Under consideration for publication in Math. Struct. in Comp. Science

Modular correspondence between dependent type theories and categories including pretopoi and topoi.

Maria Emilia Maietti

Dipartimento di Matematica Pura ed Applicata, Università di Padova

via G. Belzoni n.7, I–35131 Padova, Italy

e-mail: maietti@math.unipd.it

Received 26th January 2004, revised 3rd May 2005

We present a modular correspondence between various categorical structures and their internal languages in terms of extensional dependent type theories à la Martin-Löf. Starting from lex categories, through regular ones we provide internal languages of pretopoi and topoi and some variations of them, like for example Heyting pretopoi. With respect to the internal languages already known for some of these categories like topoi, the novelty of these calculi is that formulas corresponding to subobjects can be regained as particular types equipped with proof-terms according to the isomorphism "propositions as mono types", invisible in the previous languages.
MSC 2000: 03G30 03B15 18C50
Keywords: Categorical logic, topoi, Higher-order logic and type theory, Categorical semantics of formal languages.

1. Introduction

In order to develop constructive mathematics we can choose among different frameworks available in the literature, all sharing intuitionistic logic as the underlying logical reasoning. We can choose among theories formulated in the style of axiomatic set theory ZFC, such as the predicative Aczel-Myhill CZF (Aczel and Rathjen 2001) or the impredicative IZF (Scedrov 1985); or we can choose among theories formulated as a type theory where sets are considered as data types, such as the predicative Constructive Type Theory (Martin-Löf 1975; Martin-Löf 1998; Nordström *et al.* 1990) by Martin-Löf or the impredicative Calculus of Constructions (Coquand and Huet 1988) by Coquand; or we can develop mathematics inside a categorical universe such as a pretopos (whose underlying logic is predicative) or a topos (whose underlying logic is impredicative).

Topoi or pretopoi can be thought as universes where to develop mathematics because they provide models for certain kinds of set theory. The concept of elementary topos was introduced by Lawvere and Tierney to provide an axiomatization of a Grothendieck

topos free of set-theoretic assumptions. Then, it was shown that suitable topoi model fragments of classical set theory, like bounded Zermelo set theory (Cole 1973; Mitchell 1972; Mac Lane and Moerdijk 1992). Later Joyal and Moerdijk in (Joyal and Moerdijk 1995) proved that suitable pretopoi, namely Heyting pretopoi with a natural numbers object, are enough to provide a setting where to model the full Zermelo-Fraenkel set theory. More recently, in (Moerdijk and Palmgren 2000; Moerdijk and Palmgren 2002) Moerdijk and Palmgren studied the categorical correspondence of suitable type-theoretic constructions to be able to provide categorical models of Aczel-Myhill CZF (Aczel and Rathjen 2001) within a locally cartesian closed pretopos.

Suitable pretopoi, called arithmetic universes, were also used to give a categorical proof of Gödel incompleteness theorems by Andrè Joyal, already in the seventies (Wraith 1985). Andrè Joyal built arithmetic universes in three stages, from a cartesian category with a parameterized natural numbers object, through a regular category, to a pretopos with parameterized list objects. In (Maietti 2003) we argued that a general definition of arithmetic universes can be taken to be a pretopos with parameterized list objects after showing that an initial arithmetic universe is equivalent to an initial pretopos with parameterized list objects.

What makes more evident that all these categorical universes are suitable frameworks where to develop mathematics is that they can be viewed directly as universe of sets by means of their internal language.

The concept of internal language was first used to view a generic elementary topos as a universe of sets with the so called Mitchell-Benabou language (see (Lambek and Scott 1986),(Bell 1988)) formulated in terms of a many-sorted logic.

In this paper we can see that not only topoi but also pretopoi and many other categorical structures enjoy internal languages formulated in terms of a dependent type theory in the style of Martin-Löf's extensional type theory in (Martin-Löf 1984).

In the literature connections between categorical universes and logic was explored also for other categories than topoi, but always in terms of a many-sorted logic. Makkai and Reyes found that pretopoi provide a sort of completeness, called "conceptual completeness", with respect to many-sorted coherent logic (Makkai and Reyes 1977; Johnstone 2002). But, as far as we know, no explicit internal calculus for pretopoi has been proposed before ours.

Internal languages formulated as a dependent type theory differ from those formulated as a many-sorted logic mainly because the first include only types (possibly dependent) with their terms and equalities, while the latter includes two syntactic entities: sorts (or simple types) with their terms and formulas depending on sorts.

For example, the many-sorted Mitchell-Benabou internal language is formulated in such a way that categorical objects correspond to types, morphisms to typed terms and subobjects to many-sorted formulas, which are terms of the subobject classifier (Mac Lane and Moerdijk 1992; Lambek and Scott 1986). There is no constructor turning formulas into types, namely no isomorphism of the sort "proposition as types" is visible: propositions are not equipped with proof-terms.

Instead, in the internal language à la Martin-Löf we provide here for topoi, all the

categorical structure is described by means of dependent types, since simple types, like those used in the Mitchell-Benabou language, do not suffice.

As a consequence with a dependent type theory we can build a syntactic category by taking closed types as objects and terms as morphisms. In contrast, with a many sorted logic, for example with the Mitchell-Benabou language, a syntactic topos is built by taking formulas as objects, and functional relations as morphisms.

The interpretation of a dependent type theory in a fixed category $\mathcal{C}$ is more complex than that of a many sorted logic, or obviously that of a simple typed calculus where types are interpreted as objects of $\mathcal{C}$ and terms as morphisms of $\mathcal{C}$. The idea is to interpret a dependent type under a context as a sequence of morphisms of $\mathcal{C}$ and a dependent term as a morphism of a suitable slice category of $\mathcal{C}$. The notion of model we adopt combines the categorical semantics based on display maps (Seely 1984; Hyland and Pitts 1989) together with the tools provided by contextual categories to interpret substitution correctly (Cartmell 1986) by making use of the codomain fibration. Here, we require that the category must be at least lex since we want to interpret substitution via pullback. Then, to overcome the well known coherence problems due to the lack of a general functorial choice of pullbacks, we use the split fibration associated to the codomain fibration of the category (Benabou 1985; Hofmann 1995; Blackwell *et al.* 1989; Power 1989).

Actually, also the interpretation of formulas in an internal many-sorted logic, like that of topoi, can be equivalently given by making use of a fibration, namely the subobject fibration associated to the category. 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.*

Having in mind the above categorical semantics we can recognize how a dependent type theory à la Martin-Löf can describe the structure of subobjects avoiding the use of formulas and sequents. Indeed, it turns out that a *mono type*, which is defined as a type with at most one proof, i.e. formally a type $B(x)$ $[\Gamma]$ for which we can derive

$$y = z \in B(x) \ [\Gamma, y \in B(x), z \in B(x)]$$

gets interpreted by a sequence of morphisms the last one of which (interpreting the type itself, whilst the rest interprets the context) is monic. Hence, mono types correspond to monomorphisms as one can see by looking at the syntactic category built out from a dependent type theory. This is crucial to capture the subobject classifier of a topos or the quotients restricted to monic equivalence relations of a pretopos, or in other terms to capture the structure of the subobject fibration only speaking about types. Therefore, we deduce that interpreting formulas as subobjects yields the isomorphism *propositions as mono types, i.e. types with at most one proof* or more generally *formulas as mono dependent types*. As a consequence of these isomorphisms in the internal dependent type theories propositions, or more generally formulas, become *proof-relevant*, whilst with a unique proof-term. This is in contrast with many-sorted logic, where we are interested only in the provability of a formula and not in the shape of its proofs.

In extracting internal languages of categorical structures in terms of type theories à la Martin-Löf we are facilitated by the fact that the correspondence between dependent type theories and the categories considered is modular in the sense that we can single out a precise correspondence between universal categorical properties and type-theoretic constructors.

Hence, it is possible to get internal languages of various categories, from lex to regular categories, pretopoi, arithmetic universes, topoi and variations of them.

In more detail, universal categorical properties correspond to already known type constructors of Martin-Löf's extensional type theory or new ones as follows:

- Finite limits correspond to the terminal type, indexed sum types, extensional equality types in (Martin-Löf 1984). These types describing the type theory of lex categories form the basic module of an extensional dependent type theory internal to a category, since this category must be at least lex.
- The right adjoint to the pullback functor between slice categories corresponds to the dependent product type in (Martin-Löf 1984; Nordström *et al.* 1990).
- The left adjoint to the pullback functor between subobjects (or stable images) corresponds to the indexed sum type made mono and restricted to mono types.
- Stable quotients of monic equivalence relations correspond to extensional quotient types based only on mono equivalence relations with the addition of an axiom expressing effectiveness.
- The stable initial object corresponds to the false type in (Martin-Löf 1984; Nordström *et al.* 1990).
- Stable binary disjoint coproducts correspond to the disjoint sum types in (Martin-Löf 1984; Nordström *et al.* 1990) with the addition of an axiom expressing disjointness.
- The parameterized natural numbers object corresponds to the natural numbers type in (Martin-Löf 1984; Nordström *et al.* 1990) and parameterized list objects to list types in (Martin-Löf 1984; Nordström *et al.* 1990).
- Finally, the subobject classifier corresponds to an extensional universe type encoding mono types up to equiprovability.

The established link between dependent type theories and categories is stronger than the link established by a soundness and completeness theorem between a calculus and its models. Indeed, for any dependent typed calculus corresponding to a certain class of categorical structures we can prove not only soundness and completeness but also a sort of equivalence between the category of theories of the calculus and the category of categorical structures. It is this sort of equivalence that lets us conclude that the typed calculus provides the internal language of the corresponding categorical structures. And this sort of equivalence does not generally hold for any class of complete models of the given typed calculus.

Knowing internal languages à la Martin-Löf allows us to compare the considered categorical universes with Martin-Löf's type theory from a type theoretic point of view. Of course, we could also compare Martin-Löf's type theory with the categorical universes by looking at the categorical semantics of Martin-Löf's type theory. But we do not have yet a complete categorical semantics for the intensional version of Martin-Löf's type theory

in (Martin-Löf 1975; Martin-Löf 1998; Nordström *et al.* 1990). We do have it only for its extensional version in (Martin-Löf 1984). For sake of clearness, we must say that only the intensional version in (Nordström *et al.* 1990; Martin-Löf 1975) is nowadays considered the correct type theory. The reason is that only the intensional version can be really thought of a functional programming language because it is strongly normalizing and its definitional equality between terms is decidable. Instead, the extensional version can not be considered a functional programming language since it does not enjoy such properties as its definitional equality between terms is undecidable and it is also inconsistent with the formal Church Thesis (see (Hofmann 1997; Maietti and Sambin 2005) for more on the topic intensionality versus extensionality). Then, the intensional version of type theory, not even enjoying extensionality of functions, is not suitable to develop mathematics as it is. The actual development of mathematics in type theory is instead done inside an intermediate theory of extensional concepts, as that in (Sambin and Valentini 1998). This is a many sorted logic built upon the intensional type theory and equipped with set-theoretic notations as those commonly used in every day mathematics, and hence with proof-irrelevant propositions. Similarly, we can view the many-sorted logic internal to a category as obtained by abstraction from the internal dependent type theory in the same spirit of how the many-sorted logic in (Sambin and Valentini 1998) is built over Martin-Löf's type theory.

If we compare the internal type theory of a topos with Martin-Löf's one only from the "type perspective", one of the main differences is that all the internal dependent type theories of the categories considered so far are extensional according to the extensional version of Martin-Löf's type theory in (Martin-Löf 1984). These internal languages are necessarily extensional if we interpret the definitional equality as equality of morphisms in the category and the propositional equality as an equalizer. According to this type perspective the internal dependent type theory of a topos includes the fragment without universes of the type theory in (Martin-Löf 1984) as a subsystem and hence it is an extension of first order Martin-Löf's extensional type theory.

If we want to compare Martin-Löf's type theory with the internal type theory of a topos from the "proposition perspective" things go very differently. First of all in Martin-Löf's type theory propositions are built by interpreting them as types (where here with "type" we mean what nowadays is called set in (Nordström *et al.* 1990)) according to the isomorphism *proposition as types*. Instead, in the categorical universes considered, like topoi, *propositions as mono types* holds, as observed. One big consequence is immediately visible: Martin-Löf thinking of propositions as types conceives a strong existential quantifier that yields the validity of the propositional axiom of choice, since this turns out to correspond to a distributivity property between the dependent product type and the indexed sum type. Instead, topoi, being governed by propositions as mono types, admits only the usual existential quantifier of intuitionistic logic with no internal existence property, and hence no propositional axiom of choice. However, in topoi we can derive the axiom of unique choice and by our internal dependent language we can see why: in this case the existential quantifier becomes equivalent to the strong one.

Moreover, there are also constructions that in the presence of *proposition as mono types* do preserve constructivity and that do not in general with *propositions as types* and that

make Martin-Löf's type theory and the internal theory of a generic topos constructively incompatible. For example, extensional powersets, or effective quotients can not be added to Martin-Löf's type theory in the presence of the uniqueness of identity proofs (see (Maietti and Valentini 1999; Maietti 1998b; Maietti 1999)) if we follow *propositions as types*, since they are not constructively compatible with the axiom of choice. Indeed, by mimicking the well-known Diaconescu's argument (Diaconescu 1975) we can prove that even in the intensional type theory the choice function does not preserve constructively the extensionality reproduced in type theory by allowing the above constructions.

In conclusion from the proposition perspective it is no longer true that the internal theory of a generic topos is an extension of first order Martin-Löf's type theory.

A careful analysis of pros and cons of the two frameworks could also reveal some compromises to weaken a topos to be constructively compatible with the axiom of choice (for example by allowing some intensional impredicative universe with no effective quotients) or to extend Martin-Löf's Constructive Type Theory with stronger constructors closer to those of a topos as much as possible. In (Maietti and Sambin 2005) we singled out a type theoretic kernel common to both frameworks in terms of a predicative calculus of constructions.

So far we have discussed how the internal languages à la Martin-Löf prepare the grounds for a type-theoretic comparison between categories and type theories. We also recall that an obvious use of the internal type theory of a category is to perform categorical proofs in a logical way and to transfer techniques from type theory to category theory and the other way around. For example, in (Maietti 2003; Maietti 2005) we performed constructions inside an arithmetic universe by employing the internal type theory of an arithmetic universe. Having the internal type theory available allowed an extensive use of proofs by induction. And this is a clear advantage since it would have been much more difficult to perform such proofs with categorical diagram chasing.

Another application of the internal language of an arithmetic universe could be to provide a type-theoretic version of the proof of Gödel's incompleteness done categorically by André Joyal within an initial arithmetic universe.

Finally, having dependent typed calculi available as internal languages of categorical universes lets us analyze their computational contents, for example by investigating the validity of canonical normal form theorems.

The work reported here is mostly based on the PhD-thesis (Maietti 1998b). The internal language of Heyting pretopoi also appeared in (Maietti 1998a).

2. Categorical structures

We proceed by recalling the definitions of the categorical structures we consider, starting with lex categories to end with topoi.

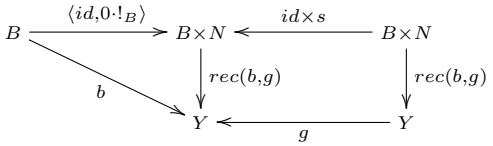
Def. 2.1. A **lex category** is a category with finite limits (or finitely complete category), i.e. with a terminal object, binary products and equalizers (Mac Lane 1971).

We recall that for a category having a terminal object and pullbacks is equivalent to being finitely complete, and that having finite products is equivalent to having a terminal object and binary products.

Def. 2.2. A lex category is said to be **arithmetic** if it has a parameterized natural numbers object.

where

Def. 2.3. *A* **parameterized natural numbers object** *in a category with finite products is an object N together with morphisms $0 : 1 \to N$ and $s : N \to N$ such that for every $b : B \to Y$ and $g : Y \to Y$ there is an unique $rec(b, g)$ making the following diagrams commute*



with $!_B : B \to 1$ the unique map towards the terminal object.

It is worth to recall here that in presence of function spaces (or exponentials), like in a cartesian closed category (see (Mac Lane and Moerdijk 1992; Lambek and Scott 1986) for a definition), this parameterized version of natural numbers object is equivalent to the usual natural numbers object.

Def. 2.4. A **regular category** is a finitely complete category with stable images (Jacobs 1999; Taylor 1997).

Def. 2.5. A **lextensive category** is a finitely complete category with stable finite disjoint coproducts (Carboni *et al.* 1993). †.

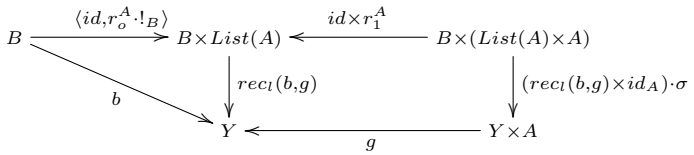
Def. 2.6. A **locos** is a lextensive category with parameterized list objects (Cockett 1990).

where parameterized list objects are defined as follows:

Def. 2.7. A finitely complete category $\mathcal{C}$ has **parameterized list objects** if for any object $A \in Ob\mathcal{C}$, there is an object $List(A)$ equipped with morphisms $r_o^A : 1 \to List(A)$ and $r_1^A : List(A) \times A \to List(A)$ such that for every $b : B \to Y$ and $g : Y \times A \to Y$ there

† In previous versions of this paper or in (Maietti 2001) we used the term "distributive" as in (Cockett 1990) for what most of the authors call "lextensive" to reserve the word "distributive" for a more general concept (see (Carboni *et al.* 1993)).

is an unique $rec_l(b,g)$ making the following diagrams commute



where $\sigma : B \times (List(A) \times A) \to (B \times List(A)) \times A$ is the associative isomorphism $\langle\langle \pi_1, \pi_1 \cdot \pi_2 \rangle, \pi_2 \cdot \pi_2 \rangle$.

In (Cockett 1990) there is an equivalent definition of finitely complete categories with list objects (also called list-arithmetic lex categories) in terms of recursive objects and preservation of recursive objects by the pullback functor $!_D^* : \mathcal{C} \to \mathcal{C}/D$ sending an object B to $\pi_1 : D \times B \to D$.

Finally, we recall the categorical definition of pretopos, locally cartesian closed category, topos and variations of them:

Def. 2.8. A **pretopos** is a category equipped with finite limits, stable finite disjoint sums and stable effective quotients of monic equivalence relations (Makkai and Reyes 1977; Joyal and Moerdijk 1995).

Def. 2.9. An **Heyting pretopos** is a pretopos where the pullback functor on subobjects has a right adjoint (Joyal and Moerdijk 1995).

Def. 2.10. An **arithmetic universe** is a pretopos with parameterized list objects (see (Maietti 2003) for a justification of such a definition).

Def. 2.11. A **locally cartesian closed** category is a category equipped with finite limits and right adjoints to pullback functors (Jacobs 1999; Taylor 1997).

Def. 2.12. A **topos** is a category equipped with finite limits, exponentials and a subobject classifier (Makkai and Reyes 1977; Mac Lane and Moerdijk 1992).

Since the aim of this paper is to describe internal dependent type theories of the categories mentioned so far, we list necessary conditions that a category $\mathcal{C}$ has to satisfy in order to enjoy a dependent typed internal language:

1 $\mathcal{C}$ has to be finitely complete. This is because we want to interpret substitution of terms by means of pullbacks.
2 The structure of $\mathcal{C}$ necessary to interpret the type constructors on closed types has to be *local*, i.e. for every object $A \in Ob\mathcal{C}$ the slice category $\mathcal{C}/A$ (see (Mac Lane 1971) for its definition) must enjoy the same structure of $\mathcal{C}$ (for example, if $\mathcal{C}$ is a topos then $\mathcal{C}/A$ should be a topos for every $A \in Ob\mathcal{C}$). This is because a dependent type is interpreted in a suitable slice category and hence any slice category has to be equipped with all the structure to interpret the type constructors under a certain dependency.
3 The structure of $\mathcal{C}$ has *to be preserved by the pullback functor* $f^* : \mathcal{C}/A \to \mathcal{C}/B$ for every morphism $f : B \to A$ of $\mathcal{C}$. This is because, if we interpret substitution

via pullback, then the structure needs to be preserved under pullbacks to make the interpretation of a type constructor closed under substitution.

The categories mentioned so far satisfy the necessary conditions to enjoy an internal dependent type theory:

Proposition 2.13. Lex, arithmetic lex, regular, lextensive categories, locoi, pretopoi, arithmetic universes, Heyting pretopoi, locally cartesian closed categories and topoi are local and their structures are preserved by pullbacks in the above sense.

Proof. The proof of this statement is modular on the universal categorical properties such categories enjoy. The local property of a finitely complete category follows from the fact that in $\mathcal{C}/A$ the identity on A is a terminal object and that the forgetful functor $U : \mathcal{C}/A \to \mathcal{C}$ creates pullbacks.

To show that regular categories and pretopoi are local is enough to note that the above forgetful functor U creates colimits and that given in $\mathcal{C}/D$ an equivalence relation $R \xrightarrow[r]{\rho} A \times_D A \xrightarrow[D]{a\cdot\pi_1}$ then $\langle \pi_1 \cdot \rho, \pi_2 \cdot \rho \rangle : R \to A \times A$ is also an equivalence relation in $\mathcal{C}$, where $\pi_i : A \times_D A \to A$ for $i : 1, 2$ are the projections of the pullback of $a : A \to D$ along itself in $\mathcal{C}$.

The proof that the natural numbers object is local follows easily: for every object A of a category $\mathcal{C}$ with products, a parameterized natural numbers object in $\mathcal{C}/A$ is $\pi_1 : A \times \mathcal{N} \to A$, where $\mathcal{N}$ is a natural numbers object of $\mathcal{C}$.

An Heyting pretopos $\mathcal{P}$ is local because the subobjects in the slice category correspond to subobjects in $\mathcal{P}$. Then, to check that right adjoints are stable under pullbacks means that Beck-Chevalley conditions are satisfied, which also follows.

The proof that list objects are local is more delicate. We can prove locality of list objects in a locos (the locality of a locos is also stated in (Cockett 1990)) as follows. First, we assume to know the internal dependent type theory of a lextensive category. Then, we get the internal type theory of a locos by simply adding list types restricted to closed types, which are interpreted as parameterized list objects. By means of this internal language we can show how to build list types on arbitrary dependent types, corresponding to list-objects in a slice category, and that in the suitable slice syntactic category they are stable under pullbacks.

A locally cartesian closed category $\mathcal{C}$ can be easily proved to be local considering its equivalent characterization (for example in (Seely 1984)) saying that for every object B the slice category C/B is cartesian closed, and knowing that for every $f : B \to A$ in $\mathcal{C}$ the slice category $(\mathcal{C}/A)/f$ is equivalent to $\mathcal{C}/B$.

Finally, the proof that a topos is local can be found in (Mac Lane and Moerdijk 1992) or (Johnstone 1977). □

We anticipate here that, in order to interpret a typed calculus in the corresponding categorical structure, a given choice of the categorical structure needs to be made given that the categorical structure is usually defined up to isomorphism. However, this is not enough to make the interpretation. Indeed, to interpret substitution we need to find

a canonical choice of pullbacks that is functorial, and that is strictly preserved by the categorical structure. But even in the category *Set* of ZFC sets and functions we do not know any canonical functorial choice of pullbacks. To overcome this difficulty we will make use of the machinery of fibred functors (Jacobs 1999; Hofmann 1997).

3. Extensional dependent type theories

For each categorical structure described in the previous section we give here the description of the corresponding typed calculus meant to provide its internal language in the style of Martin-Löf's extensional dependent type theory (Martin-Löf 1984).

Any typed system is equipped with types, which should be thought of as sets or data types, and with typed terms which represent proofs (or elements) of the types to which they belong. In order to describe them in the style of Martin-Löf's type theory, we have four kinds of judgements (Nordström *et al.* 1990):

$$A \; type \; [\Gamma] \qquad A = B \; [\Gamma] \qquad a \in A \; [\Gamma] \qquad a = b \in A \; [\Gamma]$$

that is the judgements about type formation and their terms, the equality between types and the equality between terms of the same type (called *definitional* equality of terms in contrast to the propositional equality of terms that is a type).

The contexts of these judgements are telescopic (de Bruijn 1991), since types are allowed to depend on variables of other types. The contexts are generated by the following rules

$$\frac{}{\emptyset \quad cont} \; 1c) \qquad \frac{\Gamma \quad cont \qquad A \; type \; [\Gamma]}{\Gamma, x \in A \quad cont} \; 2c) \quad (x \in A \notin \Gamma)$$

plus the rules of equality between contexts (Streicher 1991), (Pitts 2000). Then, we need to add all the inference rules that express reflexivity, symmetry and transitivity of the equality between types and terms together with the type equality rules *conv)* and *conv-eq)* and the assumption of variables

$$\frac{a \in A \; [\Gamma] \quad A = B \; [\Gamma]}{a \in B \; [\Gamma]} \; conv) \qquad \frac{a = b \in A \; [\Gamma] \quad A = B \; [\Gamma]}{a = b \in B \; [\Gamma]} \; conv\text{-}eq) \qquad \frac{\Gamma, x \in A, \Delta \quad cont}{x \in A \; [\Gamma, x \in A, \Delta]} \; var)$$

The structural rules of weakening, substitution and of a suitable exchange are not added since they are derivable.

We also adopt the usual definitions of bound and free occurrences of variables and we identify two terms under α-conversion. Moreover, we use the expression $(x)b(x)$ to mean the equivalence class of $b(x) \in B(x) \, [x \in A]$ under renaming of variables, and we also write b for $(x)b(x)$. Actually, such expressions belong to the type theory with higher ariety in (Nordström *et al.* 1990), with the warning that what is called a type here is called a set in (Nordström *et al.* 1990). Indeed, by adding the so called function type, from $b(x) \in B(x) \, [x \in A]$ we can form the abstraction, that is $(x)b(x) \in (x \in A)B(x)$, and the corresponding application that satisfy β and η conversions.

Now, we give the formation rule for types specific to the various calculi with the introduction, elimination and conversion rules of their terms. Beside them we should add the corresponding formation equality rules for types with the introduction and elimination equality rules for their terms as in (Martin-Löf 1984), but to be short we omit them. Note

that the piece of context common to all judgements involved in a rule will be omitted. The typed variables appearing in a context are meant to be added to the implicit context as the last one.

