Theory and Applications of Categories, Vol. 24, No. 3, 2010, pp. 39–83.

JOYAL'S ARITHMETIC UNIVERSE AS LIST-ARITHMETIC PRETOPOS

MARIA EMILIA MAIETTI

Abstract. We explain in detail why the notion of list-arithmetic pretopos should be taken as the general categorical definition for the construction of arithmetic universes introduced by André Joyal to give a categorical proof of Gödel's incompleteness results.

We motivate this definition for three reasons: first, Joyal's arithmetic universes are list-arithmetic pretopoi; second, the initial arithmetic universe among Joyal's constructions is equivalent to the initial list-arithmetic pretopos; third, any list-arithmetic pretopos enjoys the existence of free internal categories and diagrams as required to prove Gödel's incompleteness.

In doing our proofs we make an extensive use of the internal type theory of the categorical structures involved in Joyal's constructions.

The definition of list-arithmetic pretopos is equivalent to the general one that I came to know in a recent talk by André Joyal.

1. Introduction

The categories of topoi and pretopoi can be viewed as universes of abstract sets in which to develop mathematics (see [LR03, Joh77, JM95, MM92, Hyl82]). Joyal's arithmetic universes provide further examples of such universes.

André Joyal introduced arithmetic universes in some lectures given in the seventies, all still unpublished, to provide a categorical proof of Gödel's incompleteness theorems. He defined arithmetic universes by giving a general construction of examples including the free initial one. He then intended to prove incompleteness by mimicking the diagonal argument of Cantor's theorem within the initial arithmetic universe [Joy05]. Indeed, the initial arithmetic universe supports the amount of self-reference needed to perform the mentioned argument because it contains an internal copy of itself.

At that time it was not clear what to take as the most general categorical structure behind the construction of arithmetic universes by Joyal. It was only clear that the desired categorical structure should support the construction of free internal categories and diagrams generated from graphs. In [Mor96, Wra85, Tay05] it is more or less said that the general definition of an arithmetic universe should be a pretopos with free internal categories and diagrams. Here we propose the notion of *list-arithmetic pretopos* as the

---

Received by the editors 2009-06-15 and, in revised form, 2010-02-16.

Transmitted by Martin Hyland. Published on 2010-02-20.

2000 Mathematics Subject Classification: 03G30, 03B15, 18C50, 03B20, 03F55.

Key words and phrases: Pretopoi, dependent type theory, categorical logic.

general notion of *arithmetic universe*, as first announced in [Mai03] and used in [Mai05b].

We think that our proposal is justified by the following reasons:

1) all the examples of arithmetic universe built by Joyal are list-arithmetic pretopoi;

2) the construction of the initial arithmetic universe by Joyal is equivalent to the initial list-arithmetic pretopos;

3) list-arithmetic pretopoi enjoy free internal categories and diagrams as Joyal proved for any of his arithmetic universes.

In order to prove 1) and 2) we briefly describe how Joyal built his arithmetic universes.

An arithmetic universe à la Joyal is a category of the form $(\mathrm{Pred}(\mathcal{S}))_{\mathrm{ex}}$ built in 3 steps as follows:

- take a Skolem theory $\mathcal{S}$, namely a cartesian category with a parameterized natural numbers object where all the objects are finite products of the natural numbers object;
- take the category $\mathrm{Pred}(\mathcal{S})$ of predicates in $\mathcal{S}$, which gives a regular category;
- make its exact completion $(\mathrm{Pred}(\mathcal{S}))_{\mathrm{ex}}$ on a regular category (see [CV98]).

Now, given that the category of predicates $\mathrm{Pred}(\mathcal{S})$ is not only regular but also enjoys stable finite disjoint coproducts and parameterized list objects, as shown by Joyal and in [Rol76, Mor96, Wra85], then its exact completion $(\mathrm{Pred}(\mathcal{S}))_{\mathrm{ex}}$ inherits stable finite disjoint coproducts and parameterized list objects, and hence it turns out to be a list-arithmetic pretopos, namely fact 1) holds.

To prove fact 2), we first observe that the category of predicates of the initial Skolem theory $\mathcal{S}_{\mathrm{in}}$ is equivalent to the initial regular locos. From this we derive that the initial arithmetic universe, which is $(\mathrm{Pred}(\mathcal{S}_{\mathrm{in}}))_{\mathrm{ex}}$ built on the initial Skolem theory, is equivalent to the initial list-arithmetic pretopos.

Finally, to prove fact 3), namely that free internal categories and diagrams exist in any list-arithmetic pretopos, we employ list objects. In particular, to prove the universal properties of the free constructions we build the needed morphisms by iteration on natural numbers.

It is worth mentioning that, when proving fact 2) above, we notice that the category of predicates of the initial Skolem category is also equivalent to the construction of the initial arithmetic lextensive category. This implies that the initial arithmetic universe is also equivalent to the construction of the initial pretopos with a parameterized natural numbers object. All this says that the notion of pretopos with a parameterized natural numbers object, called *arithmetic pretopos*, surely satisfies corresponding facts 1) and 2). But we are not able to prove the corresponding fact 3), namely that any arithmetic pretopos supports free internal categories and diagrams or it is list-arithmetic. We leave this as an open problem.

In showing our results we employ internal languages of the categorical structures involved and these are taken from [Mai05a]. Also in [Rol76] and in [Mor96, Wra85] a term language is presented to reason within a Skolem theory and to build the category of predicates on it.

Here we reason within all the three stages of Joyal's constructions by adopting internal languages that are defined in a modular way as dependent type theories in the style of Martin-Löf's extensional type theory in [Mar84]. These languages are obtained by combining type constructors corresponding to properties defining the various categorical structures thanks to the modular correspondence between them described in [Mai05a].

Finally, we want to remark that the notion of list-arithmetic pretopos as a general definition of arithmetic universe is equivalent to the one that I came to know in a recent talk [Joy05] by André Joyal as a pretopos with free monoid actions (the notion of free monoid actions can be found in [Rol76]).

2. The definition of list-arithmetic pretopos and related categorical structures

Here we recall the definition of list-arithmetic pretopos and of some weaker categorical structures that we will use in the next. We also remind readers of a key preservation property that such structures have in order to enjoy an internal language as a dependent type theory according to [Mai05a].

Note that when we speak of functor preserving some categorical structure we mean preservation up to isomorphism.

Among the basic concepts to build an arithmetic universe there are the notions of Skolem category and Skolem theory. A Skolem category is equipped with the minimum structure needed to interpret primitive recursion on natural numbers. We start by reminding the notion of parameterized natural numbers object:

2.1. Definition. A *parameterized natural numbers object* in a category with finite products (i.e. a category with terminal object 1 and binary products) is an object $\mathcal{N}$ together with maps $0 : 1 \to \mathcal{N}, s : \mathcal{N} \to \mathcal{N}$ such that for every $b : B \to Y$ and $g : Y \to Y$ there is a unique $\mathrm{rec}(b, g)$ making the following diagrams commute

$$
\begin{tikzcd}
B \arrow[r, "\langle \mathrm{id}, 0\cdot !_B\rangle"] \arrow[dr, "b"'] & B \times \mathcal{N} \arrow[d, "\mathrm{rec}(b,g)"] & B \times \mathcal{N} \arrow[l, "\mathrm{id}\times s"'] \arrow[d, "\mathrm{rec}(b,g)"] \\
& Y & Y \arrow[l, "g"]
\end{tikzcd}
$$

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

It is worth recalling here that in the presence of function spaces, as in a cartesian closed category, this parameterized version of natural numbers object is equivalent to the usual natural numbers object [Joh02a].

2.2. Definition. A *Skolem category* is a category with finite products and a parameterized natural numbers object.

A *Skolem theory* is a Skolem category whose objects are finite products of the natural numbers object.

Next, we consider more complex categorical structures that enables one to interpret primitive recursion:

2.3. Definition. A *lextensive category* is a finitely complete category, that is a category with a terminal object and pullbacks, equipped with stable finite disjoint coproducts [CLW93].

If a lextensive category has a parameterized natural numbers object it is said to be *arithmetic lextensive*.

2.4. Definition. A finitely complete category $\mathcal{U}$ has *parameterized list objects* if for any object $A \in \mathsf{Ob}\mathcal{U}$, there is an object $\mathrm{List}(A)$ with morphisms $r_o^A : 1 \to \mathrm{List}(A)$ and $r_1^A : \mathrm{List}(A) \times A \to \mathrm{List}(A)$ such that for every $b : B \to Y$ and $g : Y \times A \to Y$ there is a unique $\mathrm{rec}(b, g)$ making the following diagrams commute

$$\begin{tikzcd} B \arrow[r, "\langle \mathrm{id}, r_o^A \cdot !_B \rangle"] \arrow[dr, "b"'] & B \times \mathrm{List}(A) \arrow[d, "\mathrm{rec}(b,g)"] & B \times (\mathrm{List}(A) \times A) \arrow[l, "\mathrm{id} \times r_1^A"'] \arrow[d, "(\mathrm{rec}(b,g) \times \mathrm{id}_A) \cdot \sigma"] \\ & Y & Y \times A \arrow[l, "g"] \end{tikzcd}$$

where $\sigma : B \times (\mathrm{List}(A) \times A) \to (B \times \mathrm{List}(A)) \times A$ is the associative isomorphism defined in detail as $\langle \langle \pi_1, \pi_1 \cdot \pi_2 \rangle, \pi_2 \cdot \pi_2 \rangle$.

In [Coc90] there is an equivalent definition of parameterized list objects in terms of recursive objects and preservation of recursive objects by the pullback functor $!_D^* : \mathcal{U} \to \mathcal{U}/D$ sending an object B to $\pi_1 : D \times B \to D$.

2.5. Definition. A *locos* is a lextensive category with parameterized list objects.

If a locos is also a regular category it is called a *regular locos*.

Finally, we recall the categorical definition of pretopos [MR77], [JM95].

2.6. Definition. A *pretopos* is a category equipped with finite limits, stable finite disjoint coproducts and stable effective quotients of monic equivalence relations.

If a pretopos has a parameterized natural numbers object it is said to be an *arithmetic pretopos*.

If a pretopos has parameterized list objects it is called a *list-arithmetic pretopos*.

Note that a list-arithmetic pretopos is an arithmetic pretopos since the parameterized list object of the terminal object gives a parameterized natural numbers object.

An important property that all these categorical structures enjoy is that their structure is local.

We say that a structure on a finitely complete category $\mathcal{U}$ is *local* when it satisfies the following:

1) If $\mathcal{U}$ has the considered structure then so does the slice category $\mathcal{U}/A$ for every object $A \in \mathsf{Ob}\,\mathcal{U}$.

2) for every morphism $f : A \to B$ in $\mathcal{U}$ the pullback functor $f^* : \mathcal{U}/B \to \mathcal{U}/A$ preserves the considered structure of the corresponding slice categories.

We recall that all the above mentioned categorical structures are local (see [Mai05a] for a proof):

2.7. Proposition. *The structural properties of being*

- *arithmetic lextensive*
- *a regular locos*
- *an arithmetic pretopos*
- *a list-arithmetic pretopos*

are all local.

The property of being local for a categorical structure is a prerequisite in order to enjoy an internal dependent type theory as described in [Mai05a]. In particular property 1) of a local structure is needed to model types with dependencies. Indeed a dependent type is modelled in a slice category and hence any slice category has to be equipped with all the structure of the starting category. Instead property 2) is needed to make valid the interpretation of substitution via pullback. Indeed, given that substitution of terms both in types and in terms is interpreted via a functor isomorphic to the pullback pseudofunctor, then all the structure used to interpret types and terms has to be preserved under the interpretation of substitution, and hence under pullbacks.

3. An internal language for list-arithmetic pretopoi and related structures

Here we describe internal languages of the categorical structures presented in the previous section in terms of a dependent type theory in the style of Martin-Löf's extensional type theory in [Mar84].

What is an internal language? When we say that a calculus $\mathcal{T}$ provides an internal language of some categorical structures we mean not only that $\mathcal{T}$ is valid and complete with respect to the considered categorical structures, but also that the category of the theories associated to $\mathcal{T}$ is in a sort of equivalence with the category of the considered categorical structures. Usually such an equivalence is shown on one hand by mapping a category to its internal language, which is obtained by augmenting $\mathcal{T}$ with some specific axioms, and on the other hand by mapping a theory of $\mathcal{T}$ to the syntactic category built out of the theory (the one built to prove completeness!). For more details and examples see [Mai05a].

The fact that the described correspondence gives rise to a sort of equivalence means in particular that any categorical structure is equivalent to the syntactic category built out of its internal language. Therefore, we can perform categorical constructions inside any categorical structure by using its internal language.

Note that the link between a typed calculus and a class of categorical structures in terms of the internal language theorem is much stronger than the link established by a soundness and completeness theorem. Indeed, a simple validity and completeness theorem between a calculus and a class of categorical structures does not generally guarantee that the calculus provides the internal language of the considered categories (for an example of this see [MMdPR05]).

