C}), \mathsf{sk}(\mathcal{D}))$ satisfies rules in Theorem 4.6.0.2 and in Theorem 4.6.0.2.

Definition 4.7.2.6 (Skeletal fuzzy display-map category)**.** A *skeletal fuzzy display-map category* is a fuzzy display-map category $(\mathcal{C}, \mathcal{D})$ where $\mathcal{C}$ is integrally skeletal.

We finally show that all rules for definitional equality are valid in a skeletal fuzzy display-map category. We omit mentioning those expressing reflexivity, symmetry, and transitivity of $\equiv$ for contexts, types, and terms, as their validity is trivial.

Proposition 4.7.2.7 (Soundness IIIb)**.** Let $(\mathcal{C}, \mathcal{D})$ be a skeletal fuzzy display-map category. Then it verifies the following rules for fuzzy type theory,

$$\frac{\vdash \Gamma \equiv \Delta \qquad \Gamma \vdash A \equiv B}{\vdash \Gamma.A \equiv \Delta.B} \quad \text{(C-Ext-Eq)}$$

$$\frac{\vdash \Gamma \equiv \Delta \qquad \Gamma \vdash B}{\Delta \vdash B} \quad \text{(Ty-Conv)}$$

$$\frac{\Gamma \vdash a :_\alpha A \qquad \vdash \Gamma \equiv \Delta \qquad \Gamma \vdash A \equiv B}{\Delta \vdash a :_\alpha B} \quad \text{(Tm-Conv)}$$

in addition to (C-R), (C-S), (C-T), (Ty-R), (Ty-S), (Ty-T), (Tm-R), (Tm-S), (Tm-T) meaning those expressing reflexivity, symmetry, and transitivity of $\equiv$ for contexts, types, and terms.

Proof. Everything but terms work as in Proposition 4.7.1.1. Reflexivity, symmetry, and transitivity are here trivial. As for (Tm-Conv), consider the discussion in Remark 4.7.1.2. □

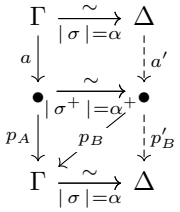
The problem with skeletons is that they seem to overlook many differences, and they are generally frowned upon. In this case it might not be as bad, since we are only picking isomorphism of the top-most value, but if we look at what happens with sets, namely when $\mathbb{M} = \{1\}$, it becomes problematic again.

4.7.3 Fuzzy equality

So we turn to the second possibility, namely that of isomorphisms of all possible values in $\mathbb{M}$, and have definitional equality be fuzzy as well.

$$\Gamma \equiv_{\alpha\cdot\alpha'} \Delta \quad \text{iff} \quad \Gamma \underset{|\sigma^{-1}|=\alpha'}{\overset{|\sigma|=\alpha}{\rightleftarrows}} \Delta$$

When looking back at the diagram in Remark 4.7.1.2,



one quickly sees that now we need to allow for fuzzy types too, because the new presentation for B would produce

$$|\, p'_B \,| = |\, \sigma \,| \cdot |\, p_B \,| \geq \alpha.$$

We believe this to be very promising and in fact a strong argument in favor of fuzzy types. We claim that dropping 4 in Definition 4.5.2.1 produces the following rules for *fully fuzzy* dependent types. In light of this new choice, we move the values measuring confidence, meaning we write

$$\begin{array}{rcl} \Gamma \vdash_\alpha A & \text{iff} & |\, p_A \,| \geq \alpha \\ \Gamma \vdash_\alpha a :_{\alpha'} A & \text{iff} & |\, p_A \,| \geq \alpha,\ |\, a \,| \geq \alpha',\ p_A a = \mathrm{id} \end{array}$$

Nevertheless, we leave this for a later time.

4.7.4 The trivial option

Finally, we could simply consider identity of objects and commutativity of appropriate triangles, and produce something very similar to Section 4.7.1. In this case, Remark 4.7.1.2 is immediately resolved. With this interpretation of definitional equality, we have the following.

