Revisiting the categorical interpretation of dependent type theory

Pierre-Louis Curien

πr^2 team, PPS Laboratory, CNRS, Université Paris Diderot, and INRIA, France

Richard Garner

Macquarie University, Sydney, Australia

Martin Hofmann

Ludwig-Maximilian's-Universität, München, Germany

Abstract

We show that Hofmann's and Curien's interpretations of Martin-Löf's type theory, which were both designed to cure a mismatch between syntax and semantics in Seely's original interpretation in locally cartesian closed categories, are related via a natural isomorphism. As an outcome, we obtain a new proof of the coherence theorem needed to show the soundness after all of Seely's interpretation.

1. Introduction

About thirty years ago, Seely [23] explained how to interpret extensional Martin-Löf's type theory in locally cartesian closed categories, using the substitution-as-pullback paradigm of categorical logic. But there was a coherence issue arising in this interpretation from the pseudo-functoriality of pullbacks, that had not been addressed by Seely.

In [7], the first author of the present paper studied this problem carefully. He proved the soundness of Seely's interpretation by first designing a syntax with explicit coercions (thus mirroring the pseudo-functoriality at the level of the language being modelled), and then by showing the coherence as a syntactic result, using rewriting techniques. The observation made by Huet in his (unpublished) lecture notes on category theory [14] that Mac Lane's proof of coherence for monoidal categories was a "categorification" of Knuth-Bendix lemma was instrumental for this proof.

In [12], the third author of this work circumvented the coherence issue by showing how to obtain a split model (that is, a model in which composition of substitutions in types and terms is associative "on the nose" rather than up to isomorphism) of Martin-Löf's type theory from a locally cartesian closed

category. Then the original type theory can be interpreted straightforwardly in this "strictified" model. The strictification consisted in taking a well-known construction in fibred category theory, going back to Giraud [11] and Bénabou [2], of a right adjoint to the forgetful functor from split fibrations and strict morphisms on a fixed base category to the category of fibrations and fibration morphisms (which are required to preserve the chosen structure only up to isomorphism), and in showing that this construction carries over to deal with additional structure required for the interpretation of Martin-Löf's type theory. Hofmann worked not with fibrations explicitly but rather with the more "syntax-friendly" framework of Cartmell's categories with attributes.

Therefore, in retrospect, the first and the third author had taken "dual" routes to cure the mismatch between the (strict) syntax and the (non-strict) models: either "unstrictify" the syntax, or strictify the model. The genesis of this work lies there: we wanted to understand the conceptual architecture in which these two approaches can be linked. In conversations with the second author, it soon became clear that three large categories were involved:

1. a category of non-strict structures and functors preserving the structure up to iso: this is where locally cartesian closed categories and Seely's original interpretation live;
2. a category of strict structures and strict morphisms (i.e., preserving the structure exactly): this is where Hofmann's interpretation lives;
3. a category of non-strict structures and strict morphisms: this is where Curien's modified syntax lives as a free structure.

In pictures, we shall represent the respective morphisms pictorially using

This three-fold superstructure comes up in various contexts, starting with monoidal categories (and indeed the monoidal case served us as a very useful test bed for the results presented here). In our case, the structures under consideration are the comprehension categories that have products and strong sums, and support extensional identity types, or $\mathbb{ML}$-categories for short. These are fibrations with additional structure, which we shall recall later (Sections 5.1 and 9). But the global picture can emerge without opening this "black box". Let us denote the corresponding three large categories by $\mathbb{ML}$, $\mathbf{S}\mathbb{ML}_{\mathbf{s}}$ and $\mathbb{ML}_{\mathbf{s}}$, respectively. Let $\mathtt{Synt}^e$ be the classifying $\mathbb{ML}$-category and let $\mathtt{Synt}$ be the classifying strict $\mathbb{ML}$-category, which are built up from the syntax with explicit coercions and from the original syntax of Martin-Löf's type theory, respectively (see Section 5.3). They are initial in $\mathbb{ML}_{\mathbf{s}}$ and $\mathbf{S}\mathbb{ML}_{\mathbf{s}}$, respectively. Our story then goes as follows.

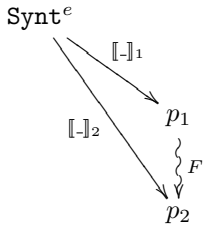
I. Let p_1 and p_2 be $\mathbb{ML}$-categories. Let $[\![\_]\!]_1$ and $[\![\_]\!]_2$ be the interpretation functions of the explicit syntax in p_1, p_2, respectively. Thus we have (cf. item (3) above):

$$[\![\_]\!]_1 \in \mathbb{ML}_{\mathbf{s}}[\mathtt{Synt}^e, p_1] \qquad [\![\_]\!]_2 \in \mathbb{ML}_{\mathbf{s}}[\mathtt{Synt}^e, p_2]$$

(note that by design interpretation functions are strict). Consider further a morphism

$$F \in \mathbb{ML}[p_1, p_2] \ .$$

Since F is not required to be strict, we do not have $F \circ [\![\text{-}]\!]_1 = [\![\text{-}]\!]_2$, but, as we shall show, the two functors are still related through a natural isomorphism γ. In picture:

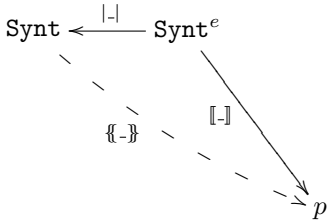


where the triangle commutes up to the isos γ.

II. Let us denote with $\{\!\{\text{-}\}\!\} \in \mathbf{S}\mathbb{ML}_{\mathbf{s}}[\mathtt{Synt}, p]$ the interpretation function of the original syntax in a strict $\mathbb{ML}$-category p (cf. item (2) above), and let us write more suggestively $|\text{-}|$ for $[\![\text{-}]\!]_{\mathtt{Synt}} \in \mathbb{ML}_{\mathbf{s}}[\mathtt{Synt}^e, \mathtt{Synt}]$ (indeed, $|\text{-}|$ removes the explicit coercions from the syntax). Then, by initiality in $\mathbb{ML}_{\mathbf{s}}$ (noting that a fortiori $\{\!\{\text{-}\}\!\} \in \mathbb{ML}_{\mathbf{s}}[\mathtt{Synt}, p]$), we have the following factorisation:

$$[\![\text{-}]\!] = \{\!\{|\text{-}|\}\!\}$$

or, pictorially:



where the triangle commutes exactly.

III. Let $\mathbb{C}$ be a locally cartesian closed category, which viewed as a fibration $p_1 = \mathtt{cod} : \mathbb{C}^{\to} \to \mathbb{C}$ endowed with a trivial identity comprehension structure is an object of $\mathbb{ML}$. Let p_2 be the Giraud-Bénabou-Hofmann strictification of p_1. Then there is a faithful and *non-strict* functor $F = (F_t, F_b) : p_1 \to p_2$ over $\mathbb{C}$ (i.e., $p_2 \circ F_t = F_b \circ p_1$ and $F_b = id$).

Then we can instantiate the situation in I as indicated in III: via II, we obtain natural isos relating Curien's interpretation $[\![\text{-}]\!]$ and Hofmann's interpretation $\{\!\{\}\!\}$, and as a bonus we get a *new proof* of the coherence theorem for the interpretation of Martin-Löf's type theory in locally cartesian closed categories. This theorem states that if we have (in the explicit syntax)

$$\Gamma \vdash M_1 : \sigma \qquad \Gamma \vdash M_2 : \sigma \qquad |M_1| = |M_2|$$

then $[\![M_1]\!] = [\![M_2]\!]$ (in any LCCC). Indeed, we have:

$$[\![M_1]\!] = F_b([\![M_1]\!]) = \gamma_{\Gamma,\sigma} \circ \{\!|\,|M_1|\,|\!\} \circ \gamma_\Gamma^{-1} = \gamma_{\Gamma,\sigma} \circ \{\!|\,|M_2|\,|\!\} \circ \gamma_\Gamma^{-1} = F_b([\![M_2]\!]) = [\![M_2]\!]\,.$$

The plan of the paper is as follows. Sections 2 through 6 collect background material to make the paper as self-contained as possible. We recall from [7] the formulation of Martin-Löf's type theory in variable-free style (Section 2), and with explicit coercions (Section 4). We recall Seely's interpretation in Section 3, and show a typical instance of the coherence problem (again recollected from [7]). In Section 5, we recall from [15] the interpretation of core (type constructor free) dependent type theory in comprehension categories, and we sketch the construction of $\mathtt{Synt}$ and $\mathtt{Synt}^e$. In Section 6, we recall the Giraud-Bénabou construction, and formulate the transfer of additional structure in the language of comprehension categories (as done in Warren's PhD thesis [25, Section 2.4]). We then make a pause in Section 7: we examine the above superstructure in the case of monoidal categories, and distill Joyal-Street's proof of coherence [18], of which our new proof of coherence is an instance of a (yet to be properly formulated) generalisation. In section 8, we assemble the material along the lines suggested above. For the smoothness of exposition, we delay the (non-problematic) treatment of products, sums and extensional identity types to Section 9. We conclude in Section 10.

2. Type theory with variable-free syntax

We recall that de Bruijn numbers record the binding depth of a variable (including the typing context in this count). We also recall that in a syntax of explicit substitutions, reduction is fine-grained: the one-step substitution involved in a β-reduction is replaced by a process that lets the substituted term eventually reach the variables.

We limit here the syntax to the bare minimum needed to present our basic core superstructure and coherence proof. In particular, we do not even consider type constructors until Section 9.

Perhaps even more disturbing is the fact that the syntax does not introduce any actual dependent types. One of the simplest ways to overcome this is to add some basic types and constants, like $\mathtt{Nat}$, $0 : \mathtt{Nat}$, $\mathtt{suc} : \mathtt{Nat} \to \mathtt{Nat}$, and then a basic dependent type, say $\mathtt{Natlist}(n)$ with constructors $\mathtt{nil} : \mathtt{Natlist}(0)$ and $\mathtt{cons}$ taking $M : \mathtt{Nat}$ and $N : \mathtt{Natlist}(n)$ to $\mathtt{cons}(M, N) : \mathtt{Natlist}(\mathtt{suc}(n))$. In fact, all that follows is parameterised over a *dependent signature*, as defined, say in Cartmell's original work [5].

With these provisos, the core syntax of [7] (see also [8]) for types, contexts, terms, and substitutions, is:

$$\begin{array}{l}
\sigma ::= \text{base types, possibly dependent} \mid \sigma[s] \\
\Gamma ::= \emptyset \mid (\Gamma, \sigma) \\
M ::= 1 \mid M[s] \mid \text{cases given by the signature} \\
s ::= id \mid \uparrow \mid M \cdot s \mid s \circ s
\end{array}$$

The typing judgements are:

$$\Gamma\ \texttt{context} \qquad \Gamma \vdash \sigma\ \texttt{type} \qquad \Gamma \vdash M : \sigma \qquad \Gamma' \vdash s : \Gamma$$

The typing rules are:

$$\frac{}{\emptyset\ \texttt{context}} \qquad \frac{\Gamma \vdash \sigma\ \texttt{type}}{\Gamma, \sigma\ \texttt{context}} \qquad \frac{\Gamma' \vdash s : \Gamma \quad \Gamma \vdash \sigma\ \texttt{type}}{\Gamma' \vdash \sigma[s]\ \texttt{type}}$$

$$\frac{}{\Gamma, \sigma \vdash 1 : \sigma[\uparrow]} \qquad \frac{\Gamma' \vdash s : \Gamma \qquad \Gamma \vdash M : \sigma}{\Gamma' \vdash M[s] : \sigma[s]}$$

$$\frac{}{\Gamma \vdash id : \Gamma} \qquad \frac{}{\Gamma, \sigma \vdash \uparrow : \Gamma} \qquad \frac{\Gamma_1 \vdash s_1 : \Gamma_2 \qquad \Gamma_2 \vdash s_2 : \Gamma_3}{\Gamma_1 \vdash s_2 \circ s_1 : \Gamma_3}$$

$$\frac{\Gamma' \vdash s : \Gamma \qquad \Gamma \vdash \sigma\ \texttt{type} \quad \Gamma' \vdash M' : \sigma[s]}{\Gamma' \vdash M' \cdot s : \Gamma, \sigma}$$

To illustrate variable-free notation, consider $\Gamma\,,\, x : \sigma\,,\, y : \tau \vdash x : \sigma$. In the variable-free version, this judgement becomes $\Gamma, \sigma, \tau \vdash 1[\uparrow] : \sigma[\uparrow]$, by "composing" $\Gamma, \sigma, \tau \vdash \uparrow : (\Gamma, \sigma)$ and $\Gamma, \sigma \vdash 1 : \sigma[\uparrow]$ (note that the two occurrences of $\uparrow$ have *different* types). The term $1[\uparrow]$ corresponds to de Bruijn number 2, the term $1[\uparrow \circ \uparrow]$ corresponds to de Bruijn number 3, etc. . . .