Modular correspondence type constructors/categorical properties. In order to single out the dependent typed theories of the categorical structures mentioned in the previous section we will make use of the modular correspondence between type constructors and categorical properties described in [Mai05a]. We just recall here the correspondence for the categorical properties of our interest:

Type constructors	**Categorical properties**
terminal type + indexed sum types + extensional equality types	finite limits
quotient types on the total equivalence relation	stable images
quotient types on mono equivalence relations + effectiveness axiom	stable quotients on monic equivalence relations + effectiveness of quotients
false type + disjoint sum types + disjointness axiom	stable initial object stable binary coproducts + disjointness of coproducts
natural numbers type	parameterized natural numbers object
list types	parameterized list objects

We also recall that, to interpret the above type constructors in a finitely complete category with the corresponding categorical properties, in [Mai05a] we made use of a split fibration associated to the codomain one following [Hof95]. This makes the original naive interpretation in [See84] (see also [Joh02b]) correct when interpreting substitution.

3.1. The typed calculus for list-arithmetic pretopoi. Now we describe in detail the rules of the dependent typed calculus that provides an internal language for list-arithmetic pretopoi (it was first introduced in [Mai99b] and reported in [Mai03, Mai05a]). We call such a calculus $\mathcal{T}_{\mathrm{au}}$ because we will identify the general notion of arithmetic universe with that of list-arithmetic pretopos. Moreover, we will use the pure calculus $\mathcal{T}_{\mathrm{au}}$ to build the initial list-arithmetic pretopos that will be shown to be equivalent to the initial arithmetic universe among Joyal's constructions.

The calculus $\mathcal{T}_{\mathrm{au}}$ is equipped with types, which should be thought of as sets or data types, and with typed terms which represent elements of the types to which they belong. In addition types may depend on typed terms. Hence, dependent types should be better thought of set families.

In the style of Martin-Löf's type theory, we have four kinds of judgements [NPS90]:

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

that is the type judgement, the equality between types, the term judgement and the equality between terms of the same type. The contexts Γ of these judgements are telescopic [dB91], since types are allowed to depend on variables of other types. The contexts are generated by the following rules

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

plus the rules of equality between contexts [Str91], [Pit00].

In the following, we present the inference rules to construct type judgements and term judgements with their equality judgements by recursion. One should also 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:

$$\frac{a \in A\ [\Gamma] \quad A = B\ [\Gamma]}{a \in B\ [\Gamma]}\ \text{conv)} \qquad \frac{a = b \in A\ [\Gamma] \quad A = B\ [\Gamma]}{a = b \in B\ [\Gamma]}\ \text{conv-eq)}$$

$$\frac{\Gamma, x \in A, \Delta \ \ \text{cont}}{x \in A\ [\Gamma, x \in A, \Delta]}\ \text{var)}$$

We can derive then the structural rules of weakening and of a suitable exchange.

In the following we give the formation rules for types specific to $\mathcal{T}_{\mathrm{au}}$ with the corresponding introduction, elimination and conversion rules of their terms. But we omit the equality rules expressing that all the type and term constructors preserve equality as in [Mar84] and that are necessary to derive the substitution rules. Moreover, we adopt the usual definitions of bound and free occurrences of variables and we identify two terms under α-conversion. Note that the context common to all judgements involved in a rule will be omitted. The typed variable appearing in a context is meant to be added to the implicit context as the last one.

The rules to generate $\mathcal{T}_{\mathrm{au}}$'s types and terms are all present in the extensional version of Martin-Löf's type theory [Mar84] except for the disjointness axiom, the rules about quotients types restricted to *mono* equivalence relations and the effectiveness axiom.

Supposing A type and $R(x,y)$ type $[x,y \in A]$, we will write $\mathsf{Equiv}(R)$ to mean the following three judgements:

$$\begin{aligned} \mathsf{refl}(x) &\in R(x,x)\ [x \in A] \\ \mathsf{sym}(x,y,z) &\in R(y,x)\ [x \in A, y \in A, z \in R(x,y)] \\ \mathsf{trans}(x,y,z,u,v) &\in R(x,z)\ [x \in A, y \in A, z \in A, u \in R(x,y), v \in R(y,z)] \end{aligned}$$

Moreover, we will write $\mathsf{Mono}(R)$ to mean

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

The $\mathcal{T}_{\mathrm{au}}$ dependent typed calculus

Terminal type

Tr) $\top$ type $\qquad$ I-Tr) $\star \in \top$ $\qquad$ C-Tr) $\dfrac{t \in \top}{t = \star \in \top}$

False type

Fs) $\bot$ type $\qquad$ E-Fs) $\dfrac{a \in \bot \quad B \text{ type}}{\mathsf{r}_\bot(a) \in B}$

Indexed Sum type

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

E-Σ) $\dfrac{d \in \Sigma_{x\in B}C(x) \quad m(x,y) \in M(\langle x,y\rangle)\ [x \in B, y \in C(x)]}{\mathrm{El}_\Sigma(d,m) \in M(d)}$

C-Σ) $\dfrac{b \in B \quad c \in C(b) \quad m(x,y) \in M(\langle x,y\rangle)\ [x \in B, y \in C(x)]}{\mathrm{El}_\Sigma(\langle b,c\rangle,m) = m(b,c) \in M(\langle b,c\rangle)}$

Equality type

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

E-Eq) $\dfrac{p \in \mathsf{Eq}(C,c,d)}{c = d \in C}$ $\qquad$ C-Eq) $\dfrac{p \in \mathsf{Eq}(C,c,d)}{p = \mathsf{eq} \in \mathsf{Eq}(C,c,d)}$

Disjoint Sum type

+) $\dfrac{B \text{ type} \quad C \text{ type}}{B+C \text{ type}}$ $\qquad$ I$_1$-+) $\dfrac{b \in B \quad B+C \text{ type}}{\mathsf{inl}(b) \in B+C}$ $\qquad$ I$_2$-+) $\dfrac{c \in C \quad B+C \text{ type}}{\mathsf{inr}(c) \in B+C}$

$$\text{E-+)}\ \frac{\begin{array}{l}A(z)\ [z \in B+C] \\ w \in B+C \quad a_B(x) \in A(\mathsf{inl}(x))\ [x \in B] \quad a_C(y) \in A(\mathsf{inr}(y))\ [y \in C]\end{array}}{\mathrm{El}_+(w, a_B, a_C) \in A(w)}$$

$$\text{C}_1\text{-+)}\ \frac{\begin{array}{l}A(z)\ [z \in B+C] \\ b \in B \quad a_B(x) \in A(\mathsf{inl}(x))\ [x \in B] \quad a_C(y) \in A(\mathsf{inr}(y))\ [y \in C]\end{array}}{\mathrm{El}_+(\mathsf{inl}(b), a_B, a_C) = a_B(b) \in A(\mathsf{inl}(b))}$$

$$\text{C}_2\text{-+)}\ \frac{\begin{array}{l}A(z)\ [z \in B+C] \\ c \in C \quad a_B(x) \in A(\mathsf{inl}(x))\ [x \in B] \quad a_C(y) \in A(\mathsf{inr}(y))\ [y \in C]\end{array}}{\mathrm{El}_+(\mathsf{inr}(c), a_B, a_C) = a_C(c) \in A(\mathsf{inr}(c))}$$

Disjointness

$$\text{dis-+)}\ \frac{b \in B \quad c \in C \quad \mathsf{inl}(b) = \mathsf{inr}(c) \in B+C}{\mathsf{dsj}(b,c) \in \bot}$$

Quotient type

$$\text{Q)}\ \frac{R(x,y)\ \text{type}\ [x \in A, y \in A] \quad \mathsf{Mono}(R) \quad \mathsf{Equiv}(R)}{A/R\ \text{type}}$$

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

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

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

Effectiveness

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

List type

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

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

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

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

Note that we can represent the type of natural numbers N as lists of the terminal type since this represents a chosen singleton. Therefore, we define $N \equiv \mathrm{List}(\top)$ with $0 \equiv \epsilon$ and successor $\mathsf{s}(n) \equiv \mathsf{cons}(n,*)$ for $n \in \mathrm{List}(\top)$.

3.2. Remark. Note also that the elimination rule of the Indexed Sum type can be equivalently replaced by the following projections

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

and corresponding β and η conversion rules

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

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

3.3. Remark. [Notation on lists] Given $s, s' \in \mathrm{List}(A)$ we abbreviate

$$\lfloor s, s' \rfloor \quad \text{for} \quad \mathsf{Rec}(s, \mathsf{cons}, s') \in \mathrm{List}(A)$$

which is the operation appending a list to another one.

Moreover, for $s \in \mathrm{List}(A)$ and $a \in A$ we will write

$$\lfloor s, a \rfloor \quad \text{for} \quad \mathsf{cons}(s,a) \qquad \text{and} \qquad \lfloor a \rfloor \quad \text{for} \quad \mathsf{cons}(\epsilon, a)$$

Mono types and quotients. In [Mai05a] we introduced the notion of *mono type*, namely a type $B\ [\Gamma]$ for which

$$w = z \in B\ [\Gamma, w \in B, z \in B]$$

is derivable. Since the interpretation of a mono type as given in [Mai05a] turns out to be a monomorphism, we can then represent the quotient of a monic equivalence relation in the internal language of a pretopos as the quotient type of a mono equivalence relation. Then, effectiveness of quotients is represented by adding a specific axiom. Note that the fact that effectiveness holds for mono equivalence relations is crucial. Indeed effectiveness of quotients on generic relations may lead to classical logic [Mai99a].

Finally we anticipate here that in $\mathcal{T}_{\mathrm{au}}$ we can define quotients on arbitrary relations that are not necessarily equivalences. Categorically this corresponds to the fact that in any list-arithmetic pretopos arbitrary coequalizers exist (see next section).

Coproducts. Coproducts are represented by disjoint sums as in [Mar84] and to represent disjointness we need to add a specific axiom. Indeed, disjointness is not generally derivable by using the same argument in [Smi88] for Peano's fourth axiom.

Coherent logic in a pretopos. Mono types in $\mathcal{T}_{au}$ inherit enough structure to validate coherent logic by interpreting falsum, equality, conjunction, disjunction and existential quantification on a type as follows: for ϕ, ψ mono types

Interpretation of connectives

$$falsum \equiv \bot \qquad\qquad s =_A t \equiv \mathsf{Eq}(A, s, t)$$

$$\phi \wedge \psi \equiv \phi \times \psi \qquad\qquad \phi \vee \psi \equiv (\phi \oplus \psi)/\top$$

$$\exists_{x \in A}\phi(x) \equiv (\Sigma_{x\in A}\phi(x))/\top$$

Thanks to the way connectives are interpreted, they inherit elimination rules that are stronger than the usual ones in intuitionistic logic (see the discussion on the calculus of regular categories on page 22 in [Mai05a]).

Now we describe the internal languages of the categorical structures defined in section 2 and weaker than that of list-arithmetic pretopos. These can be deduced from the table previously described.

The typed calculus of Skolem categories $\mathcal{T}_{sk}$. The calculus $\mathcal{T}_{sk}$ is a type theory with no dependent types including the following type constructors: terminal type, product types and the natural numbers type with the rules of List($\top$) in $\mathcal{T}_{au}$ restricted to non-dependent types that is:

Product type

$$\frac{B \text{ type} \quad C \text{ type}}{B \times C \text{ type}}\ \times) \qquad\qquad \frac{b \in B \quad c \in C}{\langle b, c\rangle \in B \times C}\ \text{I-}\times)$$

$$\frac{d \in B \times C}{\pi_1(d) \in B}\ \text{E}_1\text{-}\times) \qquad\qquad \frac{d \in B \times C}{\pi_2(d) \in C}\ \text{E}_2\text{-}\times)$$

$$\frac{b \in B \quad c \in C}{\pi_1(\langle b, c\rangle) = b \in B}\ \text{C}_1\text{-}\Sigma) \qquad\qquad \frac{b \in B \quad c \in C}{\pi_2(\langle b, c\rangle) = c \in C}\ \text{C}_2\text{-}\times)$$

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

Natural Numbers type

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

$$\frac{L \text{ type} \quad n \in N \quad a \in L \quad l(y) \in L\ [y \in L]}{\mathrm{El}_N(a,l,n) \in L} \text{ E-nat}$$

$$\frac{L \text{ type} \quad a \in L \quad l(y) \in L\ [y \in L]}{\mathrm{El}_N(a,l,0) = a \in L} \text{ C}_1\text{-nat} \qquad \frac{L \text{ type} \quad n \in N \quad a \in L \quad l(y) \in L\ [y \in L]}{\mathrm{El}_N(a,l,\mathsf{s}(n)) = l(\mathrm{El}_N(a,l,n)) \in L} \text{ C}_2\text{-nat}$$

The typed calculus of arithmetic lextensive categories $\mathcal{T}_{\mathbf{alxt}}$. The calculus $\mathcal{T}_{\mathrm{alxt}}$ includes the following type constructors: terminal type, indexed sum types, extensional equality types, false type, disjoint sum types, disjointness axiom, the natural numbers type having the rules of List($\top$) in $\mathcal{T}_{\mathrm{au}}$ that is:

Natural Numbers type

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

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

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

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

The typed calculus of locoi $\mathcal{T}_l$. The calculus $\mathcal{T}_l$ is obtained by extending $\mathcal{T}_{\mathrm{alxt}}$ with list types. Hence it includes the following type constructors: terminal type, indexed sum types, extensional equality types, false type, disjoint sum types, disjointness axiom, list types.