For example, we simply write

$$\frac{C(x)\ type \quad [x \in B]}{\Sigma_{x \in B} C(x)\ type}$$

instead of

$$\frac{C(x)\ type \quad [\Gamma, x \in B]}{\Sigma_{x \in B} C(x)\ type\ [\Gamma]}$$

The calculus for lex categories $\mathcal{T}_{lex}$

$=$

Terminal type

$$\frac{}{\top\ type}\ \text{Tr} \qquad \frac{}{\star \in \top}\ \text{I-Tr} \qquad \frac{t \in \top}{t = \star \in \top}\ \text{C-Tr}$$

Indexed Sum type

$$\frac{C(x)\ type \quad [x \in B]}{\Sigma_{x \in B} C(x)\ type}\ \Sigma \qquad \frac{b \in B \quad c \in C(b) \qquad \Sigma_{x \in B} C(x)\ type}{\langle b, c \rangle \in \Sigma_{x \in B} C(x)}\ \text{I-}\Sigma$$

$$\frac{\begin{array}{l} M(z)\ type\ [z \in \Sigma_{x \in B} C(x)] \\ d \in \Sigma_{x \in B} C(x) \qquad m(x,y) \in M(\langle x, y \rangle)\ [x \in B, y \in C(x)] \end{array}}{El_{\Sigma}(d, m) \in M(d)}\ \text{E-}\Sigma$$

$$\frac{\begin{array}{l} M(z)\ type\ [z \in \Sigma_{x \in B} C(x)] \\ b \in B \qquad c \in C(b) \qquad m(x,y) \in M(\langle x, y \rangle)\ [x \in B, y \in C(x)] \end{array}}{El_{\Sigma}(\langle b, c \rangle, m) = m(b, c) \in M(\langle b, c \rangle)}\ \text{C-}\Sigma$$

Extensional Equality type

$$\frac{C\ type \quad c \in C \quad d \in C}{\mathsf{Eq}(C, c, d)\ type}\ \text{Eq} \qquad \frac{c \in C}{\mathsf{eq_C}(c) \in \mathsf{Eq}(C, c, c)}\ \text{I-Eq}$$

$$\frac{p \in \mathsf{Eq}(C, c, d)}{c = d \in C}\ \text{E-Eq} \qquad \frac{p \in \mathsf{Eq}(C, c, d)}{p = \mathsf{eq_C}(c) \in \mathsf{Eq}(C, c, d)}\ \text{C-Eq}$$

The calculus of regular categories $\mathcal{T}_{reg}$

$=$

Lex calculus $+$

Mono Existential type

$$\frac{C(x)\ type\ \ [x\in B]\quad y=z\in C(x)\ [x\in B, y\in C(x), z\in C(x)]}{\exists_{x\in B}C(x)\ type}\ \exists \qquad \frac{b\in B\quad c\in C(b)\qquad \exists_{x\in B}C(x)\ type}{(b,c)\in \exists_{x\in B}C(x)}\ \text{I-}\exists$$

$$\frac{\exists_{x\in B}C(x)\ type \quad b\in B\quad c\in C(b)\quad d\in B\quad t\in C(d)}{(b,c)=(d,t)\in \exists_{x\in B}C(x)}\ \text{eq-}\exists$$

$$\frac{M(z)\ type\ [z\in \exists_{x\in B}C(x)] \quad y=z\in M(w)\ [w\in \exists_{x\in B}C(x), y\in M(w), z\in M(w)] \quad d\in \exists_{x\in B}C(x) \quad m(x,y)\in M((x,y))\ [x\in B, y\in C(x)]}{\mathsf{Ex}(d,m)\in M(d)}\ \text{E-}\exists$$

The calculus of arithmetic lex categories $\mathcal{T}_{alex}$

$=$

Lex calculus $+$

Natural Numbers type

$$\frac{}{N\ type}\ \text{nat} \qquad \frac{}{0\in N}\ \text{I}_1\text{-nat} \qquad \frac{n\in N}{\mathsf{s}(n)\in N}\ \text{I}_2\text{-nat}$$

$$\frac{L(z)\ type\ [z\in N] \quad n\in N\quad a\in L(0)\qquad l(x,y)\in L(\mathsf{s}(x))\ [x\in N, y\in L(x)]}{El_N(a,l,n)\in L(n)}\ \text{E-nat}$$

$$\frac{L(z)\ type\ [z\in N] \quad a\in L(0)\qquad l(x,y)\in L(\mathsf{s}(x))\ [x\in N, y\in L(x)]}{El_N(a,l,0)=a\in L(0)}\ \text{C}_1\text{-nat}$$

$$\frac{L(z)\ type\ [z\in N] \quad n\in N\quad a\in L(0)\qquad l(x,y)\in L(\mathsf{s}(x))\ [x\in N, y\in L(x)]}{El_N(a,l,\mathsf{s}(n))=l(n,El_N(a,l,n))\in L(\mathsf{s}(n))}\ \text{C}_2\text{-nat}$$

The calculus of lextensive categories $\mathcal{T}_{ext}$

$=$

Lex calculus $+$

Disjoint Sum type

$$\frac{C\ type \quad D\ type}{C + D\ type}\ + \qquad \frac{c \in C \qquad C + D\ type}{\mathsf{inl}(c) \in C + D}\ \mathrm{I}_1\text{-}+ \qquad \frac{d \in D \qquad C + D\ type}{\mathsf{inr}(d) \in C + D}\ \mathrm{I}_2\text{-}+$$

$$\frac{\begin{array}{l}A(z)\ type\ [z \in C + D] \\ p \in C + D \qquad a_C(x) \in A(\mathsf{inl}(x))\ [x \in C] \quad a_D(y) \in A(\mathsf{inr}(y))\ [y \in D]\end{array}}{El_+(p, a_C, a_D) \in A(p)}\ \mathrm{E}\text{-}+$$

$$\frac{\begin{array}{l}A(z)\ type\ [z \in C + D] \\ c \in C \qquad a_C(x) \in A(\mathsf{inl}(x))\ [x \in C] \quad a_D(y) \in A(\mathsf{inr}(y))\ [y \in D]\end{array}}{El_+(\mathsf{inl}(c), a_C, a_D) = a_C(c) \in A(\mathsf{inl}(c))}\ \mathrm{C}_1\text{-}+$$

$$\frac{\begin{array}{l}A(z)\ type\ [z \in C + D] \\ d \in D \qquad a_C(x) \in A(\mathsf{inl}(x))\ [x \in C] \quad a_D(y) \in A(\mathsf{inr}(y))\ [y \in D]\end{array}}{El_+(\mathsf{inr}(d), a_C, a_D) = a_D(d) \in A(\mathsf{inr}(d))}\ \mathrm{C}_2\text{-}+$$

False type **Disjointness**

$$\frac{}{\bot\ type}\ \mathrm{Fs} \qquad \frac{a \in \bot \quad A\ type}{\mathsf{r}_\bot(a) \in A}\ \mathrm{E}\text{-Fs} \qquad \frac{c \in C \quad d \in D \quad \mathsf{inl}(c) = \mathsf{inr}(d) \in C + D}{\mathsf{dsj}(c, d) \in \bot}$$

The calculus of locoi $\mathcal{T}_{loc}$

$=$

Lextensive calculus $+$

List type

$$\frac{C\ type}{List(C)\ type}\ \text{list} \qquad \frac{List(C)\ type}{\epsilon \in List(C)}\ \text{I}_1\text{-list} \qquad \frac{s \in List(C) \quad c \in C}{\mathsf{cons}(s,c) \in List(C)}\ \text{I}_2\text{-list}$$

$$\frac{\begin{array}{l} L(z)\ type\ [z \in List(C)] \\ s \in List(C) \qquad a \in L(\epsilon) \qquad l(x,y,z) \in L(\mathsf{cons}(x,y))\ [x \in List(C), y \in C, z \in L(x)] \end{array}}{El_{List}(a,l,s) \in L(s)}\ \text{E-list}$$

$$\frac{\begin{array}{l} L(z)\ type\ [z \in List(C)] \\ a \in L(\epsilon) \qquad l(x,y,z) \in L(\mathsf{cons}(x,y))\ [x \in List(C), y \in C, z \in L(x)] \end{array}}{El_{List}(a,l,\epsilon) = a \in L(\epsilon)}\ \text{C}_1\text{-list}$$

$$\frac{\begin{array}{l} L(z)\ type\ [z \in List(C)] \\ s \in List(C) \quad c \in C \quad a \in L(\epsilon) \quad l(x,y,z) \in L(\mathsf{cons}(x,y))\ [x \in List(C), y \in C, z \in L(x)] \end{array}}{El_{List}(a,l,\mathsf{cons}(s,c)) = l(s,c,El_{List}(a,l,s)) \in L(\mathsf{cons}(s,c))}\ \text{C}_2\text{-list}$$

The calculus of pretopoi $\mathcal{T}_{ptop}$

$=$

Lextensive calculus $+$

Extensional Quotient type

$$\frac{\begin{array}{l}R(x,y)\ type\ [x\in A, y\in A] \quad z=w\in R(x,y)\ [x\in A, y\in A, z\in R(x,y), w\in R(x,y)]\\ c_1\in R(x,x)\ [x\in A] \quad c_2\in R(y,x)\ [x\in A, y\in A, z\in R(x,y)]\\ c_3\in R(x,z)\ [x\in A, y\in A, z\in A, w\in R(x,y), w'\in R(y,z)]\end{array}}{A/R\ type}\ \text{Q}$$

$$\frac{a\in A \quad A/R\ type}{[a]\in A/R}\ \text{I-Q} \qquad \frac{a\in A \quad b\in A \quad d\in R(a,b) \quad A/R\ type}{[a]=[b]\in A/R\ type}\ \text{eq-Q}$$

$$\frac{\begin{array}{l}L(z)\ type\ [z\in A/R]\\ p\in A/R \quad l(x)\in L([x])\ [x\in A] \quad l(x)=l(y)\in L([x])\ [x\in A, y\in A, d\in R(x,y)]\end{array}}{El_Q(l,p)\in L(p)}\ \text{E-Q}$$

$$\frac{\begin{array}{l}L(z)\ type\ [z\in A/R]\\ a\in A \quad l(x)\in L([x])\ [x\in A] \quad l(x)=l(y)\in L([x])\ [x\in A, y\in A, d\in R(x,y)]\end{array}}{El_Q(l,[a])=l(a)\in L([a])}\ \text{C-Q}$$

Effectiveness

$$\frac{a\in A \quad b\in A \quad [a]=[b]\in A/R}{\mathsf{eff}(a,b)\in R(a,b)}$$

The calculus of Heyting pretopoi $\mathcal{T}_{hptop}$

$=$

Pretopos calculus $+$

Forall type

$$\frac{C(x)\ type\ [x\in B] \quad y=z\in C(x)\ [x\in B, y\in C(x), z\in C(x)]}{\forall_{x\in B}C(x)\ type}\ \forall$$

$$\frac{c\in C(x)\ [x\in B] \quad y=z\in C(x)\ [x\in B, y\in C(x), z\in C(x)]}{\lambda x^B.c\in\forall_{x\in B}C(x)}\ \text{I-}\forall$$

$$\frac{b\in B \quad f\in\forall_{x\in B}C(x)}{\mathsf{Ap}(f,b)\in C(b)}\ \text{E-}\forall$$

$$\frac{f\in\forall_{x\in B}C(x)}{\lambda x^B.\mathsf{Ap}(f,x)=f\in\forall_{x\in B}C(x)}\ \eta\text{C-}\forall \quad (x\ not\ free\ in\ f)$$

The calculus of arithmetic universes $\mathcal{T}_{au}$

=

Pretopos calculus +

List types

The calculus of locally cartesian closed categories $\mathcal{T}_{lcc}$
=
The first order fragment of Martin-Löf's extensional type theory

=

Lex calculus +

Extensional Dependent Product type

$$\frac{C(x)\ type\ [x \in B]}{\Pi_{x\in B}C(x)\ type}\ \text{F-}\Pi$$

$$\frac{c \in C(x)\ [x \in B]}{\lambda x^B.c \in \Pi_{x\in B}C(x)}\ \text{I-}\Pi \qquad \frac{b \in B \quad f \in \Pi_{x\in B}C(x)}{\mathsf{Ap}(f,b) \in C(b)}\ \text{E-}\Pi$$

$$\frac{b \in B \quad c \in C(x)\ [x \in B]}{\mathsf{Ap}(\lambda x^B.c, b) = c(b) \in C(b)}\ \beta\text{C-}\Pi \qquad \frac{f \in \Pi_{x\in B}C(x)}{\lambda x^B.\mathsf{Ap}(f,x) = f \in \Pi_{x\in B}C(x)}\ \eta\text{C-}\Pi \quad (x\ not\ free\ in\ f)$$

The calculus of topoi $\mathcal{T}_{top}$
=

Locally cartesian closed calculus +

Omega type

$$\frac{}{\Omega\ type}\ \Omega \qquad \frac{B\ type \quad y = z \in B\ [y \in B, z \in B]}{\{B\} \in \Omega}\ \text{I-}\Omega$$

$$\frac{\begin{array}{ll} B\ type & y = z \in B\ [y \in B, z \in B] \\ C\ type & y = z \in C\ [y \in C, z \in C] \\ f \in B \leftrightarrow C & \end{array}}{\{B\} = \{C\} \in \Omega}\ \text{eq-}\Omega$$

$$\frac{\{B\} = \{\top\} \in \Omega}{\mathsf{ch}(B) \in B}\ \beta\text{C-}\Omega \qquad \frac{q \in \Omega}{\{\mathsf{Eq}(\Omega, q, \{\top\})\} = q \in \Omega}\ \eta\text{C-}\Omega$$

where $B \leftrightarrow C \equiv B \to C \times C \to B$

From now on we shall often omit the word *type* in the type judgements.
In the dependent typed calculi presented so far, the types and corresponding terms coming from extensional Martin-Löf's type theory (Martin-Löf 1984) are those in the calculus for locally cartesian closed categories together with the disjoint sum types (but without the disjointness axiom). The extensional quotient types on any (equivalence) relation - hence not restricted to mono equivalence relations as presented here - and without the effectiveness axiom, appeared in Nuprl (Constable 1986).

Why categories correspond to extensional type theories.
We recall that the extensional type theory is characterized by the presence of the extensional equality type. Instead, the intensional type theory (Nordström *et al.* 1990) is characterized by the fact that all the type constructors have only β-conversions and the equality type is intensional:

Intensional Equality type

$$\frac{A\ type \quad a \in A \quad b \in A}{\mathsf{Id}(A,a,b)\ type}\ \mathsf{Id} \qquad \frac{a \in A}{\mathsf{id}(a) \in \mathsf{Id}(A,a,a)}\ \text{I-Id}$$

$$\frac{\begin{array}{l} C(x,y,z)\ [x \in A, y \in A, z \in \mathsf{Id}(A,x,y)] \\ d \in \mathsf{Id}(A,a,b) \qquad\qquad c(x) \in C(x,x,\mathsf{id}(x)))\ [x : A] \end{array}}{\mathsf{idpeel}(d,c) \in C(a,b,d)}\ \text{E-Id}$$

$$\frac{\begin{array}{l} C(x,y,z)\ [x \in A, y \in A, z \in \mathsf{Id}(A,x,y)] \\ a \in A \qquad\qquad c(x) \in C(x,x,\mathsf{id}(x))\ [x : A] \end{array}}{\mathsf{idpeel}(\mathsf{id}(a),c) = c(a) \in C(a,a,\mathsf{id}(a))}\ \text{C-Id}$$

A big difference between intensional and extensional versions of type theory is that in the intensional version the definitional equality $a = b \in A$ is decidable, while this is no longer true for the extensional version (see for example (Hofmann 1997) for discussions about this).

In our correspondence between categories and dependent typed languages the internal dependent type theories are all extensional. The reason is that the propositional equality must be extensional if we interpret the definitional equality between terms as the equality between morphisms of the category and the propositional equality as an equalizer.

Equivalent formulation for the indexed sum type.
Actually, from now on, we will refer to an equivalent formulation of the lex calculus where the elimination and conversion rules for the indexed sum type are replaced by projections satisfying β and η conversions. This formulation with projections is equivalent to that

of the indexed sum type given previously only thanks to the presence of the extensional equality type (see (Martin-Löf 1984) for a proof of the equivalence), that is the equivalence does not hold in the intensional version of Martin-Löf's type theory in (Martin-Löf 1975; Martin-Löf 1998; Nordström *et al.* 1990) and this is why we add the adjective "extensional" to this new formulation of indexed sum type.

The calculus for lex categories $\mathcal{T}_{lex}$

=

Terminal type + Extensional Equality type

+

Extensional Indexed Sum type (with projections)

$$\frac{C(x)\ type \quad [x \in B]}{\Sigma_{x\in B}C(x)\ type}\ \Sigma \qquad \frac{b \in B \quad c \in C(b) \qquad \Sigma_{x\in B}C(x)\ type}{\langle b, c\rangle \in \Sigma_{x\in B}C(x)}\ \text{I-}\Sigma$$

$$\frac{d \in \Sigma_{x\in B}C(x)}{\pi_1^B(d) \in B}\ \text{E}_{\mathbf{1}}\text{-}\Sigma \qquad \frac{d \in \Sigma_{x\in B}C(x)}{\pi_2^{C(\pi_1(d))}(d) \in C(\pi_1(d))}\ \text{E}_{\mathbf{2}}\text{-}\Sigma$$

$$\frac{b \in B \quad c \in C(b)}{\pi_1^B(\langle b, c\rangle) = b \in B}\ \beta_1\ \text{C-}\Sigma \qquad \frac{b \in B \quad c \in C(b)}{\pi_2^{C(b)}(\langle b, c\rangle) = c \in C(b)}\ \beta_2\ \text{C-}\Sigma$$

$$\frac{d \in \Sigma_{x\in B}C(x)}{\langle \pi_1^B(d), \pi_2^{C(\pi_1(d))}(d)\rangle = d \in \Sigma_{x\in B}C(x)}\ \eta\ \text{C-}\Sigma$$

The two existential quantifiers: strong and weak.

One of the main points of Martin-Löf's type theory is the validity of the isomorphism *propositions as types*. As a consequence the existential quantifier is put in correspondence with the indexed sum type as presented in the lex calculus. This means that we can define the two projections as just seen (and this holds in the intensional version, too [‡]), with the consequence of having the existence property internalized in the calculus. This is stronger than the usual formalization of the intuitionistic existential quantifier, like that present in a topos, which instead corresponds to Howard's weak indexed sum type according to the isomorphism propositions as types:

[‡] In the intensional version of Martin-Löf's type theory (Nordström *et al.* 1990) we can define projections that satisfy β conversions but η conversion only at the propositional level, i.e. in the form $\mathsf{Id}(\Sigma_{x\in B}\, C(x),\, \langle \pi_1^B(d), \pi_2^{C(\pi_1(d))}(d)\rangle\, ,\, d)$.

Howard's Weak Indexed Sum type

$$\frac{C(x)\ type\ \ [x \in B]}{\Sigma^w_{x\in B}C(x)\ type}\ \text{w}\Sigma \qquad \frac{b \in B \quad c \in C(b) \qquad \Sigma^w_{x\in B}C(x)\ type}{\langle b, c\rangle \in \Sigma^w_{x\in B}C(x)}\ \text{I-w}\Sigma$$

$$\frac{\begin{array}{l} M\ type \\ d \in \Sigma^w_{x\in B}C(x) \qquad m(x,y) \in M\ [x \in B, y \in C(x)] \end{array}}{El_{\Sigma w}(d, m) \in M}\ \text{E-w}\Sigma$$

$$\frac{\begin{array}{l} M\ type \\ b \in B \quad c \in C(b) \qquad m(x,y) \in M\ [x \in B, y \in C(x)] \end{array}}{El_{\Sigma w}(\langle b, c\rangle, m) = m(b, c) \in M}\ \text{C-w}\Sigma$$

The main difference between the rules of the indexed sum and those of the weak one is that in the elimination rule of the weak one the type M must not depend on $\Sigma^w_{x\in B}C(x)$. As a relevant consequence, in the presence of the product type while the indexed sum type allows to derive a proof of the following type, read as the axiom of choice in (Martin-Löf 1984)

$$(\Pi x \in A)(\Sigma y \in B)\ C(x, y) \to (\Sigma f \in A \to B)(\Pi x \in A)\ C(x, f(x))$$

the weak indexed sum does not (see (Swaen 1991; Swaen 1992)).

Mono types.
We call *mono* every type for which we can prove

$$\frac{B(x)\ type\ [x \in A]}{y = z \in B(x)\ [x \in A, y \in B(x), z \in B(x)]}$$

also called proof-irrelevant, for example in (Hofmann 1997). These are dependent types that can be inhabitated by at most one proof. This is the central concept to characterize the structure of subobjects of a category, like stable images of a regular category captured by the mono existential types, or the right adjoints restricted to subobjects of an Heyting pretopos captured by the forall types, or the quotient of a monic equivalence relation in a pretopos captured by the quotient type restricted to mono equivalence relations, or the subobject classifier of a topos captured by the Omega type classifying only mono types.

Observe that we can prove that the mono existential type and the forall type are mono respectively in $\mathcal{T}_{reg}$ and in $\mathcal{T}_{hptop}$. Moreover, we did not add the β conversion rule for the mono existential type and the forall type since they are derivable because they are equalities between terms of a mono type.

Remark on how to weaken the elimination rules on dependent types.
In the presence of indexed sum types and extensional equality types, the usual dependent elimination rule of a type A, which is toward types, like $M(z)\ [z \in A]$, depending on A itself is equivalent to an elimination rule toward types not depending on A, provided that we also add an extra η conversion rule stating uniqueness of the weakened elimination

constructor. This added η conversion rule can be proved to be valid by turning it into a propositional equality, which can be derived by using the dependent (!) elimination rule on the type A. The weakened elimination constructor can be proved to be as much as expressive as the starting one, by using it on the type $\Sigma_{z\in A}M(z)$ and then recovering the original dependent elimination constructor by means of the second projection of the indexed sum and the help of the added η conversion.

We pay attention to these weakened elimination rules since these are the rules that more directly correspond to the universal properties of the categorical constructors interpreting the corresponding type constructors. Instead, the dependent elimination rule corresponds to a universal property of the categorical constructor which concerns various fibres.

Now, we make explicit the weakened elimination rules for the quotient type, the list type and the disjoint sum type, by sketching the proof of the equivalence only for the quotient type.

In $\mathcal{T}_{ptop}$ the elimination and conversion rules of the quotient type are equivalent to a valid restricted elimination rule toward types not depending on A/R,

$$\frac{\begin{array}{l} M\ type \\ d \in A/R \qquad m(x) \in M\ [x \in A] \quad m(x) = m(y) \in M\ [x \in A, y \in A, d \in R(x,y)] \end{array}}{El_{Q\,s}(m,d) \in M}\ \mathrm{E}_s\text{-Q}$$

together with the following two conversion rules, also derivable in $\mathcal{T}_{ptop}$: one is the β-conversion

$$\frac{\begin{array}{l} M\ type \\ a \in A \qquad m(x) \in M\ [x \in A] \quad m(x) = m(y) \in M\ [x \in A, y \in A, d \in R(x,y)] \end{array}}{El_{Q\,s}(m,[a]) = m(a) \in M}\ \beta_s\text{C-Q}$$

and the other one is the η conversion stating the uniqueness of $El_{Q\,s}$:

$$\frac{p \in A/R \qquad t(z) \in M\ [z \in A/R]}{El_{Q\,s}(\,(x)\,t([x]),\,p) = t(p) \in M}\ \eta_s\text{C-Q}$$

Indeed, in order to derive the dependent elimination rule from the weakened one, given $l(x) \in L([x])\ [x \in C]$ and $l(x) = l(y) \in L([x])\ [x \in A, y \in A, d \in R(x,y)]$, we use the E_s-Q rule on $\langle [x], l(x)\rangle \in \Sigma_{z\in A/R}\,L(z)\ [x \in A]$ and we define $El_Q(l,p) \in L(p)$ for $p \in A/R$ as the second projection of the indexed sum type applied to $El_{Q\,s}((x)\langle [x], l(x)\rangle, p)$. It turns out to be well defined since by β_s and η_s conversion rules we can prove that

$$\pi_1(El_{Q\,s}((x)\langle [x], l(x)\rangle, z)) = z \in A/R\ [z \in A/R]$$

Analogously, in $\mathcal{T}_{loc}$ the elimination and conversion rules of the list type can be proved equivalent to the following:

$$\frac{\begin{array}{l} L\ type \\ s \in List(C) \qquad a \in L \qquad l(z,x) \in L\ [z \in L, x \in C] \end{array}}{El_{List\,s}(a,l,s) \in L}\ \mathrm{E}_s\text{-list}$$

$$\frac{\begin{array}{l} L\ type \\ a \in L \qquad l(z,x) \in L\ [z \in L, x \in C] \end{array}}{El_{List_s}(a,l,0) = a \in L}\ \beta_s\mathrm{C}_1\text{-list}$$

$$\frac{\begin{array}{l}L\ type\\ s \in List(C) \qquad c \in C \qquad a \in L \quad l(z,y) \in L\ [z \in L, y \in C]\end{array}}{El_{List_s}(a, l, \mathsf{cons}(s,c)) = l(El_{List_s}(a,l,s), c) \in L}\ \beta_s\text{C}_2\text{-list}$$

$$\frac{\begin{array}{l}L\ type \quad a \in L \quad l(z,y) \in L\ [z \in L, y \in C] \quad t(z) \in L\ [z \in List(C)]\\ s \in List(C) \qquad t(\epsilon) = a \in L \qquad t(\mathsf{cons}(x,y)) = l(t(x), y) \in L\ [x \in List(C), y \in C]\end{array}}{El_{List_s}(a,l,s) = t(s) \in L}\ \eta_s\text{C-list}$$

Similarly, in $\mathcal{T}_{alex}$ we can replace the elimination and the conversion rules of the natural numbers type with the weakened elimination rule and corresponding conversions since the natural numbers type can be represented as $List(\top)$.
Finally, also in $\mathcal{T}_{ext}$ the elimination and conversion rules of the disjoint sum type can be proved equivalent to the following

$$\frac{\begin{array}{l}A\ type\\ p \in C + D \qquad a_C(x) \in A\ [x \in C] \quad a_D(y) \in A\ [y \in D]\end{array}}{El_{+_s}(p, a_C, a_D) \in A}\ \text{E}_s\text{-}+$$

$$\frac{\begin{array}{l}A\ type\\ c \in C \qquad a_C(x) \in A\ [x \in C] \quad a_D(y) \in A\ [y \in D]\end{array}}{El_{+_s}(\mathsf{inl}(c), a_C, a_D) = a_C(c) \in A}\ \text{C}_{1s}\text{-}+$$

$$\frac{\begin{array}{l}A\ type\\ d \in D \qquad a_C(x) \in A\ [x \in C] \quad a_D(y) \in A\ [y \in D]\end{array}}{El_{+_s}(\mathsf{inr}(d), a_C, a_D) = a_D(d) \in A}\ \text{C}_{2s}\text{-}+$$

$$\frac{p \in C + D \qquad t(z) \in A\ [z \in C + D]}{El_{+_s}(p,\ (x)\, t(\mathsf{inl}(x))\,,\ (y)\, t(\mathsf{inr}(x))\,) = t(p) \in A}\ \eta\text{-}+$$

About the calculus of regular categories.
In order to interpret the rules of the mono existential type in $\mathcal{T}_{reg}$ into a regular category it is useful to note first that the mono existential type is actually mono, by using the elimination rule on the mono type $\mathsf{Eq}(\exists_{x\in B}C(x),\ z, w)$ for $z \in \exists_{x\in B}C(x)$ and $w \in \exists_{x\in B}C(x)$. Then, to see that mono existential types correspond to stable images, i.e. to left adjoints of pullback functors on subobjects, it is convenient to prove that the rules of the mono existential type as formulated in $\mathcal{T}_{reg}$ are equivalent to the following ones where we weaken the elimination rule to act on mono types not depending on the existential type as explained in the previous paragraph. For the mono existential type we do not need to add the β conversion since this is derivable, because it equates terms of a mono type. Moreover, instead of adding an η conversion rule, we put the stronger condition that $\exists_{x\in B}C(x)$ is mono, otherwise this would not be derivable:

$$\frac{\begin{array}{l}C(x)\ type \quad [x \in B] \quad y = z \in C(x)\ [x \in B, y \in C(x), z \in C(x)]\end{array}}{\exists_{x\in B}C(x)\ type}\ m\exists$$

$$\frac{b \in B \quad c \in C(b) \qquad \exists_{x\in B}C(x)\ type}{(b,c) \in \exists_{x\in B}C(x)}\ \text{I-m}\exists \qquad \frac{e \in \exists_{x\in B}C(x) \quad d \in \exists_{x\in B}C(x)}{e = d \in \exists_{x\in B}C(x)}\ \text{eq-m}\exists$$

$$\frac{\begin{array}{ll} M\ type & y = z \in M\ [w \in \exists_{x\in B}C(x), y \in M, z \in M] \\ d \in \exists_{x\in B}C(x) & m(x,y) \in M\ [x \in B, y \in C(x)] \end{array}}{\mathsf{Ex}_m(d,m) \in M}\ \text{E-m}\exists$$

Of course, the above rules are valid for the formulation of the mono existential type in $\mathcal{T}_{reg}$. To prove that from them we can derive the elimination rule on a mono dependent type $M(z)$ $[z \in \exists_{x\in B}C(x)]$ with $m(x,y) \in M((x,y))$ $[x \in A, y \in C(x)]$, as in the previous paragraph, we apply the elimination rule E-m$\exists$ on $\Sigma_{z\in\exists_{x\in B}C(x)}M(z)$, which is mono since both $\exists_{x\in B}C(x)$ and $M(z)$ are mono. Then, we use the second projection to define $\mathsf{Ex}(z,m) \in M(z)$. This is well defined as $\pi_1(\mathsf{Ex}_m(\, z\,,\, (x)(y)\langle (x,y)\,,\, m(x,y)\rangle\,)) = z$ for $z \in \exists_{x\in B}C(x)$ holds being $\exists_{x\in B}C(x)$ mono.