Theorem 4.7.4.1 (Soundness)**.** Let $(\mathcal{C}, \mathcal{D})$ be a fuzzy display-map category, then it satisfies the following axioms for fuzzy type theory.

$$\frac{\vdash \Gamma \equiv \Delta \qquad \Gamma \vdash A \equiv B}{\vdash \Gamma.A \equiv \Delta.B} \quad \text{(C-Ext-Eq)}$$

$$\frac{\vdash \Gamma \equiv \Delta \qquad \Gamma \vdash B}{\Delta \vdash B} \quad \text{(Ty-Conv)}$$

$$\frac{\Gamma \vdash a :_\alpha A \qquad \vdash \Gamma \equiv \Delta \qquad \Gamma \vdash A \equiv B}{\Delta \vdash a :_\alpha B} \quad \text{(Tm-Conv)}$$

$$\frac{}{\vdash \diamond\,\mathtt{ctx}}(\text{C-Emp}) \qquad \frac{\Gamma \vdash A\,\mathtt{Type}}{\vdash \Gamma, x : A\,\mathtt{ctx}}(\text{C-Ext}) \qquad \frac{\vdash \Gamma, x : A, \Delta\,\mathtt{ctx}}{\Gamma, x : A, \Delta \vdash x :_1 A}(\text{Var})$$

$$\frac{\Gamma, \Delta \vdash B\,\mathtt{Type} \quad \Gamma \vdash A\,\mathtt{Type}}{\Gamma, x : A, \Delta \vdash B\,\mathtt{Type}}(\text{Weak}_{ty}) \qquad \frac{\Gamma, \Delta \vdash b :_\beta B \quad \Gamma \vdash A\,\mathtt{Type}}{\Gamma, x : A, \Delta \vdash b :_\beta B}(\text{Weak}_{tm})$$

$$\frac{\Gamma, x : A, \Delta \vdash B\,\mathtt{Type} \quad \Gamma \vdash a :_\alpha A}{\Gamma, \Delta[a/x] \vdash B[a/x]\,\mathtt{Type}}(\text{Subst}_{ty}) \qquad \frac{\Gamma, x : A, \Delta \vdash b :_\beta B \quad \Gamma \vdash a :_\alpha A}{\Gamma, \Delta[a/x] \vdash b[a/x] :_\beta B[a/x]}(\text{Subst}_{tm})$$

Proof. It follows from Theorem 4.6.0.2, Theorem 4.6.0.4, Proposition 4.7.1.1, Remark 4.7.1.2. □

We have shied away from this option up to this point because identity of objects is in and of itself quite a delicate matter in foundations of category theory, see for example [Bén85]. But if we start from an enriched setting, most problems are avoided by the existence of the identity picking functor $\mathrm{id}_\Gamma \colon I \to \mathcal{C}(\Gamma, \Gamma)$.

4.7.5 In conclusion

Before we take stock of this whole discussion, we care to explain why we have saved the discussion on definitional equality for last. The reasons are at least two:

- it is closely related to both the foundations and the metatheory one operates in, and as such it is the one that most requires understanding how the inner theory works;
- we believe that the fact that the informative content of equality can be in our theory *modulated* to different levels of confidence, depending on the interpretation of opinions one is most comfortable working with, is not detrimental but an interesting feature of our proposed structure.

Our discussions in Section 4.7.1 and Section 4.7.4 can be pointed at to be perhaps the most "natural" ones, though the first suggests a little flaw in the interpretation; that in Section 4.7.3 paves the way to an interesting extension of our theory, and one we have all tools for; Section 4.7.2 just makes it work. Of course many more drastic choices could have been made, for example considering double categories or enriching over sets not only with fuzzy membership, but with fuzzy equality, as well. We leave this development to future works.

4.8 Future developments

Many questions remain open in Section 4.7, in the future we hope to explore and compare all possible choices for definitional equality, and we are in particular very excited to write fuzzy judgements such as those in 4.7.3. Moreover, now that the structural rules are well established, the next step would be to read type constructors and connectives in our theory.

Of course a different, but equivalent, approach would be considering *enriched fibrations* as in [Vas18], and rephrase our structure in term of judgemental theories, in a way much similar to 2.3.

Another interesting comparison to do would be that of our fuzzy type theory with *quantitative type theory* as in [Atk18]: though different in spirit, since it aims to model resource usage instead of confidence, hence taking value in a non-ordered structure, it has many connections to ours and its algebraic perspective could be quite interesting to import.

Finally, we want to recall that the very motivation of this project is modeling opinions and in particular opinion dynamics, so that we soon hope to use this to extend *cellular sheaves* as in [GH20] to the fuzzy context.