The typed calculus of regular locoi $\mathcal{T}_{\mathbf{rl}}$. The calculus $\mathcal{T}_{\mathrm{rl}}$ is obtained by extending $\mathcal{T}_l$ with quotient types on the total relation namely:

Quotient types on the total relation

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

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

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

Hence $\mathcal{T}_{\mathrm{rl}}$ includes the following type constructors: terminal type, indexed sum types, extensional equality types, false type, disjoint sum types, disjointness axiom, list types and quotient types of the kind $A/\top$.

The typed calculus of arithmetic pretopoi $\mathcal{T}_{\mathbf{pn}}$. The calculus $\mathcal{T}_{\mathrm{pn}}$ includes the following type constructors: terminal type, indexed sum types, extensional equality types, false type, disjoint sum types, disjointness axiom, quotient types on mono equivalence relations with the effectiveness axiom and the natural numbers type.

In other words $\mathcal{T}_{\mathrm{pn}}$ is the fragment of the dependent type theory $\mathcal{T}_{\mathrm{au}}$ in section 3 without list types but with the natural numbers type $N \equiv \mathrm{List}(\top)$.

By using the typed calculi mentioned so far we can build the initial structures among the categorical structures they describe.

From [Mai05a] we recall that given a typed calculus $\mathcal{T}$ for a certain kind of categorical structures, let us say S-structures, the initial S-structure amounts to the category $\mathcal{C}_{\mathcal{T}}$ defined as follows:

3.4. Definition. The objects of $\mathcal{C}_{\mathcal{T}}$ are the closed types $A, B, C \dots$ of $\mathcal{T}$ modulo their equality, and the morphisms between two types, A and B, are the expressions $(x)\, b(x)$ (see [NPS90]) corresponding to the judgement $b(x) \in B\ [x \in A]$ — where the type B does not depend on A — modulo their definitional equality, that is we state that $(x)\, b(x) \in \mathcal{C}_{\mathcal{T}}(A, B)$ and $(x)\, b'(x) \in \mathcal{C}_{\mathcal{T}}(A, B)$ are equal if and only if we can derive in $\mathcal{T}$ the judgement $b(x) = b'(x) \in B\ [x \in A]$. The composition in $\mathcal{C}_{\mathcal{T}}$ is defined by substitution, that is given $(x)\, b(x) \in \mathcal{C}_{\mathcal{T}}(A, B)$ and $(y)\, c(y) \in \mathcal{C}_{\mathcal{T}}(B, C)$ their composition is $(x)\, c(b(x))$. The identity is $(x)\, x \in \mathcal{C}_{\mathcal{T}}(A, A)$ obtained from $x \in A\ [x \in A]$.

Then by following the technique used in [Mai05a] we can prove that the above calculi provide internal languages of the corresponding structures and give rise to the initial structures in a modular way:

3.5. Theorem. *The following hold:*

- $\mathcal{T}_{sk}$ *provides an internal language for Skolem categories and* $\mathcal{C}_{\mathcal{T}_{sk}}$*, also called* $\mathcal{S}_{in}$*, is the initial Skolem category, and it is also the initial Skolem theory.*

- $\mathcal{T}_{alxt}$ *provides an internal language for arithmetic lextensive categories and* $\mathcal{C}_{\mathcal{T}_{alxt}}$ *is the initial arithmetic lextensive category.*

- $\mathcal{T}_{rl}$ *provides an internal language for regular locoi and* $\mathcal{C}_{\mathcal{T}_{rl}}$ *is the initial regular locos.*

- $\mathcal{T}_{pn}$ *provides an internal language for arithmetic pretopoi and* $\mathcal{C}_{\mathcal{T}_{pn}}$ *is the initial arithmetic pretopos.*

- $\mathcal{T}_{au}$ *provides an internal language for list-arithmetic pretopoi and* $\mathcal{C}_{\mathcal{T}_{au}}$ *is the initial list-arithmetic pretopos.*

3.6. Remark. It is worth mentioning that in [Wra85, Mor96] an initial Skolem category is built and used to model a programming language representing primitive recursive functions. Also in [Rol76] a term language including a primitive recursive operator is presented to reason within a Skolem theory.

3.7. First applications of our internal languages. Before entering into the main topic of our paper we give two applications regarding the use of internal languages of the mentioned categorical structures to prove some of their categorical properties.

The first application is a simple proof that parameterized list objects are local in a locos as first proved in [Coc90]. The second application is the proof that list-arithmetic pretopoi are closed under coequalizers.

3.8. Locality of locoi. Here we apply the internal type theory of a locos to show that its structure is local with a simple proof alternative to that in [Coc90]. First of all, note that the difficulty in proving that a locos $\mathcal{C}$ is local relies in proving that parameterized list objects are local. Indeed, the structure of lextensive category is local because the forgetful functor from $\mathcal{C}/A$ to $\mathcal{C}$ creates finite limits and stable finite disjoint coproducts.

To prove that $\mathcal{C}/A$ enjoys parameterized list objects we use the internal language of a locos and we anticipate a technique to define operations by iteration that we will also use to build free internal categories and diagrams in a list-arithmetic pretopos.

First note that the notion of parameterized list object corresponds in type theory to the notion of list type of a closed type. Hence, we can easily prove that the typed calculus for locoi can be taken to be a fragment $\mathcal{T}_l^*$ of the typed calculus $\mathcal{T}_l$ already presented. The fragment $\mathcal{T}_l^*$ is obtained from $\mathcal{T}_l$ by allowing *list types* $\mathrm{List}(A)$ only if A is a *closed type*. List objects are needed to interpret such list types, while their parameterization is needed to interpret the structural rule of weakening applied to such list types with their terms.

Now, we prove that in $\mathcal{T}_l^*$ we can represent list types $\mathrm{List}(B(x))\ [x \in A]$ of types $B(x)$ type $[x \in A]$ depending on at most one type. This is enough to deduce categorically that any slice category of a locos $\mathcal{C}$ is equipped with list objects. Indeed, for any object A in $\mathcal{C}$ and for any $\mathcal{C}/A$-object $b : B \to A$, thought of as a term $b(y) \in A\ [y \in B]$, we can form the list object $\mathrm{List}(b)$ as the interpretation of the list type of the dependent type $\Sigma_{y\in B}\ b(y) =_A x$ for $x \in A$. In $\mathcal{T}_l^*$ the type of lists of a dependent type $B(x)\ [x \in A]$ with A closed type can be defined by using the indexed sum type as follows: for $x \in A$

$$\mathrm{List}(B(x)) \equiv \Sigma_{w\in \mathrm{List}(\Sigma_{x\in A} B(x))} \quad \overline{\pi_1}(w) =_{\mathrm{List}(A)} \mathsf{mult}(x, \mathsf{lh}(w))$$

where $\overline{\pi_1} \equiv \mathsf{Lst}(\pi_1)$ is the lifting on lists of the first projection, $\mathsf{lh}(w)$ is the length of the list w (see the appendix for related definitions) and $\mathsf{mult}(x, \mathsf{lh}(w)) \equiv \underbrace{\lfloor x, x, \ldots, x \rfloor}_{n\text{-times}}$ is the list with n-times x, formally defined by induction on natural numbers as follows: for $x \in A$ and $n \in N$

$$\mathsf{mult}(x, n) \equiv \begin{cases} \epsilon & \text{if } n = 0 \\ \lfloor \mathsf{mult}(x, m), x \rfloor & \text{if } n = m + 1 \end{cases}$$

Then, the list constructors are the following: for $x \in A$

$$\begin{aligned} &\epsilon^{B(x)} \equiv \langle \epsilon^{\Sigma}, \mathsf{eq} \rangle \\ &\mathsf{cons}^{B(x)}(s, b) \equiv \langle \lfloor \pi_1(s), \langle x, b \rangle \rfloor, \mathsf{eq}\ \rangle \qquad \text{for } s \in \mathrm{List}(B(x)), b \in B(x) \end{aligned}$$

where ϵ^{Σ} is the empty list in $\mathrm{List}(\Sigma_{x\in A} B(x))$.

Finally, in order to prove the validity of the elimination rule on $\mathrm{List}(B(x))$ in the context of an extensional type theory, as described in [Mai05a], it is sufficient to prove the validity of the elimination rule towards types not depending on $\mathrm{List}(B(x))$ for $x \in A$

$$\text{E-list)}\ \frac{\begin{array}{l} C\ \text{type} \\ s \in \mathrm{List}(B(x)) \quad c \in C \quad l(y, z) \in C\ [y \in C, z \in B(x)] \end{array}}{\mathrm{El}_{\mathrm{List}}(c, l, s) \in C}$$

with corresponding β and η conversion rules. However, since we can not use the full elimination on $\mathrm{List}(\Sigma_{x\in A} B(x))$, given that $\mathrm{List}(B(x))$ is defined as a proper subtype of $\mathrm{List}(\Sigma_{x\in A} B(x))$, we define the elimination constructor by iteration, i.e. by induction on natural numbers.

Therefore, given a type $C\ [x \in A]$ — not depending on $\mathrm{List}(B(x))$ — together with a term $c \in C\ [x \in A, w \in \Gamma]$ (representing the value on the empty list, that is the base step) and $l(y, z) \in C\ [x \in A, w \in \Gamma, y \in C, z \in B(x)]$ (representing the inductive step) we define

$$\mathsf{El}_{\mathsf{list}}(c, l, z) \equiv \mathsf{Itr}(c, l, z, \mathsf{lh}(z_1)) \in C\ [x \in A, w \in \Gamma, z \in \mathrm{List}(B(x))]$$

where in turn

$$\mathsf{Itr}(c, l, z, n) \in C\ [x \in A, w \in \Gamma, z \in \mathrm{List}(B(x)), n \in N]$$

is defined as follows

$$\begin{aligned} &\mathsf{Itr}(c, l, z, 0) \equiv c \\ &\mathsf{Itr}(c, l, z, n + 1) \equiv \begin{cases} l(\mathsf{Itr}(c, l, z, n), \mathsf{p}_{\mathsf{n+1}}(z_1)) & \text{if } n + 1 \leq \mathsf{lh}(z_1) \\ \mathsf{Itr}(c, l, z, n) & \text{if } n + 1 > \mathsf{lh}(z_1) \end{cases} \end{aligned}$$

where $z_1 \equiv \pi_1(z)$ and $\mathsf{p_n}(z_1)$ is the nth element of the list z_1 (see the appendix). Then, we can easily prove that this elimination constructor enjoys the corresponding β and η conversions.

Moreover, we can show that such list objects in $\mathcal{C}/A$ are preserved by pullbacks. In particular the pullback of the interpretation of $\mathrm{List}(B(x))$ $[x \in A]$ in $\mathcal{C}/A$ along $f : C \to A$, thought of as a term $f(y) \in A$ $[y \in C]$, turns out to be isomorphic to the interpretation of

$$\mathrm{List}(B(x))[x/f] \equiv \Sigma_{w \in \mathrm{List}(\Sigma_{x\in A} B(x))} \quad \overline{\pi_1}(w) =_{\mathrm{List}(A)} \mathsf{mult}(f(y), \mathsf{lh}(w))$$

obtained by substituting x with $f(y)$ for $y \in C$. Analogously the pullback of the interpretation of $B(x)$ $[x \in A]$ in $\mathcal{C}/A$ along $f : C \to A$ turns out to be isomorphic to the interpretation of $B(f(y))$ $[y \in C]$ and then, its list object turns out to be the interpretation of

$$\mathrm{List}(B(x)[x/f]) \equiv \Sigma_{z \in \mathrm{List}(\Sigma_{y\in C} B(f(y)))} \quad \overline{\pi_1}(w) =_{\mathrm{List}(C)} \mathsf{mult}(y, \mathsf{lh}(w))$$

for $y \in C$. Now we show that $\mathrm{List}(B(x))[x/f]$ is isomorphic to $\mathrm{List}(B(x)[x/f])$ by just defining a morphism in the slice category over C

$$\phi : \mathrm{List}(B(x))[x/f] \to \mathrm{List}(B(x)[x/f])$$

as $\phi(w) \equiv \widetilde{\phi}(w, \mathsf{lh}(w_1))$ where $\widetilde{\phi}(w, n)$ is in turn defined by iteration on natural numbers as follows: for $y \in C$, $w \in \mathrm{List}(B(x))[x/f]$

$$\widetilde{\phi}(w, 0) \equiv \langle \epsilon^{\Sigma}, \mathsf{eq}\rangle$$
$$\widetilde{\phi}(w, n+1) \equiv \begin{cases} \langle \lfloor \pi_1(\widetilde{\phi}(w, n)), \langle y, \mathsf{p_{n+1}}(w_1)\rangle \rfloor, \mathsf{eq}\rangle & \text{if } n+1 \le \mathsf{lh}(w_1) \\ \widetilde{\phi}(w, n) & \text{if } n+1 > \mathsf{lh}(w_1) \end{cases}$$

where $w_1 \equiv \pi_1(w)$.