Finally, the axioms for equality (for types, terms, and substitutions) are as follows:

$$\begin{array}{l}
\sigma[s][t] = \sigma[s \circ t] \\
\sigma[id] = \sigma \\
\\
1[M \cdot s] = M \\
M[s][t] = M[s \circ t] \\
\\
\uparrow \circ (M \cdot s) = s \\
(s_1 \circ s_2) \circ s_3 = s_1 \circ (s_2 \circ s_3) \\
id \circ s = s \circ id = s \\
(M \cdot s) \circ t = M[t] \cdot (s \circ t) \\
1[s] \cdot (\uparrow \circ s) = s
\end{array}$$

In fact, these equality axioms must be more carefully formulated as judgements

$$\vdash \Gamma = \Delta \qquad \Gamma \vdash \sigma = \tau \qquad \Gamma \vdash M = N : \sigma \qquad \Gamma' \vdash s = t : \Gamma$$

i.e., as typed equality rules (the first judgement arises by congruence, see below). For example, the first equality above should be read as

$$\frac{\Gamma'' \vdash t : \Gamma' \qquad \Gamma' \vdash s : \Gamma \qquad \Gamma \vdash \sigma\ \texttt{type}}{\Gamma'' \vdash \sigma[s][t] = \sigma[s \circ t]}$$

etc.... One also has to give rules expressing that $=$ is an equivalence relation, and a congruence, for example:

$$\frac{\Gamma \vdash \sigma = \sigma'}{\Gamma \vdash \sigma' = \sigma} \qquad \frac{\Gamma \vdash \sigma = \sigma'}{\vdash (\Gamma, \sigma) = (\Gamma, \sigma')} \qquad \frac{\Gamma' \vdash s_1 = s_2 : \Gamma \quad \Gamma \vdash \sigma}{\Gamma' \vdash \sigma[s_1] = \sigma[s_2]}$$

Finally, there are rules allowing to change the type or context of a term:

$$\frac{\vdash \Gamma = \Gamma' \quad \Gamma \vdash M : \sigma}{\Gamma' \vdash M : \sigma} \qquad \frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash \sigma = \sigma'}{\Gamma \vdash M : \sigma'}$$

and similarly for substitutions:

$$\frac{\vdash \Gamma = \Gamma' \quad \Gamma \vdash s : \Delta}{\Gamma' \vdash s : \Delta} \qquad \frac{\Gamma \vdash s : \Delta \quad \Gamma \vdash \Delta = \Delta'}{\Gamma \vdash s : \Delta'}$$

This completes the description of the typing system for the above core syntax.

A subtle issue is that typing both $1[M \cdot s]$ and M with the *same* type σ involves itself an equality rule on types (more on this in Section 3).

3. Reviewing Seely's interpretation in LCCCs

We recall that for a category $\mathbb{C}$ and an object C of $\mathbb{C}$, the slice category $\mathbb{C}/C$ has as objects pairs $(D, f : D \to C)$ (f for short), with morphisms defined as follows:

$$(\mathbb{C}/C)[(D_1, f), (D_2, g)] = \{h \in \mathbb{C}[D_1, D_2] \mid g \circ h = f\} \ .$$

A locally cartesian closed category (LCCC) is a category $\mathbb{C}$ with a terminal object and such that all slice categories $\mathbb{C}/C$ are cartesian closed (in particular, $\mathbb{C}$ is cartesian closed). It is well known (see e.g. [9]) that the requirement that all slices are cartesian closed is equivalent to the existence, for any $k : C_1 \to C_2$ in $\mathbb{C}$, of two successive right adjoints to the functor $\Sigma k : \mathbb{C}/C_1 \to \mathbb{C}/C_2$ defined by $(\Sigma k)(f) = k \circ f$:

$$\Sigma k \dashv k^* \dashv \Pi k \ ,$$

these adjunctions being related through the so-called Beck-Chevalley conditions (see Section 9). In particular, unrolling the definition of adjunction, the functor k^* determines a choice of pullbacks along k. There are many locally cartesian closed categories, since in particular every elementary topos is an LCCC. In what follows, the reader can safely think of $\mathbb{C}$ as the category **Set** of sets and functions.

The interpretation of the syntax of Section 2 is defined as follows:

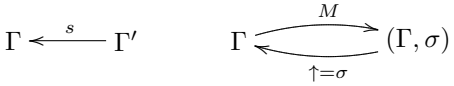
- Contexts are mapped to objects, substitutions to morphisms.

- Types are interpreted as objects in a slice category, and terms as sections. More precisely the interpretations of $\Gamma \vdash \sigma\ \mathtt{type}$ and of $\Gamma, \sigma \vdash \uparrow : \Gamma$ coincide, and the interpretation of $\Gamma \vdash M : \sigma$ is a morphism in $\mathbb{C}[\Gamma, (\Gamma, \sigma)]$ such that $\uparrow \circ M = id$. For example, if $\mathbb{C}$ is **Set**, then one interprets the context $n : \mathtt{Nat}$ as $\mathbb{N}$, and the context $(n : \mathtt{Nat},\ l : \mathtt{Natlist}(n))$ as the set of all finite lists of natural numbers, presented as the disjoint union of all lists of a given length. The interpretation of $\uparrow$ is then the projection function given by the length of the list, and the constraint on the interpretation of $n : \mathtt{Nat} \vdash M : \mathtt{Natlist}(n)$ says that $M(n)$ must have length n (typically, M computes the first n values of a series).

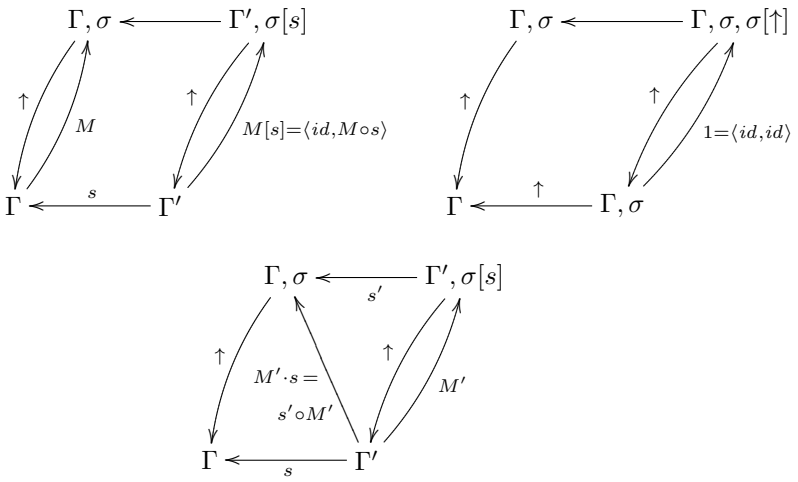
We represent the interpretation of the respective judgements

$$\Gamma' \vdash s : \Gamma \qquad \Gamma \vdash \sigma\ \mathtt{type} \qquad \Gamma \vdash M : \sigma$$

as follows:



We use this pictorial presentation to describe the interpretation of the three basic constructs $M[s]$, 1, and $M' \cdot s$. The three pictures below involve pullbacks (of $\uparrow$ along s, $\uparrow$ along $\uparrow$, and $\uparrow$ along s, respectively). In particular, $\sigma[s]$ is obtained by pulling-back $\sigma = \uparrow$ along s.



where we use angle brackets for pullback universal morphisms. We have omitted the semantic brackets for brevity. We should also note that what we interpret by induction is not a judgement, but a derivation tree π of a judgement (more on this later).

The coherence problem arises when it comes to checking the soundness of the equality rules. Let us first note that some of the equality rules for substitutions do not raise difficulties:

- The composition and identities satisfy the monoid laws exactly in a category.
- For the first axiom in our list, we have (with the notation of the last picture above): $\uparrow \circ (M' \cdot s) = (\uparrow \circ s') \circ M' = s \circ (\uparrow \circ M') = M'$.

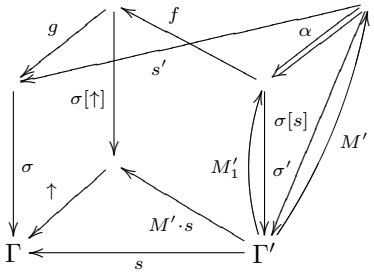
However, the type conversion rule $\sigma[s][t] = \sigma[s \circ t]$ cannot be modelled by an equality, since in an LCCC the composition of chosen pullbacks is only isomorphic to the chosen pullback of the composite, in general. For example, in **Set**, the chosen pullback of $(f : b \to a, g : c \to a)$ is given by $\{(y, z) \in B \times C \mid fy = gz\}$, so that, for $h : b' \to b$, the pullback of g along $h \circ f$ is a subset of $B' \times C$, while the pullback along h of the pullback along f of g is a (isomorphic) subset of $B' \times (B \times C)$.

This indicates that we should interpret (derivations of) type equalities as *isomorphisms*. These isomorphisms will be uniquely defined in the model thanks to universal properties. We shall make this move in Section 5.

Let us now examine the "equality" $1[M' \cdot s] = M'$. The left-hand side is typed as follows:

$$\dfrac{\Gamma, \sigma \vdash 1 : \sigma[\uparrow] \qquad \dfrac{\Gamma' \vdash s : \Gamma \quad \Gamma' \vdash M' : \sigma[s]}{\Gamma' \vdash M' \cdot s : \Gamma, \sigma}}{\Gamma' \vdash 1[M' \cdot s] : \sigma[\uparrow][M' \cdot s]}$$

while M' has type $\sigma[s]$. We have just seen that $\sigma[\uparrow][M' \cdot s]$ is only isomorphic to $\sigma[\uparrow \circ (M' \cdot s)]$ which is in turn (truly) equal to $\sigma[s]$. We show below, as an instance of the coherence theorem, that in an LCCC the equality $1[M' \cdot s] = M'$ holds precisely up to this iso. We have:



where $\sigma' = \sigma[\uparrow][M \cdot s]$, $M'_1 = 1[M \cdot s] = \langle id, 1 \circ s' \circ M' \rangle$, where g, f, s' complete pullback diagrams, and where α is the unique (iso)morphism between the two choices $(s', \sigma[s])$ and $(g \circ f, \sigma')$ of limit for σ, s, i.e., α is characterised by

$$\sigma' \circ \alpha = \sigma[s] \qquad g \circ f \circ \alpha = s' \ .$$

Our goal is to show $M'_1 = \alpha \circ M'$. And we have indeed:

$$\sigma' \circ (\alpha \circ M') = \sigma[s] \circ M' = id = \sigma' \circ M'_1$$
$$(g \circ f) \circ (\alpha \circ M') = s' \circ M' = (g \circ 1) \circ (s' \circ M') = g \circ (f \circ M'_1)$$

This suggests to introduce *explicit coercions* in the syntax, so as to be able to track down, already at the syntactic level, all the isos involved in the interpretation of a dependent judgement.

4. Type theory with explicit coercions

In [7], the first author augmented the syntax of dependent types with explicit coercions:

$$M ::= \ldots \mid \mathsf{c}(M, \sigma, \sigma')$$

The typing rule for explicit coercions is:

$$\frac{\Gamma \vdash M : \sigma \qquad \Gamma \vdash \sigma \cong \sigma'}{\Gamma \vdash \mathsf{c}(M, \sigma, \sigma') : \sigma'}$$

where $\Gamma \vdash \sigma \cong \sigma'$ is a new judgement (even with respect to [7]). The typing judgements with explicit coercions can if needed be emphasized with a subscript $_e$ on the turnstyle.

We next describe the conversions. Besides equality conversions that concern all syntactic categories (contexts, substitutions, types, and terms), denoted as before with the symbol $=$, there are now as anticipated above so-called iso conversions for types (and types only), denoted with the symbol $\cong$. The type conversion axioms of Section 2 are now iso conversions. All other axioms are kept as equality conversions and adapted to type-check properly. And some more equality conversions that deal with coercions are also added.

$$\begin{array}{l}
\sigma[s][t] \cong \sigma[s \circ t] \\
\sigma[id] \cong \sigma
\end{array}$$

$$\begin{array}{l}
\mathsf{c}(1[M \cdot s], \sigma[\uparrow][M \cdot s], \sigma[s]) = M \\
\mathsf{c}(M[s][t], \sigma[s][t], \sigma[s \circ t]) = M[s \circ t]
\end{array}$$

$$\begin{array}{l}
\uparrow \circ (M \cdot s) = s \\
(s_1 \circ s_2) \circ s_3 = s_1 \circ (s_2 \circ s_3) \\
id \circ s = s \circ id = s \\
(M \cdot s) \circ t = \mathsf{c}(M[t], \sigma[s][t], \sigma[s \circ t]) \cdot (s \circ t) \\
\mathsf{c}(1[s], \sigma[\uparrow][s], \sigma[\uparrow \circ s]) \cdot (\uparrow \circ s) = s
\end{array}$$

$$\begin{array}{l}
\mathsf{c}(\mathsf{c}(M, \sigma_1, \sigma_2), \sigma_2, \sigma_3) = \mathsf{c}(M, \sigma_1, \sigma_3) \\
\mathsf{c}(M, \sigma, \sigma) = M \\
\mathsf{c}(M, \sigma_1, \sigma_2)[t] = \mathsf{c}(M[t], \sigma_1[t], \sigma_2[t])
\end{array}$$

(Again, these stand for typed axioms $\Gamma \vdash \sigma[s][t] \cong \sigma[s \circ t]$, etc.... .) There is a coercion from equal types to isomorphic types:

$$\frac{\Gamma \vdash \sigma = \sigma'}{\Gamma \vdash \sigma \cong \sigma'}$$

The congruence rules for equalities follow the structure of all syntactic categories, while for isomorphisms they are limited to the structure of types (and contexts – we omit this here for concision). There is no such thing as $M \cong N$ – instead, type isomorphisms are encapsulated in the explicit coercions that adjust the types so as to maintain the invariant that equal terms have the same type.