Acknowledgements

The content of Chapter 2 is based on a joint work with Ivan Di Liberti and a version of it has appeared as [CD22]. The author is especially grateful to Nathanael Arkor, Jacopo Emmenegger and Francesco Dagnino for their comments and for their guidance through the literature. Moreover, the author is indebted to Pino Rosolini, Milly Maietti and Mike Shulman for greatly inspiring discussions.

The content of Chapter 3 is based on a joint work with Jacopo Emmenegger. The author is deeply grateful to Francesco Dagnino for suggesting subtyping in the setting of judgemental theories, for his sharp questions, and for some help with a proof in Section 3.3.5.

The content of Chapter 4 is based on a work in progress joint with Shreya Arya, Ana Luiza da Conceição Tenorio, Paige North, Sean O'Connor and Hans Reiss. The author is grateful to the founders of the Adjoint School, an annual research school in applied category theory, and to the organizers of the 2022 edition, Angeline Aguinaldo, Elena Di Lavore, Sophie Libkind, and David Jaz Myers. A great part of this work came to be during the research week for the school, which took place in July 2022 at the University of Strathclyde in Glasgow, Scotland.

The author is, moreover, deeply grateful to the reviewers of the present work for their thoughtful comments, which led to greatly improve this work in its present form, and their references to relevant bibliography. Their careful patience in tolerating the strange notation of Chapter 2 is not to be underestimated.

This work would not have been possible without the many *many* stimulating conversations with all the people in the logic group at the University of Genoa – many of which have already been mentioned here. You made discussing math and logic a joy.

Additionally, the author would like to thank the ItaCa community, for these years would have been much less inspiring without it.

Finally, the author is deeply indebted to their advisor, Pino Rosolini, for witnessing so clearly his love for math, logic, and category theory. It is a testament to how important it is that such knowledge is spread to students and to people in general. I will not forget it.

Segue quella parte di ringraziamenti che difficilmente interesserà chi per qualche motivo vorrà consultare gli atti ufficiali di questa discussione, e che pertanto non ne fa parte. Se questi anni stremanti sono stati anche gioiosi e memorabili è merito delle persone qui sotto. Essendo io piuttosto fortunata, queste sono molte.

Non ringrazierò mai abbastanza tutti gli amici genovesi – di nascita o d'adozione – che hanno reso per me meravigliosa una città di suo già straordinaria.

Grazie a Cecio per gli inopportuni gruppi WhatsApp, che mi hanno salvata, a Paolo e Giacomo per la loro fratellanza e per aver condiviso sempre le mail più imporanti. Grazie a Gianvi per le parole sagge, l'ospitalità, e i cappuccini da Luca. Grazie a Rosanna e Noemi perché hanno dato una casa a me e alla mia passione per le più orribili competizioni musicali in TV. Grazie ad Andrea, Lorena, Sara, Bastiano, Chiara, Emilia, Luca, Martina, Marco, Xavi, Kyveli. Grazie a tutto il MaLGa, che mi ha ospitato quasi mai controvoglia (ciao Larbi, sempre forza Marocco). Grazie a Matte e Nico per l'ufficio, grazie agli amici dell'ottavo piano, a Hongmiao e Francesco Z. Grazie (di nuovo) a Francesco D., Jacopo ed Enrico per la loro amicizia, le risate, e il loro eccezionale gusto in fatto di birra.

Grazie di nuovo pure agli amici di ItaCa: Beppe, Ivan, Fosco, Paolo, Daniele, Edoardo, Franci, Bob, Jacopo, Francesco, Matteo, Andrea. Senza di voi, semplicemente, non sarei arrivata alla fine di questa cosa. Soprattutto non mi sarei divertita tanto. Con loro ringrazio anche tutti gli amici categoristi: conoscervi è stata un dono inaspettato. Un grazie particolare va a Nicolas, che oltre ad aver letto buona parte di questa tesi, è anche la persona perfetta da chiamare per parlare di matematica, vita, o burlesque; a Ivan, che ha condiviso con me alcune delle meraviglie che vede, e che mi ha insegnato che per avere una buona conversazione è utile dirsi quanto si crede una cosa; e Fosco e Davide, per le foto di gattini. Ci vediamo per il cammino di cui si parlava in occasione del prossimo CT a Santiago.

Grazie ai matematici milanesi, che saranno sempre casa per me, a Franci, Pao e Cami. Grazie agli amici più Truci, a Vivi perché ama la vita senza fare troppo casino. Grazie a Sofi, Lavi, Roma, Ferra e Lucy per le avventure. Grazie ai colleghi politici, che nonostante le fatiche non smettono di provarci, e che alla fine donano un po' di speranza anche a me che, notoriamente, sono senza cuore.