The equivalence in the definition of regular categories between stable images and stable quotients of kernel pairs suggests another equivalent internal language of regular categories:

The calculus of regular categories $\mathcal{T}_{reg}$

$=$

Lex calculus $+$

Quotient types on the terminal type

$$\frac{A\ type}{A/\top\ type}\ \text{Qtr}$$

$$\frac{a \in A}{[a] \in A/\top}\ \text{I-Qtr} \qquad \frac{a \in A \quad b \in A}{[a] = [b] \in A/\top}\ \text{eq-Qtr}$$

$$\frac{\begin{array}{ll} L(z)\ type\ [z \in A/\top] & \\ p \in A/\top & l(x) \in L([x])\ [x \in A] \quad l(x) = l(y) \in L([x])\ [x \in A, y \in A] \end{array}}{El_Q(l,p) \in L(p)}\ \text{E-Qtr}$$

$$\frac{\begin{array}{ll} L(z)\ type\ [z \in A/\top] & \\ a \in A & l(x) \in L([x])\ [x \in A] \quad l(x) = l(y) \in L([x])\ [x \in A, y \in A] \end{array}}{El_Q(l,[a]) = l(a) \in L([a])}\ \text{C-Qtr}$$

Indeed, we can prove that the quotient types on the terminal type are equivalent to the mono existential types in the typed calculus for lex categories. For this purpose, first note that from mono existential types we can derive the following stronger elimination rule for general (not necessarily mono) types and corresponding β-conversion. They express the fact that $\exists_{x\in B}C(x)$ is a quotient of $\Sigma_{x\in B}C(x)$ making all the proofs equal, that is $\exists_{x\in B}C(x) \equiv \Sigma_{x\in B}C(x)/\top$:

$$\frac{\begin{array}{ll} M(z)\ type\ [z \in \exists_{x\in B}C(x)] & m(x,y) \in M((x,y))\ [x \in B, y \in C(x)] \\ d \in \exists_{x\in B}C(x) & m(x,y) = m(z,w) \in M((x,y))\ [x \in B, z \in B, y \in C(x), w \in C(z)] \end{array}}{\mathsf{Ex}_g(d,m) \in M(d)}\ \text{E-g}\exists$$

$$\frac{\begin{array}{ll} M(z)\ \mathit{type}\ [z \in \exists_{x\in B} C(x)] & m(x,y) \in M((x,y))\ [x \in B, y \in C(x)] \\ b \in B \quad c \in C(b) & m(x,y) = m(z,w) \in M((x,y))\ [x \in B, z \in B, y \in C(x), w \in C(z)] \end{array}}{\mathsf{Ex}_g((b,c),m) = m(b,c) \in M((b,c))}\ \text{C-g}\exists$$

To prove that the elimination rule E-g$\exists$ is valid, we apply the elimination rule E-$\exists$ of $\mathcal{T}_{reg}$ on the following mono type

$$\Sigma_{w\in M(z)}\ \exists_{x\in B}\exists_{y\in C(x)}\ \mathsf{Eq}(M(z), w, m(x,y))$$

for $z \in \exists_{x\in B} C(x)$. Then, we use the first projection of the indexed sum type to define the elimination constructor.

Thanks to the validity of the rules E-g$\exists$ and C-g$\exists$ it is immediate to prove that by means of the mono existential type we can define the quotient on the terminal type by putting

$$A/\top \equiv \exists_{x\in A}\top$$

Viceversa, from the quotient type $A/\top$ we can define the existential type by putting

$$\exists_{x\in B} C(x) \equiv (\Sigma_{x\in B} C(x))/\top$$

Finally, observe that in $\mathcal{T}_{reg}$ the quotient of a kernel pair, which we would define as the quotient type $A/\mathsf{Eq}(B, f(x), f(y))$ for $f(x) \in B$ $[x \in A]$, can be defined as

$$A/\mathsf{Eq}(B, f(x), f(y)) \equiv \Sigma_{y\in B}\ (\exists_{x\in A}\mathsf{Eq}(B, f(x), y))$$

On the other hand, from the existence of the quotient type of a kernel pair we can prove the existence of the mono existential type because, by what has just been said, its existence follows from the existence of the quotient type on the terminal type, which is a particular quotient on the kernel pair of the unique term of the terminal type.

In (Awodey and Bauer 2004) there is an equivalent formulation of the calculus for regular categories with bracket types $[A]$ for a type A corresponding to our quotients on terminal type, i.e. $[A] \equiv A/\top$ (or $[A] \equiv \exists_{x\in A}\top$). There, the elimination rule for bracket types, which acts on not dependent types, can be enforced to act on dependent types and -once this has been done- the equality rule between terms of $[A]$ can be weakened to the equality rule only on introductory elements, as in our equivalent formulations of the rules for $\exists_{x\in A}\top$.

About disjointness.
Observe that in the calculi for lextensive categories, locoi, pretopoi, Heyting pretopoi and arithmetic universes, the disjointness axiom is not derivable from the other rules. Indeed, for each calculus we can obtain a model that falsifies disjointness by using a domain with only one element (see (Smith 1988)), where the quotient type A/R is interpreted simply as A.

Comparison with the many-sorted language of topoi.
The many sorted internal language of topoi in (Lambek and Scott 1986) (also called higher order logic) has *two kinds* of syntactic entities: sorts which are represented by

simple types with corresponding terms

$$A\ type \qquad a : A\ [\Gamma]$$

and formulas which are terms of the subobject classifier Ω not equipped with proof-terms and depending on sorts

$$\phi(x) \in \Omega\ [x \in A]$$

Instead, in our calculus $\mathcal{T}_{top}$ for topoi there is only *one kind* of syntactic entities, namely dependent types $B(x_1, ..., x_n)\ type\ [x_1 \in A_1, ..., x_n \in A_n]$ with corresponding terms and equalities organized in the four kinds of judgements

$$A\ type\ [\Gamma] \qquad A = B\ [\Gamma] \qquad a \in A\ [\Gamma] \qquad a = b \in A\ [\Gamma]$$

Categorically, the internal language as a many-sorted logic captures the one dimensional properties of a topos in terms of a simple type theory and the properties of the subobject fibration in terms of logical formulas (and hence two interpretations are necessary: one for types and corresponding terms as objects and morphisms of the category, and the other one for formulas as terms of the subobject classifier, or equivalently as subobjects).

Instead, as we will see in section 5, our internal dependent typed language of topoi captures the properties of the codomain fibration by regaining the properties of the subobject fibration via mono types, since these are interpreted by monomorphisms.

If we look at the typed calculus $\mathcal{T}_{top}$ only from the type perspective, $\mathcal{T}_{top}$ is an extension of first order Martin-Löf's extensional type theory in (Martin-Löf 1984), this being the calculus $\mathcal{T}_{lcc}$ for locally cartesian closed categories.

But if we consider that in a topos propositions corresponds to subobjects, then in $\mathcal{T}_{top}$ we need to interpret the logic according to *formulas as mono types*. Following this isomorphism, the universal quantification is represented by the forall type, like in the calculus for Heyting pretopoi (we recall that a topos is an Heyting pretopos (Mac Lane and Moerdijk 1992)!), which can be defined in $\mathcal{T}_{top}$ as the dependent product type, since the dependent product of a mono type remains mono. Analogously, the existential quantifier corresponds to the mono existential type, as in the calculus for regular categories, and it can be defined by means of the quantification on the Omega type as follows:

$$\exists y \in B\ C(y) \equiv \Pi_{p\in\Omega}\ (\Pi_{y\in B}(C(y) \to \mathsf{Eq}(\Omega, \{\top\}, p)))\ \to \mathsf{Eq}(\Omega, \{\top\}, p)$$

Hence, the existential quantifier does not coincide with the indexed sum type as in Martin-Löf's type theory. Indeed, as in $\mathcal{T}_{reg}$, we can prove that it is a quotient of the indexed sum type over the terminal type:

$$\exists y \in B\ C(y) \equiv (\Sigma y \in B\ C(y))/\top$$

where the quotient type on the terminal type can be defined as $A/\top \equiv \exists x \in A\ \top$.

As a consequence, for example, we can realize why in any topos only the axiom of unique choice holds while the full one does not (see below). Hence, from the logic point of view we can not think of $\mathcal{T}_{top}$ as an extension of Martin-Löf's type theory.

Axiom of choice in the typed calculus for topoi.
In the calculus for locally cartesian closed categories $\mathcal{T}_{lcc}$, and hence in that for topoi,

we can easily derive the distributivity property of the dependent product type with the indexed sum type as in Martin-Löf's extensional type theory (Martin-Löf 1984):

$$(\Pi x \in A)(\Sigma y \in B)\ C(x,y) \to (\Sigma f \in A \to B)(\Pi x \in A)\ C(x, f(x))$$

where we recall that $A \to B \equiv (\Pi x \in A)B$.

According to the isomorphism formulas as types by Martin-Löf the above distributivity property is recognized as the propositional axiom of choice. However, this is not recognized as such if we follow *formulas as mono types* as in the logic of a topos. In fact, in topoi or Heyting pretopoi, the propositional axiom of choice should be instead expressed with the forall type and the mono existential type acting only on a mono type $C(x,y)$:

$$(\forall x \in A)(\exists y \in B)\ C(x,y) \to (\exists f \in A \to B)(\forall x \in A)\ C(x, f(x))$$

Now, recalling that the mono existential type is a quotient of the indexed sum type, that is $\exists y \in B\ C(x,y) \equiv (\Sigma y \in B\ C(x,y))/\top$, then the propositional axiom of choice can be rewritten as

$$(\forall x \in A)\,(\Sigma y \in B\ C(x,y))/\top \to (\,(\Sigma f \in A \to B)(\forall x \in A)\ C(x, f(x))\,)/\top$$

It is well known, as proved by Diaconescu (Diaconescu 1975) (see also (Lambek and Scott 1986; Mac Lane and Moerdijk 1992)), that the above propositional axiom of choice can not be generally derived in a topos because it makes the logic classical. From the above formulation in the dependent typed calculus of topoi we can perceive why this is so: indeed, to prove the axiom of choice we would need to access to a proof of $\Sigma y \in B\ C(x,y)$ from $(\Sigma y \in B\ C(x,y))/\top$, but we can not unless we have a choice operator from the quotient $A/\top$ to A.
Instead, in a topos (and also in a Heyting pretopos) we can derive the axiom of unique choice written as

$$\begin{aligned}(\forall x \in A)\quad (\Sigma y \in B\ C(x,y))/\top\ \ \wedge\ \ \forall y \in B\ \ \forall z \in B\ \ C(x,y) \wedge C(x,z) \to \mathsf{Eq}(B,y,z)\\ \to\ ((\Sigma f \in A \to B)(\forall x \in A)\ C(x, f(x)))/\top\end{aligned}$$

where $A \wedge B \equiv A \times B$ with A, B mono types. Indeed, in this case $\Sigma y \in B\ C(x,y)\ [x \in A]$ is a mono type and becomes isomorphic to its quotient on the terminal type

$$(\Sigma y \in B\ C(x,y))/\top\ \simeq\ \Sigma y \in B\ C(x,y)$$

Therefore, we can prove the axiom of unique choice as we prove the axiom of choice in Martin-Löf's type theory.

The Omega type seen as a quotient type.
The Omega type corresponds to the subobject classifier. Given that monomorphisms correspond to mono types, then the Omega type encodes mono types, which represent propositions, up to isomorphisms. Hence, the impredicativity of a topos is restricted to mono types and the Omega type is not necessarily itself mono to avoid inconsistencies.
In the Omega type mono types are encoded up to equiprovability. Indeed, the Omega type can be seen as the quotient of an intensional classifier of mono types over the

equiprovability relation. But before showing this, we explain why in the formulation of the Omega type an elimination rule is not present and where the β and η conversion rules come from. For this purpose we consider the following alternative formulation of the rules for the Omega type: recall that $B \leftrightarrow C \equiv B \to C \times C \to B$

Alternative formulation of the Omega type

$$\frac{}{\Omega\ type}\ \mathrm{F}$$

$$\frac{B\ type \quad y = z \in B\ [y \in B, z \in B]}{\{B\} \in \Omega}\ \mathrm{I} \qquad \frac{\begin{array}{ll} B\ type & y = z \in B\ [y \in B, z \in B] \\ C\ type & y = z \in C\ [y \in C, z \in C] \\ f \in B \leftrightarrow C & \end{array}}{\{B\} = \{C\} \in \Omega}\ \mathrm{eq}$$

$$\frac{q \in \Omega}{T(q)\ type}\ \mathrm{E} \qquad \frac{q \in \Omega \quad c \in T(q) \quad d \in T(q)}{c = d \in T(q)}\ \text{eq-E}$$

$$\frac{B\ type \quad y = z \in B\ [y \in B, z \in B]}{\langle \mathsf{r}_B, \mathsf{r}_B^{-1} \rangle \in T(\{B\}) \leftrightarrow B}\ \beta\text{-C} \qquad \frac{q \in \Omega}{\{T(q)\} = q \in \Omega}\ \eta\text{-C}$$

In the above alternative formulation of the Omega type its elimination rule is explicit. Now, if we add the above rules to $\mathcal{T}_{lcc}$, then for every $q \in \Omega$ we can derive a proof term of $T(q) \leftrightarrow \mathsf{Eq}(\Omega, q, \{\top\})$ because $q = \{\top\} \in \Omega$ is provable if and only if $T(q)$ is provable. Hence, we can put

$$T(q) \equiv \mathsf{Eq}(\Omega, q, \{\top\})$$

Then, we can simply keep only the corresponding β and η conversion rules which are equivalent to those in the original formulation of the Omega type. In particular, the βC-Ω rule follows from the β-C rule by putting $\mathsf{ch}(B) \equiv r_B(\mathsf{eq})$. Viceversa, the β-C rule follows from the βC-Ω rule since for any mono type B we can derive

$$\lambda x.\mathsf{ch}(B) \in \mathsf{Eq}(\Omega, \{B\}, \{\top\}) \to B$$

whose inverse is $\lambda z.\mathsf{eq} \in B \to \mathsf{Eq}(\Omega, \{B\}, \{\top\})$.

Now, we show that the Omega type is a quotient of an intensional Omega type whose rules are the intensional version of those in the above alternative formulation of the Omega type. More precisely, we show that the rules of $\mathcal{T}_{top}$ can be derived in an extension of $\mathcal{T}_{lcc}$ with extensional effective quotients restricted to mono equivalence relations, as in the type theory of pretopoi $\mathcal{T}_{ptop}$, and with an intensional Omega type encoding mono

types as follows:

The intensional Omega type

$$\frac{}{\Omega^i \ type}\ \mathrm{F}$$

$$\frac{B\ type \quad y = z \in B\ [y \in B, z \in B]}{c(B) \in \Omega^i}\ \mathrm{I} \qquad \frac{\begin{array}{l} B\ type \quad y = z \in B\ [y \in B, z \in B] \\ C\ type \quad y = z \in C\ [y \in C, z \in C] \\ B = C \end{array}}{c(B) = c(C) \in \Omega^i}\ \mathrm{eq}$$

$$\frac{p \in \Omega^i}{D(p)\ type}\ \mathrm{E} \qquad \frac{p \in \Omega^i \quad e \in D(p) \quad d \in D(p)}{e = d \in D(p)}\ \text{eq-E}$$

$$\frac{B\ type \quad y = z \in B\ [y \in B, z \in B]}{D(c(B)) = B}\ \beta\text{C-}\Omega^i \qquad \frac{p \in \Omega^i}{c(D(p)) = p \in \Omega^i}\ \eta\text{C-}\Omega^i$$

In this extension we can show that the Omega type is the quotient of the intensional Omega type Ω^i under the relation of equiprovability between elements of Ω^i:

$$\Omega \equiv \Omega^i / \leftrightarrow$$

where for $d_1 \in \Omega^i$ and $d_2 \in \Omega^i$ we define $d_1 \leftrightarrow d_2 \equiv D(d_1) \leftrightarrow D(d_2)$.

Indeed, we can see that the equality rule of the Omega type corresponds to the usual rule of equality for canonical terms of a quotient type and that the βC-Ω rule is equivalent to the following rule expressing effectiveness of the Omega type seen as a quotient:

$$\frac{\{B\} = \{C\} \in \Omega}{\mathsf{eff}(B, C) \in B \leftrightarrow C}\ \text{eff-}\Omega$$

Finally, the ηC-Ω rule can be proved by finding a proof of the corresponding propositional equality by means of the elimination rule of the quotient type $\Omega^i / \leftrightarrow$ since we can derive $B \leftrightarrow \mathsf{Eq}(\Omega, \{B\}, \{\top\})$ for any mono type B (for more details see (Maietti 1998b)).

4. The modular correspondence categorical property/type constructor

Here, we describe the correspondence between universal categorical properties and type constructors.

Observe that from the fact that such a correspondence is modular on lex categories, beside the calculi presented previously, we can then deduce the dependent type theory of other lex categories enjoying a combination of the categorical properties in the table below by combining the corresponding type constructors. For example, the type theory of locally cartesian closed pretopoi is the calculus of pretopoi with extensional dependent product types, and the type theory of locally cartesian closed categories with stable finite

coproducts is the calculus of locally cartesian closed categories with the false type and disjoint sum types.

Categorical properties	**Type-theoretic constructors**
Finite limits	terminal type extensional equality type indexed sum types
+	
stable coproducts stable initial object + coproduct disjointness	disjoint sum types false type + disjointness axiom
stable images	mono existential type
stable quotients of kernel pairs	extensional quotient types on the terminal type
stable quotients of monic equivalence relations + quotient effectiveness	extensional quotient types on mono equivalence relations + effectiveness axiom
parameterized natural numbers object	natural numbers type
parameterized list objects	list types
right adjoints to pullback functors	extensional dependent product types
right adjoints to pullback functors restricted to subobjects	forall types
subobject classifier	omega type

5. The categorical semantics

The seminal idea of the semantics goes back to (Seely 1984; Lawvere 1969; Lawvere 1970) and consists in interpreting a dependent typed calculus by means of the codomain fibration (Jacobs 1999). There is a readable informal account of the interpretation of the typed calculus for locally cartesian closed categories in (Johnstone 2002). A reader acquainted with that treatment will more readily understand what we do here.

Making sound the interpretation for dependent type theories requires one to deal with issues of coherence. In principle this can be dealt with using the general result of Power (Power 1989) based on the setting of Blackwell-Kelly-Power (Blackwell *et al.* 1989). But that is indirect and uses very heavy machinery; it seems good to have a simple direct treatment.

Our notion of model for a dependent typed calculus, first presented in (Maietti 1998b), combines the semantics based on display maps (Seely 1984; Hyland and Pitts 1989) together with the tools provided by contextual categories to interpret substitution correctly (Cartmell 1986). In particular, our models can be organized into contextual categories by means of the split fibration associated to the codomain fibration. As treated in (Hofmann 1995), the use of a split fibration is required to interpret substitution correctly, since the natural use of the codomain fibration as in (Seely 1984) gives rise to coherence problems.

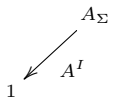
While the validity and completeness theorem is straightforward for contextual categories in general, the same theorem with respect to our particular contextual categories requires more care. However, we want to point out that it holds, hence *in order to prove the validity and completeness theorem between a dependent type theory and corresponding categories $\mathcal{C}$ it is sufficient to use models based on the split fibration of the codomain fibration of a category $\mathcal{C}$.* Moreover, we will prove a stronger link than the validity and completeness theorem, namely that the calculi considered in this paper provide internal languages of the corresponding categorical structures they are supposed to describe. Finally, we can also build free categorical structures by means of their internal dependent type theory.

5.1. *The interpretation and the validity*

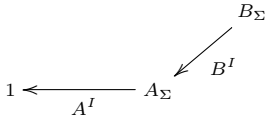
Before defining the details of the interpretation of a typed calculus in the corresponding category, that is supposed to model it, we explain the general idea of the interpretation and the coherence problem encountered to interpret substitution.

Let us suppose that we want to interpret the dependent typed calculus $\mathcal{T}_{lex}$ in a lex category $\mathcal{C}$. The idea is to interpret the type judgement $B\ [\Gamma]$ as a suitable sequence of morphisms of $\mathcal{C}$ to the terminal object 1 and the judgement $b \in B[\Gamma]$ as a section of the last morphism of the sequence interpreting the dependent type B under a context.

In particular, a closed type $A\ [\ ]$ will be interpreted as the unique morphism $A^I = !_{A_\Sigma}$ from an object A_Σ of $\mathcal{C}$ (interpreting A only) toward the terminal object 1 of $\mathcal{C}$ (interpreting the empty context):

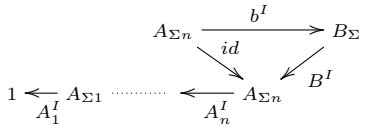


Then, a dependent type $B(x)\ [x \in A]$ will be interpreted as the sequence of two arrows

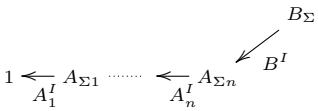


while in general a term $b \in B(x_1, ..., x_n)\ [x \in A_1, .., x_n \in A_n]$ will be interpreted as a

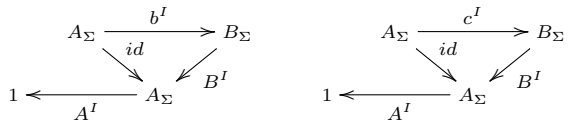
section of B^I



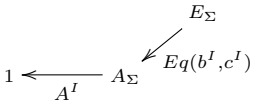
provided that $B(x_1, ..., x_n)$ $[x \in A_1, .., x_n \in A_n]$ is interpreted as



For example, supposing that $b \in B$ $[x \in A]$ and $c \in B$ $[x \in A]$ are interpreted as



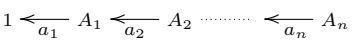
the idea is to interpret the extensional equality type $\mathsf{Eq}(B, b, c)$ $[x \in A]$ as



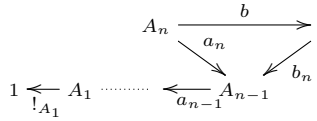
where $Eq(b^I, c^I)$ is the equalizer of b^I and c^I.

To place the interpretation into a category we define the following category of path-graphs of $\mathcal{C}$:

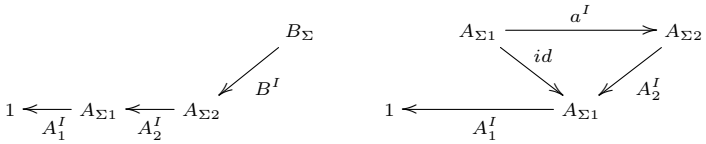
Def. 5.1. Given a category $\mathcal{C}$ with terminal object 1, the objects of the category $\boldsymbol{Pgr(\mathcal{C})}$ are finite sequences $a_1, a_2, ..., a_n$ of morphisms of $\mathcal{C}$



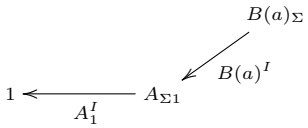
and a morphism from $a_1, a_2, ..., a_n$ to $b_1, b_2, ..., b_m$ is a morphism b of $\mathcal{C}$ such that $b_n \cdot b = a_n$ provided $n = m$ and $a_i = b_i$ for $i = 1, ..., n-1$.