ϕ is an isomorphism whose inverse can be defined in an analogous way by iteration on natural numbers.

3.9. A LIST-ARITHMETIC PRETOPOS HAS COEQUALIZERS. Now we are going to prove that any list-arithmetic pretopos $\mathcal{U}$ has coequalizers by using its internal language. The key point is to show how to make the relation needed to coequalize two given maps into an equivalence relation. To do this we use lists in a crucial way.

3.10. PROPOSITION. *In any list-arithmetic pretopos $\mathcal{U}$ the coequalizer of any two given morphisms*

$$C \underset{b}{\overset{a}{\rightrightarrows}} A$$

exists.

PROOF. First, we consider the following reflexive and symmetric relation on A: for elements $z, z' \in A$

$$\begin{aligned} R(z,z') \equiv \;& z =_A z' \\ & \vee\ \exists_{c\in C}\,(a(c) =_A z \ \wedge\ b(c) =_A z') \\ & \vee\ \exists_{c\in C}\,(b(c) =_A z \wedge a(c) =_A z') \end{aligned}$$

Then, to define its transitive closure the idea is the following: in order to express that an element $z \in A$ is connected to $z' \in A$ through a finite list of elements x_1, x_2, x_3 such that

$$R(z,x_1)\ \wedge\ R(x_1,x_2)\ \wedge\ R(x_2,x_3)\ \wedge\ R(x_3,z')$$

we require the existence of a list $s \equiv \lfloor \langle z,x_1\rangle, \langle x_1,x_2\rangle, \langle x_2,x_3\rangle, \langle x_3,z'\rangle \rfloor$ of pairs related by R with the property that the list $\lfloor x_1,x_2,x_3 \rfloor$, containing the *second* components of elements in the *back* list $\lfloor \langle z,x_1\rangle, \langle x_1,x_2\rangle, \langle x_2,x_3\rangle \rfloor$, is equal to the list with the *first* components of elements in the *front* list $\lfloor \langle x_1,x_2\rangle, \langle x_2,x_3\rangle, \langle x_3,z'\rangle \rfloor$. Moreover, z must be the first component of the first element in s, while z' must be the second component of the last element in s.

Hence, we define:

$$\begin{aligned} R_t(z,z') \equiv \exists_{s\in List^*(Q)}\ & (\mathsf{Lst}(\widetilde{\pi_2})(\mathsf{bck}(s)) =_{List(A)} \mathsf{Lst}(\widetilde{\pi_1})(\mathsf{frt}(s)) \\ & \wedge\ z =_A \widetilde{\pi_1}(\mathsf{fst}(s))\ \wedge\ z' =_A \widetilde{\pi_2}(\mathsf{las}(s))) \end{aligned}$$

where $Q \equiv \Sigma_{w\in A\times A} R(\pi_1(w), \pi_2(w))$ and $\widetilde{\pi_1} \equiv \pi_1 \cdot \pi_1$ and $\widetilde{\pi_2} \equiv \pi_2 \cdot \pi_1$ (see the appendix for precise definitions of non-empty lists $\mathrm{List}^*(Q)$ and operations $\mathsf{Lst}(-)$, fst, las, bck and frt).

It follows easily that R_t is an equivalence relation and that $[-] : A \to A/R_t$, that is the map assigning to every $x \in A$ its equivalence class in A/R_t, is the coequalizer of a and b. ∎

4. Joyal's arithmetic universes

In this section we describe the construction of arithmetic universes given by André Joyal in the seventies. The construction can be read in unpublished notes by Joyal himself, Gavin Wraith [Wra85], in [Mor96] and partly in [Rol76].

Joyal built examples of arithmetic universes by taking the exact completion of the category of predicates built out of a Skolem theory.

Before giving the definition of predicate we define some primitive recursive operations and review some key properties of the natural numbers object in a Skolem category.

4.1. DEFINITION. Given a Skolem category $\mathcal{S}$, the predecessor, truncated subtraction, order, equality are defined as follows

$$x \dot{-} 1 \equiv \begin{cases} 0 \dot{-} 1 \equiv 0 \\ (n+1) \dot{-} 1 \equiv n \end{cases} \qquad\qquad x \dot{-} y \equiv \begin{cases} x \dot{-} 0 \equiv x \\ x \dot{-} (n+1) \equiv (x \dot{-} n) \dot{-} 1 \end{cases}$$

$$x \vee y \equiv x + (y \dot{-} x) \qquad x \wedge y \equiv x \dot{-} (x \dot{-} y) \qquad \mathsf{eq}(x,y) \equiv 1 \dot{-} ((x \dot{-} y) \vee (y \dot{-} x))$$

4.2. Definition. [Wra85, Mor96] A predicate in a Skolem category $\mathcal{S}$ is an $\mathcal{S}$-morphism $P : N \to N$ between natural numbers satisfying $P * P =_{\mathcal{S}} P$ where $* : N \times N \to N$ is the multiplication morphism of natural numbers.

In essence a predicate is an $\mathcal{S}$-morphism with values $0, 1$. It defines a decidable subobject of N whose elements can be thought of those with value 1. By using the above definitions as proved in classical recursion (for example see [Odi89]) we can show:

4.3. Proposition. *In any Skolem category $\mathcal{S}$ the collection of predicates forms a boolean algebra with bounded existential and universal quantifications where the order is defined as follows*

$$P \leq Q \equiv P \dot{-} Q =_{\mathcal{S}} 0$$

namely the truncated subtraction of them is the zero constant morphism in $\mathcal{S}$.

In particular, the equality induced by the order is the equality predicate above defined and it amounts to the equality of $\mathcal{S}$-morphisms, i.e. we have

$$P \leq Q \text{ and } Q \leq P \quad \textit{iff} \quad \mathsf{eq}(P,Q) =_{\mathcal{S}} 1 \quad \textit{iff} \quad P =_{\mathcal{S}} Q$$

Moreover, the conjunction is the multiplication, i.e. $P \wedge Q =_{\mathcal{S}} P * Q$, and the complement of P is $1 \dot{-} P$.

4.4. Proposition. *In any Skolem category $\mathcal{S}$ the following hold:*

- *the natural numbers object N is isomorphic to the binary product of itself $N \times N$, namely there exist $\mathcal{S}$-morphisms*

$$\mathsf{pair} : N \times N \to N \qquad \mathsf{pr}_1 : N \to N \qquad \mathsf{pr}_2 : N \to N$$
$$\textit{such that} \quad \langle \mathsf{pr}_1 \cdot \mathsf{pair}, \mathsf{pr}_2 \cdot \mathsf{pair} \rangle =_{\mathcal{S}} id \qquad \mathsf{pair} \cdot \langle \mathsf{pr}_1, \mathsf{pr}_2 \rangle =_{\mathcal{S}} id$$

- *in $\mathcal{S}$ the natural numbers object N is a parameterized list object over itself.*

Proof. We employ the internal language of a Skolem theory.

By following standard ideas in recursion theory (see for example [Odi89]) we define the pairing map as follows

$$\mathsf{pair}(x,y) \equiv 2^x * (2 * y + 1) \dot{-} 1$$

which provides the required isomorphism with the following projections

$$\mathsf{pr}_1(z) \equiv \min\{\, x \leq z \mid \exists y \leq z \; z =_{\mathcal{S}} \mathsf{pair}(x,y) \,\}$$
$$\mathsf{pr}_2(z) \equiv \min\{\, y \leq z \mid \exists x \leq z \; z =_{\mathcal{S}} \mathsf{pair}(x,y) \,\}$$

N turns out to be a list object over itself by using the binary representation of natural numbers. The idea is to represent the empty list as zero, and a list like $\lfloor n_1, n_2, \ldots, n_m \rfloor$ as the number represented in binary digits as the list starting with 1 followed by a number of zeros equal to the last element n_m of the list, and then again 1 followed by a number of zeros equal to the last but one element n_{m-1} of the list and so on. For example:

$$\begin{aligned} \epsilon &\longmapsto 0 \\ \lfloor 0 \rfloor &\longmapsto 1 \\ \lfloor 3 \rfloor &\longmapsto 1000 = 2^3 \\ \lfloor 0, 1, 3 \rfloor &\longmapsto 1000101 = 2^6 + 2^2 + 1 \end{aligned}$$

Conversely any natural number in binary digits represents a list where each nth element counts the nested zeros after the nth-1 counted from the right.

More formally, the list constructors are defined as follows: for $s, n \in N$

1) $\epsilon \equiv_{\mathrm{List}(N)} 0$

2) $\mathsf{cons}_{\mathrm{List}(N)}(s, n) \equiv 2^c \cdot 2^n + s \qquad$ where $c \equiv \min\{\, x \leq s \mid 2^x > s \,\}$.

After defined the nth-projection of any natural number thought of as a list, we define the list elimination constructor $\mathrm{rec}_{\mathrm{List}(N)}(b, g, z)$ by induction on natural numbers, by starting from b and then iterating the application of g up to the length of z, if this is not zero. This construction is analogous to the corresponding one for list objects in a slice category of a locos in section 3.8. ■

4.5. Remark. In [Rol76] and in a draft by Joyal, a predicate is defined as an $\mathcal{S}$-morphism $P : N^k \to N$ from a finite product of N to N satisfying $P \wedge 1 =_{\mathcal{S}} P$. If $\mathcal{S}$ is a *Skolem theory*, then these predicates represent the decidable subobjects of $\mathcal{S}$-objects, given that objects in $\mathcal{S}$ are only finite products of N. However thanks to proposition 4.4, predicates on N^k with $k \geq 1$ are in bijective correspondence with predicates on N by precomposing with the isomorphism between N^k and N. Hence, without loss of generality, we can restrict to predicates from N to N. In this case definition 4.7 is equivalent to that by Joyal since $P * P =_{\mathcal{S}} P$ iff $P \wedge 1 =_{\mathcal{S}} P$.

Now, we are ready to define the category of predicates in a Skolem theory $\mathcal{S}$:

4.6. Definition. [Mor96, Wra85] Given a Skolem theory $\mathcal{S}$ the category $\mathrm{Pred}(\mathcal{S})$ of *predicates* in $\mathcal{S}$ is defined as follows:

- $\mathsf{Ob}(\mathrm{Pred}(\mathcal{S}))$: predicates in $\mathcal{S}$, namely morphisms $P : N \to N$ such that $P * P =_{\mathcal{S}} P$;
- $\mathsf{Hom}(P, Q)$: $\mathcal{S}$-morphisms $f : N \to N$ such that $P \leq Q \cdot f$ (where $Q \cdot f$ is the composition of Q with f) and two such $\mathcal{S}$-morphisms $f : N \to N$ and $g : N \to N$ are equal iff $P * f =_{\mathcal{S}} P * g$.

Note that, if the Skolem theory is the full subcategory $\mathcal{S}_{ZFC}$ of the usual category of classical ZFC-sets with only finite products of natural numbers as objects, then $\mathrm{Pred}(\mathcal{S}_{ZFC})$ is the category having subsets of natural numbers as objects. Indeed a subset is characterized by the elements with value 1 through a predicate P that is its characteristic function. Moreover, morphisms of $\mathrm{Pred}(\mathcal{S}_{ZFC})$ turn out to be functions mapping the domain subset to the codomain subset. Finally, two maps are considered equal if they are equal on the domain subset.

Joyal proved that the category of predicates $\mathrm{Pred}(\mathcal{S})$ is a regular locos:

4.7. PROPOSITION. [Wra85, Mor96, Rol76] *The category* $\mathrm{Pred}(\mathcal{S})$ *of a Skolem theory* $\mathcal{S}$ *is regular with stable finite disjoint coproducts and parameterized list objects. Moreover, epimorphisms of image factorizations split.*

PROOF. We describe the claimed structure of $\mathrm{Pred}(\mathcal{S})$.

The *terminal object* is given by the predicate $\top_{\mathrm{Pred}} : N \to N$ defined as the singleton zero for $n \in N$

$$\top_{\mathrm{Pred}}(n) \equiv 1 \dot{-} n$$

The *binary product* of two predicates P and Q is given by

$$(P \times Q)(n) \equiv P(\mathsf{pr}_1(n)) * Q(\mathsf{pr}_2(n)) \quad \text{for } n \in N$$

The *two projections* of $P \times Q$ are $\pi_P : P \times Q \to P$ and $\pi_Q : P \times Q \to Q$ defined as: $\pi_P(n) \equiv \mathsf{pr}_1(n)$ and $\pi_Q(n) \equiv \mathsf{pr}_2(n)$ for $n \in N$.

The *pairing morphism* of $f : C \to P$ and $g : C \to Q$ is defined as $\langle f, g\rangle_{\mathrm{Pred}}(n) \equiv \mathsf{pair}(f(n), g(n))$ for $n \in N$.