Grazie ai nonni Paolina e Turi, che fanno le domande più difficili come se niente fosse, alla nonna Rita e al suo senso dell'umorismo, e al nonno Sergio che non c'è più. A tutti i parenti, veri e acquisiti. Grazie alla mamma, al papà, a Filo e a Benny, che sono sempre una parte di me – quella che sa ridere e cucinare.

Grazie a Ivan, senza il quale la vita sarebbe per me troppo veloce.

Infine grazie a M., che in un breve viaggio in macchina mi ha forse salvato la vita.

Bibliography

The items are ordered with respect to how their citation is coded. While we believe having short place-holders makes reading the thesis a bit smoother, it also means that a given author's work might not appear consecutively in this list.

[AENR21] Benedikt Ahrens, Jacopo Emmenegger, Paige Randall North, and Egbert Rijke. B-systems and C-systems are equivalent. *ArXiv e-prints*, 2021.

[AGH21] Steve Awodey, Nicola Gambino, and Sina Hazratpour. Kripke-Joyal forcing for type theory and uniform fibrations. *arXiv preprint arXiv:2110.14576*, 2021.

[AHS90] Jiri Adamek, Horst Herrlich, and George E. Strecker. *Abstract and Concrete Categories: The Joy of Cats*. Wiley, New York, 1990.

[AL17] Benedikt Ahrens and Peter LeFanu Lumsdaine. Displayed categories. *Logical Methods in Computer Science*, Volume 15, Issue 1, 05 2017.

[Atk18] Robert Atkey. The syntax and semantics of quantitative type theory. In *LICS '18: 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, July 9–12, 2018, Oxford, United Kingdom*, 2018.

[Awo18] Steve Awodey. Natural models of homotopy type theory. *Mathematical Structures in Computer Science*, 28(2):241–286, 2018.

[BB66] Michael Barr and Jonathan Beck. Acyclic models and triples. 1966.

[BBD+97] William P. Barse, Linda J. Brown, John E. Douglas, James Feathers, C. Vance Haynes, Andrew Henderson, Maura Imazio da Silveira, Judy Kemp, Marconales Lima da Costa, Christiane Lopes Machado, Matthew O'Donnell, Ellen Quinn, Richard E. Reanier, and A. C. Roosevelt. Dating a paleoindian site in the amazon in comparison with clovis culture. *Science*, 275(5308):1948–1952, 1997.

[BCF08] Véronique Benzaken, Giuseppe Castagna, and Alain Frisch. Semantic subtyping: Dealing set-theoretically with function, union, intersection, and negation types. *J. ACM*, 55(4), sep 2008.

[Bén67] Jean Bénabou. Introduction to bicategories. In *Reports of the Midwest Category Seminar*, pages 1–77, Berlin, Heidelberg, 1967. Springer Berlin Heidelberg.

[Bén68] Jean Bénabou. Structures algébriques dans les catégories. *Cahiers de topologie et géométrie différentielle*, 10(1):1–126, 1968.

[Bén85] Jean Bénabou. Fibered categories and the foundations of naive category theory. *The Journal of Symbolic Logic*, 50(1):10–37, 1985.

[BJ81] André Boileau and André Joyal. La logique des topos. *The Journal of Symbolic Logic*, 46(1):6–16, 1981.

[Bor94] Francis Borceux. *Handbook of Categorical Algebra*, volume 1 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1994.

[BSSS21] Filippo Bonchi, Alessio Santamaria, Jens Seeber, and Paweł Sobociński. On Doctrines and Cartesian Bicategories. In Fabio Gadducci and Alexandra Silva, editors, *9th Conference on Algebra and Coalgebra in Computer Science (CALCO 2021)*, volume 211 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:17, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[Car86] John Cartmell. Generalised algebraic theories and contextual categories. *Annals of Pure and Applied Logic*, 32:209–243, 1986.

[CD11] Pierre Clairambault and Peter Dybjer. The biequivalence of locally cartesian closed categories and Martin-Löf type theories. In Luke Ong, editor, *Typed Lambda Calculi and Applications*, pages 91–106, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[CD22] Greta Coraglia and Ivan Di Liberti. Context, judgement, deduction. *ArXiv e-prints*, 2022.

[Che03] Gang Chen. Coercive subtyping for the calculus of constructions. In *Proceedings of the 30th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '03, page 150–159, New York, NY, USA, 2003. Association for Computing Machinery.