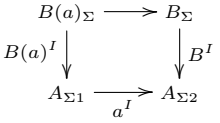
Then, in order to interpret substitution we want to use pullbacks as follows. Given a dependent type $B(x_1, x_2)$ $[x_1 \in A_1, x_2 \in A_2]$ and a term $a \in A_2$ $[x_1 \in A_1]$ interpreted as



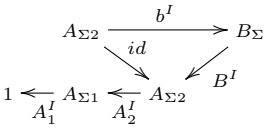
then the idea is to interpret $B(x_1, x_2)[x_2/a] \equiv B(x_1, a)\ [x_1 \in A_1]$ as



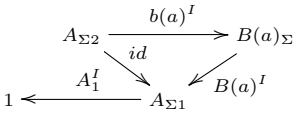
where $B(a)^I$ is the first projection of the pullback



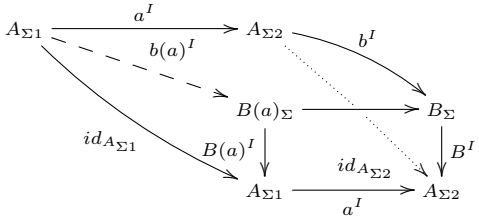
Analogously, given a term $b(x_1, x_2) \in B(x_1, x_2)\ [x_1 \in A_1, x_2 \in A_2]$ interpreted as



the idea is to interpret $b(x_1, a) \in B(x_1, a)\ [x_1 \in A_1]$ as



where $b(a)^I$ is the unique morphism induced by $id_{A_{\Sigma 1}}$ and $b^I \cdot a^I$ toward the vertex of the pullback of B^I along a^I



However we can immediately realize a coherence problem: not for all choices of pullback the pullback of B^I with the identity has the identity as first projection. This property is needed to validate $B(x)[x/x] = B(x)$. Moreover, we need a functorial choice of pullback in order to validate $(B(y)[y/a(x)])[x/c]) = B(y)[y/a(c)]$ and hence to interpret the substitution of a term both in types and in terms correctly.

Abstractly, the situation can be rephrased by saying that we wanted to interpret the calculus by using the codomain fibration associated to the category $\mathcal{C}$. But we encountered the problem that the reindexing pullback pseudofunctor associated to the codomain fibration - and used to interpret substitution of term into types (and terms) - is *not* a functor. However, there is a way out (for a general solution to coherence problems see (Power 1989)). One solution (following (Hofmann 1995)) is to use the reindexing functor $S : \mathcal{C}^{OP} \longrightarrow Cat$ associated to the split fibration of the codomain fibration (for more

details see (Benabou 1985; Jacobs 1999)). Since the fibres of S are equivalent to those of the pullback pseudofunctor we can consider the functor S as the correction of the pullback pseudofunctor into a functor! This reindexing functor associates to any object A in $\mathcal{C}$ the category $S(A) \equiv Fib(\mathcal{C}/A, \mathcal{C}^{\to})$ of fibred functors and natural transformations.

Before proceeding with the definition of the category of fibred functors, we first recall the definition of the arrow category $\mathcal{C}^{\to}$ of a given category $\mathcal{C}$: the objects of $\mathcal{C}^{\to}$ are the $\mathcal{C}$-morphisms $X \xrightarrow{\phi} A$ and the morphisms of $\mathcal{C}^{\to}$ from $X \xrightarrow{\phi} A$ to $X \xrightarrow{\psi} A$ are pairs of $\mathcal{C}$-morphisms $f : X \to Y$ and $u : A \to B$ such that the following diagram commutes:

$$\begin{array}{ccc} X & \xrightarrow{f} & Y \\ {\scriptstyle\phi}\downarrow & & \downarrow{\scriptstyle\psi} \\ A & \xrightarrow[u]{} & B \end{array}$$

Then we give the definition of fibred functor indexed on an object in a lex category:

Def. 5.2. Given a lex category $\mathcal{C}$, and an object A in $\mathcal{C}$, a fibred functor $\sigma : \mathcal{C}/A \to \mathcal{C}^{\to}$ is a functor from the slice category $\mathcal{C}/A$ to the category $\mathcal{C}^{\to}$ which sends cartesian morphisms of the domain fibration $dom_{\mathcal{C}}$ to cartesian morphisms of the codomain fibration $cod_{\mathcal{C}}$, (see (Jacobs 1999) for corresponding definitions), i.e. σ associates to every commutative triangle $\begin{array}{ccc} C & \xrightarrow{t} & B \\ & {\scriptstyle b'}\searrow \quad \swarrow{\scriptstyle b} & \\ & A & \end{array}$ a pullback diagram $\begin{array}{ccc} C' & \xrightarrow{q(t,\sigma(b))} & B' \\ {\scriptstyle\sigma(b\cdot t)}\downarrow & & \downarrow{\scriptstyle\sigma(b)} \\ C & \xrightarrow[t]{} & B \end{array}$

Then, we define the category of fibred functors related to an object of a lex category $\mathcal{C}$:

Def. 5.3. For every object A of a lex category $\mathcal{C}$, we call $Fib(\mathcal{C}/A, \mathcal{C}^{\to})$ the category of fibred functors $\sigma : \mathcal{C}/A \to \mathcal{C}^{\to}$ whose morphisms from σ to τ are natural transformations ρ such that for every $b : B \to A$ the second member of $\rho(b)$ is the identity (recall that $\rho(b)$ is a morphism of $\mathcal{C}^{\to}$), that is the triangle $\begin{array}{ccc} & \xrightarrow{\rho_1(b)} & \\ {\scriptstyle\sigma(b)}\searrow & & \swarrow{\scriptstyle\tau(b)} \\ & B & \end{array}$ commutes.

In the following when we speak of a fibred functor σ in $Fib(\mathcal{C}/A, \mathcal{C}^{\to})$ we call $i(\sigma) = A$ the object indexing σ.

Note that we can specify the definition of a fibred functor only on objects of $\mathcal{C}/A$, since its action on $\mathcal{C}/A$-morphisms is determined by the pullback property, once we know that the defined object part fits well into a pullback. Therefore, in the following *to define a fibred functor we will only specify its action on objects.*

Moreover, observe also that any component $\rho(b)$ of a morphism ρ of $Fib(\mathcal{C}/A, \mathcal{C}^{\to})$ is determined by $\rho(id_A)$ thanks to naturality of ρ and the properties of pullbacks. Indeed, if we consider $\begin{array}{ccc} B & \xrightarrow{b} & A \\ & {\scriptstyle b}\searrow \quad \swarrow{\scriptstyle id} & \\ & A & \end{array}$, we get that $\rho(b)$ from $\sigma(b)$ to $\tau(b)$ is equal to the unique morphism to the pullback of $\tau(id)$ along b, according to the functorial choice of pull-

backs of τ, induced by $\sigma(b)$ and $\rho(id_A) \cdot q(b, \sigma(id))$, which we indicate with the notation $\langle \sigma(b),_A \ \rho(id_A) \cdot q(b, \sigma(id)) \rangle$. Therefore, in the following *to define a natural transformation between fibred functors indexed on A we will only specify its action on the identity of A.*

Now, how can fibred functors help to interpret the calculus? To answer this question, we show how the category of fibred functors $Fib(\mathcal{C}/A, \mathcal{C}^{\to})$ is equivalent to the slice category $\mathcal{C}/A$ (that is the fibres of S are equivalent to the fibres of the pullback pseudofunctor).

The functor

$$\widehat{(-)} : \mathcal{C}/A \to Fib(\mathcal{C}/A, \mathcal{C}^{\to})$$

establishes one side of the equivalence, by associating to an object $b : B \to A$ of $\mathcal{C}/A$ the fibred functor $\widehat{b}$, defined as $\widehat{b}(t) \equiv t^*(b)$, namely the first projection of the pullback of b along t $\begin{array}{ccc} B_t & \xrightarrow{b^*(t)} & B \\ {\scriptstyle t^*(b)}\downarrow & & \downarrow{\scriptstyle b} \\ D & \xrightarrow[t]{} & A \end{array}$ for every $t : D \to A$. Then, $\widehat{(-)}$ is extended to morphisms by the universal property of pullback. For every $a : C \to A$ and $b : B \to A$, the morphism part of $\widehat{(-)}$ associates to every $\begin{array}{ccc} C & \xrightarrow{g} & B \\ & {\scriptstyle a}\searrow \quad \swarrow{\scriptstyle b} & \\ & A & \end{array}$ the natural transformation $\overline{g} : \widehat{a} \to \widehat{b}$ defined in this way: for every $t : D \to A$, we put $\overline{g}(t) \equiv \langle \widehat{a}(t),_A \ g \cdot a^*(t) \rangle)$ which is the unique morphism to the pullback of b along t induced by $\widehat{a}(t)$ and $g \cdot a^*(t)$.

The other side of the equivalence is established by the functor

$$(-)(id) : Fib(\mathcal{C}/A, \mathcal{C}^{\to}) \to \mathcal{C}/A$$

which is defined exactly as the evaluation of a fibred functor and of a natural transformation on the identity on A.

The above equivalence suggests that instead of interpreting a type $B(x)$ $[x \in A]$ directly into the slice category $\mathcal{C}/A_\Sigma$ - supposing to simplify the interpretation of a closed type A into an object A_Σ of $\mathcal{C}$ - we can preinterpret it into a fibred functor $\beta : \mathcal{C}/A_\Sigma \to \mathcal{C}^{\to}$. The idea is that the evaluation of β on the identity, that is $\beta(id_{A_\Sigma})$, represents its interpretation with the advantage that the fibred functor provides also how to interpret $B(x)$ $[x \in A]$ after any possible substitution:

$$\begin{array}{ccc} B(a)_\Sigma(& \xrightarrow{q(a^I, \beta(id))} & B_\Sigma \\ {\scriptstyle B(a)^I = \beta(a)}\downarrow & & \downarrow{\scriptstyle B^I = \beta(id)} \\ C_\Sigma & \xrightarrow[a^I]{} & A_\Sigma \end{array}$$

More in general a type $B(x_1, ..., x_n)$ $[x \in A_1, .., x_n \in A_n]$ will be preinterpreted in a sequence of fibred functors

$$\alpha_1 \, , \, \alpha_2 \, , ..., \, \alpha_n \, , \, \beta$$

with $1 = i(\alpha_1)$ and $A_{\Sigma i-1} = i(\alpha_i)$ for $i = 2, ..., n$ and $i(\beta) = A_{\Sigma n}$, and then interpreted as

$$1 \xleftarrow[\alpha_1(id)]{} A_{\Sigma 1} \cdots\cdots \xleftarrow[\alpha_n(id)]{} A_{\Sigma n} \xleftarrow[\beta(id)]{} B_\Sigma$$

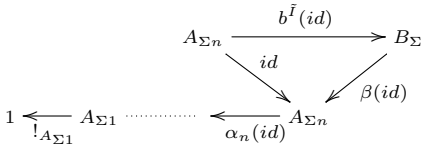
This idea suggests to define a category of path-graphs of fibred functors analogously to $Pgr(\mathcal{C})$ provided that their evaluation on the identity gives an object of $Pgr(\mathcal{C})$:

Def. 5.4. Given a lex category $\mathcal{C}$, the objects of the category $\boldsymbol{Pgf(\mathcal{C})}$ are finite sequences $\sigma_1, \sigma_2, ..., \sigma_n$ of fibred functors $\sigma_i : \mathsf{Ob}Fib(\mathcal{C}/A_i, \mathcal{C}^{\to})$ with $1 = i(\sigma_1)$ and $A_{i-1} \equiv i(\sigma_i)$ for $i = 2, ..., n$ such that $\sigma_1(id_1), \sigma_2(id_{A_1}), ..., \sigma_n(id_{A_{n-1}})$ is an object of $Pgr(\mathcal{C})$. The morphisms of $Pgf(\mathcal{C})$ from $\sigma_1, \sigma_2, ..., \sigma_n$ to $\tau_1, \tau_2, ..., \tau_m$ are defined only if $n = m$ and $\sigma_i = \tau_i$ for $i = 1, ..., n-1$ and $\sigma_n, \tau_n \in \mathsf{Ob}Fib(\mathcal{C}/A_{n-1}, \mathcal{C}^{\to})$ and they are natural transformations ρ from the fibred functor σ_n to τ_n.

Then, we will use morphisms of $Pgf(\mathcal{C})$ to preinterpret dependent terms. Before giving the precise idea of the preinterpretation we define the fibred functor $i_A : \mathcal{C}/A \to \mathcal{C}^{\to}$ on the objects as follows: for any $t : D \to A$

$$i_A(t) \equiv id_D$$

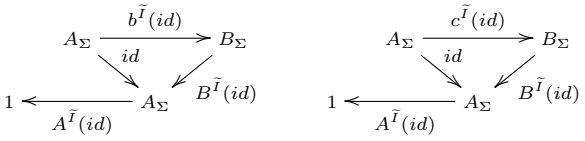
A term $b \in B(x_1, ..., x_n)$ $[x \in A_1, .., x_n \in A_n]$ will be preinterpreted as a natural transformation $b^{\tilde{I}}$ from $i_{A_{\Sigma n}}$ to β, that is a morphism from from α_1 , α_2 , ..., α_n , $i_{A_{\Sigma n}}$ to α_1 , α_2 , ..., α_n , β in $Pgf(\mathcal{C})$, and hence interpreted as $b^{\tilde{I}}(id)$, that is a section of $\beta(id)$



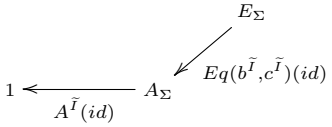
provided that the type judgement $B(x_1, ..., x_n)$ $[\Gamma_n]$ is interpreted as $1 \xleftarrow[\alpha_1(id)]{} A_{\Sigma 1} \cdots\cdots \xleftarrow[\alpha_n(id)]{} A_{\Sigma n} \xleftarrow[\beta(id)]{} B_\Sigma$. Also for the terms, the idea is that the natural transformation $b^{\tilde{I}}$ interprets the term under all the possible substitutions.

In practice we pass from the preinterpretation of a type or of a term in $Pgf(\mathcal{C})$ to its interpretation by the functor $(-)(id)$ in $Pgr(\mathcal{C})$.

For example, given $b \in B$ $[x \in A]$ and $c \in B$ $[x \in A]$ preinterpreted by the natural transformation $a^{\tilde{I}}$ and $b^{\tilde{I}}$ and interpreted as



the idea is to preinterpret the extensional equality type $\mathsf{Eq}(B, b, c)$ $[x \in A]$ as the sequence $A^{\tilde{I}}$, $Eq(b^{\tilde{I}}, c^{\tilde{I}})$ in $Pgf(\mathcal{C})$ and to interpret it as



where $Eq(b^{\tilde{I}}, c^{\tilde{I}}) : \mathcal{C}/A_\Sigma \to \mathcal{C}^{\to}$ is the fibred functor defined on objects as follows (recall

that a fibred functor is determined by its action on objects): for any $t : D \to A_\Sigma$

$$Eq(\, c^{\widetilde{I}}\,,\, d^{\widetilde{I}}\,)(t) \;\equiv\; eq(\, c^{\widetilde{I}}(t)\,,\, d^{\widetilde{I}}(t)\,)$$

that is the chosen equalizer in $\mathcal{C}$ of $c^{\widetilde{I}}(t)$ and $d^{\widetilde{I}}(t)$. This is a well defined fibred functor since equalizers are stable under pullbacks.

Moreover, to preinterpret basic types and their terms, we will make use of the functor $\widehat{(-)}$. For example, we will preinterpret the False type into the fibred functor $\widehat{(!_0)}$ where $!_0 : 0 \to 1$ is the unique morphism from the initial object 0 of $\mathcal{C}$ to the terminal object 1.

To preinterpret weakening and substitution we will make use of the morphism part of the reindexing functor $S : \mathcal{C}^{OP} \longrightarrow Cat$, which is defined as follows: for a morphism $f : B \to A$ of $\mathcal{C}$, the functor $S(f) : Fib(\mathcal{C}/A, \mathcal{C}^{\to}) \to Fib(\mathcal{C}/B, \mathcal{C}^{\to})$ associates to every fibred functor σ a fibred functor $\sigma[f]$ defined as $\sigma[f](t) \equiv \sigma(f \cdot t)$ for every morphism $t : C \to B$. And $S(f)$ associates to any natural transformation ρ the natural transformation $S(f)(\rho) \equiv \rho[f]$ defined as $\rho[f](t) \equiv \rho(f \cdot t)$ for every $t : C \to B$. Observe that S is the reindexing functor with respect to the Grothendieck fibration $p : Gr(S) \to \mathcal{C}$, which is the projection of the Grothendieck completion of the functor S (see (Jacobs 1999) for definitions) and is equivalent to the codomain fibration.

In more detail, we define a preinterpretation $\widetilde{\mathcal{I}_\mathcal{C}} : \mathcal{T} \longrightarrow Pgf(\mathcal{C})$ for each typed calculus $\mathcal{T}$ presented in section 3 into the category $Pgf(\mathcal{C})$, where $\mathcal{C}$ is a categorical structure that the calculus is supposed to describe. For example for $\mathcal{T}_{top}$ we suppose that $\mathcal{C}$ is a topos.

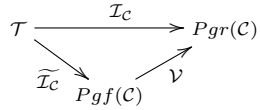
Since the preinterpretation essentially says how to interpret a dependent type and a typed term after any possible substitution, then, we define the interpretation of type and term judgements as the evaluation of their preinterpretations on the identical substitution by means of $\mathcal{V} : Pgf(\mathcal{C}) \longrightarrow Pgr(\mathcal{C})$ defined in this manner:

$$\mathcal{V}(\sigma_1, \sigma_2, ..., \sigma_n) \equiv \sigma_1(id_1), \sigma_2(id_{A_1}), ..., \sigma_n(id_{A_{n-1}}) \qquad \mathcal{V}(\rho) \equiv \rho(id_{A_{n-1}})$$

for every $Pgf(\mathcal{C})$-object $\sigma_1, \sigma_2, ..., \sigma_n$ with $1 = i(\sigma_1)$ and $A_{i-1} = i(\sigma_i)$ for $i = 2, ..., n$ and for every $Pgf(\mathcal{C})$-morphism ρ from $\sigma_1, \sigma_2, ..., \sigma_n$ to $\tau_1, \tau_2, ..., \tau_n$. Hence, the interpretation

$$\mathcal{I}_\mathcal{C} : \mathcal{T} \longrightarrow Pgr(\mathcal{C})$$

is defined as $\mathcal{I}_\mathcal{C} \equiv \mathcal{V} \cdot \widetilde{\mathcal{I}_\mathcal{C}}$.



In defining the preinterpretation we need to overcome a difficulty, namely that we can not define it by induction on the derivations of type formation and of term introduction and elimination as we do with simple type theory. Indeed, in the presence of type dependencies some typing rules require the validity of equality judgements, like for example the type equality rule *conv*) or the elimination rule of the extensional propositional equality type, or the formation rule of the forall type or the introduction rule of the Omega type. We overcome this difficulty following the approach in (Streicher 1991) by defining an *a priori partial preinterpretation*

$$\widetilde{\mathcal{I}_\mathcal{C}} : pseudo(\mathcal{T}) \longrightarrow Pgf(\mathcal{C})$$

on the pseudo-judgements of a dependent typed calculus $\mathcal{T}$ - that are raw types, raw equal types, raw terms and raw equal terms (under a context) obtained from the signature associated to each dependent typed calculus in section 3 - by induction on their complexity (Pitts 2000; Streicher 1991). Only when we prove the validity theorem we also prove that the preinterpretation is well defined on the type and term judgements derivable in the theory (see (Maietti 1998b) for more details).

Def. 5.5 ((Pre)interpretation). A *dependent type* (pseudo)judgement

$$B(x_1, ..., x_n)\ [x_1 \in A_1, ..., x_n \in A_{n-1}(x_1, ..., x_{n-1})]$$

is preinterpreted as an object of $Pgf(\mathcal{C})$

$$\alpha_1\ ,\ \alpha_2\ , ...,\ \alpha_n\ ,\ \beta$$

and then interpreted as $1 \xleftarrow[\alpha_1(id)]{} A_{\Sigma 1} \cdots\cdots \xleftarrow[\alpha_n(id)]{} A_{\Sigma n} \xleftarrow[\beta(id)]{} B_\Sigma$.

The *equality between types* is preinterpreted as equality between objects of $Pgf(\mathcal{C})$ and hence interpreted as the equality between objects of $Pgr(\mathcal{C})$.

A term (pseudo)judgement $b \in B(x_1, ..., x_n)\ [\Gamma_n]$ is preinterpreted as a natural transformation $b^{\widetilde{I}}$ from $\alpha_1\ ,\ \alpha_2\ , ...,\ \alpha_n\ ,\ i_{A_{\Sigma n}}$ to $\alpha_1\ ,\ \alpha_2\ , ...,\ \alpha_n\ ,\ \beta$, and then interpreted as $b^{\widetilde{I}}(id)$, that is a section of $\beta(id)$

$$\begin{array}{ccccc} & & A_{\Sigma n} & \xrightarrow{b^{\widetilde{I}}(id)} & B_\Sigma \\ & & & \searrow^{id} \quad \swarrow_{\beta(id)} & \\ 1 & \xleftarrow[!_{A_{\Sigma 1}}]{} & A_{\Sigma 1} \cdots\cdots & \xleftarrow[\alpha_n(id)]{} & A_{\Sigma n} \end{array}$$

provided that the type judgement $B(x_1, ..., x_n)\ \ [\Gamma_n]$ is interpreted as $1 \xleftarrow[\alpha_1(id)]{} A_{\Sigma 1} \cdots\cdots \xleftarrow[\alpha_n(id)]{} A_{\Sigma n} \xleftarrow[\beta(id)]{} B_\Sigma$.

The *equality between terms* is preinterpreted as equality between natural transformations and hence interpreted as the equality between morphisms of $Pgr(\mathcal{C})$.

In the following we give the interpretation of type and term (pseudo)judgements, specifying only for types their preinterpretation as sequences of fibred functors. Indeed, recalling that a natural transformation between fibred functors is determined by its evaluation on the identity and being a term preinterpreted into a natural transformation and interpreted into its evaluation on the identity, we conclude that *the interpretation of a term determines its preinterpretation.*

The partial (pre)interpretation of the various dependent typed calculi of section 3 follows the correspondence in the table of section 4 and hence, we assume that each time we interpret a type constructor then the category $\mathcal{C}$ has the corresponding property according to the table. In defining such a (pre)interpretation of type and term constructors under a (pseudo)context, we assume that $\Gamma \equiv x_1 \in A_1, ..., x_n \in A_n(x_1, ..., x_{n-1})$ and $\vec{\sigma}(id) \equiv \alpha_1(id)\ ,\ \alpha_2(id)\ , ...,\ \alpha_n(id)$ and that $dom(\alpha_n(id)) = A_{\Sigma n}$.

Note also that, when we interpret a term constructor, like for example $\langle b, c \rangle$, we also

assume to know the interpretation of the terms b and c, but we omit the corresponding details for simplicity.

Assumption of variable:

$\mathcal{I}_{\mathcal{C}}(\ x \in B(x_1,...,x_n)\ [\Gamma,\ x \in B(x_1,...,x_n)]\ )(id) \equiv \triangle_{B_\Sigma}$
$\mathcal{I}_{\mathcal{C}}(\ x \in B(x_1,...,x_n)\ [\Gamma,\ x \in B(x_1,...,x_n)\,,\ y \in C(x_1,...,x_n,x)]\ )(id) \equiv \gamma(id)^*(\triangle_{B_\Sigma})$
where $\triangle_{B_\Sigma}$ is the diagonal with respect to $\beta(id)$ in $\mathcal{C}/A_{\Sigma n}$ with $B_\Sigma \equiv dom(\beta(id))$
provided that $\mathcal{I}_{\mathcal{C}}(\ B(x_1,...,x_n)\ [\Gamma]\ ) \equiv \vec{\sigma}(id)\ ,\ \beta(id)$
and $\mathcal{I}_{\mathcal{C}}(\ C(x_1,...,x_n,x)\ [\Gamma,\ x \in B(x_1,...,x_n)\ ) \equiv \vec{\sigma}(id)\ ,\ \beta(id)\ ,\ \gamma(id)$
and that the interpretations $\mathcal{I}_{\mathcal{C}}(\ B(x_1,...,x_n)\ [\Gamma,\ x \in B(x_1,...,x_n)\,,\ y \in C(x_1,...,x_n,x)]\ )$
and $\mathcal{I}_{\mathcal{C}}(\ B(x_1,...,x_n)\ [\Gamma,\ x \in B(x_1,...,x_n)]\ )$ are defined by means of the semantic operation of weakening according to lemma 5.6.
The assumption of variable when the context Δ is made of more than one typed variable (see the rule in section 3) is interpreted by repeating the semantic operation of weakening according to lemma 5.6.

Terminal type:

$\mathcal{I}_{\mathcal{C}}(\ \top\ [\ ]\ ) \equiv \widehat{id_1}(id_1)$ and $\mathcal{I}_{\mathcal{C}}(\ \top\ [\Gamma_n]\ ) \equiv \vec{\sigma}(id)\ ,\ \widehat{id_1}(!_{A_{\Sigma n}})$
where 1 is the terminal object in $\mathcal{C}$ and $!_{A_{\Sigma n}}$ is the unique morphism from $A_{\Sigma n}$ to 1.

$\mathcal{I}_{\mathcal{C}}(\ \star \in \top\ [\ ]\ ) \equiv \langle id_1\ ,_1\ id_1\rangle$ and $\mathcal{I}_{\mathcal{C}}(\ \star \in \top\ [\Gamma_n]\ ) \equiv \langle id_{A_{\Sigma n}}\ ,_1\ !_{A_{\Sigma n}}\rangle$
which are the unique morphisms toward the pullback respectively of id_1 along id_1 and of id_1 along $!_{A_{\Sigma n}}$.

False type:

$\mathcal{I}_{\mathcal{C}}(\ \bot\ [\ ]\ ) \equiv \widehat{!_0}(id_1)$ $\mathcal{I}_{\mathcal{C}}(\ \bot\ [\Gamma_n]\ ) \equiv \vec{\sigma}(id)\ ,\ \widehat{!_0}(!_{A_{\Sigma n}})$
where 0 is the initial object in $\mathcal{C}$.