The *equalizer* of two morphisms $f, g : P \to Q$ is

$$\mathrm{Eq}(f, g)(n) \equiv P(n) * \mathsf{eq}(f(n), g(n)) \quad \text{for } n \in N$$

and the equalizer injection $e : \mathrm{Eq}(f, g) \to P$ is defined as $e(n) \equiv n$ for all $n \in N$. Obviously, the morphism factoring any $h : C \to P$ equalizing f and g is h itself but with $\mathrm{Eq}(f, g)$ as codomain.

The *initial object* is given by the predicate $\perp_{\mathrm{Pred}} : N \to N$ defined as

$$\perp_{\mathrm{Pred}}(n) \equiv 0 \quad \text{for } n \in N$$

The *binary coproduct* of two predicates P and Q is defined as follows: for $n \in N$

$$(P + Q)(n) \equiv \mathsf{even}(n) * P(n/2) + \mathsf{odd}(n) * Q(n \dot{-} 1/2)$$

where $n/2$ is the quotient of the division by 2 and the operations $\mathsf{even}, \mathsf{odd}$ are defined as follows:

$$\mathsf{even}(n) \equiv \begin{cases} \mathsf{even}(0) \equiv 1 \\ \mathsf{even}(n+1) \equiv 1 \dot{-} \mathsf{even}(n) \end{cases} \qquad \mathsf{odd}(n) \equiv \begin{cases} \mathsf{odd}(0) \equiv 0 \\ \mathsf{odd}(n+1) \equiv 1 \dot{-} \mathsf{odd}(n) \end{cases}$$

The *injections* $i_P : P \to P + Q$ and $i_Q : Q \to P + Q$ are defined as $i_P(n) \equiv 2 * n$ and $i_Q(n) \equiv 2 * n + 1$ for $n \in N$. Given $f : P \to C$ and $g : Q \to C$ their coproduct morphism $f + g : P + Q \to C$ is defined as follows: for $n \in N$

$$(f + g)(n) \equiv \mathsf{even}(n) * f(n/2) + \mathsf{odd}(n) * g(n \dot{-} 1/2)$$

It follows easily that such binary coproducts are stable under pullbacks.

The *image object* of a morphism $f : P \to Q$ is defined as follows: for $n \in N$

$$(\mathsf{Im} f)(n) \equiv P(n) * \mathsf{eq}(n, p(n))$$

where

$$p(n) \equiv \min\{\, x \leq n \;\mid\; P(x) * \mathsf{eq}(f(x), f(n)) =_{\mathcal{S}} 1 \,\}$$

Its *image morphism* $\iota_f : \mathsf{Im} f \to Q$ is defined as $\iota_f(n) \equiv f(n)$ for $n \in N$ and this factors f through $p : P \to \mathsf{Im} f$ defined as the above $p(n)$ for $n \in N$, i.e. $f = \iota_f \cdot p$ in $\mathrm{Pred}(\mathcal{S})$. Moreover, p has a section $p^{-1} : \mathsf{Im} f \to P$ defined as $p^{-1}(n) \equiv n$, that is $p \cdot p^{-1} = \mathrm{id}$ in $\mathrm{Pred}(\mathcal{S})$. Since epimorphisms of image factorizations split then images are stable under pullbacks.

The *parameterized list object* of a predicate is defined by using the fact that N is isomorphic to $\mathrm{List}(N)$ by prop 4.4. Hence, given a predicate P its *parameterized list object* is

$$\mathrm{List}(P)(n) \equiv \begin{cases} 1 & \text{if } n \equiv 0 \\ \mathrm{List}(P)(m) * P(k) & \text{if } n \equiv \mathsf{cons}_{\mathrm{List}(N)}(m, k) \end{cases}$$

The list structure of $\mathrm{List}(P)$ is induced by that of $\mathrm{List}(N)$ as follows:

${r_0}^{\mathrm{List}(P)} : \top \to \mathrm{List}(P)$ is defined for $n \in N$ as

$${r_0}^{\mathrm{List}(P)}(n) \equiv 0$$

and ${r_1}^{\mathrm{List}(P)} : \mathrm{List}(P) \times P \to \mathrm{List}(P)$ is defined for $n \in N$ as

$${r_1}^{\mathrm{List}(P)}(n) \equiv \mathsf{cons}_{\mathrm{List}(N)}(\mathsf{pr_1}(n), \mathsf{pr_2}(n))$$

Then, given $b : C \to Q$ and $g : Q \times P \to Q$ we put $\mathrm{rec}_{\mathrm{List}(P)}(b, g)(n) \equiv \mathrm{rec}_{\mathrm{List}(N)}(b, g, n)$ for $n \in N$. ∎

Now, we are finally ready to give Joyal's definition of arithmetic universe. It amounts to the exact completion $(\mathrm{Pred}(\mathcal{S}))_{\mathrm{ex}}$ of the category of predicates built out of a Skolem theory $\mathcal{S}$. Given that the category of predicates in a Skolem theory is regular as proved in proposition 4.7, then we can use the exact completion performed on a regular category (see [CV98] and loc. cit.). However, since in $\mathrm{Pred}(\mathcal{S})$ epimorphisms of image factorizations split, then we can define the morphisms in such exact completions $(\mathrm{Pred}(\mathcal{S}))_{\mathrm{ex}}$ just as $\mathrm{Pred}(\mathcal{S})$-morphisms preserving the equivalence relations. Therefore, we put:

4.8. Definition. A *Joyal-arithmetic universe* is the category $(\mathrm{Pred}(\mathcal{S}))_{\mathrm{ex}}$ built out of a Skolem theory $\mathcal{S}$ as follows:

- $\mathsf{Ob}(\mathrm{Pred}(\mathcal{S}))_{\mathrm{ex}}$: (X, R) where X is an object of $\mathrm{Pred}(\mathcal{S})$ and $R : dom(R) \to X \times X$ is a monic categorical equivalence relation on X;
- $\mathsf{Hom}((X, R), (Y, S))$: $\mathrm{Pred}(\mathcal{S})$-morphisms $f : X \to Y$ preserving the equivalence relations, that is $R \leq (f \times f)^*(S)$ as subobjects where $(f \times f)^*(S)$ is the first projection of the pullback of S along $f \times f$.

  Moreover, two arrows $f, g : (X, R) \to (Y, S)$ in $\mathsf{Hom}((X, R), (Y, S))$ are equal iff $R \leq (f \times g)^*(S)$.

We also recall the definition of the embedding of $\mathrm{Pred}(\mathcal{S})$ into $(\mathrm{Pred}(\mathcal{S}))_{\mathrm{ex}}$:

4.9. Definition. The embedding functor

$$\mathcal{Y} : \mathrm{Pred}(\mathcal{S}) \longrightarrow (\mathrm{Pred}(\mathcal{S}))_{\mathrm{ex}}$$

is defined as follows:

$$\mathcal{Y}(P) \equiv (P, x =_P y) \qquad\qquad \mathcal{Y}(f) \equiv f$$

where $x =_P y \equiv \mathrm{Eq}(\pi_1, \pi_2)$ is the identity relation on P, that is the equalizer of the projections π_1, π_2 from $P \times P$ in $\mathrm{Pred}(\mathcal{S})$ (categorically $\mathrm{Eq}(\pi_1, \pi_2)$ is isomorphic to the diagonal $\langle \mathrm{id}, \mathrm{id} \rangle : P \to P \times P$).

We now give the main result of this section:

4.10. Proposition. $(\mathrm{Pred}(\mathcal{S}))_{ex}$ *is a list-arithmetic pretopos.*

Proof. In [Wra85, Mor96] it is shown that $(\mathrm{Pred}(\mathcal{S}))_{\mathrm{ex}}$ is a pretopos with list objects but we can show that list objects are also parameterized. We know from [CV98] that this category is exact and hence we just describe finite coproducts and list objects (see [Car95, BCRS98]).

Note that in doing this we employ the internal language of a regular locos provided by $\mathcal{T}_{\mathrm{rl}}$ according to which we can consider the equivalence relation R as a type-theoretic equivalence relation.

The *initial object* of $(\mathrm{Pred}(\mathcal{S}))_{\mathrm{ex}}$ is $(\perp_{\mathrm{Pred}}, x =_{\perp_{\mathrm{Pred}}} x)$ where $x =_{\perp_{\mathrm{Pred}}} x$ is the identity relation.

The *binary coproduct* of two objects (X, R) and (Y, S) is $(X + Y, R + S)$ where $X + Y$ is the coproduct in $\mathrm{Pred}(\mathcal{S})$ and $R + S$ is defined as follows: for $z, z' \in X + Y$

$$(R + S)(z, z') \equiv \begin{cases} \exists_{x \in X}\, \exists_{x' \in X}\, (z =_X i_X(x)) \wedge (z' =_X i_X(x')) \wedge R(x, x') \\ \vee \\ \exists_{y \in Y}\, \exists_{y' \in Y}\, (z = i_Y(y)) \;\wedge\; (z' =_Y i_Y(y')) \wedge S(y, y') \\ \vee \\ \perp \wedge \exists_{x \in X}\, \exists_{y \in Y}\, ((z =_X i_X(x) \wedge z' = i_Y(y)) \\ \qquad\qquad\qquad \vee\, (z' =_X i_X(x) \wedge z = i_Y(y))) \end{cases}$$

$R+S$ can be proved to be an equivalence relation since in $\mathcal{T}_{\mathrm{rl}}$, as proved in [Mai05a], given an element $z \in X+Y$ we can derive a proof of

$$(\exists_{x\in X}\ \mathsf{inl}(x) =_X z) \vee (\exists_{y\in Y}\ \mathsf{inr}(y) =_Y z)$$

The *injections* are $i_{(X,R)} : (X,R) \to (X+Y, R+S)$ and $i_{(Y,S)} : (Y,S) \to (X+Y, R+S)$ defined as $i_{(X,R)}(x) \equiv i_X(x)$ for $x \in X$ and $i_{(Y,S)}(y) \equiv i_Y(y)$ for $y \in Y$.

The list object on (X,R) is $(\mathrm{List}(X), \mathrm{List}(R))$ where $\mathrm{List}(X)$ is the list object on X in $\mathrm{Pred}(\mathcal{S})$ and $\mathrm{List}(R)$ is defined as follows:

$$\mathrm{List}(R)(s,s') \equiv \begin{cases} \exists_{t\in \mathrm{List}(\Sigma_{x\in X}\Sigma_{x'\in X} R(x,x'))} \quad (\overline{\pi_1}(t) =_{\mathrm{List}(X)} s \wedge \overline{\pi_2}(t) =_{\mathrm{List}(X)} s' \\ \qquad\qquad\qquad\qquad\qquad\qquad \wedge\ \mathsf{lh}(s) =_{\mathrm{List}(X)} \mathsf{lh}(s')) \\ \vee \\ \bot\ \wedge\ (\mathsf{lh}(s) < \mathsf{lh}(s')\ \vee\ \mathsf{lh}(s') < \mathsf{lh}(s)) \end{cases}$$

where $\mathsf{lh}(l)$ is the length of the list l and $\overline{\pi_1} \equiv \mathsf{Lst}(\pi_1)$ and $\overline{\pi_2} \equiv \mathsf{Lst}(\pi_1 \cdot \pi_2)$ are obtained by lifting the first and second projections on lists (see the appendix) and for $x, x' \in N$ we put $x < x' \equiv \exists_{y\in N}\ x' =_N x + (y+1)$ for $x, x' \in N$.

$\mathrm{List}(R)$ can be proved to be an equivalence relation since in $\mathcal{T}_{\mathrm{rl}}$, given $s, s' \in \mathrm{List}(X)$, we can derive a proof of

$$\mathsf{lh}(s) =_{\mathrm{List}(X)} \mathsf{lh}(s')\ \vee\ \mathsf{lh}(s) < \mathsf{lh}(s')\ \vee\ \mathsf{lh}(s') < \mathsf{lh}(s)$$

Moreover, the list structure on $(\mathrm{List}(X), \mathrm{List}(R))$ is defined as follows:

$$\begin{array}{lll} r_o^{(X,R)} : (\top, x =_\top x) \to (\mathrm{List}(X), \mathrm{List}(R)) & \text{as} & r_o^{(X,R)}(x) \equiv r_o^X(x) \\ & & \text{for } x \in X \\ r_1^{(X,R)} : (\mathrm{List}(X), \mathrm{List}(R)) \times (X,R) \to (\mathrm{List}(X), \mathrm{List}(R)) & \text{as} & r_1^{(X,R)}(s,x) \equiv r_1^X(s,x) \\ & & \text{for } s \in \mathrm{List}(X), x \in X \end{array}$$

The list object is parameterized. Indeed, given the morphisms $f : (Z,H) \to (C,M)$ and $g : (C,M) \times (X,R) \to (C,M)$, by using the property of the list object $\mathrm{List}(X)$ in $\mathrm{Pred}(\mathcal{S})$ we obtain a $\mathrm{Pred}(\mathcal{S})$-morphism

$$\mathrm{rec}(f,g) : Z \times \mathrm{List}(X) \longrightarrow C$$

satisfying $\mathrm{rec}(f,g) \cdot \langle \mathrm{id}, r_o^X \cdot !_Z \rangle = f$ and $\mathrm{rec}(f,g) \cdot (\mathrm{id} \times r_1^X) = (g \cdot (\mathrm{rec}(f,g) \times \mathrm{id})) \cdot \sigma$ in $\mathrm{Pred}(\mathcal{S})$ and hence in $(\mathrm{Pred}(\mathcal{S}))_{\mathrm{ex}}$ if $\mathrm{rec}(f,g)$ is a $(\mathrm{Pred}(\mathcal{S}))_{\mathrm{ex}}$-morphism.