[Cos72] Michel Coste. Langage interne d'un topos. *Seminaire Bénabou, Université Paris-Nord*, 1972.

[CS10] Miguel de Cervantes Saavedra. *Don Quijote de la Mancha*. New York, 2010.

[CVST10] Joan B. Climent Vidal and Juan Carlos Soliveres Tur. Kleisli and Eilenberg-Moore constructions as parts of biadjoint situations. *Extracta mathematicae*, 25(1):1–61, 2010.

[CZ21] Olivia Caramello and Riccardo Zanfa. On the dependent product in toposes. *Mathematical Logic Quarterly*, 67(3):282–294, 2021.

[DR21] Francesco Dagnino and Giuseppe Rosolini. Doctrines, modalities and comonads, 2021.

[DT87] Roy Dyckhoff and Walter Tholen. Exponentiable morphisms, partial products and pullback complements. *Journal of Pure and Applied Algebra*, 49(1):103–116, 1987.

[Dyb96] Peter Dybjer. Internal type theory. In Stefano Berardi and Mario Coppo, editors, *Types for Proofs and Programs*, pages 120–134, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[EM65] Samuel Eilenberg and John C. Moore. Adjoint functors and triples. *Illinois Journal of Mathematics*, 9(3):381 – 398, 1965.

[EML45] Samuel Eilenberg and Saunders Mac Lane. General theory of natural equivalences. *Transactions of the American Mathematical Society*, 58(2):231–294, 1945.

[EPR20] Jacopo Emmenegger, Fabio Pasquali, and Giuseppe Rosolini. Elementary doctrines as coalgebras. *Journal of Pure and Applied Algebra*, 224(12):106445, 2020.

[Eve17] Caleb Everett. *Numbers and the Making of Us*. Harvard University Press, Cambridge, MA and London, England, 2017.

[EW67] Samuel Eilenberg and Jesse B. Wright. Automata in general algebras. *Information and Control*, 11(4):452–470, 1967.

[FK69] Solomon Feferman and Georg Kreisel. Set-theoretical foundations of category theory. In *Reports of the Midwest Category Seminar III*, pages 201–247, Berlin, Heidelberg, 1969. Springer Berlin Heidelberg.

[Fre66] Peter Freyd. Algebra valued functors in general and tensor products in particular. *Colloquium Mathematicae*, 14(1):89–106, 1966.

[Fre72] Peter Freyd. Aspects of topoi. *Bulletin of Australian Mathematical Society*, 7:1–76, 1972.

[Gen35] Gerhard Gentzen. Untersuchungen über das logische Schließen. ii. *Mathematische Zeitschrift*, 39:405–431, 1935.

[Gen64] Gerhard Gentzen. Investigations into logical deduction. *American Philosophical Quarterly*, 1(4):288–306, 1964.

[GH20] Robert Ghrist and Jakob Hansen. Opinion dynamics on discourse sheaves, 2020.

[GK13] Nicola Gambino and Joachim Kock. Polynomial functors and polynomial monads. *Mathematical Proceedings of the Cambridge Philosophical Society*, 154(1):153–192, 2013.

[GL23] Nicola Gambino and Marco F. Larrea. Models of Martin-Löf type theory from algebraic weak factorisation systems. *The Journal of Symbolic Logic*, 88(1):242–289, 2023.

[GR22] Robert Ghrist and Hans Riess. Cellular sheaves of lattices and the Tarski laplacian. *Homology, Homotopy and Applications*, 24(1):325–345, 2022.

[Gra66] John W. Gray. Fibred and cofibred categories. In S. Eilenberg, D. K. Harrison, S. Mac Lane, and H. Röhrl, editors, *Proceedings of the Conference on Categorical Algebra*, pages 21–83, Berlin, Heidelberg, 1966. Springer Berlin Heidelberg.

[Gra10] Antonio Gramsci. *Lettere dal carcere; Prefazione di Luciano Canfora*. I classici del pensiero libero. RCS Quotidiani, Milano, 2010.

[Gro61] Alexander Grothendieck. Categoriés fibrées et descente (Exposé VI). *Revêtements étales et groupe fondamental - SGA1*, 1960-61.