$\mathcal{I}_{\mathcal{C}}(\ \mathsf{r}_\bot(a) \in A\ [\Gamma_n]\ ) \equiv ?_{A_\Sigma} \cdot (\widetilde{a^I}(id))$
where $?_{A_\Sigma}$ is the unique morphism from $\widehat{!_0}(!_{A_{\Sigma n}})$ to $\alpha(id)$ in $\mathcal{C}/A_{\Sigma n}$ provided that $\mathcal{I}_{\mathcal{C}}(\ A\ [\Gamma_n]\ ) \equiv \vec{\sigma}(id)\ ,\ \alpha(id)$.

Indexed Sum type:

$\mathcal{I}_{\mathcal{C}}(\ \Sigma_{y \in B} C(y)\ [\Gamma_n]\ ) \equiv \vec{\sigma}(id)\ ,\ \Sigma_\beta(\gamma)(id)$
where $\Sigma_\beta(\gamma)(t) \equiv \beta(t) \cdot \gamma(q(t, \beta(id)))$ for $t : D \to A_{\Sigma n}$ provided that
$\mathcal{I}_{\mathcal{C}}(\ B\ [\Gamma_n]\ ) \equiv \vec{\sigma}(id)\ ,\ \beta(id)$ and $\mathcal{I}_{\mathcal{C}}(\ C\ [\Gamma_n, z \in B]]\ ) \equiv \vec{\sigma}(id)\ ,\ \beta(id)\ ,\ \gamma(id)$.

$\mathcal{I}_{\mathcal{C}}(\ \langle b, c\rangle \in \Sigma_{y \in B} C(y))\ [\Gamma_n]\ ) \equiv q(\widetilde{b^I}(id), \gamma(id)) \cdot (\widetilde{c^I}(id))$

$\mathcal{I}_{\mathcal{C}}(\ \pi_1(d) \in B\ [\Gamma_n]\ )\ \equiv\ \gamma(id) \cdot (d^{\widetilde{I}}(id))$

$\mathcal{I}_{\mathcal{C}}(\ \pi_2(d) \in C(\pi_1(d))\ [\Gamma_n]\ )\ \equiv\ \langle id_{A_{\Sigma n}}, _{B_\Sigma}\, d^{\widetilde{I}}(id)\rangle$
which is the unique morphism to the pullback of $\gamma(id)$ along $\gamma(id) \cdot (d^{\widetilde{I}}(id))$.

Extensional Propositional Equality type:

$\mathcal{I}_{\mathcal{C}}(\ \mathsf{Eq}(C, c, d)\ [\Gamma_n]\ )\ \equiv\ \vec{\sigma}(id)\ ,\ Eq(\ c^{\widetilde{I}}\ ,\ d^{\widetilde{I}}\ )(id)$
where $Eq(\ c^{\widetilde{I}}\ ,\ d^{\widetilde{I}}\ )(t)\ \equiv\ eq(\ c^{\widetilde{I}}(t)\ ,\ d^{\widetilde{I}}(t)\ )$ which is the equalizer of $c^{\widetilde{I}}(t)$ and $d^{\widetilde{I}}(t)$ in $\mathcal{C}$ for $t : D \to A_{\Sigma n}$.

$\mathcal{I}_{\mathcal{C}}(\ \mathsf{eq}_{\mathcal{C}}(c) \in \mathsf{Eq}(C, c, c)\ [\Gamma_n]\ )\ \equiv\ t$
which is the unique morphism in $\mathcal{C}/A_{\Sigma n}$ toward the equalizer $eq(\ c^{\widetilde{I}}(id)\ ,\ c^{\widetilde{I}}(id)\ )$ induced by $id_{A_{\Sigma n}}$ (which equalizes $c^{\widetilde{I}}(id)$ with itself!).

Disjoint Sum type:

$\mathcal{I}_{\mathcal{C}}(\ C + D\ [\Gamma_n]\ )\ \equiv\ \vec{\sigma}(id)\ ,\ (\gamma \oplus \delta)(id)$
where for $t : D \to A_{\Sigma n}$ we put $(\gamma \oplus \delta)(t)\ \equiv\ \gamma(t) \oplus \delta(t)$ which is the coproduct in $\mathcal{C}/A_{\Sigma n}$ provided that $\mathcal{I}_{\mathcal{C}}(\ C\ [\Gamma_n]\ )\ \equiv\ \vec{\sigma}(id)\ ,\ \gamma(id)$ and $\mathcal{I}_{\mathcal{C}}(\ D\ [\Gamma_n]\ )\ \equiv\ \vec{\sigma}(id)\ ,\ \delta(id)$.

$\mathcal{I}_{\mathcal{C}}(\ \mathsf{inl}(c) \in C + D\ [\Gamma_n]\ )\ \equiv\ \epsilon_1(c^{\widetilde{I}}(id))\quad$ and $\quad\mathcal{I}_{\mathcal{C}}(\ \mathsf{inr}(d) \in C + D\ [\Gamma_n]\ )\ \equiv\ \epsilon_2(d^{\widetilde{I}}(id))$
where ϵ_1, ϵ_2 are the injections of the coproduct $\gamma(id) \oplus \delta(id)$ in $C/A_{\Sigma n}$.

$\mathcal{I}_{\mathcal{C}}(\ El_+(p, a_C, a_D) \in A(p)\ [\Gamma_n])\ \equiv\ \langle id_{A_{\Sigma n}, A_{\Sigma n}}\ (\ (q_1 \cdot a_C^{\widetilde{I}}(id))\ \oplus\ q_2 \cdot a_D^{\widetilde{I}}(id))\cdot p^{\widetilde{I}}(id)\rangle$
which is the unique morphism toward the pullback of $\alpha(id)$ along $p^{\widetilde{I}}(id)$ where $q_1\ \equiv\ q(\epsilon_1, \alpha(id))$ and $q_2\ \equiv\ q(\epsilon_2, \alpha(id))$ and $(q_1 \cdot a_C^{\widetilde{I}}(id))\ \oplus\ q_2 \cdot a_D^{\widetilde{I}}(id))$ is the codiagonal morphism of $q_1 \cdot a_C^{\widetilde{I}}(id)$ and $q_2 \cdot a_D^{\widetilde{I}}(id))$ provided that $\mathcal{I}_{\mathcal{C}}(\ A(w)\ [\Gamma_n, w \in C + D]\ )\ \equiv\ \vec{\sigma}\ ,\ (\gamma \oplus \delta)(id)\ ,\ \alpha(id)$.

$\mathcal{I}_{\mathcal{C}}(\ \mathsf{dsj}(c, d) \in \bot\ [\Gamma_n]\ )\ \equiv\ \langle c^{\widetilde{I}}(id)\ ,_{\widehat{!_0}(!_{A_{\Sigma n}})}\ d^{\widetilde{I}}(id)\rangle$
which is the unique morphism to $\widehat{!_0}(!_{A_{\Sigma n}})$ which is vertex of the pullback of ϵ_1 along ϵ_2 in $\mathcal{C}/A_{\Sigma n}$ by disjointness of $\gamma(id) \oplus \delta(id)$.

The mono existential type: it is interpreted as $(\Sigma_{x \in B} C(x))/\top$ and hence we refer to the interpretation of quotient types and indexed sum types.

Forall type:

$\mathcal{I}_{\mathcal{C}}(\ \forall_{y \in B} C(y)\ [\Gamma_n]\ )\ \equiv\ \vec{\sigma}(id)\ ,\ \forall_\beta\, \gamma(id)$
where $\forall_\beta\, \gamma(t)\ \equiv\ \forall_{\beta(t)}\, \gamma(q(t, \beta(id)))$ for $t : D \to A_{\Sigma n}$ and $\forall_{\beta(t)}$ is the right adjoint of the pullback functor $\beta(t)^*$ on subobjects provided that $\mathcal{I}_{\mathcal{C}}(\ B\ [\Gamma_n]\ )\ \equiv\ \vec{\sigma}(id)\ ,\ \beta(id)$ and

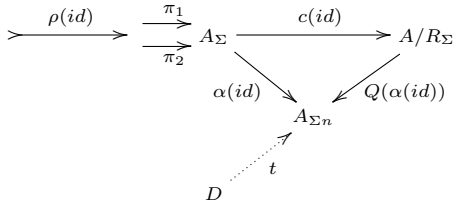
$\mathcal{I}_{\mathcal{C}}(\ C\ [\Gamma_n, z \in B]]\ )\ \equiv\ \vec{\sigma}(id)\ ,\ \beta(id)\ ,\ \gamma(id)$ with $\gamma(id)$ monomorphism.

$\mathcal{I}_{\mathcal{C}}(\ \lambda y^B.c \in \forall_{y\in B} C(y)\ [\Gamma_n]\ )\ \equiv\ \psi(\widetilde{c^I}(id))$
where $\psi\ :\ \mathcal{C}/B_\Sigma(\ id_{B_{\Sigma n}}\ ,\ \gamma(id)\ )\ \rightarrow\ \mathcal{C}/A_{\Sigma n}(\ id_{A_{\Sigma n}}\ ,\ \forall_{\beta(id)}(\gamma(id))\ )$ is the adjunction bijection considering that $\beta(id)^*(id_{A_{\Sigma n}})$ is isomorphic to $id_{B_{\Sigma n}}$.

$\mathcal{I}_{\mathcal{C}}(\ \mathsf{Ap}(f,b) \in C(b)\ [\Gamma_n]\ )\ \equiv\ \langle id_{A_{\Sigma n}, B_\Sigma}\ \psi^{-1}(\widetilde{f^I}(id)) \cdot \widetilde{b^I}(id)\rangle$
which is the morphism toward the pullback of $\gamma(id)$ along $\widetilde{b^I}(id)$ and ψ^{-1} is the inverse of ψ.

Quotient type: (with weakened elimination rule)

$\mathcal{I}_{\mathcal{C}}(\ A/R\ [\Gamma_n]\ )\ \equiv\ \vec{\sigma}(id)\ ,\ Q(\alpha)(id)$
which is defined in the following, provided that $\mathcal{I}_{\mathcal{C}}(\ A\ [\Gamma_n]\ )\ \equiv\ \vec{\sigma}(id)\ ,\ \alpha(id)$ and
$\mathcal{I}_{\mathcal{C}}(\ R(x,y)\ type\ [x \in A, y \in A]\ )\ \equiv\ \vec{\sigma}(id)\ ,\ \alpha(id)\ \ \alpha[\alpha(id)](id)\ ,\ \rho(id)$
with $\rho(id)$ an equivalence relation on $\alpha(id)$ in $\mathcal{C}/A_{\Sigma n}$. Hence, $\langle \pi_1 \cdot \rho(id)\ ,\ \pi_2 \cdot \rho(id)\rangle$ is an equivalence relation in $\mathcal{C}$ where $\pi_1\ \equiv\ \alpha(\alpha(id))$ and $\pi_2\ \equiv\ q(\alpha(id), \alpha(id))$ are the kernel pair of $\alpha(id)$; therefore we can take the quotient $c(id)$ of the equivalence relation in $\mathcal{C}$ and we define $Q(\alpha(id))$ as the unique morphism such that $\alpha(id) = Q(\alpha(id)) \cdot c(id)$ as expressed in the following commutative diagram:



Then, for any $t : D \rightarrow A_{\Sigma n}$, we define $Q(\alpha)(t)\ \equiv\ Q(\alpha(t))$ as the unique morphism such that $\alpha(t) = Q(\alpha(t)) \cdot c(t)$, where $c(t)$ is the quotient map of the equivalence relation obtained by pulling back the above diagram along t by using the corresponding fibred functors, (for example $\rho(id)$ is pulled back to $\rho(t_3)$ where $t_3\ \equiv\ q(\ t_2\ ,\ \pi_1)$ and $t_2\ \equiv\ q(t, \alpha(id))$.
$\mathcal{I}_{\mathcal{C}}(\ [a] \in A/R\ [\Gamma_n]\ )\ \equiv\ c(id) \cdot (\widetilde{a^I}(id))$

$\mathcal{I}_{\mathcal{C}}(\ El_{Q_s}(m,p) \in M\ [\Gamma_n]\ )\ \equiv\ q \cdot \widetilde{p^I}(id)$
where q is the unique morphism such that $q \cdot c(id) = q(\alpha(id), \mu(id)) \cdot \widetilde{m^I}(id)$ provided that $\mathcal{I}_{\mathcal{C}}(\ M\ [\Gamma_n]\ )\ \equiv\ \vec{\sigma}(id)\ ,\ \mu(id)$ and $\widetilde{m^I}(id) \cdot (\pi_1 \cdot \rho(id)) = \widetilde{m^I}(id) \cdot (\pi_2 \cdot \rho(id))$.

$\mathcal{I}_{\mathcal{C}}(\ \mathsf{eff}(a,b) \in R(a,b)\ [\Gamma_n]\ )\ \equiv\ \langle id_{A_{\Sigma n}}, _{A \times A_\Sigma}\ t\rangle$
which is the unique morphism toward the pullback of $\rho(id)$ along the pair morphism $\langle \widetilde{a^I}(id), \widetilde{b^I}(id)\rangle$ toward the product of $\alpha(id)$ with itself in $\mathcal{C}/A_{\Sigma n}$, where t is the unique morphism in $\mathcal{C}$ such that $(\pi_1 \cdot \rho(id)) \cdot t = \widetilde{a^I}(id)$ and $(\pi_2 \cdot \rho(id)) \cdot t = \widetilde{b^I}(id)$ obtained by effectiveness of the equivalence relation $\langle \pi_1 \cdot \rho(id), \pi_2 \cdot \rho(id)\rangle$, provided that

$c(id) \cdot a^{\widetilde{I}}(id) = c(id) \cdot b^{\widetilde{I}}(id)$.

Natural Numbers type: it is interpreted as $List(\top)$.

List type: (with weakened elimination rule)

$\mathcal{I}_{\mathcal{C}}(\ List(C)\ [\Gamma_n]\ ) \ \equiv\ \vec{\sigma}(id)\ ,\ List(\gamma)(id)$
where $List(\gamma)(t) \ \equiv\ List(\gamma(t))$ that is the list object on $\gamma(t)$ in $\mathcal{C}/D$ for $t : D \to A_{\Sigma n}$ provided that $\mathcal{I}_{\mathcal{C}}(\ C\ [\Gamma_n]\ ) \ \equiv\ \vec{\sigma}(id)\ ,\ \gamma(id)$.

$\mathcal{I}_{\mathcal{C}}(\ \epsilon \in List(C)\ [\Gamma_n]\ ) \ \equiv\ r_o^{\gamma}$
that is the empty map for $List(\gamma)(id)$ in $\mathcal{C}/A_{\Sigma n}$.

$\mathcal{I}_{\mathcal{C}}(\ \mathsf{cons}(s,c) \in List(C)\ [\Gamma_n]) \ \equiv\ {r_1}^{\gamma} \cdot \langle s^{\widetilde{I}}(id), c^{\widetilde{I}}(id)\rangle$
where ${r_1}^{\gamma} : List(\gamma)(id) \times \gamma(id) \to List(\gamma)(id)$ is the list-constructor map in $\mathcal{C}/A_{\Sigma n}$ and $\langle s^{\widetilde{I}}(id), c^{\widetilde{I}}(id)\rangle$ is the pair morphism toward $List(\gamma)(id) \times \gamma(id)$ in $\mathcal{C}/A_{\Sigma n}$.

$\mathcal{I}_{\mathcal{C}}(\ El_{List_s}(a,l,s) \in L\ [\Gamma_n]\ ) \ \equiv\ t \cdot s^{\widetilde{I}}(id)$
where t is the unique morphism such that $t \cdot {r_o}^{\gamma} = a^{\widetilde{I}}$ and $(\pi_2^L \cdot l^{\widetilde{I}}(id)) \cdot t \times id = r \cdot {r_1}^{\gamma}$ by the property of list object in $\mathcal{C}/A_{\Sigma n}$ with $\pi_2^L \ \equiv\ q(\,\xi(id) \cdot \gamma(\xi(id))\,,\,\xi(id)\,)$ provided that $\mathcal{I}_{\mathcal{C}}(\ L\ [\Gamma_n]\ ) \ \equiv\ \vec{\sigma}(id)\ ,\ \xi(id)$.

Extensional dependent product type:

$\mathcal{I}_{\mathcal{C}}(\ \Pi_{y\in B}C(y)\ [\Gamma_n]\ ) \ \equiv\ \vec{\sigma}(id)\ ,\ \Pi_{\beta}\,\gamma(id)$
where $\Pi_{\beta}\,\gamma(t) \ =\ \Pi_{\beta(t)}\,\gamma(q(t,\beta(id)))$ for $t : D \to A_{\Sigma n}$ and $\Pi_{\beta(t)}$ is the right adjoint of $\beta(t)^*$ provided that $\mathcal{I}_{\mathcal{C}}(\ B\ [\Gamma_n]\ ) \ \equiv\ \vec{\sigma}(id)\ ,\ \beta(id)$ and that $\mathcal{I}_{\mathcal{C}}(\ C\ [\Gamma_n, z \in B]]\ ) \ \equiv\ \vec{\sigma}(id)\ ,\ \beta(id)\ ,\ \gamma(id)$.

Corresponding abstraction and application are interpreted analogously to those of the forall type.

Omega type:

$\mathcal{I}_{\mathcal{C}}(\ \Omega\ type\ [\Gamma_n]\ ) \ \equiv\ \vec{\sigma}(id)\ ,\ \widehat{!_{\mathcal{P}(1)}}(!_{A_{\Sigma n}})$
where $\mathcal{P}(1)$ is the subobject classifier in $\mathcal{C}$.

$\mathcal{I}_{\mathcal{C}}(\ \{B\} \in \Omega\ [\Gamma_n]\ ) \ \equiv\ \aleph(\beta(id))$
that is the characteristic morphism in $\mathcal{C}/A_{\Sigma n}$ of $\beta(id)$ (thought of as a morphism from $\beta(id)$ to $id_{A_{\Sigma n}}$) with respect to $\widehat{!_{\mathcal{P}(1)}}(!_{A_{\Sigma n}})$ provided that $\mathcal{I}_{\mathcal{C}}(\ B\ type\ [\Gamma_n]\ ) \ \equiv\ \vec{\sigma}(id)\ ,\ \beta(id)$ with $\beta(id)$ a monomorphism in $\mathcal{C}$.

$\mathcal{I}_{\mathcal{C}}(\ \mathsf{ch}(B) \in B\ [\Gamma_n]\ ) \ \equiv\ \rho_{B_{\Sigma}}$
where $\rho_{B_{\Sigma}}$ is the isomorphism in $\mathcal{C}/A_{\Sigma n}$ from the identity on $A_{\Sigma n}$ to $\beta(id)$ that exists

provided that $\mathcal{I}_{\mathcal{C}}(\ B\ type\ [\Gamma_n]\ ) \equiv \vec{\sigma}(id)\ ,\ \beta(id)$ with $\beta(id)$ a monomorphism in $\mathcal{C}$ such that $\aleph(\beta(id)) = \mathcal{I}_{\mathcal{C}}(\ \{\top\} \in \Omega\ [\Gamma_n]\ )$, which yields that $\beta(id)$ is isomorphic to $id_{A_{\Sigma_n}}$.

In the next lemma we use the abbreviation $\Gamma_{j+1-\rangle n} \equiv x_{j+1} \in A_{j+1}, ..., x_n \in A_n$ for a given context Γ_n denoting with Γ_o the empty context.

Lemma 5.6. The *weakening* of a variable in type and term judgements is interpreted as follows:
if $\mathcal{I}_{\mathcal{C}}(\ B(x_1, ..., x_n)\ [\Gamma_n]\ ) \equiv \alpha_1(id)\ ,\ \alpha_2(id)\ , ...,\ \alpha_n(id)\ ,\ \beta(id)$
and $\mathcal{I}_{\mathcal{C}}(\ b \in B(x_1, ..., x_n)\ [\Gamma_n]\ ) \equiv b^{\widetilde{I}}(id)$
and $\mathcal{I}_{\mathcal{C}}(\ D(x_1, ..., x_j)\ [\Gamma_j]\ ) \equiv \alpha_1(id)\ ,\ \alpha_2(id)\ , ...,\ \alpha_j(id)\ ,\ \delta(id)$ with $j \leq n$, then

$B(x_1, ..., x_n)\ [\Gamma_j, y \in D, \Gamma_{j+1-\rangle n}]$ is interpreted as

$$\alpha_1(id)\ , ...,\ \alpha_j(id),\ \delta(id),\ \alpha_{j+1}[t_j](id)\ , ...,\ \alpha_n[t_{n-1}](id)\ ,\ \beta[t_n](id)$$

(where $\alpha_1(id)\ , ...,\ \alpha_j(id)$ appears only if Γ_j is not empty and $\alpha_{j+1}[t_j](id)\ , ...,\ \alpha_n[t_{n-1}](id)$ appears only if $\Gamma_{j+1-\rangle n}$ is not empty)

and $b \in B(x_1, ..., x_n)\ [\Gamma_j, y \in D, \Gamma_{j+1-\rangle n}]$ is interpreted as

$$b^{\widetilde{I}}(t_n)$$

where $t_j \equiv \delta(id)$ and $t_i \equiv q(t_{i-1}, \alpha_i(id))$ for $i = j+1, ..., n$.

In the next lemma we use the abbreviation $\Gamma'_{j+1-\rangle n} \equiv x'_{j+1} \in A'_{j+1}, ..., x'_n \in A'_n$ for a given context Γ_n where if $n \geq j+1$ we define $A'_{j+k} \equiv A_{j+k}\ [x_j/a_j][x_i/x'_i]_{i=j+1,...,j+k-1}$ for $k = 1, ...n-j$ for a given $a_j \in A_j\ [\Gamma_{j-1}]$.

Lemma 5.7. The *substitution* of variables in type and term judgements is interpreted as follows:
if $\mathcal{I}_{\mathcal{C}}(\ B(x_1, ..., x_n)\ [\Gamma_n]\ ) \equiv \alpha_1(id)\ ,\ \alpha_2(id)\ , ...,\ \alpha_n(id)\ ,\ \beta(id)$
and $\mathcal{I}_{\mathcal{C}}(\ a_j \in A_j\ [\Gamma_{j-1}]\ ) \equiv a_j^{\widetilde{I}}(id)$ and $\mathcal{I}_{\mathcal{C}}(\ b \in B(x_1, ..., x_n)[\Gamma_n]\ ) \equiv b^{\widetilde{I}}(id)$ where $n \geq j$ then

$B(x_1, ..., x_{j-1}, a_j, x'_{j+1}, ..., x'_n)\ [\Gamma_{j-1}, \Gamma'_{j+1-\rangle n}]$ is interpreted as

$$\alpha_1(id)\ , ...\alpha_{j-1}(id)\ ,\ \alpha_{j+1}[q_j](id)\ ,\ ...,\ \alpha_n[q_{n-1}](id)\ ,\ \beta[q_n](id)$$

(where $\alpha_1(id)\ , ...\alpha_{j-1}(id)$ appears only if Γ_{j-1} is not empty and $\alpha_{j+1}[q_j](id)\ ,\ ...,\ \alpha_n[q_{n-1}](id)$ appears only if $\Gamma_{j+1-\rangle n}$ is not empty)

and $b(x_1, .., x_{j-1}, a_j, x'_{j+1}, .., x'_n) \in B(x_1, .., x_{j-1}, a_j, x'_{j+1}, .., x'_n)\ [\Gamma_{j-1}, \Gamma'_{j+1-\rangle n}]$ as

$$b^{\widetilde{I}}(q_n)$$

where $q_j \equiv a_j^{\widetilde{I}}(id)$ and $q_i \equiv q(q_{i-1}, \alpha_i(id))$ for $i = j+1, ..., n$.

Now, we just state the validity and completeness for *pretopoi* with respect to $\mathcal{T}_{ptop}$ as a paradigmatic example:

Theorem 5.8 (Validity). If $A\ type\ [\Gamma_n]$ is derivable in $\mathcal{T}_{ptop}$ then $\mathcal{I}_{\mathcal{P}}(\ A\ type\ [\Gamma_n]\ )$ is well defined for any pretopos $\mathcal{P}$.
If $a \in A\ [\Gamma_n]$ is derivable in $\mathcal{T}_{ptop}$, then also $\mathcal{I}_{\mathcal{P}}(\ a \in A\ [\Gamma_n]\ )$ is well defined for any pretopos $\mathcal{P}$.
Suppose that $A\ type\ [\Gamma_n]$ and $B\ type\ [\Gamma_n]$ are derivable in $\mathcal{T}_{ptop}$. Then, if $A = B\ [\Gamma_n]$ is derivable in $\mathcal{T}_{ptop}$, we have $\mathcal{I}_{\mathcal{P}}(\ A\ type\ [\Gamma_n]\ ) = \mathcal{I}_{\mathcal{P}}(\ B\ type\ [\Gamma_n]\ )$ for any pretopos $\mathcal{P}$.
Suppose that $a \in A\ [\Gamma_n]$ and $b \in A\ [\Gamma_n]$ are derivable in $\mathcal{T}_{ptop}$. Then, if $a = b \in A\ [\Gamma_n]$ is derivable in $\mathcal{T}_{ptop}$, we have $\mathcal{I}_{\mathcal{P}}(\ a \in A\ [\Gamma_n]\ ) = \mathcal{I}_{\mathcal{P}}(\ b \in A\ [\Gamma_n]\ )$ for any pretopos $\mathcal{P}$.

Proof. The proof proceeds by induction on the derivation of type and term judgements by means of weakening and substitution lemmas.

We just underline that a mono type $B(x)\ [x \in A]$ turns out to be interpreted by a sequence of morphisms whose last one $\beta(id)$ is a *monomorphism*, thanks to the fact that the valid interpretation of the judgement

$$y = z \in B(x)\ [x \in A, y \in B(x), z \in B(x)]$$

says that the kernel pair of $\beta(id)$ is the identity relation. □

5.2. *The syntactic categories out of the type theories*

Now we show that every typed calculus proposed so far gives rise to a syntactic category with the categorical properties it intends to capture, modularly as in the table of section 4. Actually, this holds for any theory of a given typed calculus, where by a theory we mean the following:

Def. 5.9. Given a typed calculus $\mathcal{T}$, then a *theory* T of $\mathcal{T}$ is an extension of $\mathcal{T}$ with:

- new (w.r.t. $\mathcal{T}$) type judgements $A\ type\ [\Gamma]$,
- new (w.r.t. $\mathcal{T}$) type equality judgements $A = B\ [\Gamma]$ provided that $A\ type\ [\Gamma]$ and $B\ type\ [\Gamma]$ are derivable type judgements in T,
- new (w.r.t. $\mathcal{T}$) term judgements $a \in A\ [\Gamma]$, provided that $A\ type\ [\Gamma]$ is a derivable type judgement in T,
- new (w.r.t. $\mathcal{T}$) term equality judgements $a = b \in A\ [\Gamma]$, provided that $a \in A\ [\Gamma]$ and $b \in A\ [\Gamma]$ are derivable type judgements in T.