We can prove that $\mathrm{rec}(f,g)$ is a morphism in $(\mathrm{Pred}(\mathcal{S}))_{\mathrm{ex}}$ from $(Z,H) \times (\mathrm{List}(X), \mathrm{List}(R))$ to (C,M). Indeed, given $z, z' \in Z$ for which $H(z,z')$ holds and $s, s' \in \mathrm{List}(X)$ for which $\mathrm{List}(R)(s,s')$ holds, we prove that $M(\,\mathrm{rec}(f,g)(\langle z,s\rangle)\,,\, \mathrm{rec}(f,g)(\langle z',s'\rangle)\,)$ holds by showing by induction on $n \in N$ the validity of

$$M(\,\mathrm{rec}(f,g)(\langle z, \mathsf{part_n}(s)\rangle)\,,\, \mathrm{rec}(f,g)(\langle z', \mathsf{part_n}(s')\rangle)\,)$$

where $\mathsf{part_n}(s)$ is a list operation defined in the appendix. This lets us prove what wanted since $\mathsf{part_k}(s) =_{\mathrm{List}(X)} s$ holds for $k \geq \mathsf{lh}(s)$.

Finally, we can show uniqueness of $\mathrm{rec}(f,g)$ in an analogous way. Indeed, for any morphism h with the same domain and codomain of $\mathrm{rec}(f,g)$ satisfying

$$h \cdot \langle \mathrm{id}, r_o^{(X,R)} \cdot !_{(Z,H)} \rangle = f$$
$$h \cdot (\mathrm{id} \times r_1^{(X,R)}) = (g \cdot (h \times \mathrm{id})) \cdot \sigma$$

in $(\mathrm{Pred}(\mathcal{S}))_{\mathrm{ex}}$, we can then prove by induction on $n \in N$ the validity of

$$M(\,\mathrm{rec}(f,g)(\langle z, \mathsf{part_n}(s)\rangle)\,,\; h(\langle z', \mathsf{part_n}(s')\rangle)\,)$$

from which $\mathrm{rec}(f,g) =_{(\mathrm{Pred}(\mathcal{S}))_{\mathrm{ex}}} h$ follows. ■

Observe that $(\mathrm{Pred}(\mathcal{S}_{\mathrm{in}}))_{\mathrm{ex}}$ built out of the initial Skolem category $\mathcal{S}_{\mathrm{in}}$ amounts to the *initial arithmetic universe* $(\mathrm{Pred}(\mathcal{S}_{in}))_{ex}$ in the category of Joyal-arithmetic universes and functors induced from functors between Skolem theories, with a fixed choice of their structure, preserving the Skolem structure. Note that $\mathrm{Pred}(\mathcal{S}_{\mathrm{in}})$ turns out to be the category of *primitive recursive predicates.*

In the following we simply call $\mathcal{A}_{\mathrm{in}}$ the initial arithmetic universe $(\mathrm{Pred}(\mathcal{S}_{\mathrm{in}}))_{\mathrm{ex}}$.

5. The category of primitive recursive predicates via type theory

Here we outline how the category of primitive recursive predicates is equivalent to the initial arithmetic lextensive category $\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}$ and also to the initial regular locos $\mathcal{C}_{\mathcal{T}_{\mathrm{rl}}}$.

To this purpose, we define two embeddings

$$\mathcal{E}_{\mathrm{ad}} : \mathrm{Pred}(\mathcal{S}_{\mathrm{in}}) \longrightarrow \mathcal{C}_{\mathcal{T}_{\mathrm{ad}}} \qquad\qquad \mathcal{E}_{\mathrm{rl}} : \mathrm{Pred}(\mathcal{S}_{\mathrm{in}}) \longrightarrow \mathcal{C}_{\mathcal{T}_{\mathrm{rl}}}$$

where $\mathcal{E}_{\mathrm{ad}}$ is defined as follows:

- $\mathcal{E}_{\mathrm{ad}}(P) \equiv \Sigma_{x\in N} P(x) =_N 1$

  for any predicate P in $\mathsf{Ob}\,\mathrm{Pred}(\mathcal{S}_{\mathrm{in}})$.

- $\mathcal{E}_{\mathrm{ad}}(f)(z) \equiv \langle f(\pi_1(z)), \mathsf{eq}\rangle \in \Sigma_{x\in N} Q(x) =_N 1 \; [z \in \Sigma_{x\in N} P(x) =_N 1]$

  for $f : P \to Q$ in $\mathrm{Pred}(\mathcal{S}_{\mathrm{in}})$, that is $\mathcal{E}_{\mathrm{ad}}(f)$ is the function associating $f(x)$ satisfying $Q(f(x)) =_N 1$ to any $x \in N$ such that $P(x) =_N 1$ holds.

The functor $\mathcal{E}_{\mathrm{rl}}$ can be defined essentially in the same way. Indeed observe that the calculus $\mathcal{T}_{\mathrm{ad}}$ is a fragment of $\mathcal{T}_{\mathrm{rl}}$ after recalling to represent the natural numbers object as the list type of the terminal type. This means that the category $\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}$ is a subcategory of $\mathcal{C}_{\mathcal{T}_{\mathrm{rl}}}$ and that the embedding $\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}} \xhookrightarrow{i^{\mathrm{rl}}_{\mathrm{ad}}} \mathcal{C}_{\mathcal{T}_{\mathrm{rl}}}$, which is the identity on objects and morphisms, preserves the arithmetic lextensive structure. Then, we define $\mathcal{E}_{\mathrm{rl}}$ as the composition of the embedding $i^{\mathrm{rl}}_{\mathrm{ad}}$ with $\mathcal{E}_{\mathrm{ad}}$

$$\mathcal{E}_{\mathrm{rl}} \equiv i^{\mathrm{rl}}_{\mathrm{ad}} \cdot \mathcal{E}_{\mathrm{ad}}$$

In the next we prove that the embeddings $\mathcal{E}_{\mathrm{ad}}$ and $\mathcal{E}_{\mathrm{rl}}$ preserve the relevant structure of their domain categories. To prove this we will make use of the fact that $0 =_N n+1$ is false:

5.1. Lemma. *In the typed calculus $\mathcal{T}_{ad}$ from a proof of $0 =_N n+1$ for $n \in N$, we can derive a proof of falsum $\bot$. Therefore, also in any arithmetic lextensive category we can prove that $0 =_N n+1$ is false for $n \in N$.*

Proof. By the elimination rule on the natural numbers $n \in N$ we can define the term

$$(\mathsf{tt} \oplus \mathsf{ff})\ (n) \in \top \oplus \top$$

as follows

$$(\mathsf{tt} \oplus \mathsf{ff})\ (n) \equiv \begin{cases} \mathsf{tt} & \text{if } n \equiv 0 \\ \mathsf{ff} & \text{if } n \equiv m+1 \end{cases}$$

where $\mathsf{tt} \equiv \mathsf{inl}(\star)$ and $\mathsf{ff} \equiv \mathsf{inr}(\star)$.

Hence, if there existed a proof of $0 =_N n+1$, then by the rule E-Eq) of the extensional propositional equality we would get $0 = n+1 \in N$. Hence, by equality preservation of $\mathsf{tt} \oplus \mathsf{ff}$ we would obtain that $(\mathsf{tt} \oplus \mathsf{ff})(0) = (\mathsf{tt} \oplus \mathsf{ff})(n+1) \in \top \oplus \top$. Now, after recalling that $(\mathsf{tt} \oplus \mathsf{ff})(0) = \mathsf{tt}$ and $(\mathsf{tt} \oplus \mathsf{ff})(n+1) = \mathsf{ff}$, we would also get

$$\mathsf{ff} = \mathsf{tt} \in \top \oplus \top$$

By disjointness of sum, namely by rule dis-+), we would conclude a proof of $\bot$ as claimed.

Now, we are ready to prove that the embeddings $\mathcal{E}_{\mathrm{ad}}$ and $\mathcal{E}_{\mathrm{rl}}$ have the following preservation properties:

5.2. Lemma. *The embedding $\mathcal{E}_{ad}$ is an arithmetic lextensive functor and $\mathcal{E}_{rl}$ is a regular locos functor.*

Proof. We first show that $\mathcal{E}_{\mathrm{ad}}$ preserves the arithmetic lextensive structure. The structure of $\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}$ is described in [Mai05a].

$\mathcal{E}_{\mathrm{ad}}$ preserves the terminal object because, thanks to lemma 5.1, $\Sigma_{x\in N}\top_{\mathrm{Pred}}(x) =_N 1$ is isomorphic to $\top$ which can be chosen as a terminal object in $\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}$. Indeed, the key point is to prove that from $\langle x, \mathsf{eq}\rangle \in \Sigma_{x\in N}\top_{\mathrm{Pred}}(x) =_N 1$ we obtain $\langle x, \mathsf{eq}\rangle =_{\mathcal{E}_{\mathrm{ad}}(\top_{\mathrm{Pred}})} \langle 0, \mathsf{eq}\rangle$. This follows by elimination of disjunction from

$$x =_N 0 \ \vee\ \exists_{y\in N} x =_N y+1$$

that can be proved by induction on natural numbers. In the case $x =_N 0$ we conclude trivially, and in the case $x =_N y+1$ for some $y \in N$ we conclude by lemma 5.1.

$\mathcal{E}_{\mathrm{ad}}$ preserves binary products. Recall from [Mai05a] that the indexed sum type allows to define the binary product $\mathcal{E}_{\mathrm{ad}}(P) \times \mathcal{E}_{\mathrm{ad}}(Q)$ in $\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}$, for P, Q predicates, with projections $\pi_1(z) \in \mathcal{E}_{\mathrm{ad}}(P)$ and $\pi_2(z) \in \mathcal{E}_{\mathrm{ad}}(Q)$, respectively, for $z \in \mathcal{E}_{\mathrm{ad}}(P) \times \mathcal{E}_{\mathrm{ad}}(Q)$.

Then, observe that the morphism induced by the product projections in $\mathrm{Pred}(\mathcal{S}_{\mathrm{in}})$

$$\langle \mathcal{E}_{\mathrm{ad}}(\pi_P), \mathcal{E}_{\mathrm{ad}}(\pi_Q)\rangle : \mathcal{E}_{\mathrm{ad}}(P\times Q) \longrightarrow \mathcal{E}_{\mathrm{ad}}(P)\times \mathcal{E}_{\mathrm{ad}}(Q)$$

has an inverse defined as follows: for $z\in \mathcal{E}_{\mathrm{ad}}(P)\times \mathcal{E}_{\mathrm{ad}}(Q)$

$$\mathrm{In}_{\times}(z) \equiv \langle \mathsf{pair}(\pi_1(z_1), \pi_1(z_2)), \mathsf{eq}\rangle$$

where $z_1 \equiv \pi_1(z)$ and $z_2 \equiv \pi_2(z)$.

$\mathcal{E}_{\mathrm{ad}}$ preserves equalizers. Recall that the equalizer of $\mathcal{E}_{\mathrm{ad}}(f)$ and $\mathcal{E}_{\mathrm{ad}}(g)$ in $\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}$, for $f,g : P \to Q$ in $\mathrm{Pred}(\mathcal{S}_{\mathrm{in}})$, is defined as

$$\mathrm{Eq}_{\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}}(\mathcal{E}_{\mathrm{ad}}(f), \mathcal{E}_{\mathrm{ad}}(g)) \equiv \Sigma_{z\in\mathcal{E}_{\mathrm{ad}}(P)}\ \mathcal{E}_{\mathrm{ad}}(f)(z) =_{\mathcal{E}_{\mathrm{ad}}(Q)} \mathcal{E}_{\mathrm{ad}}(g)(z)$$

and the equalizer map is $\pi_1 : \mathrm{Eq}_{\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}}(\mathcal{E}_{\mathrm{ad}}(f), \mathcal{E}_{\mathrm{ad}}(g)) \longrightarrow \mathcal{E}_{\mathrm{ad}}(P)$. Since the equalizer embedding $e : \mathrm{Eq}(f,g) \to P$ gives $\mathcal{E}_{\mathrm{ad}}(f)\cdot\mathcal{E}_{\mathrm{ad}}(e) = \mathcal{E}_{\mathrm{ad}}(g)\cdot\mathcal{E}_{\mathrm{ad}}(e)$ in $\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}$, it induces a morphism

$$\mathrm{Un}_{eq} : \mathcal{E}_{\mathrm{ad}}(\mathrm{Eq}(f,g)) \longrightarrow \mathrm{Eq}_{\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}}(\mathcal{E}_{\mathrm{ad}}(f), \mathcal{E}_{\mathrm{ad}}(g))$$

defined as $\mathrm{Un}_{eq}(z) \equiv \langle \mathcal{E}_{\mathrm{ad}}(e)(z), \mathsf{eq}\rangle$ for $z \in \mathcal{E}_{\mathrm{ad}}(\mathrm{Eq}(f,g))$, which is indeed an isomorphism with inverse In_{eq} defined as $\mathrm{In}_{eq}(z) \equiv \langle \pi_1(\pi_1(z)), \mathsf{eq}\rangle$ for $z \in \mathrm{Eq}_{\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}}(\mathcal{E}_{\mathrm{ad}}(f), \mathcal{E}_{\mathrm{ad}}(g))$.