[Gro60] Alexander Grothendieck. Technique de descente et théorèmes d'existence en géométrie algébrique. I. Généralités. Descente par morphismes fidèlement plats. In *Séminaire Bourbaki : années 1958/59 - 1959/60, exposés 169-204*, number 5 in Séminaire Bourbaki. Société mathématique de France, 1960. talk:190.

[GW07] John A. Gifford and Rachel K. Wentz. Florida's deep past: The bioarchaeology of little salt spring (8so18) and its place among mortuary ponds of the archaic. *Southeastern Archaeology*, 26(2):330–337, 2007.

[HHP93] Robert Harper, Furio Honsell, and Gordon Plotkin. A framework for defining logics. *Journal of the ACM (JACM)*, 40(1):143–184, 1993.

[Hof97] Martin Hofmann. *Syntax and Semantics of Dependent Types*, page 79–130. Publications of the Newton Institute. Cambridge University Press, 1997.

[HP87] Martin Hyland and Andrew M. Pitts. The theory of constructions: Categorical semantics and topos-theoretic models. 01 1987.

[HR13] Dirk Hofmann and Carla D. Reis. Probabilistic metric spaces as enriched categories. *Fuzzy Sets and Systems*, 210:1–21, 2013. Theme : Topology and Algebra.

[Hub61] Peter J. Huber. Homotopy theory in general categories. *Mathematische Annalen*, 1961.

[HWW14] John Harding, Carol Walker, and Elbert Walker. Categories with fuzzy sets and relations. *Fuzzy Sets and Systems*, 256:149–165, 2014. Special Issue on Enriched Category Theory and Related Topics (Selected papers from the 33rd Linz Seminar on Fuzzy Set Theory, 2012).

[Isb64] John R. Isbell. *Subobjects, adequacy, completeness and categories of algebras*. Instytut Matematyczny Polskiej Akademi Nauk, 1964.

[Jac93] Bart Jacobs. Comprehension categories and the semantics of type dependency. *Theoretical Computer Science*, 107(2):169–207, 1993.

[Jac99] Bart Jacobs. *Categorical logic and type theory*. Elsevier, 1999.

[Jec07] Tomáš Jech. *Set Theory: The Third Millennium Edition, revised and expanded*. Springer Monographs in Mathematics. Springer Berlin Heidelberg, 2007.

[Joh77] Peter T. Johnstone. Topos theory, volume 10 of. *London Mathematical Society Monographs*, 1977.

[Kel82] Maxwell Kelly. *Basic concepts of enriched category theory*, volume 64. CUP Archive, 1982.

[Kle65] Heinrich Kleisli. Every standard construction is induced by a pair of adjoint functors. *Proceedings of the American Mathematical Society*, 16(3):544–546, 1965.

[Kle67] Stephen Cole Kleene. *Mathematical Logic*. John Wiley & Sons, 1967.

[KR77] Anders Kock and Gonzalo E. Reyes. Doctrines in categorical logic. In *Studies in Logic and the Foundations of Mathematics*, volume 90, pages 283–313. Elsevier, 1977.

[KS74] G. M. Kelly and Ross Street. Review of the elements of 2-categories. In Gregory M. Kelly, editor, *Category Seminar*, pages 75–103, Berlin, Heidelberg, 1974. Springer Berlin Heidelberg.

[Law63] F. William Lawvere. *Functorial Semantics of Algebraic Theories: And, Some Algebraic Problems in the Context of Functorial Semantics of Algebraic Theories*. Mount Allison University, 1963.

[Law64] F. William Lawvere. An elementary theory of the category of sets. *Proceedings of the National Academy of Sciences U.S.A.*, 52:1506–1511, 1964.

[Law65] F. William Lawvere. Algebraic theories, algebraic categories, and algebraic functors. *The theory of models, Proceedings of the 1963 International Symposium at Berkeley*, page 413–418, 1965.

[Law69] F. William Lawvere. Adjointness in foundations. *Dialectica*, 23(3/4):281–296, 1969.

[Law70] F. William Lawvere. Equality in hyperdoctrines and comprehension schema as an adjoint functor. *Proceedings of the American Mathematical Society*, pages 1–14, 1970.

[Law04] F. William Lawvere. *Functorial Semantics of Algebraic Theories: And, Some Algebraic Problems in the Context of Functorial Semantics of Algebraic Theories*. Number 5. Reprints in Theory and Applications of Categories, 2004.

[Lei98] Tom Leinster. Basic bicategories, 1998.