In Figure 1, we list all the rules for iso conversions in the core syntax.

One also defines a stripping function that removes these coercions:

$$|1| = 1 \qquad |M[s]| = |M|[|s|] \qquad |\mathsf{c}(M, \sigma, \sigma')| = |M|$$

$$|M \cdot s| = |M| \cdot |s| \quad \ldots \quad |\sigma[s]| = |\sigma|[|s|]$$

Note that the stripped version of the equations of this section are the equations of Section 2 (also collapsing $\cong$ to $=$).

The coherence theorem is the following (for an arbitrary target locally cartesian closed category):

1. For any two derivation trees π_1, π_2 of $\Gamma \vdash_e \sigma_1 \cong \sigma_2$, we have $[\![\pi_1]\!] = [\![\pi_2]\!]$.
2. If $\Gamma \vdash_e M_1 : \sigma$, $\Gamma \vdash_e M_2 : \sigma$, and $|M_1| = |M_2|$, then $[\![M_1]\!] = [\![M_2]\!]$ (and likewise for two substitutions s_1, s_2 with the same source and target and the same stripping).

As for 1. the typical situation is comparing the two possible derivations of

$$\sigma[s_1][s_2][s_3] \cong \sigma[s_1 \circ (s_2 \circ s_3)]$$

(a variant of Mac Lane's pentagon!).

The proof in [7] was using the rewriting technique (cf. Section 7). A new proof via strictification will be given in Section 8.

We end the section with a terminological convention, and a remark.

Convention. In the rest of the paper, we shall refer to the syntax of Section 3 (resp. of this section) as the *original* (resp. *explicit*) syntax. Therefore, "explicit" will refer to explicit coercions (not to explicit substitutions, which are quite standard in formal treatments of type theory).

Figure 1: Iso conversion rules (core syntax)

$$\frac{\Gamma \vdash \sigma = \sigma'}{\Gamma \vdash \sigma \cong \sigma'} \qquad \frac{\Gamma \vdash \sigma \cong \sigma'}{\Gamma \vdash \sigma' \cong \sigma} \qquad \frac{\Gamma \vdash \sigma \cong \sigma' \quad \Gamma \vdash \sigma' \cong \sigma''}{\Gamma \vdash \sigma \cong \sigma''} \qquad \frac{\Gamma' \vdash s : \Gamma \quad \Gamma \vdash \sigma \cong \sigma'}{\Gamma' \vdash \sigma[s] \cong \sigma'[s]}$$

$$\frac{\Gamma'' \vdash t : \Gamma' \qquad \Gamma' \vdash s : \Gamma \qquad \Gamma \vdash \sigma\ \mathtt{type}}{\Gamma'' \vdash \sigma[s][t] \cong \sigma[s \circ t]} \qquad \frac{\Gamma \vdash \sigma\ \mathtt{type}}{\Gamma \vdash \sigma[id] \cong \sigma}$$

Judgements versus derivations of judgements. A virtue of the explicit syntax is that a judgement, if provable, admits a unique derivation, if one does not count the derivations of type (or context) isomorphisms. (Well, this must be taken with a piece of salt, as one would have to provide additional information in the terms like the intermediate context in a composition of substitutions, etc...). Therefore the interpretation function $[\![\_]\!]$, that is a priori defined by induction on the derivations of judgements, is actually well-defined as a function of the judgements of the explicit syntax (using property (1) above). This is also true of the semantics of original judgements in an LCCC: with any derivation of a judgement one can associate a judgement of the explicit syntax, and property (2) tells us that its interpretation does not depend on the chosen derivation. Thus it is safe to speak about the interpretation of judgements (in the original or in the explicit syntax), as a by-product of the coherence theorem.

5. Fibrations with comprehension

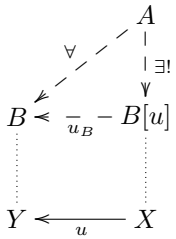
For the reader's convenience, we start from the beginning, recalling the definition of fibration, and then of comprehension category. We pay particular attention to the definition of non-strict and strict morphisms between them, respectively. We then "lift" the interpretation of Section 3 to this more general setting. Finally we sketch the construction of the strict and non-strict classifying categories.

5.1. Categories of fibrations with comprehension

We recall briefly the definition of a Grothendieck fibration. Let $p : \mathbb{E} \to \mathbb{B}$ be a functor. A morphism $f : A \to B$ of $\mathbb{E}$ is called cartesian if for every C and $h : C \to B$ such that $p(h) = p(f)$ there exists a unique $g : C \to B$ such that $p(g) = id$ and $h = f \circ g$ The functor is called a fibration if:

1. for every $u : X \to Y$ in $\mathbb{B}$ and every B over Y (i.e., $p(B) = Y$), there exists a cartesian morphism u_B with codomain B such that $p(u_B) = u$, called cartesian lifting of u at B. We (opportunely) write $B[u]$ for the domain of u_B;
2. the composition of two cartesian morphisms is again cartesian.

We call $\mathbb{B}$ the base category, and the objects of $\mathbb{E}$ local objects: if $p(A) = X$, we say that A lies over X. Schematically, we thus have:



where we use dash arrows for morphisms in $\mathbb{E}$ and vertical dots to record that B is above Y and similarly that $B[u]$ is above X.

Note that in a fibration cartesian morphisms are in fact hypercartesian, i.e. they satisfy the stronger condition that for every C, $u : p(C) \to p(A)$, and $h : C \to B$ such that $p(h) = p(f) \circ u$ there exists a unique $g : C \to B$ such that $p(g) = u$ and $h = f \circ g$. Morphisms having this property are stable under composition, making condition 2 then superfluous.

The fibration is *cloven* when choices $u_B : B[u] \to B$ have been made for every u, B. We shall always assume our structures to be cloven.

The fibration is called *split* when composites of chosen cartesian morphisms are chosen cartesian morphisms, i.e., when

$$u : X \to Y, v : Y \to Z, p(C) = Z \Rightarrow C[v][u] = C[v \circ u] \text{ and } v_C \circ u_{C[v]} = (v \circ u)_C$$

(in general, by definition of a fibration, we have that $v_C \circ u_{C[v]}$ is cartesian, hence "equal" to $(v \circ u)_C$ only up to iso). Also, one requires, for all A over X, $A[id] = A$ and $(id_X)_A = id_A$.

A morphism between two fibrations $p_1 : \mathbb{E}_1 \to \mathbb{B}_1$ and $p_2 : \mathbb{E}_2 \to \mathbb{B}_2$ is a pair $F = (F_t, F_b)$ of functors such that $p_2 \circ F_t = F_b \circ p_1$ and F_t maps cartesian morphisms to cartesian morphisms. It is called *strict* if it maps the chosen cartesian morphisms of p_1 to chosen cartesian morphisms of p_2.

Basic examples of fibrations are ($\to$ stands for the category with two objects and one non-identity arrow):

- $\mathtt{dom} : \mathbb{C}^{\to} \to \mathbb{C}$ is a split fibration for any category $\mathbb{C}$, as well as its restrictions $\mathtt{dom}_X : \mathbb{C}/X \to \mathbb{C}$ (for every object X of $\mathbb{C}$).
- $\mathtt{cod} : \mathbb{C}^{\to} \to \mathbb{C}$ is a fibration if and only if $\mathbb{C}$ has pullbacks (and hence this fibration is usually non-split).

A comprehension structure [15] on a fibration $p : \mathbb{E} \to \mathbb{B}$, where $\mathbb{B}$ has pullbacks and has a terminal object (i.e., has all finite limits), is given by a morphism from p to $\mathtt{cod}$ over $\mathbb{B}$, i.e., a functor $\mathcal{P} : \mathbb{E} \to \mathbb{B}^{\to}$ such that $p = \mathtt{cod} \circ \mathcal{P}$, that moreover preserves cartesian morphisms (i.e., that sends cartesian morphisms to pullback squares). Thus for $f \in \mathbb{E}[A, B]$, $\mathcal{P}(f)$ is a morphism $(t, s) : \mathcal{P}(A) \to \mathcal{P}(B)$ such that $s = p(f)$ and $\mathcal{P}(B) \circ t = s \circ \mathcal{P}(A)$ is a pullback square. We shall usually write $t = \mathcal{P}(f)$. For A in $\mathbb{E}$ over X, we write (X, A) for the domain of $\mathcal{P}(A)$:

$$\mathcal{P}(A) : (X, A) \to X \ .$$

We call *comprehension category* such a pair $(p, \mathcal{P})$ of a fibration equipped with a comprehension structure. It is called full if $\mathcal{P}$ is full and faithful.

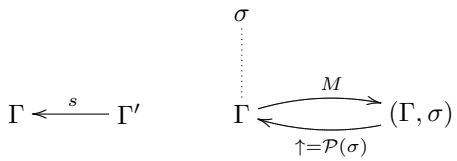
Morphisms between comprehension structures $(p_1, \mathcal{P}_1)$, $(p_2, \mathcal{P}_2)$ are morphisms $F = (F_t, F_b)$ between the underlying fibrations that commute with the comprehension structure up to isomorphism, i.e., there is a natural equivalence between $\tilde{F} \circ \mathcal{P}_1$ and $\mathcal{P}_2 \circ F$ (where $\tilde{F} : \mathtt{cod}_1 \to \mathtt{cod}_2$ is induced by F_b). We call

a morphism strict if the latter two composites are equal and if it is strict as a morphism between the underlying fibrations.

Obviously, the fibration `cod` has a trivial comprehension structure, given by the identity functor.

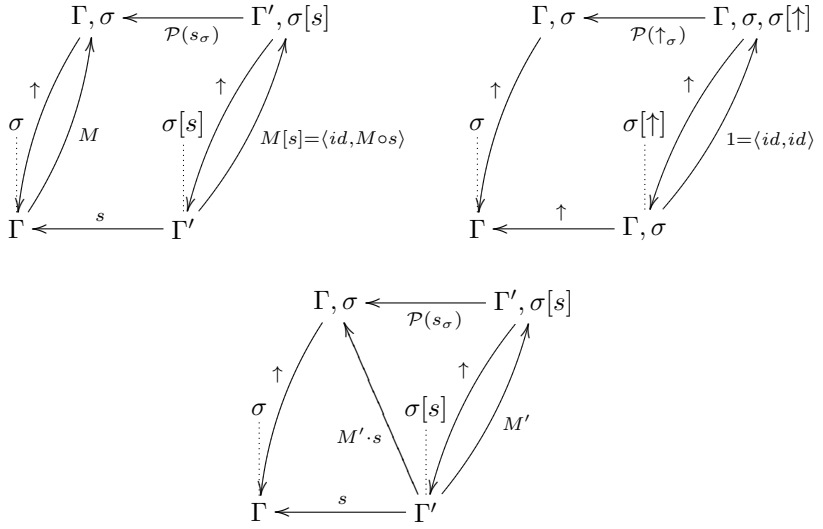
5.2. Interpretation of the core syntax in comprehension categories

We can now reformulate the interpretation of Section 3 more generally in any comprehension category. In this interpretation, types are no longer base morphisms but *local objects*, while the other syntactic categories are interpreted as before. Also, $\uparrow$ is no longer assimilated to σ but is rather defined as $\mathcal{P}(\sigma)$.



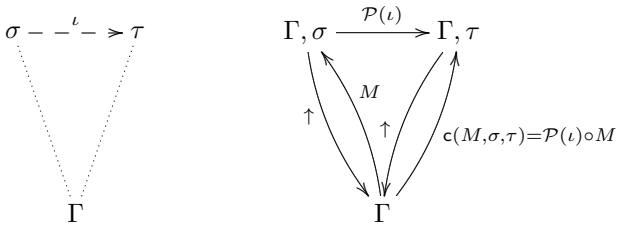
Given $\Gamma' \vdash s : \Gamma$ and $\Gamma \vdash \sigma\ \mathtt{type}$, $\sigma[s]$ is interpreted as the domain of s_σ, whence our notation for cartesian morphisms.

The constructs $M[s]$, 1, and $M' \cdot s$ are interpreted as follows (generalising the diagrams of Section 3, and still involving pullbacks in the base category):



The new judgement $\Gamma \vdash \sigma \cong \tau$ is interpreted as an isomorphism ι in $\mathbb{E}$ over (the interpretation of) Γ (that is, $p(\iota) = id$), and the coercions are interpreted by

composing with $\mathcal{P}(\iota)$:



Note that we have indeed $\uparrow \circ \mathsf{c}(M, \sigma, \tau) = (\uparrow \circ \mathcal{P}(\iota)) \circ M = \uparrow \circ M = id$. And, last but not least, the terminal object in the base category serves to interpret the empty context.

Typically, the type isomorphism $\sigma[s][t] \cong \sigma[s \circ t]$ is interpreted using the universal property of $(s \circ t)_\sigma$.

Finally, equality judgements such as $\Gamma \vdash \sigma = \tau$ etc... are interpreted as equality, i.e. the interpretations of σ and τ coincide in the model.