(In other terms no inference rules are admitted to form new type or term judgements.)

From now on, we indicate with T_{lex} a theory of the calculus $\mathcal{T}_{lex}$ and we do the same with the other calculi.

We start by showing how the syntactic category built out of a theory T_{lex} is a lex category to end with the syntactic topos out of a theory of the calculus $\mathcal{T}_{top}$. These categories will be useful to prove a completeness theorem between each calculus presented in section 3 and the class of categories we mean to describe with it.
It is worth recalling that contrary to the syntactic topos in (Lambek and Scott 1986), using dependent types we build a *proof-relevant* topos. Indeed, here morphisms are terms

instead of functional relations there. The same can be said for our syntactic lex and regular categories with respect to those built out of a many-sorted logic in the literature. From now on we refer to $\mathcal{C}_T$ as the syntactic category built from the dependent typed calculus T and defined as follows:

Def. 5.10. The objects of $\mathcal{C}_T$ are closed types $A, B, C...$ of T (modulo their equality), and the morphisms between two types, A and B, are expressions $(x)\, b(x)$ (see (Nordström *et al.* 1990)) corresponding to $b(x) \in B\ [x \in A]$ where the type B does not depend on A, modulo their definitional equality, that is we state that $(x)\, b(x) \in \mathcal{C}_T(\, A\, ,\, B\,)$ and $(x)\, b'(x) \in \mathcal{C}_T(\, A\, ,\, B\,)$ are equal iff we can derive $b(x) = b'(x) \in B\ [x \in A]$ in T. The composition in $\mathcal{C}_T$ is defined by substitution, that is given $(x)\, b(x) \in \mathcal{C}_T(\, A\, ,\, B\,)$ and $(y)\, c(y) \in \mathcal{C}_T(\, B\, ,\, C\,)$ their composition is $(x)\, c(b(x))$. The identity is $(x)\, x \in \mathcal{C}_T(\, A\, ,\, A\,)$ obtained from $x \in A\ [x \in A]$.

In the following we often write a morphism t in $\mathcal{C}_T(\, A\, ,B\,)$ as $t : A \to B$.

Proposition 5.11 (Finite limits). The category $\mathcal{C}_{T_{lex}}$ is finitely complete.

Proof. The *terminal object* is $\top$ and from any object A the morphism toward $\top$ is $(x) \star \in \mathcal{C}_{T_{lex}}(\, A\, ,\, \top\,)$ which is unique by the conversion rule for $\top$.
Given $c \in \mathcal{C}_{T_{lex}}(\, A\, ,\, C\,)$ and $d \in \mathcal{C}_{T_{lex}}(\, B\, ,\, C\,)$ the *pullback* is given by

$$\Sigma_{x \in A}\, \Sigma_{y \in B}\ \mathsf{Eq}(\, C,\, c(x),\, d(y)\,)$$

where the first projection to A is $(z)\, \pi_1^A(z) : \Sigma_{x \in A}\, \Sigma_{y \in B}\, \mathsf{Eq}(C, c(x), d(y)) \longrightarrow A$ and the second projection to B is $(z)\, \pi_1^B(\pi_2^A(z)) : \Sigma_{x \in A}\, \Sigma_{y \in B}\, \mathsf{Eq}(C, c(x), d(y)) \longrightarrow B$. □

In the following, we will often write $a =_A b$ to mean $\mathsf{Eq}(A, a, b)$ and eq_C, instead of $\mathsf{eq}_C(c)$.

Proposition 5.12 (Finite disjoint coproducts). $\mathcal{C}_{T_{ext}}$ has finite disjoint coproducts.

Proof. The initial object is the false type $\bot$. Given an object A, the morphism $(x)\, r_\bot(x)$ from $\bot$ to A is unique because, given any other morphism $t : \bot \to A$, we can derive a proof of $r_\bot(x) =_A t(x)$ for $x \in \bot$ by the elimination of false type.

The *coproduct* of A and B is defined by $A + B$, equipped with the injections $(x)\ \mathsf{inl}(x) : A \to A + B$ and $(y)\ \mathsf{inr}(y) : B \to A + B$. Given $c : A \to C$ and $d : B \to C$ the mediating morphism $c \oplus d$ from $A + B$ to C is $(w)\, El_+(w, c, d)$. Coproducts are disjoint by the rule of disjointness. □

Moreover, finite coproducts are stable under pullbacks. The stability of the initial object follows easily and to show stability of binary coproducts we first prove that

Lemma 5.13. $A + B$ is isomorphic in $\mathcal{C}_{T_{ext}}$ to

$$\Sigma_{w \in A+B}\, (\Sigma_{x \in A}\ \mathsf{inl}(x) =_A w) + (\Sigma_{y \in B}\ \mathsf{inr}(y) =_B w)$$

and in particular $(z)\pi_1(z) : \Sigma_{w \in A+B}\, \widetilde{A}(w) + \widetilde{B}(w) \longrightarrow A + B$ is equipped with an inverse $\delta : A + B \longrightarrow \Sigma_{w \in A+B}\, \widetilde{A}(w) + \widetilde{B}(w)$ where $\widetilde{A}(w) \equiv \Sigma_{x \in A}\ \mathsf{inl}(x) =_{A+B} w$ and $\widetilde{B}(w) \equiv \Sigma_{y \in B}\ \mathsf{inr}(y) =_{A+B} w$.

Hence, we are ready to prove:

Proposition 5.14. In $\mathcal{C}_{T_{ext}}$ coproducts are stable under pullbacks.

Proof. Given the following pullbacks

$$\begin{array}{ccc} P_1 & \xrightarrow{\pi_2^1} & A \\ {\scriptstyle \pi_1^1}\downarrow & & \downarrow{\scriptstyle a} \\ D & \xrightarrow{m} & C \end{array} \qquad \begin{array}{ccc} P_2 & \xrightarrow{\pi_2^2} & B \\ {\scriptstyle \pi_1^2}\downarrow & & \downarrow{\scriptstyle b} \\ D & \xrightarrow{m} & C \end{array} \qquad \begin{array}{ccc} P & \xrightarrow{\pi_2^P} & A+B \\ {\scriptstyle \pi_1^P}\downarrow & & \downarrow{\scriptstyle a\oplus b} \\ D & \xrightarrow{m} & C \end{array}$$

we have to show that $\pi_1^1 \oplus \pi_1^2 \simeq \pi_1^P$ in $\mathcal{C}_{T_{ext}}/D$. For this purpose we define $\gamma : P_1 + P_2 \to P$ as $\gamma \equiv (w)\, El_+(w, d_1, d_2)$ where d_1 corresponds to

$$\langle \pi_1^1(w_1)\,,\, \langle \mathsf{inl}(\pi_2^1(w_1)), \mathsf{eq}_C \rangle \rangle \in P \ [w_1 \in P_1]$$

and d_2 corresponds to

$$\langle \pi_1^2(w_2)\,,\, \langle \mathsf{inr}(\pi_2^2(w_2)), \mathsf{eq}_C \rangle \rangle \in P \ [w_2 \in P_2]$$

We can notice that $\pi_1^P \cdot \gamma = \pi_1^1 \oplus \pi_1^2$ and that $\pi_2^P \cdot \gamma = (\mathsf{inl} \cdot \pi_2^1) \oplus (\mathsf{inr} \cdot \pi_2^2)$.
Moreover, we want to define $\gamma^{-1} : P \to P_1 + P_2$. First of all, we consider that, given $w \in P$, we get $\pi_2^P(w) \in A + B$, hence, by δ defined in the above lemma we deduce

$$\pi_2(\delta(\pi_2^P(w))) \in \widetilde{A}(\pi_2^P(w)) + \widetilde{B}(\pi_2^P(w))$$

Now, we use the elimination rule with respect to $\widetilde{A}(\pi_2^P(w)) + \widetilde{B}(\pi_2^P(w))$ and we define $\gamma^{-1} \equiv (w)\, El_+(\, \pi_2(\delta(\pi_2^P(w)))\,,\, d_1'\,,\, d_2'\,)$ where d_1' corresponds to

$$\mathsf{inl}(\langle \pi_1(w)\,,\, \langle \pi_1(x'), \mathsf{eq}_C \rangle \,\rangle) \in P_1 + P_2 \ [w \in P, x' \in \widetilde{A}(\pi_2^P(w))]$$

Indeed, from $w \in P$ and $x' \in \widetilde{A}(\pi_2^P(w))$ we get $m(\pi_1(w)) = (a \oplus b)(\pi_2^P(w))$ and $\pi_2^P(w) = \mathsf{inl}(\pi_1(x'))$ and hence $m(\pi_1(w)) = a(\pi_1(x'))$. In an analogous way, we define d_2' as

$$\mathsf{inr}(\langle \pi_1(w)\,,\, \langle \pi_1(y'), \mathsf{eq}_C \rangle \rangle) \in P_1 + P_2 \ [w \in P, y' \in \widetilde{B}(\pi_2^P(w))]$$

We can prove that γ^{-1} is the inverse morphism of γ by the elimination rule of the disjoint sum type. □

Proposition 5.15 (The list object). The syntactic category $\mathcal{C}_{T_{loc}}$ is equipped with list-objects.

Proof. The empty map is $(x)\,\epsilon : B \to List(A)$ and the list-constructor map is $\widetilde{\mathsf{cons}} \equiv (z)\,\mathsf{cons}(\pi_1(z), \pi_2(z)) : List(A) \times A \to List(A)$. Then, given a closed type Y and the morphisms $f : B \to Y$ and $g : Y \times A \to Y$ we can prove that there exists a unique morphism $t \in \mathcal{C}_{T_{loc}}(\, B \times List(A)\,,\, Y\,)$ such that the following diagram commutes in all its parts:

$$\begin{array}{ccccc} B & \xrightarrow{id\times((x)\,\epsilon)} & B\times List(A) & \xleftarrow{id\times\widetilde{\mathsf{cons}}} & B\times(List(A)\times A) \\ & \searrow^{f} & \downarrow{\scriptstyle t} & & \downarrow{\scriptstyle (t\times id)\cdot\sigma} \\ & & Y & \xleftarrow[g]{} & Y\times A \end{array}$$

Indeed, by the weakened elimination rule of the list type we can define $t \equiv (z)\, El_{List_s}(\, f(\pi_1(z)),\, \tilde{g},\, \pi_2(z)) : B \times List(A) \to Y$, with $\tilde{g}(x,y) \equiv g(\langle x,y\rangle)$, which is the unique map making the diagrams commute by β_s and η_s C-list conversion rules. □

Note that the existence of a natural numbers object in $\mathcal{C}_{T_{alex}}$ can be proved specializing the proof for list objects just seen to the list on the terminal object.

Proposition 5.16 (Images). $\mathcal{C}_{T_{reg}}$ has images.

Proof. The image of a morphism $f : A \to B$ is

$$\mathcal{I}m(f) \;\equiv\; \pi_1 : I(f) \to B$$

where $I(f) \;\equiv\; \Sigma_{y\in B}\, \exists_{x\in A}\, \mathsf{Eq}(B, f(x), y)$.
Indeed, there exists a map q such that $f = \mathcal{I}m(f) \cdot q$ and defined as $q(z) \equiv \langle f(z), (z, \mathsf{eq})\rangle \in I(f)\ [z \in A]$.
Moreover, $\mathcal{I}m(f)$ satisfies the universal property of an image. Indeed, given another factorization $f = i \cdot p$ with $i : D \to B$ monic, we define by the elimination rule of the mono existential type toward a general dependent type

$$t(z) \;\equiv\; \mathsf{Ex}_g(\, \pi_2(z)\,,\, (x)(y)p(x)\,) \in D$$

for $z \in I(f)$. This is well defined because, if $f(x) = \pi_1(z) = f(x')$, then $i(p(x)) = f(x) = f(x') = i(p(x'))$, and since i is monic we conclude $p(x) = p(x')$. (Note that i is monic if and only if i is injective on elements of A, that is from $i(z) = i(z') \in B$ for $z, z' \in D$ we conclude $z = z' \in D$, since free variables correspond to projections.) Finally, we get that $i \cdot t = \pi_1$ since for $z \in I(f)$ we can derive a proof of

$$(i(\mathsf{Ex}_g(\, w\,,\, (x)(y)p(x)\,)) =_B \pi_1(z)) \;\times\; (\pi_2(z) =_J w) \quad [w \in \exists_{x\in A}\, \mathsf{Eq}(B, f(x), \pi_1(z))]$$

by the elimination rule of the mono existential type, where $J \equiv \exists_{x\in A}\, \mathsf{Eq}(B, f(x), \pi_1(z))$. □

Moreover, we can prove:

Proposition 5.17. In $\mathcal{C}_{T_{reg}}$ the image of any $f : A \to B$ is stable under pullbacks.

Proof. Given $h : C \to B$ we need to show that the image of $\pi_1^{D\times A}$ is $\pi_1^{D\times I(f)}$ in the following pullback squares

$$\begin{array}{ccc} P & \xrightarrow{\pi_2^{D\times A}} & A \\ {\scriptstyle \pi_1^{D\times A}}\downarrow & & \downarrow{\scriptstyle f} \\ D & \xrightarrow[h]{} & B \end{array} \qquad\qquad \begin{array}{ccc} Q & \xrightarrow{\pi_2^{D\times I(f)}} & I(f) \\ {\scriptstyle \pi_1^{D\times I(f)}}\downarrow & & \downarrow{\scriptstyle \mathcal{I}m(f)} \\ D & \xrightarrow[h]{} & B \end{array}$$

with $P \;\equiv\; \Sigma_{y\in D}\, \Sigma_{x\in A}\, \mathsf{Eq}(B, h(y), f(x))$ and $Q \;\equiv\; \Sigma_{y\in D}\, \Sigma_{x\in I(f)}\, \mathsf{Eq}(B,\, \mathcal{I}m(f)(x)\,,\, h(y))$.
It is sufficient to prove that there is an isomorphism between Q and $I(\pi_1^{D\times A})$ such that $\mathcal{I}m(\pi_1^{D\times A})$ is isomorphic to $\pi_1^{D\times I(f)}$ in $\mathcal{C}_{T_{reg}}/D$.
We define $\delta(z) \in Q\ [z \in I(\pi_1^{D\times A})]$ as

$$\delta(z) \;\equiv\; \langle\, \pi_1(z),\; \mathsf{Ex}_g(\, \pi_2(z)\,,\, (w)(w')\langle\langle f(\pi_2^{D\times A}(w))\,,\, (\pi_2^{D\times A}(w), \mathsf{eq})\,\rangle,\, \mathsf{eq}\rangle\,)\,\rangle$$

and we put $\delta^{-1}(w) \equiv \langle\, \pi_1(w),\ \mathsf{Ex}_g(\pi_2(\pi_2^{D\times I(f)}(w))\,, (x)(y)(\langle\, \pi_1(w)\,, \langle x, \mathsf{eq}\rangle\,\rangle\,, \mathsf{eq}\,)\,)\,\rangle$ for $w \in Q$. We can prove that δ and δ^{-1} are well defined and that they are inverse to each other by the elimination rule of the mono existential type and the fact that the existential type is mono. □

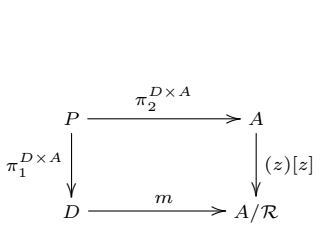
Proposition 5.18 (Quotients of equivalence relations). $\mathcal{C}_{T_{ptop}}$ has got effective quotients of monic equivalence relations.

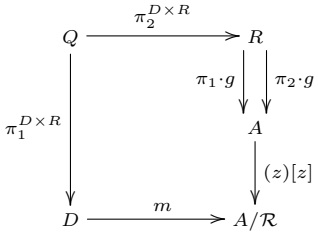
Proof. Given an equivalence relation $R \rightarrowtail^{g} A \times A$ we consider the following mono type: $\mathcal{R}(x, x') \equiv \Sigma_{y\in R}\ g(y) =_{A\times A} \langle x, x'\rangle\ [x \in A, x' \in A]$. It is easy to check that the categorical definition of equivalence relation implies that $\mathcal{R}(x, x')\,[x \in A, x' \in A]$ is an equivalence relation from the type-theoretical point of view. Let $A/\mathcal{R}$ be the quotient with respect to $\mathcal{R}(x, x')\,[x \in A, x' \in A]$. We can prove that $(z)[z] : A \to A/\mathcal{R}$ is the coequalizer of $\pi_1 \cdot g : R \to A$ and $\pi_2 \cdot g \in \mathcal{C}_{T_{ptop}} : R \to A$ (also called the quotient of g) by the elimination and conversion rules of the quotient type. The uniqueness property of the coequalizer follows from the η_sC-quotient rule.
In $\mathcal{C}_{T_{ptop}}$ any categorical equivalence relation is effective, that is $\pi_1 \cdot g$ and $\pi_2 \cdot g$ are projections of the pullback of $(z)[z]$ along itself (that is, its kernel pair). The existence of a morphism toward the pullback vertex is guaranteed by the effectiveness axiom and its uniqueness follows from the fact that equivalence relations are monic. □

Moreover, we prove stability of quotients for equivalence relations.

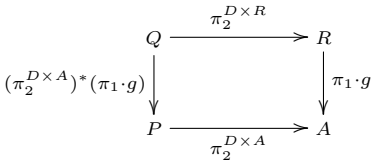
Proposition 5.19. In $\mathcal{C}_{T_{ptop}}$ quotients of monic equivalence relations are stable.

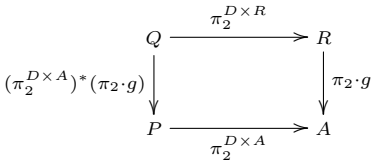
Proof. In the following we write $\pi_i^{A\times D}$ for the i'th projection from the vertex of the pullback of suitable arrows $A \to \cdot \leftarrow D$. We will omit to label the projections when their domains and codomains are clear from the context.
Given $m \in \mathcal{C}_{T_{ptop}}(D, A/\mathcal{R})$ let us consider the following pullbacks:





with $P \equiv \Sigma_{w\in D}\,\Sigma_{x\in A}\,\mathsf{Eq}(A/\mathcal{R},\ m(w), [x])$ and $Q \equiv \Sigma_{w\in D}\,\Sigma_{y\in R}\,\mathsf{Eq}(A/\mathcal{R},\ m(w), [(\pi_1 \cdot g)(y)])$.
Moreover, let us consider these two pullbacks:





where $(\pi_2^{D\times A})^*(\pi_1 \cdot g) \equiv (w)\langle \pi_1(w)\,,\, \langle \pi_1(g(\pi_2^{D\times R}(w)))\,,\, \mathsf{eq}\rangle\rangle$ and $(\pi_2^{D\times A})^*(\pi_2 \cdot g) \equiv (w)\langle \pi_1(w)\,,\, \langle \pi_2(g(\pi_2^{D\times R}(w)))\,,\, \mathsf{eq}\rangle\rangle$. We must show that in $P/\mathcal{C}_{T_{ptop}}$ we get $\pi_1^{D\times A} \simeq \mathsf{coeq}((\pi_2^A)^*(\pi_1 \cdot g), (\pi_2^A)^*(\pi_2 \cdot g))$. (We recall that the objects of the category $P/\mathcal{C}_{T_{ptop}}$ are the morphisms $b : P \to B$ of $\mathcal{C}$, and the morphisms of $P/\mathcal{C}_{T_{ptop}}$ from $b : P \to B$ to $b' : P \to B'$ are the morphisms $t : B \to B'$ of $\mathcal{C}$ such that $t \cdot b = b'$.) We can observe that the pullback given by effectiveness

$$\begin{array}{ccc} R & \xrightarrow{\pi_2 \cdot g} & A \\ {\scriptstyle \pi_1 \cdot g}\downarrow & & \downarrow{\scriptstyle (z)[z]} \\ A & \xrightarrow[(z)[z]]{} & A/\mathcal{R} \end{array}$$

can be completed in a cube

$$\begin{array}{ccc} R & \xrightarrow{\pi_2 \cdot g} & A \\ {\scriptstyle \pi_1 \cdot g}\downarrow & & \downarrow{\scriptstyle (z)[z]} \\ A & \xrightarrow[(z)[z]]{} & A/\mathcal{R} \\ & \nearrow_{m} & \\ D & & \end{array}$$

of pullbacks and hence

$$\begin{array}{ccc} Q & \xrightarrow{(\pi_2^{D\times A})^*(\pi_2 \cdot g)} & P \\ {\scriptstyle (\pi_2^{D\times A})^*(\pi_1 \cdot g)}\downarrow & & \downarrow{\scriptstyle \pi_1^{D\times A}} \\ P & \xrightarrow[\pi_1^{D\times A}]{} & D \end{array}$$

is a pullback.

Therefore, $\langle \pi_2^*(\pi_1 \cdot g)\,,\, \pi_2^*(\pi_2 \cdot g)\rangle$ is an equivalence relation as kernel pair of $\pi_1^{D\times A}$. Then, consider the coequalizer of $\pi_2^*(\pi_1 \cdot g)$ and $\pi_2^*(\pi_2 \cdot g)$ written $[-]_P : P \to P/m^*(\mathcal{R})$ where $P/m^*(\mathcal{R})$ is the quotient of the equivalence relation $\langle \pi_2^*(\pi_1 \cdot g)\,,\, \pi_2^*(\pi_2 \cdot g)\rangle$. Since $\pi_1^{D\times A} \cdot \pi_2^*(\pi_1 \cdot g) = \pi_1^{D\times A} \cdot \pi_2^*(\pi_2 \cdot g)$ there exists a map $Q^P : P/m^*(\mathcal{R}) \to D$ such that $Q^P \cdot [-]_P = \pi_1^{D\times A}$. Now, we want to prove that Q^P is an isomorphism in $P/\mathcal{C}_{T_{ptop}}$. In order to define the inverse of Q^P we need the following lemma:

Lemma 5.20. In T_{ptop}, the quotient type A/R is isomorphic to

$$\Sigma_{z\in A/R}\ (\Sigma_{x\in A}\,[x] =_{A/R} z)/\top$$

Proof.
We define the morphism $\phi : A/R \to \Sigma_{z\in A/R}\ (\Sigma_{x\in A}\,[x] =_{A/R} z)/\top$ as $\phi(z) \equiv \langle z, El_Q(z\,,\,(x)\,[\langle x, \mathsf{eq}\rangle]\,)\,\rangle$ for $z \in A/R$ and its inverse as $\phi^{-1}(z') \equiv \pi_1(z')$ for $z' \in \Sigma_{z\in A/R}\,(\Sigma_{x\in A}\,[x] =_{A/R} z)/\top$.
It is immediate to see $\phi^{-1} \cdot \phi = id$ and $\phi \cdot \phi^{-1} = id$ follows from the fact that for every fixed $z \in A/R$ then $(\Sigma_{x\in A}\,[x] =_{A/R} z)/\top$ is a mono type. □

Now, we go back to prove that Q^P is an isomorphism by finding its inverse. By means of ϕ defined in the above lemma we define: for $d \in D$

$$Q^{P^{-1}}(d) \equiv El_Q(\,\pi_2(\phi(m(d)))\,,\,(w)[\langle d\,,\,\langle \pi_1 w\,,\,\mathsf{eq}\rangle\,\rangle])$$

where the elimination constant El_Q is referred to the type $(\Sigma_{x\in A}\,([x] =_{A/R} m(d)))/\top$. This term is well typed by effectiveness. Then, by the elimination rule on the quotient type $P/m^*(\mathcal{R})$ it is easy to prove that $Q^{P^{-1}} \cdot Q^P = id$. Moreover, note that we can prove that $Q^{P^{-1}}$ is monic by lemma 5.20 and by the elimination rule on the quotient type and effectiveness. Hence $Q^P \cdot Q^{P^{-1}} = id$ also follows. This concludes the proof that ${\pi_1}^D$ is a coequalizer of $({\pi_2}^A)^*(\pi_1 \cdot g)$ and $({\pi_2}^A)^*(\pi_2 \cdot g)$. □

[of prop. 5.19]

In order to show that in $\mathcal{C}_{T_{hptop}}$ each pullback functor on subobjects has a right adjoint, we first note that the pullback functor on subobjects is isomorphic to the functor $Prop(-) : \mathcal{C}_T^{op} \to Cat$ defined in the following.

Def. 5.21. For any object $A \in Ob\mathcal{C}_{T_{hptop}}$, the objects of the category $Prop(A)$ are the equivalence classes of mono types depending on A, $B(x)$ $[x \in A]$, under the relation of equiprovability, and the morphisms are the terms $f \in B(x) \to C(x)$ $[x \in A]$ where $B(x) \to C(x) \equiv \forall_{B(x)}(C(x))$. The identity is $\lambda y.\, y \in B(x) \to B(x)$. The composition of $f \in B(x) \to C(x)$ and $g \in C'(x) \to D(x)$, supposing that $C(x)$ is equivalent to $C'(x)$ and in particular that there exists $s \in C(x) \to C'(x)$, is given by $\lambda y.\mathsf{Ap}(g, \mathsf{Ap}(s, \mathsf{Ap}(f, y))) \in B(x) \to D(x)$.

Therefore, we can define the above functor $Prop(-) : \mathcal{C}^{op}_{T_{hptop}} \longrightarrow Cat$:

Def. 5.22. For any object $A \in Ob\mathcal{C}_{T_{hptop}}$, $Prop(A)$ is the above defined category and given a morphism $m \in \mathcal{C}_{T_{hptop}}(\, D\,,\, A\,)$ we define $Prop(m)$ as the following functor: for any $B(x)$ $[x \in A]$

$$Prop(m)(\, B(x)\ [x \in A]\,) \ \equiv\ B(m(z))\ [z \in D]$$

and for every $t \in B(x) \to C(x)$ $[x \in A]$, given $z \in D$, we define

$$Prop(m)(t) \ \equiv\ \lambda w \in B(m(z)).\, \mathsf{Ap}(t[x := m(z)], w)$$

which is a term of type $B(m(z)) \to C(m(z))$.