$\mathcal{E}_{\mathrm{ad}}$ preserves the initial object, since $\perp$, which can be chosen as the initial object in $\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}$, is isomorphic to $\Sigma_{x\in N} 0 =_N 1$ thanks to lemma 5.1.

$\mathcal{E}_{\mathrm{ad}}$ preserves binary coproducts since in $\mathcal{T}_{\mathrm{ad}}$ we can prove that N is isomorphic to the coproduct of even numbers with odd ones. To prove this more in detail, recall that the type $\mathcal{E}_{\mathrm{ad}}(P)\ +\ \mathcal{E}_{\mathrm{ad}}(Q)$ is a binary coproduct of $\mathcal{E}_{\mathrm{ad}}(P)$ and $\mathcal{E}_{\mathrm{ad}}(Q)$ for given predicates P and Q, and its injections are the terms $\mathsf{inl}(z) \in \mathcal{E}_{\mathrm{ad}}(P)\ +\ \mathcal{E}_{\mathrm{ad}}(Q)$ for $z\in\mathcal{E}_{\mathrm{ad}}(P)$ and $\mathsf{inr}(w) \in \mathcal{E}_{\mathrm{ad}}(P)\ +\ \mathcal{E}_{\mathrm{ad}}(Q)$ for $w \in \mathcal{E}_{\mathrm{ad}}(Q)$.

Now, the coproduct morphism induced by the injections in $\mathrm{Pred}(\mathcal{S}_{\mathrm{in}})$

$$\mathcal{E}_{\mathrm{ad}}(j_P)\oplus\mathcal{E}_{\mathrm{ad}}(j_Q) : \mathcal{E}_{\mathrm{ad}}(P)\ +\ \mathcal{E}_{\mathrm{ad}}(Q) \longrightarrow \mathcal{E}_{\mathrm{ad}}(P+Q)$$

is an isomorphism with inverse $\mathrm{In}_{\oplus}$ defined as follows: for $z \in \mathcal{E}_{\mathrm{ad}}(P+Q)$

$$\mathrm{In}_{\oplus}(z) \equiv \begin{cases} \mathsf{inl}(\langle z_1/2, \mathsf{eq}\rangle) & \text{if } \mathsf{even}(z_1) = 1 \\ \mathsf{inr}(\langle (z_1 \dot{-} 1)/2, \mathsf{eq}\rangle) & \text{if } \mathsf{odd}(z_1) = 1 \end{cases}$$

where $z_1 \equiv \pi_1(z)$.

$\mathcal{E}_{\mathrm{rl}}$ is a lextensive functor being the composition of $\mathcal{E}_{\mathrm{ad}}$ with an embedding preserving such a structure.

In order to prove that $\mathcal{E}_{\mathrm{rl}}$ is a regular locos functor, we need to show that it preserves images and list objects.

We start by showing that $\mathcal{E}_{\mathrm{rl}}$ preserves images. We recall from [Mai05a] that the image object of a morphism $\mathcal{E}_{\mathrm{rl}}(f) : \mathcal{E}_{\mathrm{rl}}(P) \to \mathcal{E}_{\mathrm{rl}}(Q)$ in $\mathcal{C}_{\mathcal{T}_{\mathrm{rl}}}$ is defined as

$$Im(\mathcal{E}_{\mathrm{rl}}(f)) \equiv \Sigma_{w\in\mathcal{E}_{\mathrm{rl}}(Q)}\quad \exists_{z\in\mathcal{E}_{\mathrm{rl}}(P)}\ \mathcal{E}_{\mathrm{rl}}(f)(z) =_{\mathcal{E}_{\mathrm{rl}}(Q)} w$$

and the image morphism $\iota_{\mathcal{E}(f)} : Im(\mathcal{E}_{\mathrm{rl}}(f)) \longrightarrow \mathcal{E}_{\mathrm{rl}}(Q)$ is defined as

$$\iota_{\mathcal{E}(f)}(z) \equiv \pi_1(z) \in \mathcal{E}_{\mathrm{rl}}(Q)$$

for $z \in Im(\mathcal{E}_{\mathrm{rl}}(f))$.

From the image factorization of f in $\mathrm{Pred}(\mathcal{S}_{\mathrm{in}})$ it follows that $\mathcal{E}_{\mathrm{rl}}(f) = \mathcal{E}_{\mathrm{rl}}(\iota_f)\cdot\mathcal{E}_{\mathrm{rl}}(p)$ with $\mathcal{E}_{\mathrm{rl}}(\iota_f)$ monic since $\mathcal{E}_{\mathrm{rl}}$ preserves pullbacks (recall that a monomorphism is characterized by the fact that the projections of its pullback along itself are isomorphic to the identities). Hence by the universal property of images there exists a morphism

$$\mathrm{Un}_{\mathrm{im}} : \mathsf{Im}(\mathcal{E}_{\mathrm{rl}}(f)) \longrightarrow \mathcal{E}_{\mathrm{rl}}(\mathsf{Im} f)$$

that is indeed an isomorphism with inverse $\mathrm{In}_{\mathrm{im}}$ defined as follows: for $z \in \mathcal{E}_{\mathrm{rl}}(\mathsf{Im} f)$

$$\mathrm{In}_{\mathrm{im}}(z) \equiv \langle \langle f(z_1), \mathsf{eq}\rangle, [\ \langle z_1, \mathsf{eq}\rangle, \mathsf{eq}\rangle\ ]\rangle$$

where $z_1 \equiv \pi_1(z)$.

Finally, we show that $\mathcal{E}_{\mathrm{rl}}$ preserves list objects. We recall that the list object of $\mathcal{E}_{\mathrm{rl}}(P)$ in $\mathcal{C}_{\mathcal{T}_{\mathrm{rl}}}$ is $\mathrm{List}(\mathcal{E}_{\mathrm{rl}}(P))$ with list constructors

$$\begin{array}{ll} r_o^{\mathcal{E}_{\mathrm{rl}}(P)}(z) \equiv \epsilon \in \mathrm{List}(\mathcal{E}_{\mathrm{rl}}(P)) & \text{for } z \in \top \\ r_1^{\mathcal{E}_{\mathrm{rl}}(P)}(z) \equiv \mathsf{cons}(\pi_1(z), \pi_2(z)) & \text{for } z \in \mathrm{List}(\mathcal{E}_{\mathrm{rl}}(P)) \times \mathcal{E}_{\mathrm{rl}}(P) \end{array}$$

By using the list structure of N as defined in proposition 4.4, we can easily define an isomorphism

$$\mathsf{Bin} : \mathrm{List}(\mathcal{E}_{\mathrm{rl}}(P)) \longrightarrow \mathcal{E}_{\mathrm{rl}}(\mathrm{List}(P))$$

by induction on the list $z \in \mathrm{List}(\mathcal{E}_{\mathrm{rl}}(P))$ as follows:

$$\mathsf{Bin}(z) \equiv \begin{cases} \langle 0, \mathsf{eq}\rangle & \text{if } z \equiv \epsilon \\ \langle \mathsf{cons}_{\mathrm{List}(N)}(\pi_1(\mathsf{Bin}(s)), \pi_1(a)), \mathsf{eq}\rangle & \text{if } z \equiv \mathsf{cons}(s, a) \\ & \quad \text{for } s \in \mathrm{List}(\mathcal{E}_{\mathrm{rl}}(P)),\ a \in \mathcal{E}_{\mathrm{rl}}(P) \end{cases}$$

whose inverse $\mathrm{In}_{\mathrm{list}}$ is defined by iteration on natural numbers as

$$\mathrm{In}_{\mathrm{list}}(z) \equiv \widetilde{\mathrm{In}_{\mathrm{list}}}(z, \mathsf{lh}(z_1))$$

for $z \in \mathcal{E}_{\mathrm{rl}}(\mathrm{List}(P))$ where $z_1 \equiv \pi_1(z)$ and in turn $\widetilde{\mathrm{In}_{\mathrm{list}}}(z, \mathsf{lh}(z_1))$ is defined as follows

$$\widetilde{\mathrm{In}_{\mathrm{list}}}(z, 0) \equiv \epsilon$$

$$\widetilde{\mathrm{In}_{\mathrm{list}}}(z, n+1) \equiv \begin{cases} \mathsf{cons}(\widetilde{\mathrm{In}_{\mathrm{list}}}(z, n), \langle \mathsf{p_{n+1}}(z_1), \mathsf{eq}\rangle) & \text{if } n+1 \leq \mathsf{lh}(z_1) \\ \widetilde{\mathrm{In}_{\mathrm{list}}}(z, n) & \text{if } n+1 > \mathsf{lh}(z_1) \end{cases}$$

where $\mathsf{p_{n+1}}(z_1)$ is the $n+1$-th projection of the number z_1 thought of as a list in $\mathrm{List}(N)$.

■

Thanks to this lemma we are ready to prove that:

5.3. THEOREM. *The syntactic categories $\mathcal{C}_{\mathcal{T}_{ad}}$ and $\mathcal{C}_{\mathcal{T}_{rl}}$ are equivalent to* $\mathrm{Pred}(\mathcal{S}_{in})$.

PROOF. Since by proposition 4.7 we know that $\mathrm{Pred}(\mathcal{S}_{\mathrm{in}})$ is a regular locos, then by the initiality of $\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}$ and $\mathcal{C}_{\mathcal{T}_{\mathrm{rl}}}$, as stated in theorem 3.5, there exist an arithmetic lextensive functor $\mathcal{J}_{\mathrm{ad}} : \mathcal{C}_{\mathcal{T}_{\mathrm{ad}}} \longrightarrow \mathrm{Pred}(\mathcal{S}_{\mathrm{in}})$ and a regular locos functor $\mathcal{J}_{\mathrm{rl}} : \mathcal{C}_{\mathcal{T}_{\mathrm{rl}}} \longrightarrow \mathrm{Pred}(\mathcal{S}_{\mathrm{in}})$, defined through the corresponding interpretations of $\mathcal{T}_{\mathrm{ad}}$ and $\mathcal{T}_{\mathrm{rl}}$ in $\mathrm{Pred}(\mathcal{S}_{\mathrm{in}})$ as detailed in [Mai05a].

It turns out that both $\mathcal{J}_{\mathrm{ad}} \cdot \mathcal{E}_{\mathrm{ad}}$ and $\mathcal{J}_{\mathrm{rl}} \cdot \mathcal{E}_{\mathrm{rl}}$ are naturally isomorphic to the identity functor. This is because the interpretation of a predicate P turns out to be isomorphic to itself as a $\mathrm{Pred}(\mathcal{S}_{\mathrm{in}})$-morphism and $eq(P, 1) = P$ holds in $\mathcal{S}_{\mathrm{in}}$.

Moreover, being $\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}$ and $\mathcal{C}_{\mathcal{T}_{\mathrm{rl}}}$ respectively an initial up to iso arithmetic lextensive category and an initial up to iso regular locos, it follows that $\mathcal{E}_{\mathrm{ad}} \cdot \mathcal{J}_{\mathrm{ad}} : \mathcal{C}_{\mathcal{T}_{\mathrm{ad}}} \longrightarrow \mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}$ and $\mathcal{E}_{\mathrm{rl}} \cdot \mathcal{J}_{\mathrm{rl}} : \mathcal{C}_{\mathcal{T}_{\mathrm{rl}}} \longrightarrow \mathcal{C}_{\mathcal{T}_{\mathrm{rl}}}$ are naturally isomorphic to the corresponding identity functors and hence we conclude. ∎

5.4. COROLLARY. *The initial arithmetic lextensive category $\mathcal{C}_{\mathcal{T}_{ad}}$ is equivalent to the initial regular locos $\mathcal{C}_{\mathcal{T}_{rl}}$.*

From this corollary we conclude that stable images and parameterized list objects are definable in the initial arithmetic lextensive category by translating back into $\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}$ the image and list structures of $\mathcal{C}_{\mathcal{T}_{\mathrm{rl}}}$ via the equivalences in theorem 5.3. But, since these equivalences are built through interpretation functors, defined in turn by induction on type and term constructors, we can not expect to be able to construct images and lists internally to any arithmetic lextensive category. Indeed the construction of images and list objects rely on the fact that every object can be considered a predicate on natural numbers as in $\mathrm{Pred}(\mathcal{S}_{\mathrm{in}})$: this is an *external property* that does not seem necessarily valid internally in any arithmetic lextensive category. Logically it means that any type in the internal language of $\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}$, which is the pure calculus $\mathcal{T}_{\mathrm{ad}}$ without the addition of axioms, can be seen as an indexed sum of a predicate on natural numbers. But this property does not seem necessarily valid in *any theory of $\mathcal{T}_{ad}$*.