[LHTVM19] Juan Luis Lopez Hernández, Luis Turcio, and Adrian Vazquez-Marquez. Applications of the Kleisli and Eilenberg-Moore 2-adjunctions. *Categories and General Algebraic Structures with Applications*, 10(1):117–156, 2019.

[Lor21] Fosco Loregian. *(Co)end Calculus*. London Mathematical Society Lecture Note Series. Cambridge University Press, 2021.

[LS88] Joachim Lambek and Philip J. Scott. *Introduction to higher-order categorical logic*, volume 7. Cambridge University Press, 1988.

[LSX13] Zhaohui Luo, Sergei Soloviev, and Tao Xue. Coercive subtyping: Theory and implementation. *Information and Computation*, 223:18–42, 2013.

[Luo99] Zhaohui Luo. Coercive subtyping. *Journal of Logic and Computation*, 9(1):105–130, 02 1999.

[LW15] Peter LeFanu Lumsdaine and Michael A. Warren. The local universes model: An overlooked coherence construction for dependent type theories. *ACM Trans. Comput. Logic*, 16(3), jul 2015.

[Mai05] Maria E. Maietti. Modular correspondence between dependent type theories and categories including pretopoi and topoi. *Mathematical Structures in Computer Science*, 15(6):1089–1149, 2005.

[Mai17] Maria E. Maietti. On choice rules in dependent type theory. In T.V. Gopal, Gerhard Jäger, and Silvia Steila, editors, *Theory and Applications of Models of Computation*, pages 12–23, Cham, 2017. Springer International Publishing.

[Mak93] Michael Makkai. The fibrational formulation of intuitionistic predicate logic I: completeness according to Gödel, Kripke, and Läuchli, part 2. *Notre Dame J. Formal Log.*, 34:471–498, 1993.

[Mar75] Per Martin-Löf. An intuitionistic theory of types: Predicative part. In H.E. Rose and J.C. Shepherdson, editors, *Logic Colloquium '73*, volume 80 of *Studies in Logic and the Foundations of Mathematics*, pages 73–118. Elsevier, 1975.

[Mar84] Per Martin-Löf. *Intuitionistic type theory*, volume 1 of *Studies in proof theory*. Bibliopolis, 1984.

[Mar87] Per Martin-Löf. Truth of a proposition, evidence of a judgement, validity of a proof. *Synthese*, pages 407–420, 1987.

[Mar96a] Per Martin-Löf. On the meanings of the logical constants and the justifications of the logical laws. *Nordic journal of philosophical logic*, 1(1):11–60, 1996.

[Mar96b] Per Martin-Löf. On the meanings of the logical constants and the justifications of the logical laws. *Nordic Journal of Philosophical Logic*, 1(1):11–60, 1996.

[McL92] Colin McLarty. *Elementary Categories, Elementary Toposes*. Clarendon Press, 1992.

[Mit72] William Mitchell. Boolean topoi and the theory of sets. *Journal of Pure and Applied Algebra*, 2(3):261–274, 1972.

[ML78] Saunders Mac Lane. *Categories for the Working Mathematician*. Springer, 1978.

[MLM94] Saunders Mac Lane and Ieke Moerdijk. *Sheaves in Geometry and Logic*. Springer New York, 1994.

[MMdPR05] Maria E. Maietti, Paola Maneggia, Valeria C. V. de Paiva, and Eike Ritter. Relating categorical semantics for intuitionistic linear logic. *Applied Categorical Structures*, 13:1–36, 2005.

[Mog91] Eugenio Moggi. A category-theoretic account of program modules. *Mathematical Structures in Computer Science*, 1(1):103–139, 1991.

[MP89] Michael Makkai and Robert Paré. *Accessible Categories: The Foundations of Categorical Model Theory*. American Mathematical Society, 1989.

[MPR17] Maria E. Maietti, Fabio Pasquali, and Giuseppe Rosolini. Triposes, exact completions, and Hilbert's ϵ-operator. *Tbilisi Mathematical Journal*, 10:141 – 166, 2017.

[MR13] Maria E. Maietti and Giuseppe Rosolini. Quotient completion for the foundation of constructive mathematics. *Logica Universalis*, 7(3):371–402, 2013.

[MR15] Maria E. Maietti and Giuseppe Rosolini. Unifying exact completions. *Applied Categorical Structures*, 23:43–52, 2015.

[MS84] Per Martin-Löf and Giovanni Sambin. *Intuitionistic type theory*, volume 9. Bibliopolis Naples, 1984.