We can easily check that $Prop(-)$ is a well defined functor. Then, we recall that the functor $Sub(-) : \mathcal{C}^{op}_{T_{hptop}} \to Cat$ is defined as follows: for every $A \in Ob\mathcal{C}_{T_{hptop}}$, $Sub(A)$ is the poset category $Sub(A)$ of subobjects of $\mathcal{C}_{T_{hptop}}$, and for every morphism $t : A \to B$, $Sub(t)$ is the restriction of the pullback functor on subobjects. Then, we can prove that

Proposition 5.23. The functor $Sub(-) : \mathcal{C}^{op}_{T_{hptop}} \longrightarrow Cat$ is naturally isomorphic to the functor $Prop(-) : \mathcal{C}^{op}_{T_{hptop}} \longrightarrow Cat$.

Proof. Any monomorphism $t : B \rightarrowtail A$ in $\mathcal{C}_{T_{hptop}}$ gives rise to a mono type

$$\Sigma_{y \in B}\, t(y) =_A x\ [x \in A]$$

Conversely, any mono type $B(x)$ $[x \in A]$ gives rise to a monomorphism $\pi_1 : \Sigma_{x \in A} B(x) \rightarrowtail A$. □

Proposition 5.24 (Right adjoints on subobjects). In $\mathcal{C}_{T_{hptop}}$ for every morphism $m(y) \in A$ $[y \in D]$ there exists the right adjoint of the pullback functor m^*.

$$Sub(A) \underset{\forall_m}{\overset{m^*}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} Sub(D)$$

Proof. By the previous proposition, it is enough to show that $Prop(m)$ has a right adjoint. For every mono type $B(y)$ $[y \in D]$ we put similarly to (Seely 1984; Lawvere 1969)

$$\forall_m(B(y)\ [y \in D]) \ \equiv\ \forall_{y \in D}\,(x =_A m(y)) \to B(y)\ [x \in A]$$

whose value at a mono type is indeed a mono type. Then we define a bijection (where we omit the contexts of the type)

$$Prop(D)(Prop(m)(\,C(x))\,,\,B(y)\,) \underset{\psi_2}{\overset{\psi_1}{\rightleftarrows}} Prop(A)(\,C(x)\,,\,\forall_m(B(y))\,)$$

as follows: for any $t \in C(m(y)) \to B(y)$ $[y \in D]$ we put $\psi_1(t) \equiv \lambda z.\,\lambda y.\lambda w.\mathsf{Ap}(t,z)$ and for any $s \in C(x) \to \forall_m(B(y))$ $[x \in A]$ and for any $y \in D$ we put

$$\psi_2(s) \equiv \lambda z.\mathsf{Ap}(\mathsf{Ap}(\mathsf{Ap}(s[x := m(y)], z), y), \mathsf{eq}_A)$$

It is easy to see that ψ_1 and ψ_2 are inverse to each other and that they are natural on the first variable. □

Proposition 5.25 (Right adjoints to pullback functors). In $\mathcal{C}_{T_{lcc}}$ pullback functors have right adjoints.

Proof. The right adjoint to a pullback functor is described as in (Seely 1984; Lawvere 1969): for every morphism $m : D \to A$ of $\mathcal{C}_{T_{lcc}}$, for every object $b : B \to D$ of $\mathcal{C}_{T_{lcc}}/D$, we put

$$\Pi_m(b) \equiv \pi_1 : \Sigma_{x \in A}\, C(x) \longrightarrow A$$

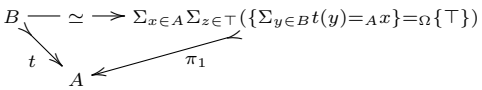
where $C(x) \equiv \Pi_{y \in D}\,(x =_A m(y)) \to \Sigma_{z \in B}\, b(z) =_D y$ for $x \in A$. □

Proposition 5.26 (Subobject classifier). In the syntactic category $\mathcal{C}_{T_{top}}$ there is a subobject classifier.

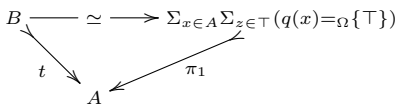
Proof. The subobject classifier is the type Ω and the *true* map is $\{\top\} \in \Omega$ $[x \in \top]$. Moreover, given a monomorphism $B \overset{t}{\rightarrowtail} A$ its characteristic map is

$$\{\Sigma_{y \in B}\, t(y) =_A x\} \in \Omega\ [x \in A]$$

since $\Sigma_{y \in B}\, t(y) =_A x$ $[x \in A]$ is a mono dependent type. Indeed, it is easy to prove that the pullback of the characteristic map with the *true* map is isomorphic to t, i.e.



Then, the uniqueness of the characteristic map can be proved as follows. Given any $q(x) \in \Omega$ $[x \in A]$ such that



we conclude $q(x) = \{\,\mathsf{Eq}(\Omega\,,\,q(x)\,,\,\{\top)\}\,)\,\} = \{\,\Sigma_{y \in B} t(y) =_A x\,\}$ by η-C conversion rule of Ω and by the equality on Ω. □

5.3. *Completeness*

We know that the completeness theorem with respect to general contextual categories with attributes is quite straightforward (see, for example, (Pitts 2000), (Streicher 1991)). Indeed, the interpretation in the syntactic contextual category is faithful, since it turns out to correspond to an identity modulo provable equality between types and between terms. But, since our models are particular contextual categories with attributes, and the interpretation of the indexed sum type is the composition of fibred functors, by taking as a paradigmatic example the calculus $\mathcal{T}_{ptop}$, we have that the interpretation $\mathcal{I}_{\mathcal{C}_{\mathcal{T}_{ptop}}}$ in the syntactic pretopos $\mathcal{C}_{\mathcal{T}_{ptop}}$ is no longer exactly an identity modulo provable equality. Anyway, this interpretation is isomorphic to an interpretation into a canonical comprehension structure that resembles the identity interpretation and hence will be useful to prove completeness.
In more detail, given a pretopos $\mathcal{P}$, we define the following category $\mathcal{P}^{\rightarrow_{fin}}$ whose objects are sequences $A_0 \xleftarrow[a_1]{} A_1 \xleftarrow[a_2]{} A_2 \cdots\cdots \xleftarrow[a_n]{} A_n$ of composable morphisms of $\mathcal{P}$ and whose morphisms between a_1 , a_2 ,..., a_n and b_1 , b_2 ,..., b_n are sequences ϕ_0 , ϕ_1 ,..., ϕ_n of morphisms of $\mathcal{P}$ such that all the following squares commute

$$\begin{array}{ccc}
A_n & \xrightarrow{\phi_n} & B_n \\
{\scriptstyle a_n}\downarrow & & \downarrow{\scriptstyle b_n} \\
A_{n-1} & \xrightarrow{\phi_{n-1}} & B_{n-1} \\
{\scriptstyle a_{n-1}}\downarrow & & \downarrow{\scriptstyle b_{n-1}} \\
\vdots & & \vdots \\
A_2 & \xrightarrow{\phi_2} & B_2 \\
{\scriptstyle a_2}\downarrow & & \downarrow{\scriptstyle b_2} \\
A_1 & \xrightarrow{\phi_1} & B_1 \\
{\scriptstyle a_1}\downarrow & & \downarrow{\scriptstyle b_1} \\
A_0 & \xrightarrow{\phi_0} & B_0
\end{array}$$

Then, we can prove that there is a kind of isomorphism between the interpretation $\mathcal{I}_{\mathcal{C}_{\mathcal{T}_{ptop}}}$ and another interpretation $\mathcal{H}_{\mathcal{C}_{\mathcal{T}_{ptop}}}$ of $\mathcal{T}_{ptop}$-judgements into $Pgr(\mathcal{C}_{\mathcal{T}_{ptop}})$ which is close to an identity interpretation modulo the equality and is clearly faithful. The idea is to define $\mathcal{H}_{\mathcal{C}_{\mathcal{T}_{ptop}}}$ on $B(x)$ $[x \in A]$ as the projection $\pi_1 : \Sigma_{x\in A}\, B(x) \to A$ and $b(x) \in B(x)$ $[x \in A]$ as the section $\langle x, b(x)\rangle \in \Sigma_{x\in A}\, B(x)$ $[x \in A]$ of π_1 in $\mathcal{C}_{\mathcal{T}_{ptop}}$, all corrected to start from the terminal type.

In more detail $\mathcal{H}_{\mathcal{C}_{\mathcal{T}_{ptop}}}$ is defined by induction on the number of assumptions in the context in this manner:

$\mathcal{H}_{\mathcal{C}_{\mathcal{T}_{ptop}}}(\ B\ type\ [x_1 \in A_1, ..., x_n \in A_n(x_1, ..., x_{n-1})]\ )$ is

$$\Sigma_{z_n \in \widetilde{A_n}}\, \overline{B} \xrightarrow{\pi_1^B} \Sigma_{z_{n-1}\in \widetilde{A_{n-1}}}\, \overline{A_n} \xrightarrow{\pi_1^n} \cdots\cdots\ \Sigma_{z\in\top}\, \overline{A_1} \xrightarrow{\pi_1^1} \top$$

where $\widetilde{A_1} \equiv \Sigma_{z\in\top} A_1$ and, for $n \geq 2$, $\widetilde{A_n}$ is the domain of the last morphism of

$\mathcal{H}_{\mathcal{C}_{\mathcal{T}_{ptop}}}(\ A_n\ type\ [\Gamma_{n-1}]\ )$ and for $z_n \in \widetilde{A_n}$

$$\overline{B} \ \equiv\ B\ [x_1 := \pi_2^1 \cdot \pi_1^2 \cdots \pi_1^n(z_n)]...[x_{n-1} := \pi_2^{n-1}(\pi_1^n(z_n))][x_n := \pi_2^n(z_n)]$$

with $\pi_i^n \ \equiv\ \lambda x.\pi_i^n(x)$ for $i = 1, 2$, where $\pi_1^n(x)$ and $\pi_2^n(x)$ are the two projections of $\Sigma_{z_{n-1}\in\widetilde{A_{n-1}}}\overline{A_n}$ and π_1^B and π_2^B are the two projections of $\Sigma_{z_n\in\widetilde{A_n}}\overline{B}$.
For $i = 1, ..., n$ we define $\overline{A_i}$ in the same manner as $\overline{B}$.
If $b \in B\ [\Gamma_n]$ is a judgement of $\mathcal{T}_{ptop}$, we put

$$\overline{b} \ \equiv\ b\ [x_1 := \pi_2^1 \cdot \pi_1^2 \cdots \pi_1^n(z_n)]...[x_{n-1} := \pi_2^{n-1}(\pi_1^n(z_n))][x_n := \pi_2^n(z_n)]$$

and we define

$$\mathcal{H}_{\mathcal{C}_{\mathcal{T}_{ptop}}}(\ b \in B\ [\Gamma_n]\ ) \ \equiv\ \langle z_n, \overline{b}\rangle \in \Sigma_{z'_n\in\widetilde{A_n}}\overline{B}\ \ [z_n \in \widetilde{A_n}]$$

In order to prove the completeness theorem, we prove a lemma by induction on the type and term judgements:

Lemma 5.27. For every judgement $B\ type\ [\Gamma_n]$ derivable in $\mathcal{T}_{ptop}$, - which we suppose to be interpreted as $\alpha_1(id)\ ,\ \alpha_2(id)\ ,...,\ \alpha_n(id)\ ,\ \beta(id)$ - there is an isomorphism

$$\phi_{A_1}\ ,...,\ \phi_{A_n}\ ,\ \phi_B$$

in $\mathcal{C}^{\rightarrow fin}_{\mathcal{T}_{ptop}}$ between $\mathcal{I}_{\mathcal{C}_{\mathcal{T}_{ptop}}}(\ B\ type\ [\Gamma_n]\ )$ and $\mathcal{H}_{\mathcal{C}_{\mathcal{T}_{ptop}}}(\ B\ type\ [\Gamma_n]\ )$ such that for every judgement $b \in B\ [\Gamma_n]$

$$\phi_B \cdot \widetilde{b^I}(id) = \langle id, \overline{b}\rangle \cdot \phi_{A_n}$$

and such that it commutes with the interpretations of weakening and substitution: namely about weakening, for every judgement $M\ type\ [\Gamma_j]$ with $n \geq j$ -supposed to be interpreted as $\alpha_1(id)\ ,\ \alpha_2(id)\ ,...,\ \alpha_j(id)\ ,\ \mu(id)$-

$$\phi_B \cdot q(t_n, \beta(id)) = (p_n \times id) \cdot \phi_{M\times B}$$

where $\phi_{M\times B} : \widetilde{(M \times B)^I} \to \Sigma_{x\in\widetilde{M}}\overline{B}$ and $\widetilde{(M \times B)^I} \ \equiv\ dom(\beta(t_n))$ and t_i for $i = j, ...n$ is defined as in the weakening lemma 5.6 and $p_j \ \equiv\ \pi_1^M$ and $p_i \ \equiv\ p_{i-1} \times id$ for $i = j + 1, ..., n$.
and about substitution, for every $a_j \in A_j\ [\Gamma_{j-1}]$ with $n \geq j$

$$\phi_B \cdot q(q_n, \beta(id)) = (s_n \times id) \cdot \phi_{B(a_j)}$$

where $\phi_{B(a_j)} : (B(a_n))^I \to \Sigma_{x\in\widetilde{A_n(a_j)}}\overline{B(a_j)}$ and $(B(a_n))^I \ \equiv\ dom(\beta(q_n))$ and q_i for $i = j, ...n$ is defined as in the substitution lemma 5.7 and $s_j \ \equiv\ \langle id, \overline{a_j}\rangle$ and $s_i \ \equiv\ s_{i-1}\times id$ for $i = j + 1, ..., n$.

Hence, we are ready to prove

Theorem 5.28 (completeness). Suppose that $A\ type\ [\Gamma_n]$ and $B\ type\ [\Gamma_n]$ are derivable in $\mathcal{T}_{ptop}$, if, for every pretopos $\mathcal{P}$, we have $\mathcal{I}_{\mathcal{P}}(\ A\ type\ [\Gamma_n]\ ) = \mathcal{I}_{\mathcal{P}}(\ B\ type\ [\Gamma_n]\ )$, then $A = B\ [\Gamma_n]$ is derivable in $\mathcal{T}_{ptop}$. Suppose that $a \in A\ [\Gamma_n]$ and $b \in A\ [\Gamma_n]$ are derivable in $\mathcal{T}_{ptop}$, if, for every pretopos $\mathcal{P}$, $\mathcal{I}_{\mathcal{P}}(\ a \in A\ [\Gamma_n]\ ) = \mathcal{I}_{\mathcal{P}}(\ b \in A\ [\Gamma_n]\ )$, then $a = b \in A\ [\Gamma_n]$ is derivable in $\mathcal{T}_{ptop}$.

In conclusion the same validity and completeness theorems formulated as before hold

- for $\mathcal{T}_{lex}$ with respect to *lex* categories,
- for $\mathcal{T}_{alex}$ with respect to *arithmetic lex* categories,
- for $\mathcal{T}_{reg}$ with respect to *regular* categories,
- for $\mathcal{T}_{ext}$ with respect to *lextensive* categories,
- for $\mathcal{T}_{loc}$ with respect to *locoi*,
- for $\mathcal{T}_{ptop}$ with respect to *pretopoi*,
- for $\mathcal{T}_{hptop}$ with respect to *Heyting pretopoi*,
- for $\mathcal{T}_{au}$ with respect to *arithmetic universes*,
- for $\mathcal{T}_{lcc}$ with respect to *locally cartesian closed* categories,
- for $\mathcal{T}_{top}$ with respect to *topoi*.

More details on the proofs can be found in (Maietti 1998b).

Comparison with contextual categories. Our notion of model for a dependent type theory corresponds to particular contextual categories with attributes (Cartmell 1986; Pitts 2000). Indeed, for each typed calculus presented in section 3 we can give a corresponding notion of model in terms of a contextual category with attributes. Our notion of model corresponds to the contextual category built out of the reindexing functor of the split fibration equivalent to the codomain fibration based on a category $\mathcal{C}$ as follows. We define $Cont(\mathcal{C})$ as the category of contexts, where the objects of $Cont(\mathcal{C})$ are finite sequences $a_1, a_2, ..., a_n$ of morphisms of $\mathcal{C}$, also written $1 \xleftarrow[a_1]{} A_1 \xleftarrow[a_2]{} A_2 \cdots\cdots \xleftarrow[a_n]{} A_n$ and a morphism in $Cont(\mathcal{C})$ from $a_1, a_2, ..., a_n$ to $b_1, b_2, ..., b_m$ is simply a morphism b of $\mathcal{C}$ such that $(b_1 \cdot b_2 ... \cdot b_n) \cdot b = a_1 \cdot a_2 ... \cdot a_n$.
Moreover, for each object of $Cont(\mathcal{C})$ $1 \xleftarrow[a_1]{} A_1 \xleftarrow[a_2]{} A_2 \cdots\cdots \xleftarrow[a_n]{} A_n$ we define

$$Type_{Cont(\mathcal{C})}(a_1, a_2, ..., a_n) \equiv Fib(\mathcal{C}/A_n, \mathcal{C}^{\rightarrow})$$

Therefore, in this case $Cont(\mathcal{C})$ turns out to be equivalent to $\mathcal{C}$ and to $Type_{Cont(\mathcal{C})}(1)$. Hence, the class of contextual categories with attributes for a dependent type theory captured by our semantics is smaller than the corresponding class of contextual categories with attributes, since it is not the case that for any contextual category we have that the base category is equivalent to the fibre over 1.

5.4. *Beyond soundness and completeness: dependent type theory as internal language.*

The correspondence between typed calculi and categories described in section 3 satisfies not only the validity and completeness theorem but also a stronger link namely that the dependent type theories provide the internal languages of the corresponding categories in which they are modelled. This is mathematically expressed by proving a sort of equivalence between the category of theories (and translations) of a type calculus $\mathcal{T}$ and the category of categorical structures that $\mathcal{T}$ is supposed to describe.

The reason why the internal language link between a calculus and some categorical structures is stronger than validity and completeness is because validity and completeness

do not guarantee that the category of theories of the calculus is in a sort of equivalence with the category of the complete categorical structures. Indeed, for a calculus $\mathcal{T}$ the category of categorical structures for which $\mathcal{T}$ provides an internal language is determined up to equivalences by the category of theories of $\mathcal{T}$. Instead we may have not equivalent categories of models both satisfying validity and completeness with respect to $\mathcal{T}$. (Think of the calculus of predicative intuitionistic logic without proof-terms: it is sound and complete both with respect to Lawvere's hyperdoctrines and with respect to models based on complete Heyting algebras. But these two kinds of models are not equivalent if organized into categories with morphisms preserving the corresponding structure!). Therefore, soundness and completeness alone do not suffice to prove that the considered calculus provides an internal language of the complete models.

Now, we take as a paradigmatic example $\mathcal{T}_{ptop}$ and we sketch the proof that it provides an internal dependent type theory for pretopoi. The same can be done for the other calculi in section 3.

To connect theories of $\mathcal{T}_{ptop}$ with pretopoi we first define the following categories:

1 $Th(\mathcal{T}_{ptop})$ whose objects are theories of $\mathcal{T}_{ptop}$ (called pretopos type theories) and whose morphisms are translations: they send derivable types to derivable types, derivable terms to derivable terms so as to preserve their types and all the type and term equalities and in addition all the type and term constructors of $\mathcal{T}_{ptop}$ (for example, a quotient type is sent into a quotient type and an equivalence class into an equivalence class etc.); we call $Th(\mathcal{T}_{ptop})^{\sim}$ the category whose objects are those of $Th(\mathcal{T}_{ptop})$, but whose morphisms are translations preserving type and term constructors up to isomorphism;

2 $Ptop_{st}$ whose objects are pretopoi with a fixed choice of pretopos structure and whose morphisms are strict logical functors, that is functors preserving the pretopos structure with respect to the fixed choices; we call $Ptop$ the category whose objects are those of $Ptop_{st}$ but whose morphisms are functors preserving the pretopos structure up to isomorphism.

These two categories are related each other by two functors. One is the functor from pretopoi to pretopos type theories

$$T : Ptop_{st} \longrightarrow Th(\mathcal{T}_{ptop})$$

that associates to a pretopos $\mathcal{P}$ its internal type theory $T(\mathcal{P})$ that is defined as an extension of the calculus $\mathcal{T}_{ptop}$ with the specific type and term judgements of $\mathcal{P}$. As in the interpretation of a typed calculus in section 5.1, the idea is that a type judgement corresponds to an object of $Pgr(\mathcal{P})$ obtained as the evaluation on the identical substitution of an object of $Pgf(\mathcal{P})$, which represents a dependent type with all its possible substitutions. Analogously a term judgement corresponds to a morphism of $Pgr(\mathcal{P})$ obtained as the evaluation on the identical substitution of a morphism of $Pgf(\mathcal{P})$, which represents a dependent term with all its possible substitutions.
In more detail, for any object $!^{A_1}, a_2, ..., a_n, t$ of $Pgr(\mathcal{P})$ we add to $T(\mathcal{P})$ a new type judgement

$$t^{-1}(x_1, .., x_n)\ type\ [x_1 \in A_1, ..., x_n \in a_n^{-1}(x_1, .., x_n)]$$

that names the evaluation on identical substitution of the following object of $Pgf(\mathcal{P})$ (and hence it is preinterpreted by it):

$$\widehat{!^{A_1}},\ \widehat{a_2}[p_1],\ \widehat{a_3}[p_2],\ ...,\widehat{a_n}[p_{n-1}],\ \widehat{t}[p_n]$$

where p_i is the second projection of the pullback of a_i and p_{i-1} for $i = 2, \ldots, n$:

$$\begin{array}{ccc}
 & & B_\Sigma \xrightarrow{t^*(p_n)} B \\
 & & \downarrow{\scriptstyle p_n^*(t)} \quad \downarrow{\scriptstyle t} \\
 & & A_{\Sigma n} \xrightarrow{p_n} A_n \\
 & & \downarrow{\scriptstyle p_{n-1}^*(a_n)} \quad \downarrow{\scriptstyle a_n} \\
 & & \vdots \qquad\quad \vdots \\
 & & A_{\Sigma 2} \xrightarrow{p_2} A_2 \\
 & & \downarrow{\scriptstyle p_1^*(a_2)} \quad \downarrow{\scriptstyle a_2} \\
 & & A_{\Sigma 1} \xrightarrow{p_1} A_1 \\
 & & \downarrow{\scriptstyle !_{A_{\Sigma 1}}} \quad \downarrow{\scriptstyle !^{A_1}} \\
 & & 1 \xrightarrow{id_1} 1
\end{array}$$

Analogously, for any dependent type $B(x_1, .., x_n)$ $[x_1 \in A_1, .., x_n \in A_n]$ preinterpreted by the sequence of fibred functors $\alpha_1, \alpha_2, ..., \alpha_n, \beta$ of $Pgf(\mathcal{P})$ and for every morphism of $Pgr(\mathcal{P})$ $\begin{array}{ccccc} & & A_n & \xrightarrow{c} & B \\ & & & \searrow^{id} & \swarrow_{\beta(id)} \\ 1 & \xleftarrow{!_{A_1}} & A_1 & \cdots & A_n \end{array}$ we introduce the new term judgement

$$c(x_1, ..c_n) \in B(x_1, .., x_n)\ [x_1 \in A_1, .., x_n \in A_n]$$

which is preinterpreted in the unique natural transformation $c^{\tilde{I}}$ such that $c^{\tilde{I}}(id) \equiv c$.
Moreover, we add all the type equality judgements regarding types that have the same preinterpretation in $Pgf(\mathcal{P})$ and all the definitional term equality judgements about terms that have the same preinterpretation in $Pgf(\mathcal{P})$.

The functor T associates to every morphism $F : \mathcal{P} \to \mathcal{D}$ of $Pretop_o$ the translation $T(F) : T(\mathcal{P}) \to T(\mathcal{D})$ defined by induction on the signature of $T(\mathcal{P})$ by sending the type and term constructors of $\mathcal{T}_{ptop}$ in $T(\mathcal{P})$ into the corresponding ones in $T(\mathcal{Q})$, that is for example $T(F)(\,\Sigma_{x\in A}B(x)\,) \equiv \Sigma_{x\in T(F)(A)}T(F)(B(x))$, and each specific type arising from $!^{A_1}$, $a_2\,, ...,\, a_n$, t is translated into the specific type of $T(\mathcal{D})$ arising from $!^{F(A_1)}$, $F(a_2)\,, ...,\, F(a_n)$, $F(t)$, and each specific term arising from a morphism c is translated into the term arising from $F(c)$. Note that this translation preserves type and term equalities because F preserves the categorical structure strictly and because for any type B preinterpreted into a fibred functor σ we have that $T(F)(B)$ is preinterpreted into a fibred functor σ^F such that $\sigma^F(id) = F(\sigma(id))$ and the translation of a term interpreted into a morphism b via $T(F)$ is a term interpreted into $F(b)$.
Moreover, we can define a functor from pretopos type theories to pretopoi

$$P : Th(\mathcal{T}_{ptop}) \longrightarrow Ptop_{st}$$

that associates to every type theory T_{ptop} the pretopos $\mathcal{C}_{T_{ptop}}$ defined in section 5.2 (with the same choice of its structure). The functor P is defined on morphisms L of $Th(\mathcal{T}_{ptop})$ as follows. For every closed type A, we put $P(L)(A) \equiv L(A)$, and for every morphism

$b(x) \in B[x \in A]$ of $P(\mathcal{T})$ we put

$$P(L)(\, b(x) \in B\ [x \in A]\,) \ \equiv\ L(b(x)) \in L(B)\ [x \in L(A)]$$

Since L is a translation, then $P(L)$ is a functor preserving the pretopos structure strictly.

Note that in the next we will need to think of T_{ptop} as a category. A categorical structure can be assigned to it by taking the dependent types of T_{ptop} as objects, and the context morphisms as morphisms from the dependent type $B\ [\Gamma]$ to $B'\ [x'_1 \in A'_1, ..., x'_n \in A'_n]$, that is terms of T_{ptop}

$$b' \in B'(a'_1, ..., a'_n)\ [\Gamma, y \in B]$$

where $a_1 \in A'_1\ [\Gamma, y \in B]$ and $a'_i \in A'_i(a'_1, ..., a'_{i-1})\ [\Gamma, y \in B]$ for $i = 1, ..., n$ are derivable terms (see also (Maietti 1998a)).