When the comprehension category is split (i.e., when its underlying fibration is split), then all type isomorphisms are interpreted as identity morphisms, so there is no need to distinguish two levels of "equalities", and we can interpret the syntax of Section 2 rightaway.

We note that the interpretation given above does not make use of the morphisms of $\mathbb{E}$. It is therefore natural to limit attention to full comprehension categories (where all morphisms of $\mathbb{E}$ are "already there" in $\mathbb{B}$).

5.3. Classifying categories

We end the section by sketching the constructions of the two classifying comprehension categories $\mathtt{Synt}^e$ and $\mathtt{Synt}$ corresponding to the explicit and the original syntax (cf. Section 1). The ingredients of the two categories are built in the same way:

- Base objects are equivalence classes of contexts.
- A base morphism from a base object with representative Γ to a base object with representative Δ is an equivalence class of a substitution $\Gamma \vdash s : \Delta$.
- Local objects are equivalence classes of types. The fibration functor p sends a class with representative σ well-typed over Γ to the class of Γ.
- A morphism from the local object with representative σ well-formed over Γ to the local object with representative τ well-formed over Δ is a pair $([s], [t])$ of equivalence classes, where $\Gamma \vdash s : \Delta$, $(\Gamma, \sigma) \vdash t : (\Delta, \tau)$, and $\uparrow \circ t = s \circ \uparrow$ is provable.

Above, equivalence classes are all meant as the congruence closure of all rules listed in Section 2 (case $\mathtt{Synt}$) and all equality (not iso!) rules in Section 4 (case $\mathtt{Synt}^e$). We check in some detail that $\mathtt{Synt}^e$ is a (*non-split*) fibration. This

verification can be viewed as a "categorification" of the verification that `Synt` is a split fibration.

That the data above provide two categories and a functor p between them should be obvious, setting $p([s],[t]) = [s]$. What the comprehension functor should be is also clear: for $\Gamma \vdash \sigma\ \mathtt{type}$, we set $\mathcal{P}([\sigma]) = [\uparrow]$ (where $\Gamma, \sigma \vdash \uparrow : \Gamma$).

Let $\Gamma' \vdash s : \Gamma$ and $\Gamma \vdash \sigma\ \mathtt{type}$. We show that

$$s_\sigma = \mathsf{c}(1, \sigma[s][\uparrow], \sigma[s \circ \uparrow]) \cdot (s \circ \uparrow)$$

exhibits (s, s_σ) as representative of a cartesian morphism, which is mapped by $\mathcal{P}$ to a pullback square. We first prove the second part of the claim. Let $\Delta \vdash t : \Gamma, \sigma$ and $\Delta \vdash u : \Gamma'$ such that $\Delta \vdash \uparrow \circ t = s \circ u$. We first show uniqueness. Let $\Delta \vdash v : \Gamma', \sigma[s]$ be a mediating morphism. We have

$$\Delta \vdash v = \mathsf{c}(1[v], \sigma[s][\uparrow][v], \sigma[s][\uparrow \circ v]).(\uparrow \circ v)\ .$$

We claim that the assumption $t = s_\sigma \circ v$ implies

$$1[t] = \mathsf{c}[1[v], \sigma[s][\uparrow][v], \sigma[\uparrow][t])\ .$$

Indeed, we have

$$1[t] = 1[s_\sigma \circ v] = \mathsf{c}(1[s_\sigma][v], \sigma[\uparrow][s_\sigma][v], \sigma[\uparrow][s_\sigma \circ v])\ .$$

We then compute (setting $1' = \mathsf{c}(1, \sigma[s][\uparrow], \sigma[s \circ \uparrow])$):

$$1[s_\sigma] = 1[1'.(s \circ \uparrow)] = \mathsf{c}(1', \sigma[s \circ \uparrow], \sigma[\uparrow][s_\sigma]) = \mathsf{c}(1, \sigma[s][\uparrow], \sigma[\uparrow][s_\sigma])\ .$$

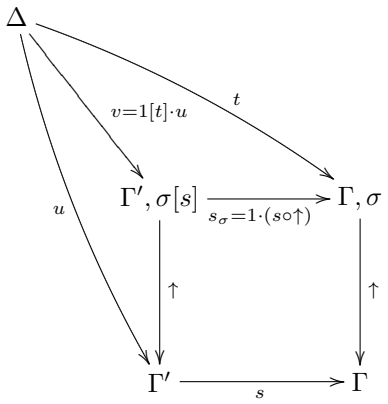
It follows that $1[s_\sigma][v] = \mathsf{c}(1[v], \sigma[s][\uparrow][v], \sigma[\uparrow][s_\sigma][v])$, from which the claim follows. From the claim, and from the other assumption $\uparrow \circ v = u$, we obtain a definition of v:

$$v = \mathsf{c}(1[v], \sigma[s][\uparrow][v], \sigma[s][\uparrow \circ v]).(\uparrow \circ v) = \mathsf{c}(1[t], \sigma[\uparrow][t], \sigma[s][u]) \cdot u$$

(it is reassuring to see that this is well defined since one can prove $\sigma[\uparrow][t] \cong \sigma[\uparrow \circ t] = \sigma[s \circ u] \cong \sigma[s][u]$). We now prove that this v satisfies the two equations. We have $\uparrow \circ v = \uparrow \circ (\ldots \cdot u) = u$, and

$$\begin{array}{rcll} s_\sigma \circ v & = & (1' \cdot (s \circ \uparrow)) \circ v & = \mathsf{c}(1'[v], \sigma[s \circ \uparrow][v], \sigma[(s \circ \uparrow) \circ v]) \cdot (s \circ \uparrow) \circ v \\ & & & = \mathsf{c}(1[v], \sigma[s][\uparrow][v], \sigma[(s \circ \uparrow) \circ v]) \cdot (s \circ \uparrow) \circ v \\ & & & = \mathsf{c}(1[t], \sigma[\uparrow][t], \sigma[(s \circ \uparrow) \circ v]) \cdot (s \circ \uparrow) \circ v \\ & & & = \mathsf{c}(1[t], \sigma[\uparrow][t], \sigma[\uparrow \circ t]) \cdot (\uparrow \circ t) \\ & & & = t \end{array}$$

The picture below summarises the data of the pullback (omitting the coercions).



To check that s_σ is cartesian, we just instantiate Δ, u as (Γ', σ') and $\uparrow$.

It is an instructive exercise to check that cartesian morphisms compose. Specifically:

$$(s \circ t)_\sigma = ((\mathsf{c}(1, \sigma[s][t][\uparrow], \sigma[s \circ t][\uparrow]) \cdot \uparrow) \circ (s_\sigma \circ t_{\sigma[s]}) \ .$$

One can also (tediously) verify that the interpretation of a type σ in this model is (the equivalence class of) σ, and similarly for contexts and substitutions, and that the interpretation of $\Gamma \vdash M : \sigma$ is the equivalence class of $\mathsf{c}(M, \sigma, \sigma[id]) \cdot id$. It follows that we can reformulate the interpretation function of the syntax into some arbitrary fibration with comprehension as the unique strict morphism from the classifying comprehension category (as anticipated in the introduction).

The same remarks apply if we revert to the original syntax (with its associated classifying split comprehension category), and to its interpretation in a split comprehension category.

6. The Giraud-Bénabou construction

The construction recalled in this section goes back to at least [11]. Let $p : \mathbb{E} \to \mathbb{B}$ be a (non-split) fibration. We define a new fibration (on the same basis)

$$R(p : \mathbb{E} \to \mathbb{B}) = p' : \mathbb{E}' \to \mathbb{B}$$

as follows:

- $Obj(\mathbb{E}')$ consists of the pairs (X, ϕ), where $\phi : \mathtt{dom}_X \to p$, where $\mathtt{dom}_X$ is the domain fibration $\mathbb{B}/X \to \mathbb{B}$ recalled in the previous section. We require ϕ to be a morphism of fibrations, i.e. to preserve cartesian morphisms, which amounts to say that $\phi(f : g_1 \to g_2)$ is cartesian for every morphism in $\mathbb{B}/X$. We set $p'(X, \phi) = X$.

- $\mathbb{E}'[(X,\phi),(Y,\psi)]$ consists of the pairs (t,μ) where $t \in \mathbb{B}[X,Y]$ and μ is a natural transformation from ϕ to $\psi \circ \mathtt{dom}_t$ over $\mathbb{B}$, i.e. $\mu_s : \phi(s) \to \psi(t \circ s)$ and $p(\mu_s) = id_Z$ for all $s : Z \to X$.

We check that p' is a fibration: for $t : X \to Y$ we set $\phi[t] = \phi \circ \mathtt{dom}_t$, i.e. $\phi[t](s) = \phi(t \circ s)$ for all $s : Z \to X$, and $t_\phi = (t, id)$. Moreover, p' is split, by the associativity of composition in $\mathbb{B}$.

If moreover p carries a comprehensions structure $\mathcal{P}$, then we define a comprehension structure on p' as follows:

$$\mathcal{P}'(\phi) = \mathcal{P}(\phi(id)) \qquad \mathcal{P}'(t,\mu) = \mathcal{P}(\psi(t : t \to id) \circ \mu_{id})$$

The name Rp for the split fibration p' comes from the fact that it fits as right adjoint to the inclusion functor from the category of split fibrations on some fixed basis $\mathbb{B}$ and strict morphisms to the category of fibrations on the same basis and fibration morphisms (i.e. neither objects nor morphisms are strict), i.e., following the terminological conventions of the introduction, we have

$$(\subseteq\colon \mathbf{SFib_s}(\mathbb{B}) \to \mathbf{Fib}(\mathbb{B})) \dashv R\ .$$

A morphism (resp. strict morphism) from $p_1 : \mathbb{E}_1 \to \mathbb{B}$ to $p_2 : \mathbb{E}_2 \to \mathbb{B}$ is a functor $G : \mathbb{E}_1 \to \mathbb{E}_2$ which sends cartesian (resp. chosen cartesian) morphisms to cartesian (resp. chosen cartesian) morphisms.

But we are in fact interested in a fibration morphism $F : p \to Rp$ that is neither the unit nor the counit of this adjunction, which we now define. Let A a local object over X. We set

$$F(A) = (X,\phi), \text{ where } \phi(f : Y \to X) = A[f]\ .$$

For $f : A_1 \to A_2$, we set $F(f) = (u,\mu)$ where $u = p(f)$ and where, for every $v : X \to X_1 = p(A_1)$, μ_v is the unique morphism from $A_1[v]$ to $A_2[u \circ v]$ such that $(u \circ v)_{A_2} \circ \mu_v = (f \circ v_{A_1})$.

This functor is faithful since one can recover f from μ_{id}. Note as a consequence that if p_1 is a full comprehension category, then so is Rp_1. We also note that F is non-strict, both as a morphism of fibrations (since we have on one hand $F(A[g])(f) = A[g][f]$ and on the other hand $(F(A)[g])(f) = A[g \circ f]$), and as a morphism of comprehension categories ($A[id]$ versus A).

As a final remark, let us note that F is right inverse to the counit of the adjunction, up to the identification of A and $A[id]$, as the counit $\epsilon : RP \to p$ maps ϕ to $\phi(id)$.

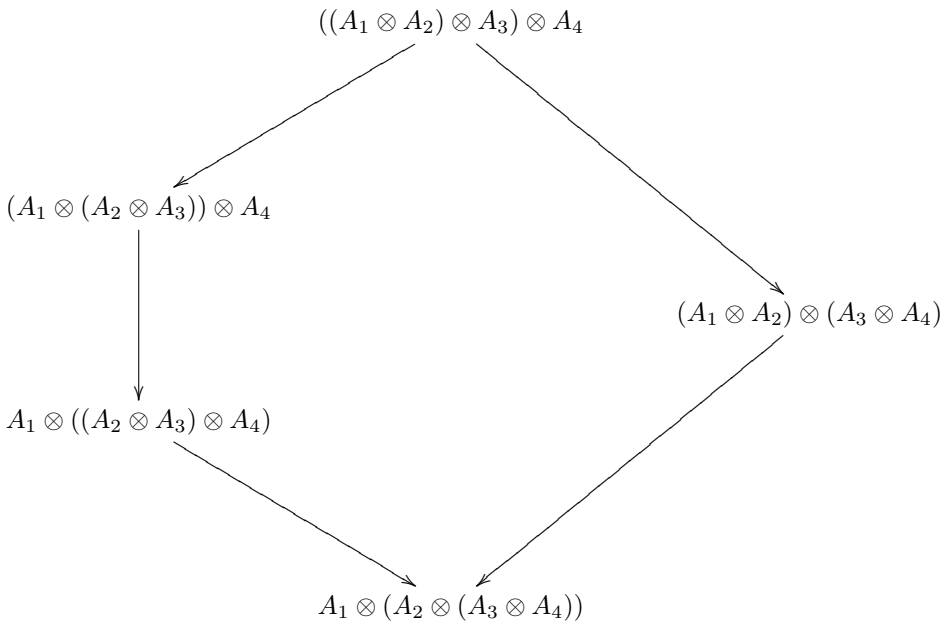
7. Intermezzo: coherence in monoidal categories

We review here coherence for monoidal categories, as a kind of warming up for the following section.