[MS21] Guy McCusker and Alessio Santamaria. Composing dinatural transformations: Towards a calculus of substitution. *Journal of Pure and Applied Algebra*, 225(10):106689, 2021.

[Mye20] David Jaz Myers. Cartesian factorization systems and Grothendieck fibrations. *arXiv: Category Theory*, 2020.

[MZ15] Paul-André Melliès and Noam Zeilberger. Functors are type refinement systems. In *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '15, page 3–16, New York, NY, USA, 2015. Association for Computing Machinery.

[NvP08] Sara Negri and Jan von Plato. *Structural proof theory*. Cambridge university press, 2008.

[Osi75a] Gerhard Osius. Logical and set theoretical tools in elementary topoi. In *Model Theory and Topoi*, pages 297–346. Springer, 1975.

[Osi75b] Gerhard Osius. A note on Kripke-Joyal semantics for the internal language of topoi. In *Model theory and topoi*, pages 349–354. Springer, 1975.

[Pit83] Andrew M. Pitts. An application of open maps to categorical logic. *Journal of Pure and Applied Algebra*, 29:313–326, 1983.

[Pit99] Andrew M. Pitts. Tripos theory in retrospect. *Electronic Notes in Theoretical Computer Science*, 23(1):111–127, 1999. Tutorial Workshop on Realizability Semantics and Applications (associated to FLoC'99, the 1999 Federated Logic Conference).

[Res02] Greg Restall. *An introduction to substructural logics*. Routledge, 2002.

[Rie17] Emily Riehl. *Category Theory in Context*. Aurora: Dover Modern Math Originals. Dover Publications, 2017.

[Sco79] Dana S. Scott. *Identity and existence in intuitionistic logic*, pages 660–696. Springer Berlin Heidelberg, Berlin, Heidelberg, 1979.

[See84] Robert A. G. Seely. Locally cartesian closed categories and type theory. In *Mathematical proceedings of the Cambridge philosophical society*, volume 95, pages 33–48. Cambridge University Press, 1984.

[See86] Robert A. G. Seely. *Modelling Computations: a 2-categorical Framework*. The College, 1986.

[Shu08] Michael A. Shulman. Set theory for category theory, 2008.

[Str72] Ross Street. The formal theory of monads. *Journal of Pure and Applied Algebra*, 2(2):149–168, 1972.

[Str80] Ross Street. Fibrations in bicategories. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, 21(2):111–160, 1980.

[Str22] Thomas Streicher. Fibred categories à la Jean Bénabou. *arXiv preprint arXiv:1801.02927*, 2022.

[SW73] Ross Street and Robert F. C. Walters. The comprehensive factorization of a functor. *Bulletin of the American Mathematical Society*, 79:936–941, 1973.

[Tar56] Alfred Tarski. The concept of truth in formalized languages. *Logic, semantics, metamathematics*, 2(152-278):7, 1956.

[Tay99] Paul Taylor. *Practical Foundations of Mathematics*. Number v. 59 in Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1999.

[Tro87] Anne S. Troelstra. On the syntax of Martin-Löf's type theories. *Theoretical Computer Science*, 51(1):1–26, 1987.

[Tro20] Davide Trotta. The existential completion. *Theory Appl. Categ.*, 35:1576–1607, 2020.

[TS00] Anne S. Troelstra and Helmut Schwichtenberg. *Basic proof theory*. Number 43. Cambridge University Press, 2000.

[Uem19] Taichi Uemura. A general framework for the semantics of type theory. *arXiv preprint arXiv:1904.04097*, 2019.

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

[Vas18] Christina Vasilakopoulou. On enriched fibrations. *Cahiers de topologique et géométrie différentielle categoriques*, LIX-4, 2018.

[Vis04] Angelo Vistoli. Notes on Grothendieck topologies, fibered categories and descent theory, 2004.

[Voe16] Vladimir Voevodsky. Dependent type theories, February 2016.

[Wad15] Philip Wadler. Propositions as types. *Communications of the ACM*, 58(12):75–84, 2015.

[Web07] Mark Weber. Yoneda structures from 2-toposes. *Applied Categorical Structures*, 15(3):259–323, 2007.

[Wit22] Ludwig Wittgenstein. Tractatus logico-philosophicus. *London: Routledge, 1981*, 1922.

[Wyl91] Oswald Wyler. *Lecture Notes on Topoi and Quasitopoi*. World Scientific, 1991.

[Zad65] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.