Therefore, we can consider equivalences of type theories and we are ready to state the internal language theorem linking pretopoi and $\mathcal{T}_{ptop}$ (where with ID we mean the identical embedding functor):

Theorem 5.29. Let $T : Ptop_{st} \to Th(\mathcal{T}_{ptop})$ and $P : Th(\mathcal{T}_{ptop}) \to Ptop_{st}$ be the functors defined above. There are two natural transformations: η from ID to $T \cdot P$ thought as functors from $Th(\mathcal{T}_{ptop})$ to $Th(\mathcal{T}_{ptop})^{\sim}$, and ϵ from $P \cdot T$ to ID thought as functors from $Ptop_{st}$ to $Ptop$ such that for every pretopos type theory T_{ptop} and for every pretopos $\mathcal{P}$, $\eta_{T_{ptop}} : T_{ptop} \to T(\mathcal{C}_{T_{ptop}})$ and $\epsilon_{\mathcal{P}} : \mathcal{C}_{T(\mathcal{P})} \to \mathcal{P}$ are equivalences.

This theorem says that $Th(\mathcal{T}_{ptop})$ and $Ptop_{st}$ are in a sort of equivalence which is a familiar feature of 2-dimensional algebra in the sense of Blackwell-Kelly-Power (Blackwell *et al.* 1989). This appears to be the correct mathematical way to express - in the case of dependent type theory[§] - that $\mathcal{T}_{ptop}$ provides an *internal dependent type theory* for *pretopoi*.

Note that the usual definition of internal language just requires to check that the syntactic category out of the internal language of a category $\mathcal{P}$ is equivalent to the category $\mathcal{P}$ itself (see for example (Barr and Wells 1990; Pitts 2000)).
In conclusion, we can prove that

- $\mathcal{T}_{lex}$ provides an internal dependent type theory for *lex* categories,
- $\mathcal{T}_{alex}$ provides an internal dependent type theory for *arithmetic lex* categories,
- $\mathcal{T}_{reg}$ provides an internal dependent type theory for *regular* categories,
- $\mathcal{T}_{ext}$ provides an internal dependent type theory for *lextensive* categories,
- $\mathcal{T}_{loc}$ provides an internal dependent type theory for *locoi*,
- $\mathcal{T}_{ptop}$ provides an internal dependent type theory for *pretopoi*,
- $\mathcal{T}_{hptop}$ provides an internal dependent type theory for *Heyting pretopoi*,
- $\mathcal{T}_{au}$ provides an internal dependent type theory for *arithmetic universes*,
- $\mathcal{T}_{lcc}$ provides an internal dependent type theory for *locally cartesian closed* categories,
- $\mathcal{T}_{top}$ provides an internal dependent type theory for *topoi*.

[§] In the case of not dependent type theories, like simply typed lambda calculus, it is possible to prove a simple categorical equivalence. For general reasons this gives a standard 2-equivalence.

Hence, we conclude that the table in section 4 establishes a correspondence "internal type theory/category" between a calculus obtained as a combination of the type constructors in the table with respect to the categories enjoying the corresponding combination of categorical properties. Alternatively, the correspondence "internal type theory/category" can be read as "type theory/category as model" linked by internal language (and not only validity and completeness).

5.5. *Free structures and initiality.*

The typed calculi for the categorical structures considered in this paper also provide a way to build the free constructions of such categorical structures generated from a given category. We study only the case of pretopoi as a paradigmatic example.

The main idea is to generate a pretopos from a given category $\mathcal{C}$ by considering its objects as closed types and its morphisms as terms with a free variable.

Def. 5.30. *Given a category $\mathcal{C}$, we consider the dependent type theory $T(\mathcal{C})$ generated by the inference rules as follows:*

1 *For every object A of $Ob\mathcal{C}$ we introduce a new type A and we state the closed type judgement A [ ].*
Given $A \in Ob\mathcal{C}$ and $B \in Ob\mathcal{C}$ we state $A = B$ [ ], if they are the same object in $Ob\mathcal{C}$.

2 *For every morphism $b : A \to B$ in $\mathcal{C}$, we introduce a new term $b(x)$ and we state $b(x) \in B$ $[x \in A]$, where A and B are closed types.*
Given $b : A \to B$ and $d : A \to B$ in $\mathcal{C}$, we state $b(x) = d(x) \in B$ $[x \in A]$, provided that b and d are equal morphisms in $\mathcal{C}$.
Given $b : A \to B$ and $a : D \to A$ in $\mathcal{C}$, we state about composition $b(x)[x := a(y)] = (b \cdot a)(y) \in B$ $[y \in D]$.

3 *Then, we add all the inference rules of the typed calculus $\mathcal{T}_{ptop}$.*

Then, we prove that the category $P(T(\mathcal{C}))$ classifies the functors from $\mathcal{C}$ to a pretopos $\mathcal{P}$, that is it classifies the models in $\mathcal{P}$ of $\mathcal{T}_{ptop}$ augmented with new types and terms naming objects and arrows of $\mathcal{C}$, since such models are in correspondence with the above functors. The classification on which we are interested is not that via strict structure preserving functors but via the usual notion. Hence, we deal with free structures in the up to isomorphism sense. (For the proper general context see (Blackwell *et al.* 1989)).

Theorem 5.31. (free pretopos) Let $\mathcal{C}$ be a category, $\mathcal{P}$ be a pretopos, $Cat(\mathcal{C}\,, \mathcal{P})$ be the category having functors from $\mathcal{C}$ to $\mathcal{P}$ as objects (and natural transformations as morphisms), $Ptop(P(T(\mathcal{C}))\,, \mathcal{P})$ be the category having functors from $P(T(\mathcal{C}))$ to $\mathcal{P}$ preserving the pretopos structure up to isomorphism as objects (and natural transformations as morphisms), and $P : Th(\mathcal{T}_{ptop}) \longrightarrow Ptop_o$ be the functor described at the beginning of the section.

There exists an equivalence between the categories $Cat(\mathcal{C}\,,\,\mathcal{P})$ and $Ptop(P(T(\mathcal{C}))\,,\,\mathcal{P})$. ¶

Proof. We know that $P(T(\mathcal{C}))$ is a pretopos from the definition of P. We can then embed $\mathcal{C}$ into $P(T(\mathcal{C}))$ via a functor $\mathcal{Y} : \mathcal{C} \to P(T(\mathcal{C}))$ defined as follows: for every object $A \in Ob\mathcal{C}$ we put $\mathcal{Y}(A) \equiv A[\,]$ and for every morphism $b : A \to B$ we put $\mathcal{Y}(b) \equiv b(x) \in B\ [x \in A]$. Now, the embedding $\mathcal{Y}$ induces a functor

$$Cat(\mathcal{Y},\mathcal{P}) : Ptop(P(T(\mathcal{C}))\,,\,\mathcal{P}) \to Cat(\mathcal{C}\,,\,\mathcal{P})$$

associating the functor $H \cdot \mathcal{Y}$ to a pretopos functor H and $\alpha_{\mathcal{Y}(-)}$ to a natural transformation α. This functor establishes an equivalence of categories, whose inverse is

$$\widetilde{(-)} : Cat(\mathcal{C}\,,\,\mathcal{P}) \to Ptop(P(T(\mathcal{C}))\,,\,\mathcal{P})$$

$\widetilde{(-)}$ applied to a functor $K : \mathcal{C} \to \mathcal{P}$ is defined as follows: we first define an interpretation $\mathcal{I}^K : T(\mathcal{C}) \to Pgr\mathcal{P}$ of the calculus $T(\mathcal{C})$ by extending the interpretation in section 5 on a type arising from an $\mathcal{C}$-object A as $\mathcal{I}^K(\,A\,[\,]\,) \equiv \widehat{!_{K(A)}}(id)$ and on a term arising from a $\mathcal{C}$-morphism $b : A \to B$ as $\mathcal{I}^K(\,b \in B\ [x \in A]\,) \equiv \langle\, id_{K(A)}\,,_1\ K(b)\,\rangle$ which is the unique morphism toward the pullback of $!_{K(B)}$ along $!_{K(A)}$ induced by $id_{K(A)}$ and $K(b)$.
Then, on generic objects and morphisms of $P(T(\mathcal{C}))$ we put

$$\widetilde{K}(A) \equiv \mathrm{dom}(\mathcal{I}^K(\,A\,[\,]\,))$$
$$\widetilde{K}(\,b(x) \in B\ [x \in A]\,) \equiv q(\mathcal{I}^K(\,A\,[\,]\,)\,,\,\mathcal{I}^K(\,B\,[\,]\,)) \cdot \mathcal{I}^K(\,b \in B\ [x \in A]\,)$$

$\widetilde{K}$ turns out to be a functor preserving the pretopos structure up to isomorphisms.

The definition of $\widetilde{(-)}$ on a natural transformation $\phi : K \Rightarrow H$ is defined by induction on the type and term judgements as the result of the following lemma:

Lemma 5.32. For every judgement $B\ type\ [\Gamma_n]$ derivable in $T(\mathcal{C})$, there is a morphism

$$\widetilde{\phi}_{A_1}\,,...,\,\widetilde{\phi}_{A_n}\,,\,\widetilde{\phi}_B$$

in $\mathcal{P}^{\to_{fin}}$ from $\mathcal{I}^K(\,B\ [\Gamma_n]\,)$ to $\mathcal{I}^H(\,B\ [\Gamma_n]\,)$ such that for every judgement $b \in B\ [\Gamma_n]$

$$\widetilde{\phi}_B \cdot b^{\mathcal{I}^K} = b^{\mathcal{I}^H} \cdot \widetilde{\phi}_{A_n}$$

and such that it commutes with the interpretations of weakening and substitution: namely about weakening, for every judgement $M\ type\ [\Gamma_j]$ with $n \geq j$ -supposed to be interpreted as $\alpha_1^K(id)$, $\alpha_2^K(id)$,..., $\alpha_j^K(id)$, $\mu^K(id)$ according to $\mathcal{I}^K$ and $\alpha_1^H(id)$, $\alpha_2^H(id)$,..., $\alpha_j^H(id)$, $\mu^H(id)$ according to $\mathcal{I}^H$-

$$\widetilde{\phi}_B \cdot q(t_n^K, \beta(id)) = q(t_n^H, \beta(id)) \cdot \widetilde{\phi}_{M \times B}$$

where $\widetilde{\phi}_{M\times B} : (M \times B)^{\mathcal{I}^K} \to (M \times B)^{\mathcal{I}^H}$ and $(M \times B)^{\mathcal{I}^K} \equiv dom(\beta^{\mathcal{I}^K}(t_n^K))$ and t_i^K and t_i^H for $i = j, ..., n$ are defined as in the weakening lemma 5.6 respectively according to $\mathcal{I}^K$ and to $\mathcal{I}^H$,

¶ The construction of free Heyting pretopos or topos in (Maietti 1998b; Maietti 1998a) works only up to isomorphisms as presented here.

and about substitution, for every $a_j \in A_j\ [\Gamma_{j-1}]$ with $n \geq j$

$$\widetilde{\phi}_B \cdot q(q_n^K, \beta(id)) = q(q_n^H, \beta(id)) \cdot \widetilde{\phi}_{B(a_n)}$$

where $\widetilde{\phi}_{B(a_n)} : (B(a_n))^{\mathcal{I}^K} \to (B(a_n))^{\mathcal{I}^H}$ and $(B(a_n))^{\mathcal{I}^K} \equiv dom(\beta^{\mathcal{I}^K}(q_n^K))$ and q_i^K and q_i^H for $i = j, ..., n$ are defined as in the substitution lemma 5.7 respectively according to $\mathcal{I}^K$ and to $\mathcal{I}^H$.

Note that in the above lemma the morphism $\widetilde{\phi}_A$ on a type arising from a $\mathcal{C}$-object A is provided by ϕ_A.

Then, the fact that $Cat(\mathcal{C}\,, \mathcal{Y}) \cdot \widetilde{(-)}$ is naturally isomorphic to the identity follows easily.

Instead, to prove that $\widetilde{(-)} \cdot Cat(\mathcal{Y}, \mathcal{P})$ is naturally isomorphic to the identity we proceed as follows. Given a pretopos functor $G : P(T(\mathcal{C})) \to \mathcal{P}$ we define an interpretation

$$\mathcal{H}^G : T(\mathcal{C}) \to Pgr(\mathcal{P}^{G(\top)})$$

of the calculus $T(\mathcal{C})$ into $Pgr(\mathcal{P}^{G(\top)})$, where $Pgr(\mathcal{P}^{G(\top)})$ is defined exactly as $Pgr(\mathcal{P})$ except for the choice of $G(\top)$ for the terminal object, by applying G to an interpretation of $T(\mathcal{C})$ into $Pgr(P(T(\mathcal{C})))$ defined analogously to $\mathcal{H}_{\mathcal{C}_{\mathcal{T}_{ptop}}}$ in section 5. More precisely, $\mathcal{H}^G(\ B\ [x_1 \in A_1, ..., x_n \in A_n(x_1, ..., x_{n-1})]\ )$ is defined as

$$G(\Sigma_{z_n \in \widetilde{A_n}}\, \overline{B}) \xrightarrow{G(\pi_1^B)} G(\Sigma_{z_{n-1} \in \widetilde{A_{n-1}}}\, \overline{A_n}) \xrightarrow{G(\pi_1^n)} \cdots\cdots G(\Sigma_{z \in \top}\, \overline{A_1}) \xrightarrow{G(\pi_1^1)} G(\top)$$

and

$$\mathcal{H}^G(\ b \in B\ [\Gamma_n]\ ) \ \equiv\ G(\ \langle z_n, \bar{b}\rangle \in \Sigma_{z'_n \in \widetilde{A_n}}\, \overline{B}\ \ [z_n \in \widetilde{A_n}]\ )$$

Then, we prove a lemma relating $\mathcal{I}^{(G \cdot \mathcal{Y})}$ with $\mathcal{H}^G$ analogously to lemma 5.27 from which we conclude that $\widetilde{G \cdot \mathcal{Y}}$ is naturally isomorphic to G. □

Remark 5.33. To connect theorem 5.31 with the notion of a free structure, note that this equivalence implies that given a functor $K : \mathcal{C} \to \mathcal{P}$, from the category $\mathcal{C}$ to the pretopos $\mathcal{P}$, there exists a functor $\widetilde{K} : P(T(\mathcal{C})) \to \mathcal{P}$ in $Ptop$ such that the diagram

$$\begin{array}{ccc} \mathcal{C} & \xrightarrow{\widetilde{I}} & P(T(\mathcal{C})) \\ & {\scriptstyle K}\searrow \quad \swarrow{\scriptstyle \widetilde{K}} & \\ & \mathcal{P} & \end{array}$$

commutes up to a natural isomorphism.

Moreover, from this we derive in particular that

Corollary 5.34. The syntactic category $\mathcal{C}_{\mathcal{T}_{ptop}}$ in definition 5.10 is an initial up to iso pretopos in the sense that for any pretopos $\mathcal{P}$ there is a functor from $\mathcal{C}_{\mathcal{T}_{ptop}}$ to $\mathcal{P}$ preserving the pretopos structure up to isomorphism and which is unique up to a natural isomorphism.

Proof. We define a pretopos functor $I_{\mathcal{P}} : \mathcal{C}_{\mathcal{T}_{ptop}} \to \mathcal{P}$ as follows:

$$I_{\mathcal{P}}(\ A\ ) \ \equiv\ \mathrm{dom}(\mathcal{I}_{\mathcal{P}}(\ A[\ ]\ ))$$
$$I_{\mathcal{P}}(\ b(x) \in B\ [x \in A]\ ) \ \equiv\ q(\mathcal{I}_{\mathcal{P}}(\ A[\ ]\ ), \mathcal{I}_{\mathcal{P}}(\ B[\ ]\ )) \cdot \mathcal{I}_{\mathcal{P}}(\ b \in B\ [x \in A]\ )$$

where $\mathcal{I}_{\mathcal{P}}$ is the interpretation of $\mathcal{T}_{ptop}$ in $Pgr(\mathcal{P})$ defined in section 5. To show that this

is unique up to a natural isomorphism, we follow the technique used in theorem 5.31. Indeed, given any functor $G : \mathcal{C}_{\mathcal{T}_{ptop}} \to \mathcal{P}$ we define an interpretation $\mathcal{H}^G$ of $\mathcal{T}_{ptop}$ into $Pgr(\mathcal{P})$ as in theorem 5.31 and we prove a lemma relating the interpretation $\mathcal{I}_{\mathcal{P}}$ with $\mathcal{H}^G$ analogously to lemma 5.27, from which we conclude that G is naturally isomorphic to $\mathcal{I}_{\mathcal{P}}$. □

Analogously we can prove the existence of *free structures* and *initial up to iso structures* for *lex categories, arithmetic lex categories, regular categories, lextensive categories, locoi, Heyting pretopoi, arithmetic universes, locally cartesian closed categories* and *topoi.*

6. Conclusions

The modular correspondence between categories and dependent type theories so far obtained can be applied:

- to reason within a categorical structure by using logical or type-theoretic proofs;
- viceversa, to import in type theory categorical proofs;
- to study the computational contents of internal languages of categories by employing type-theoretic methods;
- to relate the internal dependent type theories of categorical universes with other type theories like Martin-Löf's Constructive Type Theory.

Acknowledgements. I wish to thank all the people with whom I talked and discussed about internal languages of categories: during my work in Padova Giovanni Sambin, Pino Rosolini and Silvio Valentini, during my visits abroad Martin Hyland and Peter Johnstone and among the people I met at the Mittag-Leffler Institute during Spring 2001 especially Per Martin-Löf, Peter Aczel, Steve Awodey, Andrej Brauer and Erik Palmgren for many interesting discussions about this research topic.

References

Awodey, S and Bauer, A. (2004) Propositions as [types]. *Journal of Logic and Computation*, 14:447–471.

Aczel, P. and Rathjen, M. (2001) Notes on constructive set theory. Mittag-Leffler Technical Report No.40.

Bell, J. L. (1988) *Toposes and Local Set Theories: an introduction.* Claredon Press, Oxford.

Benabou, J. (1985) Fibred categories and the foundations of naive category theory. *Journal of Symbolic Logic*, 50:10–37.

Blackwell, R., Kelly, G. M., and Power, A. J. (1989) Two dimensional monad theory. *Journal of Pure and Applied Algebra*, 59:1–41.

Barr, M. and Wells, C. (1990) *Category Theory for Computer Science.* International Series in Computer Science. Prentice Hall.

Cartmell, J. (1986) Generalised algebraic theories and contextual categories. *Annals of Pure and Applied Logic*, 32:209–243.

Coquand, T. and Huet, G. (1988) The calculus of constructions. *Information and Computation*, 76:95–120.

Carboni, A., Lack, S. and Walters, R. F. C. (1993) Introduction to extensive and distributive category. *Journal of Pure and Applied Algebra*, 84:145–158.

Cockett, J. R. B. (1990) List-arithmetic distributive categories: locoi. *Journal of Pure and Applied Algebra*, 66:1–29.

Cole, J. C. (1973) Categories of sets and models of set theory. In *The Proceedings of the Bertrand Russell Memorial Conference (Uldum, 1971)*, pages 351–399, Leeds.

Constable, R. *et al.* (1986) *Implementing mathematics with the Nuprl Development System.* Prentice Hall.

de Bruijn, N. G. (1991) Telescopic mapping in typed lambda calculus. *Information and Computation*, 91:189–204.

Diaconescu, R. (1975) Axiom of choice and complementation. *Proc. Amer. Math. Soc.*, 51:176–178.

Hofmann, M. (1995) On the interpretation of type theory in locally cartesian closed categories. In *Computer science logic (Kazimierz, 1994)*, volume 933 of *Lecture Notes in Comput. Sci.*, pages 427–441.

Hofmann, M. (1997) *Extensional Constructs in Intensional Type Theory.* Distinguished Dissertations. Springer.

Hyland, J. M. E. and Pitts, A. M. (1989) The theory of constructions: Categorical semantics and topos theoretic models. In J. W. Gray and A. Scedrov, editors, *Categories in Computer Science and Logic*, volume 92 of *Contemporary Mathematics*, pages 137–199.

Jacobs, B. (1999) *Categorical Logic and Type Theory.*, volume 141 of *Studies in Logic.* Elsevier.

Joyal, A. and I. Moerdijk, I. (1995) *Algebraic set theory.*, volume 220 of *Lecture Note Series.* Cambridge University Press.

Johnstone, P. T. (1977) *Topos theory.* Academic Press.

Johnstone, P. T. (2002) *Sketches of an elephant: a topos theory compendium. Vol. 2.*, volume 44 of *Oxford Logic Guides.* The Clarendon Press, Oxford University Press, New York.

Lawvere, F. W. (1969) Adjointness in foundations. *Dialectica*, 23:281–296.

Lawvere, F. W. (1970) Equality in hyperdoctrines and comprehension schema as an adjoint functor. *Proc. Sympos. Pure Math.*, XVII:1–14.

Lambek, J. and Scott, P. J. (1986) *An introduction to higher order categorical logic.*, volume 7 of *Studies in Advanced Mathematics.* Cambridge University Press.

Mac Lane, S. (1971) *Categories for the working mathematician.*, volume 5 of *Graduate text in Mathematics.* Springer.

Maietti, M. E. (1998)a The internal type theory of an Heyting Pretopos. In C.Paulin-Mohring E. Gimenez, editor, *Types for Proofs and Programs. Selected papers of International Workshop Types '96, Aussois*, volume 1512 of *LNCS*, pages 216–235. Springer Verlag.

Maietti, M. E. (1998)b *The type theory of categorical universes.* PhD thesis, University of Padova, February, 1998.

Maietti, M. E. (1999) About effective quotients in constructive type theory. In W. Naraschewski T. Altenkirch and B. Reus, editors, *Types for proofs and programs. International workshop, TYPES '98. Kloster Irsee, Germany, March 27-31. 1999*, volume 1657 of *Lectures Notes in Computer Science*, pages 164–178. Springer Verlag.

Maietti, M. E. (2001) Modular correspondence between dependent type theories and categorical universes. *Mittag-Leffler Preprint Series*, 44.

Maietti, M. E. (2003) Joyal's arithmetic universes via type theory. In *Category Theory in Computer Science, 2002*, volume 69 of *Electronic Notes in Theoretical Computer Science.* Elsevier.

Maietti, M. E. (2005) Reflection into models of finite decidable fp-sketches in an arithmetic universe. In *Category Theory in Computer Science, 2004*, volume 122 of *Electronic Notes in Theoretical Computer Science.* Elsevier.

Martin-Löf, P. (1975) An intuitionistic theory of types: predicative part. In H.E. Rose and J.C. Shepherdson, editors, *Logic Colloquium '73 (Bristol, 1973)*, volume 80 of *Studies in Logic and the Foundations of Mathematics*, pages 73–118. North-Holland, Amsterdam.

Martin-Löf, P. (1984) *Intuitionistic Type Theory, notes by G. Sambin of a series of lectures given in Padua, June 1980.* Bibliopolis, Naples.

Martin-Löf, P. (1998) An intuitionistic theory of types. In G. Sambin and J. Smith, editors, *Twenty five years of Constructive Type Theory*, pages 127–172. Oxford U. P..

Mitchell, W. (1972) Boolean topoi and the theory of sets. *J. Pure Appl. Alg.*, 2:261–274.

Mac Lane, S. and Moerdijk, I. (1992) *Sheaves in Geometry and Logic. A first introduction to Topos theory.* Springer Verlag.

Moerdijk, I. and Palmgren, E. (2000) Wellfounded trees in categories. *Annals of Pure and Applied Logic*, 104(1-3):189–218. Proceedings of the Workshop on Proof Theory and Complexity, PTAC'98 (Aarhus).

Moerdijk, I. and Palmgren, E. (2002) Type theories, Toposes and Constructive Set Theory: predicative aspects of AST. *Annals of Pure and Applied Logic*, 114(1-3):155–201. Commemorative Symposium Dedicated to Anne S. Troelstra (Noordwijkerhout, 1999).

Makkai, M. and Reyes, G. (1977) *First order categorical logic.*, volume 611 of *Lecture Notes in Mathematics.* Springer Verlag.

Maietti, M. E. and Sambin, G. (2005) Toward a minimalist foundation for constructive mathematics. In L. Crosilla and P. Schuster, editor, *From Sets and Types to Topology and Analysis: Practicable Foundations for Constructive Mathematics.* Oxford University Press. Forthcoming.

Maietti, M. E. and Valentini S. (1999) Can you add powersets to Martin-Löf intuitionistic type theory? *Mathematical Logic Quarterly*, 45:521–532.

Nordström, B., Peterson, K. and Smith, J. (1990) *Programming in Martin Löf's Type Theory.* Clarendon Press, Oxford.

Pitts, A. M. (2000) Categorical logic. In Oxford University Press, editor, *Handbook of Logic in Computer Science*, volume 5 of *Oxford Sci. Publ.*, pages 39–128.

Power, A. J. (1989) A general coherence result. *Journal of Pure and Applied Algebra*, 57:165–173.

Scedrov, A. (1995) Intuitionistic set theory. In *Harvey Friedman's research on the foundations of mathematics*, volume 117 of *Stud. Logic Found. Math.*, pages 257–284. North-Holland, Amsterdam.

Seely, R. (1984) Locally cartesian closed categories and type theory. *Math. Proc. Cambr. Phyl. Soc.*, 95:33–48.

Smith, J. (1988) The independence of Peano's fourth axiom from Martin Löf's type theory without universes. *Journal of Symbolic Logic*, 53.

Streicher, T. (1991) *Semantics of type theory.* Birkhäuser.

Sambin, G. and Valentini, S. (1998) Building up a toolbox for Martin-Löf's type theory: subset theory. In G. Sambin and J. Smith, editors, *Twenty-five years of constructive type theory, Proceedings of a Congress held in Venice, October 1995*, pages 221–244. Oxford U. P..

Swaen, M. D. G. (1991) The logic of first order intuitionistic type theory with weak sigma-elimination. *J. Symbolic Logic*, 56:467–483.

Swaen, M. D. G. (1992) A characterization of ML in many-sorted arithmetic with conditional application. *J. Symbolic Logic*, 57:924–953.

Taylor, P. (1997) *Practical Foundations of Mathematics*, volume 99 of *Cambridge studies in advanced mathematics.* Cambridge University Press.

Wraith, G. C. (1985) Notes on arithmetic universes and Gödel incompleteness theorems. Unpublished manuscript.