We know of two quite different ways of proving the coherence theorem for monoidal categories (see also the final remark of this section). The first method

uses a "categorification" of the Knuth-Bendix's (KB) lemma. It is is implicitly the path followed by Mac Lane [21], since he analyses, say, the pentagon in terms of a "diamond". As far as we know, the connection with Knuth-Bendix was first remarked by Huet in his lecture notes on categories [14].

We just recall here that the KB lemma for rewriting systems asserts that if a rewriting system is terminating and the critical pairs converge, then the system is confluent. The rewriting aspect of coherence can be seen in the pentagon diagram if one looks at the vertices as formal terms and reads the morphisms as rewriting steps:



and if one reads Mac Lane's pentagonal equation $(A_1 \otimes \alpha) \circ \alpha \circ (\alpha \otimes A_4) = \alpha \circ \alpha$ (where each α is an appropriate instantiation of an associativity isomorphism) as witnessing the confluence of the critical pair out of $((A_1 \otimes A_2) \otimes A_3) \otimes A_4$. Then, decorating the proof of KB's lemma with such witnesses, one gets that any two composites of morphisms of the form $[id] \otimes \kappa \otimes [id]$ (where the brackets mean "optional", and where κ is one of the three basic canonical isomorphisms α, λ, ρ corresponding to the three monoid laws – see below) from a fixed source to a fixed target are equal. One then has to prove the same statement allowing now compositions with the inverses, but this is easy stuff, remarking by induction that any composition of these morphisms or their inverses can be organised as a composition of $([id] \otimes \kappa \otimes [id])$'s followed by a composition of $([id] \otimes \kappa^{-1} \otimes [id])$'s.

We shall not say more on this, since the emphasis in this paper is on the other method. We refer the reader to [7] where this method was applied to show the coherence properties stated in Section 4. (Incidentally, that paper made a quite intensive use of string diagrams.)

The second method is based on a "representation theory" type of argument

[18] (see also [19, Chapter 1], which we follow and expand more closely here). Before explaining and decomposing this proof in detail, we shall set the scene in order to formulate the coherence theorem in a language akin to that of type theory. We start from a set X, considered as a set of "object variables" (they are meant to be interpreted as objects in some category). We define two sets of object terms and morphism terms, respectively:

$$T ::= x \ \ (\text{where } x \in X) \mid I \mid T \otimes T$$
$$M ::= \alpha \mid \lambda \mid \rho \mid \alpha^{-1} \mid \lambda^{-1} \mid \rho^{-1} \mid M \circ M \mid id \mid M \otimes M$$

We actually only consider *well-typed* morphism terms which are the ones accepted by the following typing rules, involving judgements of the form $M : T_1 \to T_2$:

$$\frac{}{\alpha : (T_1 \otimes T_2) \otimes T_3 \to T_1 \otimes (T_2 \otimes T_3)} \qquad \frac{}{\lambda : I \otimes T \to T} \qquad \frac{}{\rho : T \otimes I \to T}$$

$$\frac{}{\alpha^{-1} : T_1 \otimes (T_2 \otimes T_3) \to (T_1 \otimes T_2) \otimes T_3} \qquad \frac{}{\lambda^{-1} : T \to I \otimes T} \qquad \frac{}{\rho^{-1} : T \to T \otimes I}$$

$$\frac{}{id : T \to T} \qquad \frac{M_1 : T_1 \to T_2 \quad M_2 : T_2 \to T_3}{M_2 \circ M_1 : T_1 \to T_3} \qquad \frac{M_1 : T_1 \to T_1' \quad M_2 : T_2 \to T_2'}{M_1 \otimes M_2 : T_1 \otimes T_2 \to T_1' \otimes T_2'}$$

One quotients the set of morphism terms by the laws of categories and of bifunctors, and by Mac Lane's coherence equations. What one then obtains is the free monoidal category $\mathit{Free}(X)$ over X i.e., for every *function* $\rho : X \to \mathbb{C}$ (mapping each x to an object of a monoidal category $\mathbb{C}$) there exists a unique strict monoidal functor $[\![\_]\!]^\rho_{\mathbb{C}} : \mathit{Free}(X) \to \mathbb{C}$ that extends it.

The coherence theorem asserts that for any two terms $M, M' : T \to T'$ of the *same type* we have $[\![M]\!]^\rho_{\mathbb{C}} = [\![M']\!]^\rho_{\mathbb{C}}$ (for any monoidal category, and any valuation).

Towards its proof, we make some observations (cf. items II, I, and III of Section 1).

A. Let us see what $[\![\_]\!]^\rho_{\mathbb{C}}$ looks like when $\mathbb{C}$ is strict. We define a "stripping" function $X \mapsto |X|$ from object terms to words over X defined by forgetting the parentheses and the I's:

$$|T_1 \otimes T_2| = |T_1||T_2| \qquad |I| = \epsilon \ \ (\text{the empty word})$$

We have the following properties (for the second and the third one, $\mathbb{C}$ is supposed strict):

- if $M : T \to T'$ is a well-typed term, then $|T| = |T'|$;
- if $|T| = |T'|$ then $[\![T]\!]^\rho_{\mathbb{C}} = [\![T']\!]^\rho_{\mathbb{C}}$;
- for any well-typed term M, one has $[\![M]\!]^\rho_{\mathbb{C}} = id$.

B. Let us now revert to monoidal categories. Let $(F, \beta) : \mathbb{C} \to \mathbb{C}'$ be a monoidal functor between monoidal categories (i.e., $\beta : FC_1 \otimes FC_2 \to F(C_1 \otimes C_2)$ are natural isos satisfying the obvious compatibility conditions). The freeness of $Free(X)$ "lives" in the category $\mathbb{M}onCat_s$ of monoidal categories and *strict* monoidal functors (i.e. those for which $\beta = id$, and hence $F\alpha = \alpha$, $F\lambda = \lambda$, and $F\rho = \rho$). Hence we cannot directly relate $[\![\_]\!]^\rho_{\mathbb{C}}$ and $[\![\_]\!]^\rho_{\mathbb{C}'}$. But the following "glueing property" can be proved: for all T there exists an isomorphism $\gamma_{\rho,T} : F([\![T]\!]^\rho_{\mathbb{C}}) \to [\![T]\!]^{F\rho}_{\mathbb{C}'}$, such that for every $M : T \to T'$ the following coherence equation holds:

$$\gamma_{\rho,T'} \circ F[\![M]\!]^\rho_{\mathbb{C}} = [\![M]\!]^{F\rho}_{\mathbb{C}'} \circ \gamma_{\rho,T} \ .$$

The isomorphisms $\gamma_{\rho,T}$ are defined by induction on the structure of T (the "types"), and then the coherence equation is proved by induction on the structure of the morphism terms M.

C. Finally, the most crucial part of Joyal-Street's proof relies on the existence of a faithful monoidal functor $F : \mathbb{C} \to \mathbb{C}^s$, where $\mathbb{C}^s$ is a strict monoidal category associated with $\mathbb{C}$ (see below). In fact, F is a monoidal equivalence, but we do not need it.

If in (B) we take F to be the faithful functor from $\mathbb{C}$ to $\mathbb{C}^s$ (given in (C)), we see that if $M_1, M_2 : T \to T'$, then (using (A)):

$$F[\![M_1]\!]^\rho_{\mathbb{C}} = \gamma^{-1}_{\rho,T'} \circ [\![M_1]\!]^{F\rho}_{\mathbb{C}'} \circ \gamma_{\rho,T} = \gamma^{-1}_{\rho,T'} \circ \gamma_{\rho,T} = \gamma^{-1}_{\rho,T'} \circ [\![M_2]\!]^{F\rho}_{\mathbb{C}'} \circ \gamma_{\rho,T} = F[\![M_2]\!]^\rho_{\mathbb{C}}$$

from which $[\![M_1]\!]^\rho_{\mathbb{C}} = [\![M_2]\!]^\rho_{\mathbb{C}}$ follows by faithfulness.

It remains to define $\mathbb{C}^s$ and the faithful embedding. The objects of $\mathbb{C}^s$ are pairs (E, δ), where E is (just) an endofunctor on $\mathbb{C}$ and $\delta : E(A) \otimes B \to E(A \otimes B)$ is an iso, natural in A, B, and is required to commute appropriately with α. The tensor product is given by

$$(E, \delta) \otimes (E', \delta') = (E \circ E', E\delta' \circ \delta) \ .$$

This tensor product is strict: composition of functors is associative!

The functor $F : \mathbb{C} \to \mathbb{C}^s$ takes A to $(A \otimes \_, \alpha)$. It is faithful since one may recover $f : A \to B$ from $F(f)_I$.

As an aside, we note that a more conceptual view of this construction is obtained by switching dimensions by one and considering our monoidal category as a bicategory (let us still call it $\mathbb{C}$) with just one object $\star$. We may now consider the Yoneda embedding

$$\mathcal{Y} : \mathbb{C} \to \mathbf{Cat}^{\mathbb{C}^{op}} \ .$$

One observes that $\mathbb{C}^s$, again considered as a bicategory, is the full sub-bicategory of $\mathbf{Cat}^{\mathbb{C}^{op}}$ having $\mathcal{Y}(\star)$ as (unique) object, and that F is the corestriction of $\mathcal{Y}$ to $\mathbb{C}^s$. Under these glasses, the strictness of $\mathbb{C}^s$ follows from the following sequence of observations: **Cat** is a 2-category, that is, a strict bicategory, and therefore so is $\mathbf{Cat}^{\mathbb{C}^{op}}$, and therefore so is $\mathbb{C}^s$. In two words, Yoneda strictifies!

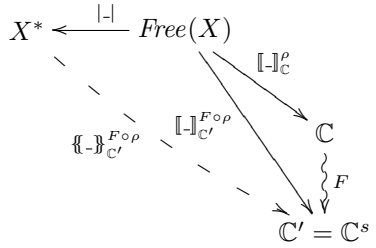
The constructions used in this proof can be organised in the following picture, which involves three large categories, of monoidal categories and monoidal functors, of monoidal categories and strict monoidal functors, and of strict monoidal categories and strict monoidal functors, respectively, with the following respective graphical conventions for their morphisms (cf. Section 1):

$$\mathbf{Mon} \rightsquigarrow \qquad \mathbf{Mon_s} \longrightarrow \qquad \mathbf{SMon_s} - - \succ$$

One checks easily that the free monoid X^*, viewed as a (discrete) category, is the free strict monoidal category over X, and that the stripping function can be recaptured as $[\![\_]\!]^{\subseteq}_{Free(X)}$ (where the valuation $\subseteq$ maps x to the object term x), extending the definition of $|\_|$ to morphisms by setting $|M| = id$, for all morphism terms. It then follows that the observations made above on $[\![\_]\!]^{\rho}_{\mathbb{C}}$ when $\mathbb{C}$ is strict amount to saying that :

$$[\![\_]\!]^{\rho}_{\mathbb{C}} = \{\!|\_|\!\}^{\rho}_{\mathbb{C}}$$

where $\{\!|\_|\!\}^{\rho}_{\mathbb{C}}$ is the unique strict monoidal functor from X^* to $\mathbb{C}$ extending ρ.



In this picture, the left triangle commutes exactly, while the right triangle commutes up to the isos $\gamma_{\rho,\_}$.

The philosophy behind all this is that the coherence information α is integrated in the definition of objects while moving from A to FA, and that the operation of tensor product is then carried out on these enriched objects. So instead of "tensoring first" and then "dealing with coherence", Joyal-Street's route "integrates coherence first" and then tensors.

We conclude this section with two remarks.

A more conceptual view of the glueing property. One can construct a category $[F]$, called the iso-glueing of (F, β) (we are back to a general monoidal functor between two monoidal categories), whose objects are triples $(A, A', \kappa : FA \to A')$ (with κ iso). One can prove that $[F]$ is a monoidal category, with

$$(A_1, A'_1, \kappa_1) \otimes (A_2, A'_2, \kappa_2) = (A_1 \otimes A_2, A'_1 \otimes A'_2, (\kappa_1 \otimes \kappa_2) \circ \beta^{-1})$$

and that the two projections from $[F]$ to $\mathbb{C}$ and $\mathbb{C}'$ are strict monoidal. Then the isos $\gamma_{\rho,\_}$ arise as the third component of $[\![\_]\!]^{[\rho]}_{[F]}$, where $[\rho](x) = (\rho(x), F(\rho(x)), id)$:

$$[\![T]\!]^{[\rho]}_{[F]} = ([\![T]\!]^{\rho}_{\mathbb{C}}, [\![T]\!]^{F\circ\rho}_{\mathbb{C}'}, \gamma_{\rho,T})$$

and the coherence conditions are wired in the monoidal structure of $[F]$.

Yet another proof of coherence. One can find in [4] a somewhat similar proof of coherence for monoidal categories. There, the rôle of $\mathbb{C}^s$ is played by N^N (where N is a set of normal forms for the free monoid over X – in bijection with X^*), considered as a discrete strict monoidal category, where the monoidal product is function composition. The object part of the unique morphism $[\![\_]\!] \in \mathbf{Mon_s}[\mathit{Free}(X), N^N]$ features normalisation by evaluation for monoids (as a matter of fact, the authors of [4] synthesised their proof of coherence from normalisation by evaluation). Their argument also involves a natural isomorphism (in their case, between $n \otimes T$ and $[\![T]\!](n)$), which is exploited in the same way as above to establish coherence.

8. A new proof of coherence for dependent types

We shall get our new proof of coherence of the interpretation of dependent type theory in comprehension categories by following the very path that we made explicit in the previous section, replacing "monoidal categories" with "comprehension categories".

In Section 5.3 we have constructed the "initial" objects $\mathtt{Synt}$ and $\mathtt{Synt}^e$ in the categories of split comprehension categories and comprehension categories, respectively (with strict morphisms in both categories). It would be more exact to say that these categories are free over the underlying dependent signature (which we have left unspecified).

Following the conventions of Sections 1 and 7, we shall use $[\![\_]\!]$ (resp. $\{\!\{\_\}\!\}$) to denote the interpretation function of the explicit (resp. original) syntax in a comprehension (resp. split comprehension) category. It follows from the remarks made in Section 5 that the interpretation of the explicit syntax in a split comprehension category q factors through the interpretation of the original syntax, by first collapsing the coercion information:

$$[\![\Gamma]\!] = \{\!\{|\Gamma|\}\!\} \qquad [\![\Gamma \vdash \sigma]\!] = \{\!\{|\Gamma| \vdash |\sigma|\}\!\} \qquad [\![M]\!] = \{\!\{|M|\}\!\} \qquad [\![s]\!] = \{\!\{|s|\}\!\}$$

We also have (still in the split case) that, for any derivation π of $\Gamma \vdash \sigma \cong \sigma'$, $[\![\sigma]\!] = [\![\sigma']\!]$ and $[\![\pi]\!] = id$. By defining the stripping of a derivation of type isomorphism to be id (proof-irrelevance!), we can extend the factorisation:

$$[\![\pi : (\Gamma \vdash \sigma \cong \sigma')]\!] \;=\; \{\!\{|\pi|\}\!\}\ (= \{\!\{id\}\!\} = id)\ .$$

Note that the stripping defines a morphism of fibrations that is not on a fixed base, i.e., if we write $\mathtt{Synt}$ and $\mathtt{Synt}^e$ explicitly as $p : \mathtt{Types} \to \mathtt{Contexts}$ and $p^e : \mathtt{Types}^e \to \mathtt{Contexts}^e$, then $|\_| = (|\_|_C, |\_|_T)$, with

$$\begin{array}{ccc} \mathtt{Types}^e & \xrightarrow{\;|\_|_T\;} & \mathtt{Types} \\ \Big\downarrow{\scriptstyle p^e} & & \Big\downarrow{\scriptstyle p} \\ \mathtt{Contexts}^e & \xrightarrow{\;|\_|_C\;} & \mathtt{Contexts} \end{array}$$

Also the interpretation morphisms $[\![\_]\!]$ and $\{\!\{\_\}\!\}$ are "base changing", i.e., come as pairs (F_t, F_b) of functors with $F_b \neq id$.

We now formulate the key glueing property in the setting of comprehension categories. Suppose that $p_1 : \mathbb{E}_1 \to \mathbb{B}_1$ and $p_2 : \mathbb{E}_2 \to \mathbb{B}_2$ are comprehension categories, and that $F = (F_t, F_b)$ is a morphism between them. Then one can build by induction on the structure of types (and hence of substitutions and terms – a difference with respect to the previous section) and contexts a family of isomorphisms

$$\gamma_\Gamma : F_b[\![\Gamma]\!]_1 \to [\![\Gamma]\!]_2 \qquad \gamma_\sigma : F_t[\![\sigma]\!]_1 \to [\![\sigma]\!]_2$$

such that if $\Gamma \vdash_e \sigma\ \mathtt{type}$, then $p_2(\gamma_\sigma) = \gamma_\Gamma$:

$$\begin{array}{ccc} F_t[\![\sigma]\!]_1 & \xrightarrow{\gamma_\sigma} & [\![\sigma]\!]_2 \\ \vdots & & \vdots \\ F_b[\![\Gamma]\!]_1 & \xrightarrow{\gamma_\Gamma} & [\![\Gamma]\!]_2 \end{array}$$

and such that for all substitutions $\Gamma' \vdash s : \Gamma$ and terms $\Gamma \vdash M : \sigma$ the following coherence equations hold:

$$\begin{array}{ccc} F_b[\![\Gamma']\!]_1 & \xrightarrow{F_b[\![s]\!]_1} & F_b[\![\Gamma]\!]_1 \\ \downarrow{\scriptstyle \gamma'_\Gamma} & & \downarrow{\scriptstyle \gamma_\Gamma} \\ [\![\Gamma']\!]_2 & \xrightarrow{[\![s]\!]_2} & [\![\Gamma]\!]_2 \end{array} \qquad \begin{array}{ccc} F_b[\![\Gamma]\!]_1 & \xrightarrow{F_b[\![M]\!]_1} & F_b[\![\Gamma, \sigma]\!]_1 \\ \downarrow{\scriptstyle \gamma_\Gamma} & & \downarrow{\scriptstyle \gamma_{\Gamma,\sigma}} \\ [\![\Gamma]\!]_2 & \xrightarrow{[\![M]\!]_2} & [\![\Gamma, \sigma]\!]_2 \end{array}$$

and finally such that for all derivations of isomorphisms $\pi : (\Gamma \vdash \sigma \cong \sigma')$, the following other coherence equations hold:

$$\begin{array}{ccc} F_t[\![\sigma]\!]_1 & \xrightarrow{F_t[\![\pi]\!]_1} & F_t[\![\sigma']\!]_1 \\ \vdots{\scriptstyle \gamma_\sigma} & & \vdots{\scriptstyle \gamma_{\sigma'}} \\ [\![\sigma]\!]_2 & \xrightarrow{[\![\pi]\!]_2} & [\![\sigma']\!]_2 \end{array}$$

Typically, here is how we define $\gamma_{\sigma[s]}$. Since $([\![s]\!]_1)_{[\![\sigma]\!]_1}$ is cartesian, so is $F_t(([\![s]\!]_1)_{[\![\sigma]\!]_1})$, and hence so is $\gamma_\sigma \circ F_t(([\![s]\!]_1)_{[\![\sigma]\!]_1})$ (isos are cartesian, and cartesian morphisms compose). By induction, we know that

$$p_2(\gamma_\sigma \circ F_t(([\![s]\!]_1)_{[\![\sigma]\!]_1}) = \gamma_\Gamma \circ F_t[\![s]\!]_1 = [\![s]\!]_2 \circ \gamma_{\Gamma'} \ .$$

Thus we have that

$$\begin{array}{lll} \gamma_\sigma \circ F_t((\llbracket s \rrbracket_1)_{\llbracket \sigma \rrbracket_1}) & \text{is cartesian over} & \llbracket s \rrbracket_2 \circ \gamma_{\Gamma'} \\ (\llbracket s \rrbracket_2)_{\llbracket \sigma \rrbracket_2} & \text{is cartesian over} & \llbracket s \rrbracket_2 \end{array}$$

Then it follows from the definition of hypercartesian morphisms and from the fact that $\gamma_{\Gamma'}$ is iso that there exists a unique iso from $F_t\llbracket \sigma[s] \rrbracket_1$ to $\llbracket \sigma[s] \rrbracket_2$, which we take as our $\gamma_{\sigma[s]}$, such that $p_2(\gamma_{\sigma[s]}) = \gamma_{\Gamma'}$ and $(\llbracket s \rrbracket_2)_{\llbracket \sigma \rrbracket_2} \circ \gamma_{\sigma[s]} = \gamma_\sigma \circ F_t((\llbracket s \rrbracket_1)_{\llbracket \sigma \rrbracket_1})$. The first equation establishes the first property required above for the coherent isos, while the second equality can be used to establish that the coherence equation holds for $\uparrow : \Gamma', \sigma[s] \to \Gamma'$.

We now can give the proof of coherence, by specialising p_2 to Rp_1. We use the notation of Section 6.

1. Let π_1, π_2 be two derivations of $\Gamma \vdash_e \sigma \cong \sigma'$. We have

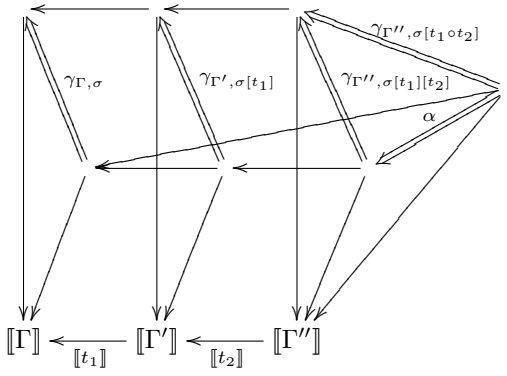
$$F_t\llbracket \pi_1 \rrbracket \;=\; \gamma_{\sigma'}^{-1} \circ \{\!|\, |\pi_1| \,|\!\} \circ \gamma_\sigma \;(\;=\; \gamma_{\sigma'}^{-1} \circ \gamma_\sigma\;) \;=\; F_t\llbracket \pi_2 \rrbracket$$

   and hence $\llbracket \pi_1 \rrbracket = \llbracket \pi_2 \rrbracket$ by faithfulness of F_t.
2. Let $\Gamma \vdash M_1 : \sigma$, $\Gamma \vdash M_2 : \sigma$ be such that $|M_1| = |M_2|$. We have (repeated from the introduction) (recall that $F_b = id$):

$$\llbracket M_1 \rrbracket = \gamma_{\Gamma,\sigma} \circ \{\!|\, |M_1| \,|\!\} \circ \gamma_\Gamma^{-1} = \gamma_{\Gamma,\sigma} \circ \{\!|\, |M_2| \,|\!\} \circ \gamma_\Gamma^{-1} = \llbracket M_2 \rrbracket$$

   and likewise for substitutions.

We end the section by linking this formal treatment of coherence with the discussion at the end of Section 3. Let $\Gamma' \vdash t_1 : \Gamma$, $\Gamma'' \vdash t_2 : \Gamma'$, and $\Gamma \vdash \sigma$ `type`. Let α be the isomorphism interpreting $\Gamma'' \vdash \sigma[t_1 \circ t_2] \cong \sigma[t_1][t_2]$ in the target LCCC $\mathbb{C}$. Let us make the simplifying assumptions that $|\Gamma| = \Gamma$, $|\Gamma'| = \Gamma'$, $|\Gamma''| = \Gamma''$, $|t_1| = t_1$, $|t_2| = t_2$, and $|\sigma| = \sigma$. In the picture



one part (referred to as the "lower part") reflects the adjustment between $\llbracket \sigma[t_1 \circ t_2] \rrbracket$ (obtained by a direct pullback along $t_1 \circ t_2$) and $\llbracket \sigma[t_1][t_2] \rrbracket$ (obtained by successive pullbacks). The top vertices and the arrows into them correspond

to what the strictification picture gives us as a third point of view. The three top vertices are, from left to right, $\{\!\{\Gamma, \sigma\}\!\}$, $\{\!\{\Gamma', \sigma[t_1]\}\!\}$, and $\{\!\{\Gamma'', \sigma[t_1][t_2]\}\!\} = \{\!\{\Gamma'', \sigma[t_1 \circ t_2]\}\!\}$ (the key being this equality). The two rectangles are also pullback squares and "reconcile" the lower part of the picture, as witnessed by the commutation of the triangle $\gamma_{\Gamma'',\sigma[t_1][t_2]} \circ \alpha = \gamma_{\Gamma'',\sigma[t_1 \circ t_2]}$.

9. Adding the type constructors

We consider successively (and independently) identity types (Section 9.1), and products and sums (Section 9.2). We recall from [16, 17] the additional structure needed to interpret them in a Grothendieck fibration. In Section 9.3, we indicate how to carry over our results to these extensions.

9.1. Identity types

In the variable-free syntax, identity types are described as follows:

$$\begin{array}{l} \sigma ::= \ldots \mid \mathtt{Id}_\sigma \\ M ::= \ldots \mid \mathtt{r} \mid \mathtt{J}(M) \end{array}$$

with the following rules:

$$\frac{\Gamma \vdash \sigma\ \mathtt{type}}{\Gamma, \sigma, \sigma[\uparrow] \vdash \mathtt{Id}_\sigma\ \mathtt{type}}$$

$$\frac{\Gamma \vdash \sigma\ \mathtt{type}}{\Gamma, \sigma \vdash \mathtt{r} : \mathtt{Id}_\sigma[1 \cdot id]} \qquad \frac{\Gamma, \sigma, \sigma[\uparrow], \mathtt{Id}_\sigma \vdash \tau\ \mathtt{type} \quad \Gamma, \sigma \vdash M : \tau[\mathtt{r} \cdot (1 \cdot id)]}{\Gamma, \sigma, \sigma[\uparrow], \mathtt{Id}_\sigma \vdash \mathtt{J}(M) : \tau}$$

and equality axioms:

$$\frac{\Gamma, \sigma, \sigma[\uparrow], \mathtt{Id}_\sigma \vdash \tau\ \mathtt{type} \quad \Gamma, \sigma \vdash M : \tau[\mathtt{r} \cdot (1 \cdot id)]}{\Gamma, \sigma \vdash \mathtt{J}(M)[\mathtt{r} \cdot (1 \cdot id)] = M : \tau[\mathtt{r} \cdot (1 \cdot id)]}$$

$$\frac{\Gamma, \sigma, \sigma[\uparrow], \mathtt{Id}_\sigma \vdash \tau\ \mathtt{type} \quad \Gamma, \sigma, \sigma[\uparrow], \mathtt{Id}_\sigma \vdash M : \tau}{\Gamma, \sigma, \sigma[\uparrow], \mathtt{Id}_\sigma \vdash \mathtt{J}(M[\mathtt{r} \cdot (1 \cdot id)]) = M : \tau}$$

$$\frac{\Gamma' \vdash s : \Gamma \quad \Gamma \vdash \sigma\ \mathtt{type}}{\Gamma', \sigma[s], \sigma[s][\uparrow] \vdash (\mathtt{Id}_\sigma)[1 \cdot ((1 \cdot (s \circ \uparrow)) \circ \uparrow)] = \mathtt{Id}_{\sigma[s]}}$$

The latter rule becomes an iso conversion rule in the explicit syntax, and coercions have to be inserted where appropriate. Finally, the extensional theory of identity types is defined by adding the following rule, called the reflection rule,

that collapses the notions of equality provided by identity types (called propositional equality) and by the conversion rules (called definitional equality):

$$\frac{\Gamma \vdash M_1 : \sigma \quad \Gamma \vdash M_2 : \sigma \quad \Gamma \vdash N : \mathtt{Id}_\sigma[M_1 \cdot (M_2 \cdot id)]}{\Gamma \vdash M_1 = M_2}$$

We now list the structure needed to interpret identity types, starting from a fibration $p : \mathbb{E} \to \mathbb{B}$. We shall need the following notions and notation. let X be an object of $\mathbb{B}$. We define the fiber $\mathbb{E}_X$ at X as the subcategory of $\mathbb{E}$ whose objects are the objects over X and whose morphisms are all the morphisms f of $\mathbb{E}$ between objects of $\mathbb{E}_X$ such that $p(f) = id$. These morphisms are called vertical (or local). Every morphism $s : Y \to X$ of $\mathbb{B}$ induces a functor $s^* = \_[s]$ from $\mathbb{E}_X$ to $\mathbb{E}_Y$.

1. We require the fibration p to have fibred terminal objects. This means one of the following two equivalent properties:
   (a) Each fibre $\mathbb{E}_X$ has a terminal object $\mathbf{1}_X$, and terminal objects are preserved by substitution, i.e., for every $s : X' \to X$, $\mathbf{1}_X[s]$ is terminal in $\mathbb{E}_{X'}$.
   (b) The functor $p : \mathbb{E} \to \mathbb{B}$, considered as a morphism from p to the trivial fibration $id : \mathbb{B} \to \mathbb{B}$, admits a fibred right adjoint $\mathbf{1}$, that is, there is an adjunction $p \dashv \mathbf{1}$ such that both the counit and the unit are vertical (which here means that the counit $\epsilon_1 : p(\mathbf{1}_X) \to X$ is the identity and that $p(\eta_1) = id$, where, for A over X, $\eta_1 : A \to \mathbf{1}_X$ gives us the unique morphism into $\mathbf{1}_X$ in the fiber.
2. One requires further that $\mathbf{1} : \mathbb{B} \to \mathbb{E}$ has an ordinary right adjoint $\{\_\}$, with unit η_2 and counit ϵ_2. This together with property (1) provides us with a comprehension structure on p, by setting $\mathcal{P}(A) = p(\epsilon_2)$, and thus setting $(X, A) = \{A\}$ for A over X. In this comprehension category, we can identify (for every Y, X in $\mathbb{B}$ and A in $\mathbb{E}_X$):
   (a) base morphisms from Y to (X, A);
   (b) pairs of a morphism s from Y to X and a section from Y to $(Y, A[s])$;
   (c) pairs of a morphism s from Y to X and a vertical morphism f from $\mathbf{1}_Y$ to $A[s]$ in $\mathbb{E}_Y$. We shall use the notation $\prec s, f \succ$ for the associated morphism from Y to (X, A).

   The equivalence between (a) and (b) holds in any comprehension category. The equivalence of (b) and (c) follows from the two adjunctions, and relies on the following commutation, for any morphism $g : \mathbf{1}_Y \to A$ in $\mathbb{E}$:

$$p((\epsilon_2)_A) \circ (\{g\} \circ (\eta_2)_X) = p(g) \ .$$

   This is proved easily by inserting explicitly the trivial inverse of ϵ_1, and then by naturality plus a triangular law to cancel out η_2 and ϵ_2.
   This commutation guarantees us in particular that in decomposing a morphism $t : Y \to (X, A)$ as $t = M.s_1$ as in (b) or as $t = \prec s_2, f \succ$ as in (c) we have $s_1 = s_2$, and that the correspondence between (b) and (c) boils down

to a correspondence between sections from Y to $(Y, A[s])$ and vertical morphism from $\mathbf{1}_Y$ to $A[s]$.
The following observation will be useful. We know from (1) that ϵ_1 is iso, hence $\mathbf{1}$ is full and faithful, and hence $(\eta_2)_X : X \to (X, \mathbf{1}_X)$ is iso (for all X). It is not difficult to see that the inverse of $(\eta_2)_X$ is $p((\epsilon_2)_{\mathbf{1}_X})$.

3. Consider, for fixed (X, A), the diagonal $\delta = \langle id, id \rangle$ from (X, A) to the pullback $(X, A) \times_X (X, A)$ of $\mathcal{P}(A)$ and $\mathcal{P}(A)$. For each such diagonal, we require $\delta^* = \_[\delta]$ to have a left adjoint $\mathtt{Eq}_A$, these adjunctions being related by the following Beck-Chevalley condition: for every $s : Y \to X$, the canonical morphism from $\mathtt{Eq}(B[1 \cdot s])$ to $(\mathtt{Eq}\, B)[t]$ is iso, where t is the morphism from $(Y, A[s]) \times_Y (Y, A[s])$ to $(X, A) \times_X (X, A)$ induced by $(1 \cdot s)$ and $(1 \cdot s)$.
Note that when $\mathcal{P} = id$, these adjoints are given by the postcomposition with δ.
4. Finally we require the notion of equality given in (3) to be strong in the following sense: for each X, A, B, the canonical morphism

$$\mathcal{P}(\delta_{\mathtt{Eq}\, B} \circ \eta) : (X, A, B) \to ((X, A) \times_X (X, A), \mathtt{Eq}\, B)$$

is iso. In the case where $B = \mathbf{1}_{X,A}$, this boils down to the requirement that

$$\prec \delta, \eta \succ : (X, A) \to ((X, A) \times_X (X, A), \mathtt{Eq}\, \mathbf{1})$$

is iso. Indeed, it is easy to check that

$$\mathcal{P}(\delta_{\mathtt{Eq}\, B} \circ \eta) = \{\delta_{\mathtt{Eq}\, B} \circ \eta\} = \prec \delta, \eta \succ \circ\, p(\epsilon_2)$$

(by taking transposes through the adjunction $\mathbf{1} \dashv \{\_\}$). But we already know (cf. above) that the projection $p(\epsilon_2) : (X, A, \mathbf{1}_{(X,A)}) \to (X, A)$ is iso. A consequence is that the projection morphism

$$\uparrow : ((X, A) \times_X (X, A), \mathtt{Eq}\, \mathbf{1}) \to (X, A) \times_X (X, A)$$

is an equaliser of the two projections π_1 and π_2 from $(X, A) \times_X (X, A)$ to (X, A) (since δ is an equaliser of these two morphisms).

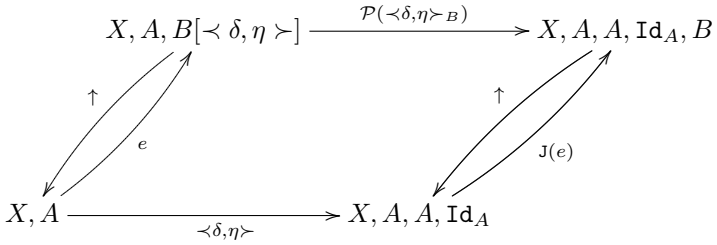
We shall say that p supports extensional identity types if it is equipped with all this structure.

We now show how to interpret identity types in a fibration that supports extensional identity types. Let $\Gamma \vdash \sigma\ \mathtt{type}$ be interpreted as A over X. Then one interprets

- $\mathtt{Id}_\sigma$ as $\mathtt{Eq}_A\, \mathbf{1}_{(X,A)}$, and
- $\mathtt{r}$ as $\prec id, \eta \succ : (X, A) \to ((X, A) \times_X (X, A), (\mathtt{Eq}\, \mathbf{1})[\delta])$, and
- if τ is interpreted as B and M is interpreted as e, then $\mathtt{J}(M)$ is interpreted as

$$\mathtt{J}(e) = \mathcal{P}(\prec \delta, \eta \succ_B) \circ e \circ \prec \delta, \eta \succ^{-1} \ .$$

This is illustrated below, where we have written X, A, A for $(X, A) \times_X (X, A)$:



We check a few properties:

- The interpretation of $\mathtt{J}(M)$ is a section:

$$(\uparrow \circ \mathcal{P}(\prec \delta, \eta \succ_B)) \circ e \circ \prec \delta, \eta \succ^{-1} = \prec \delta, \eta \succ \circ (\uparrow \circ e) \circ \prec \delta, \eta \succ^{-1} = id\ .$$

- Verification of $\mathtt{J}(M)[\mathtt{r} \cdot (1 \cdot id)] = M$. We have that $\mathtt{r} \cdot (1 \cdot id)$ gets interpreted as $\prec \delta, \eta \succ$. Then the verification boils down to the observation that $e = \langle id, \mathcal{P}(\prec \delta, \eta \succ_B) \circ e \rangle = \langle id, \mathtt{J}(e) \circ \prec \delta, \eta \succ \rangle$, the latter being precisely the interpretation of $\mathtt{J}(M)[\mathtt{r} \cdot (1 \cdot id)]$.

- Verification of $\mathtt{J}(M[\mathtt{r} \cdot (1 \cdot id)]) = M$. Let d be the interpretation of M. Then the interpretation of $M[\mathtt{r} \cdot (1 \cdot id)]$ is $\langle id, d \circ \prec \delta, \eta \succ \rangle$, and we have indeed

$$\mathcal{P}(\prec \delta, \eta \succ_B)) \circ \langle id, d \circ \prec \delta, \eta \succ \rangle \circ \prec \delta, \eta \succ^{-1} = d \circ \prec \delta, \eta \succ \circ \prec \delta, \eta \succ^{-1} = d\ .$$

- Reflection rule. Let a_1, a_2, b be the interpretations of M_1, M_2, N, respectively. Then $\mathtt{Id}_\sigma[M_1 \cdot (M_2 \cdot id)]$ is interpreted as $(\mathtt{Eq}\,\mathbf{1})[\langle a_1, a_2 \rangle]$, and we have:

$$\begin{array}{rcccl} a_1 & = & \pi_1 \circ (\langle a_1, a_2 \rangle \circ \uparrow) \circ b & = & (\pi_1 \circ \uparrow) \circ \mathcal{P}(\langle a_1, a_2 \rangle_{\mathtt{Eq}\,\mathbf{1}}) \circ b \\ & & & = & (\pi_2 \circ \uparrow) \circ \mathcal{P}(\langle a_1, a_2 \rangle_{\mathtt{Eq}\,\mathbf{1}}) \circ b \quad = \quad a_2 \end{array}$$

9.2. Π *and* Σ *types*

We add the following cases to the categories of types and terms (both in the explicit and in the original syntax).

$$\begin{array}{l} \sigma ::= \ldots \mid \Pi\sigma.\tau \mid \Sigma\sigma.\tau \\ M ::= \ldots \mid \lambda\sigma.M \mid MM \mid (M, N) \mid fst(M) \mid snd(M) \end{array}$$

We next give the added typing and conversion rules in the explicit syntax (the ones in the original syntax are obtained by stripping):

$$\frac{\Gamma, \sigma \vdash \tau\ \mathtt{type}}{\Gamma \vdash \Pi\sigma.\tau\ \mathtt{type}} \qquad \frac{\Gamma, \sigma \vdash \tau\ \mathtt{type}}{\Gamma \vdash \Sigma\sigma.\tau\ \mathtt{type}} \qquad \frac{\Gamma, \sigma \vdash M : \tau}{\Gamma \vdash \lambda\sigma.M : \Pi\sigma.\tau}$$

$$\frac{\Gamma \vdash M : \Pi\sigma.\tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau[\mathsf{c}(N, \sigma, \sigma[id]) \cdot id]} \qquad \frac{\Gamma \vdash M : \sigma \quad \Gamma \vdash N : \tau[\mathsf{c}(M, \sigma, \sigma[id]) \cdot id]}{\Gamma \vdash (M, N) : \Sigma\sigma.\tau}$$

$$\frac{\Gamma \vdash M : \Sigma\sigma.\tau}{\Gamma \vdash \mathit{fst}(M) : \sigma} \qquad \frac{\Gamma \vdash M : \Sigma\sigma.\tau}{\Gamma \vdash \mathit{snd}(M) : \tau[\mathsf{c}(\mathit{fst}(M), \sigma, \sigma[id]) \cdot id]}$$

Note that coercions are needed in the typing rules, since in an expression of the form $M \cdot s$, M must have a type of the form $\sigma[s]$ (cf. Section 4). In practice one may without loss of generality impose that $\sigma[id] = \sigma$ hold exactly, even in a non-split fibration, but stating the rules in full generality is the best way to avoid hiding this point!

$$(\Pi\sigma.\tau)[s] \cong \Pi\sigma[s].\tau[\mathsf{c}(1, \sigma[s][\uparrow], \sigma[s \circ \uparrow) \cdot (s \circ \uparrow)]$$
$$(\Sigma\sigma.\tau)[s] \cong \Sigma\sigma[s].\tau[\mathsf{c}(1, \sigma[s][\uparrow], \sigma[s \circ \uparrow) \cdot (s \circ \uparrow)]$$

$$(\lambda\sigma.M)N = M[\mathsf{c}(N, \sigma, \sigma[id]).id]$$
$$\mathit{fst}(M, N) = M$$
$$\mathit{snd}(M, N) = M[\mathsf{c}(N, \sigma, \sigma[id]).id]$$

to which one may add the η-rules:

$$\lambda\sigma.M[\uparrow]1 = M$$
$$(\mathit{fst}(M), \mathit{snd}(M)) = M$$

We collect the additional iso conversion rules (omitting the coercions) arising from products, sums, and identity types in Figure 2.

We now give the additional structure needed to interpret Π and Σ types.

1. First, we assume that our comprehension category is given via a fibred adjunction $p \dashv \mathbf{1}$ and an adjunction $\mathbf{1} \dashv p$, as described in (1) and (2) of Section 9.1.
2. For every X and A over X, we require the functor

$$\mathcal{P}(A)^* = \_[\mathcal{P}(A)] : \mathbb{E}_X \to \mathbb{E}_{(X,A)}$$

Figure 2: Iso conversion rules (type constructors)

$$\frac{\Gamma' \vdash s : \Gamma \quad \Gamma, \sigma \vdash \tau\ \mathtt{type}}{\Gamma' \vdash (\Pi\sigma.\tau)[s] \cong \Pi\sigma[s].\tau[1 \cdot (s \circ \uparrow)]} \qquad \frac{\Gamma' \vdash s : \Gamma \quad \Gamma, \sigma \vdash \tau\ \mathtt{type}}{\Gamma' \vdash (\Sigma\sigma.\tau)[s] \cong \Sigma\sigma[s].\tau[1 \cdot (s \circ \uparrow)]}$$

$$\frac{\Gamma' \vdash s : \Gamma \quad \Gamma \vdash \sigma\ \mathtt{type}}{\Gamma', \sigma[s], \sigma[s][\uparrow] \vdash (\mathtt{Id}_\sigma)[1 \cdot ((1 \cdot (s \circ \uparrow) \circ \uparrow)] \cong \mathtt{Id}_{\sigma[s]}}$$

(called weakening) to have left and right adjoints. Moreover, these adjunctions are required to satisfy the Beck-Chevalley conditions. For the Π types, this means that for every Y and u, and for every C, the canonical morphism in

$$\mathbb{E}_Y[(\Pi A.C)[u], \Pi A[u].C[\mathcal{P}(u_A)]]$$

is iso; and similarly for Σ types.

3. For Σ types, one requires additionally the following strongness assumption (similar to (4) of Section 9.1), namely that all canonical morphisms (A over X, B over (X, A)) from X, A, B to $X, \Sigma A.B$ are iso.

We leave it to the reader to spell out the interpretation of the explicit (and therefore also of the original) syntax in Grothendieck fibrations endowed with this additional structure (the strong sum assumption is needed to interpret the second projection in Σ types).

9.3. Extending our results to products, sums, and identity types

We call $\mathbb{ML}$-category a Grothendieck fibration that has products, strong sums, and supports extensional identity types, as spelled out in Sections 9.2 and 9.1.

When the fibration is split, when $\mathbf{1}_X[s] = \mathbf{1}_Y$ for every $X, Y, s : Y \to X$, and when the canonical morphisms involved in the respective Beck-Chevalley conditions are identities (one then says that the Beck-Chevalley condition holds exactly), it is called a strict $\mathbb{ML}$-category.

In [12], the third author has shown (in the language of locally cartesian closed categories and of categories with attributes) that when p is an $\mathbb{ML}$-category, the associated Rp (cf. Section 6) is a strict $\mathbb{ML}$-category. In our present framework, this goes as follows. First, the fibred terminal objects are given by

$$\mathbf{1}_X = (X, \phi) \quad \text{where } \phi(s : Y \to X) = \mathbf{1}_Y$$

where $\mathbf{1}$ on the right refers to the terminal objects relative to p. For Π types, abbreviating (X, ϕ), (Y, ψ) as ϕ, ψ, the definition of $\Pi\phi.\psi$ is as follows:

$$(\Pi\phi.\psi)(s) = \Pi\phi(s).\psi(\mathcal{P}(\phi(s : s \to id)))$$

(note that this is well typed: we have $\mathcal{P}(\phi(s)) : \mathcal{P}(\phi(s)) \to \mathcal{P}(\phi(id))$, and $p'(\psi) = (X, \phi)$, hence $\psi(\mathcal{P}(\phi(s)))$ is over $(Y, \phi(s))$.

We now check that Beck-Chevalley condition holds exactly. We first compute

$$\mathcal{P}'(t_\phi) = \mathcal{P}'(t, id) = \mathcal{P}(\phi(t : t \to id)) \ .$$

Hence we have

$$\begin{array}{rcl} (\Pi\phi[t].\psi[\mathcal{P}'(t_\phi)])(s) & = & (\Pi\phi[t].\psi[\mathcal{P}(\phi(t : t \to id))])(s) \\ & = & \Pi\phi(t \circ s).(\psi[\mathcal{P}(\phi(t : t \to id))](\mathcal{P}(\phi[t](s : s \to id))) \\ & = & (\Pi\phi.\psi)[t](s) \end{array}$$

where the last step follows from $\phi[t](s) = \phi(s : t \circ s \to t)$. Note that this proof does not make use of the Beck-Chevalley condition. The latter (or actually its consequence that the isomorphisms so obtained are natural) serves in the (tedious) proof that the fonctor $F : p \to Rp$ preserves all the structure of Π types (non strictly).

Likewise, $\Sigma\phi.\psi$ is defined as follows:

$$(\Sigma\phi.\psi)(s) = \Sigma\phi(s).\psi(\mathcal{P}(\phi(s : s \to id))) \ .$$

For identity types, let ψ be over (X, ϕ) (read here (X, ϕ), etc. . . as context extension!). Let $s : Y \to (X, \phi) \times_X (X, \phi)$. which induces $s_0 : Y \to X$, $t : (Y, \phi[s_0]) \to (X, \phi)$, and $u : Y \to (Y, \phi[s_0]) \times_Y (Y, \phi[s_0])$. We set

$$(Eq\,\phi)(s) = (Eq\,\phi(t))[u] \ .$$

where the right-hand side Eq is relative to p.

It can be checked (tediously, but without further difficulties) that all these data endow Rp with the structure of a strict $\mathbb{ML}$-category, and that all properties listed in Section 8 can be lifted from comprehension categories to $\mathbb{ML}$-categories, thus establishing the results claimed in the introduction. The definitions of morphisms and strict morphisms follow the same pattern as for comprehension categories: morphisms are required to preserve the added structure up to isomorphism, while strict morphisms, in addition to being strict morphisms between the underlying comprehension categories, have to send the chosen added structure to the chosen added structure.

It should be noted that both explicit and original classifying categories are meant here as extensional, i.e., they incorporate the reflection rule. Would we like to dispense with the reflection rule and hence account for intensional type theory, we would need a weaker notion of (non strict) model of identity types. This can probably be done, making use of ideas from weak factorisation systems (cf. e.g. [10, 1]). We leave this as future work.

10. Conclusion

We have placed the earlier constructions of the first and third author [7, 12] in some general mathematical context. The analogy with monoidal categories, which are algebras for a certain 2-monad, further suggests that some of the constructions described here (glueing, strictification) for $\mathbb{ML}$-categories could be carried out in the even larger context of two-dimensional monad theory [3]. We leave this for future investigation.

While we have focussed our attention on extensional type theory and its coherent interpretation in locally cartesian closed categories, we believe that our framework should have wider applicability. In particular, the presence of two levels of equality in the explicit syntax has a higher-dimensional category-theoretic flavour. We would like to investigate how to adapt our approach to

intensional type theory [22] (as mentioned above) and more widely to homotopy type theory [24], by further varying the notions of equality in the modelled type theory and by examining the corresponding coherence issues.

Finally, we would like to quote two related works that provide complementary information around the structures that we use in this paper. In [6], Clairambault and Dybjer establish a biequivalence between the 2-category of LCCC's and a 2-category of categories with families (a variant of the categories with attributes), which amounts in our setting to a biequivalence between $\mathbb{ML}$ and $\mathbf{S}\mathbb{ML}$, where the latter is the full sub-2-category of $\mathbb{ML}$ spanned by the strict $\mathbb{ML}$-categories. This latter category does not play a role in our analysis here, but fits clearly as the fourth item in the matrix strict/non-strict for objects/morphisms.

In [20], Lumsdaine and Warren exploit the other strictification arising as a left, rather than right, adjoint to the inclusion functor from $\mathbf{S}\mathbb{ML}_{\mathbf{s}}$ to $\mathbb{ML}$ (over a fixed basis), which they nicely explain in terms of local universes. Again, this is relevant in the context of the present homotopic developments in type theory.

Acknowledgements. We wish to thank Michael Warren for his remarks on a preliminary draft of this paper and for useful conversations on the semantics of identity types in comprehension categories, as well as the anonymous referees for their helpful corrections and recommendations.

And last but not least, we wish to thank the editors of this volume for this occasion to honour our friend and colleague Glynn Winskel, a contemporary of the first author, who remembers vividly our conversations and exchanges back when both were PhD strudents working on very similar beasts called concrete data structures and event structures, respectively.

References

[1] S. Awodey and M. Warren, Homotopy theoretic models of identity types, Mathematical Proceedings of the Cambridge Philosophical Society 146, 45-55 (2009).

[2] Des catégories fibrées, Notes by J.-R. Roisin of a course by J. Bénabou at Univ. Louvain-la-Neuve (1980).

[3] R. Blackwell, G.M. Kelly, and A.J. Power, Two-dimensional monad theory, J. Pure Appl. Algebra 59, 289-319 (1989).

[4] I. Beylin and P. Dybjer, Extracting a proof of coherence for monoidal categories from a formal proof of normalization for monoids, in Proc. TYPES'95, Lecture Notes in Computer Science 1158, 47-61 (1996).

[5] J. Cartmell, Generalised algebraic theories and contextual categories, PhD Thesis, Oxford University (1978).

[6] P. Clairambault and P. Dybjer, The biequivalence of locally cartesian closed categories and Martin-Löf type theory, Proceedings TLCA 2011, Lecture Notes in Computer Science 6690, 91-106 (2011).

[7] P.-L. Curien, Substitution up to isomorphism, Fundamenta Informaticae 19, 51-85 (1993) (preliminary version appeared as LIENS Technical report 90-9 (1990)).

[8] P. Dybjer, Internal type theory, in Proc. TYPES'95, Lecture Notes in Computer Science 1158, 120-134 (1996).

[9] P. Freyd, Aspects of topoi, Bull. Austral. Math. Soc. 7 (1972).

[10] R. Garner and N. Gambino, The identity type weak factorisation system, Theoretical Computer Science 409 (1), 94-109 (2008).

[11] J. Giraud, Cohomologie non abélienne, Springer (1971).

[12] M. Hofmann, On the interpretation of type theory in locally cartesian closed categories, Proc. Computer Science Logic (CSL'94), Lecture Notes in Computer Science 933, 427-441(1994).

[13] M. Hofmann, Syntax and semantics of dependent types, in Semantics and Logics of Computation, 79–130, Cambridge University Press (1997).

[14] G. Huet, Initiation la Théorie des Catégories", notes de cours (1985).

[15] B. Jacobs, Categorical type theory, PhD Thesis, Nijmegen (1991).

[16] B. Jacobs, Comprehension categories and the semantics of type theory, Theoretical Computer Science 107, 169-207 (1993).

[17] B. Jacobs, Categorical logic and type theory, Studies in logic and the foundations of mathematics, Elservier (1999).

[18] A. Joyal and R. Street, Braided monoidal categories, Advances in Mathematics 102(1), 20-78 (1993).

[19] T. Leinster, Higher operads, higher categories, Cambridge University Press (2004).

[20] P.L. Lumsdaine, M.A. Warren, The local universe model of type theory (preprint) (2012).

[21] S. Mac Lane, Categories for the working mathematician, Springer (1971).

[22] B. Nordström, K. Peterson, and J.M. Smith, Programming in Martin-Löf's type theory: an introduction, Oxford Science Publ. (1990).

[23] R. Seely, Locally cartesian closed categories and type theory, Math. Proc. Camb. Phil. Soc. 95, 33-48 (1984).

[24] V. Voevodsky, Notes on type systems (available from `http://www.math.ias.edu/~vladimir/Site3/Univalent_Foundations.html`) (2012).

[25] M. Warren, Homotopy theoretic aspects of constructive type theory, Ph.D. thesis, Carnegie Mellon University (2008).