6. The initial arithmetic universe via type theory

Here, we outline how the construction of the initial arithmetic universe $\mathcal{A}_{\mathrm{in}}$ is equivalent to the initial arithmetic pretopos $\mathcal{C}_{\mathcal{T}_{\mathrm{pn}}}$ and also to the initial list-arithmetic pretopos $\mathcal{C}_{\mathcal{T}_{\mathrm{au}}}$.

To this purpose, analogously to the embeddings of lemma 5.2 we define the embeddings

$$\mathcal{E}_{pn} : \mathrm{Pred}(\mathcal{S}_{\mathrm{in}}) \longrightarrow \mathcal{C}_{\mathcal{T}_{\mathrm{pn}}} \qquad\qquad \mathcal{E}_{\mathrm{au}} : \mathrm{Pred}(\mathcal{S}_{\mathrm{in}}) \longrightarrow \mathcal{C}_{\mathcal{T}_{\mathrm{au}}}$$

They are essentially defined as $\mathcal{E}_{\mathrm{ad}}$. Indeed observe that the calculus $\mathcal{T}_{\mathrm{ad}}$ is a fragment of $\mathcal{T}_{\mathrm{pn}}$ and also of $\mathcal{T}_{\mathrm{au}}$, always after recalling to represent the natural numbers object as the list type of the terminal type. This means that the category $\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}$ is a subcategory of $\mathcal{C}_{\mathcal{T}_{\mathrm{pn}}}$ and also of $\mathcal{C}_{\mathcal{T}_{\mathrm{au}}}$. Therefore the embeddings

$$\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}} \overset{i_{\mathrm{ad}}^{\mathrm{pn}}}{\hookrightarrow} \mathcal{C}_{\mathcal{T}_{\mathrm{pn}}} \qquad\qquad \mathcal{C}_{\mathcal{T}_{\mathrm{ad}}} \overset{i_{\mathrm{ad}}^{\mathrm{au}}}{\hookrightarrow} \mathcal{C}_{\mathcal{T}_{\mathrm{au}}}$$

which are the identity on objects and morphisms, preserve the arithmetic lextensive structure. Hence, we put

$$\mathcal{E}_{\mathrm{pn}} \equiv i_{\mathrm{ad}}^{\mathrm{pn}} \cdot \mathcal{E}_{\mathrm{ad}} \qquad\qquad \mathcal{E}_{\mathrm{au}} \equiv i_{\mathrm{ad}}^{\mathrm{au}} \cdot \mathcal{E}_{\mathrm{ad}}$$

Now note that $\mathcal{E}_{\mathrm{pn}}$ and $\mathcal{E}_{\mathrm{au}}$ enjoy the same preservation properties of $\mathcal{E}_{\mathrm{ad}}$ and also of $\mathcal{E}_{\mathrm{rl}}$ when applicable:

6.1. LEMMA. *The embedding $\mathcal{E}_{pn}$ is regular and preserves finite disjoint coproducts and the parameterized natural numbers object.*

The embedding $\mathcal{E}_{au}$ is regular and preserves finite disjoint coproducts and parameterized list objects.

Therefore, by the above lemma, since $\mathcal{A}_{\mathrm{in}}$ is the exact completion of $\mathrm{Pred}(\mathcal{S}_{\mathrm{in}})$, by its universal property [CV98] there exist two exact functors

$$\widehat{\mathcal{E}_{\mathrm{pn}}} : \mathcal{A}_{\mathrm{in}} \longrightarrow \mathcal{C}_{\mathcal{T}_{\mathrm{pn}}} \qquad\qquad \widehat{\mathcal{E}_{\mathrm{au}}} : \mathcal{A}_{\mathrm{in}} \longrightarrow \mathcal{C}_{\mathcal{T}_{\mathrm{au}}}$$

such that

$$\widehat{\mathcal{E}_{\mathrm{pn}}} \cdot \mathcal{Y} \simeq \mathcal{E}_{\mathrm{pn}} \qquad\qquad \widehat{\mathcal{E}_{\mathrm{au}}} \cdot \mathcal{Y} \simeq \mathcal{E}_{\mathrm{au}}$$

$\widehat{\mathcal{E}_{\mathrm{pn}}}$ can be explicitly defined as follows:

- $\widehat{\mathcal{E}_{\mathrm{pn}}}((X, R)) \equiv \mathcal{E}_{\mathrm{pn}}(X)/\mathcal{E}_{\mathrm{pn}}(R)$,
- $\widehat{\mathcal{E}_{\mathrm{pn}}}(f) \equiv \mathcal{Q}(\mathcal{E}_{\mathrm{pn}}(f))$ for $f : (X, R) \to (Y, S)$, where $\mathcal{Q}(\mathcal{E}_{\mathrm{pn}}(f))$ is the unique map from the quotient $\mathcal{E}_{\mathrm{pn}}(X)/\mathcal{E}_{\mathrm{pn}}(R)$ to $\mathcal{E}_{\mathrm{pn}}(Y)/\mathcal{E}_{\mathrm{pn}}(S)$ such that $\mathcal{Q}(\mathcal{E}_{\mathrm{pn}}(f))([x]) \equiv [\mathcal{E}_{\mathrm{pn}}(f)(x)]$ for $x \in \mathcal{E}_{\mathrm{pn}}(X)$.

The functor $\widehat{\mathcal{E}_{\mathrm{au}}}$ is defined in the same way by using $\mathcal{E}_{\mathrm{au}}$ in place of $\mathcal{E}_{\mathrm{pn}}$. However, since $\mathcal{T}_{\mathrm{pn}}$ is a fragment of $\mathcal{T}_{\mathrm{au}}$, and hence the category $\mathcal{C}_{\mathcal{T}_{\mathrm{pn}}}$ is a subcategory of $\mathcal{C}_{\mathcal{T}_{\mathrm{au}}}$ with an embedding

$$\mathcal{C}_{\mathcal{T}_{\mathrm{pn}}} \overset{i_{\mathrm{pn}}^{\mathrm{au}}}{\hookrightarrow} \mathcal{C}_{\mathcal{T}_{\mathrm{au}}}$$

preserving the arithmetic pretopos structure, it turns out that $\widehat{\mathcal{E}_{\mathrm{au}}}$ satisfies

$$\widehat{\mathcal{E}_{\mathrm{au}}} = i_{\mathrm{pn}}^{\mathrm{au}} \cdot \widehat{\mathcal{E}_{\mathrm{pn}}}$$

From [CV98] we know that $\widehat{\mathcal{E}_{\mathrm{au}}}$ as well as $\widehat{\mathcal{E}_{\mathrm{pn}}}$ are exact functors. Here we check that they preserve also finite coproducts and, respectively, list objects and the natural numbers object:

6.2. LEMMA. *The functor $\widehat{\mathcal{E}_{pn}} : \mathcal{A}_{in} \longrightarrow \mathcal{C}_{\mathcal{T}_{pn}}$ preserves the arithmetic pretopos structure as well as the functor $\widehat{\mathcal{E}_{au}} : \mathcal{A}_{in} \longrightarrow \mathcal{C}_{\mathcal{T}_{au}}$ preserves the list-arithmetic pretopos structure.*

PROOF. For simplicity, we just show that $\widehat{\mathcal{E}_{\mathrm{au}}}$ preserves finite disjoint coproducts and list objects. The proof that $\widehat{\mathcal{E}_{\mathrm{pn}}}$ is an arithmetic pretopos functor can be seen a special case of the above, given that these functors are defined in the same way and that natural numbers are lists of the terminal type.

Moreover we use the abbreviations $X^{\mathcal{E}}$ and $R^{\mathcal{E}}$ for $\mathcal{E}_{\mathrm{au}}(X)$ and $\mathcal{E}_{\mathrm{au}}(R)$ respectively.

Clearly, $\widehat{\mathcal{E}_{\mathrm{au}}}$ preserves the initial object. Indeed, given that $\mathcal{E}_{\mathrm{au}}$ is regular and preserves the initial object, then $\widehat{\mathcal{E}_{\mathrm{au}}}((\bot_{\mathrm{Pred}}, x =_{\bot_{\mathrm{Pred}}} x)) \simeq \bot/(x =_{\bot} x)$ which is in turn isomorphic to $\bot$.

Now we prove that $\widehat{\mathcal{E}_{\mathrm{au}}}$ preserves the binary coproduct structure. The binary coproduct of $\widehat{\mathcal{E}_{\mathrm{au}}}((X,R))$ and $\widehat{\mathcal{E}_{\mathrm{au}}}((Y,S))$ is $X^{\mathcal{E}}/R^{\mathcal{E}} + Y^{\mathcal{E}}/S^{\mathcal{E}}$ with the coproduct structure described in lemma 5.2 for $\mathcal{C}_{\mathcal{T}_{\mathrm{ad}}}$. Since $\mathcal{E}_{\mathrm{au}}$ is regular and preserves coproducts, for simplicity we just suppose $\widehat{\mathcal{E}_{\mathrm{au}}}((X+Y, R+S)) \equiv X^{\mathcal{E}} + Y^{\mathcal{E}}/R^{\mathcal{E}} + S^{\mathcal{E}}$. Then, injections in $(X+Y, R+S)$ induce an isomorphism

$$\widehat{\mathcal{E}_{\mathrm{au}}}(i_{(X,R)}) \oplus \widehat{\mathcal{E}_{\mathrm{au}}}(i_{(Y,S)}) : \widehat{\mathcal{E}_{\mathrm{au}}}((X,R)) + \widehat{\mathcal{E}_{\mathrm{au}}}((Y,S)) \longrightarrow \widehat{\mathcal{E}_{\mathrm{au}}}((X+Y, R+S))$$

whose inverse $\mathcal{Q}(\mathrm{In}_{\mathrm{cp}})$ is defined to be the unique morphism induced on the quotient $(X^{\mathcal{E}}+Y^{\mathcal{E}})/(R^{\mathcal{E}}+S^{\mathcal{E}})$ by a map $\mathrm{In}_{\mathrm{cp}} : X^{\mathcal{E}}+Y^{\mathcal{E}} \longrightarrow X^{\mathcal{E}}/R^{\mathcal{E}}+Y^{\mathcal{E}}/S^{\mathcal{E}}$ defined by elimination on the sum as follows: for $w \in X^{\mathcal{E}} + Y^{\mathcal{E}}$

$$\mathrm{In}_{\mathrm{cp}}(w) \equiv \begin{cases} \mathsf{inl}([x]_{X^{\mathcal{E}}/R^{\mathcal{E}}}) & \text{if } w = \mathsf{inl}(x) \text{ for } x \in X^{\mathcal{E}} \\ \mathsf{inr}([y]_{Y^{\mathcal{E}}/R^{\mathcal{E}}}) & \text{if } w = \mathsf{inr}(y) \text{ for } y \in Y^{\mathcal{E}} \end{cases}$$

Now we prove that $\widehat{\mathcal{E}_{\mathrm{au}}}$ preserves the list object structure. The list object on $\widehat{\mathcal{E}_{\mathrm{au}}}(X,R) = X^{\mathcal{E}}/R^{\mathcal{E}}$ is $\mathrm{List}(X^{\mathcal{E}}/R^{\mathcal{E}})$ with the list structure described in lemma 5.2 for $\mathcal{C}_{\mathcal{T}_{\mathrm{rl}}}$. Since $\mathcal{E}_{\mathrm{au}}$ is regular and preserves list objects, for simplicity we suppose

$$\widehat{\mathcal{E}_{\mathrm{au}}}((\mathrm{List}(X), \mathrm{List}(R)) \equiv \mathrm{List}(X^{\mathcal{E}})/\,\mathrm{List}(R^{\mathcal{E}}).$$

Now, by induction on the list structure of $\mathrm{List}(X^{\mathcal{E}}/R^{\mathcal{E}})$ we can define an isomorphism

$$\mathrm{Un}_{\mathrm{list}} : \mathrm{List}(\widehat{\mathcal{E}_{\mathrm{au}}}(X,R)) \longrightarrow \widehat{\mathcal{E}_{\mathrm{au}}}((\mathrm{List}(X), \mathrm{List}(R)))$$

as follows: for $z \in \mathrm{List}(X^{\mathcal{E}}/R^{\mathcal{E}})$

$$\mathrm{Un}_{\mathrm{list}}(z) \equiv \begin{cases} [\epsilon]_{L'} & \text{if } z = \epsilon \\ \mathcal{Q}(\mathsf{cons})(\mathrm{Un}_{\mathrm{list}}(l), c) & \text{if } w = \mathsf{cons}(l,c) \text{ for } l \in \mathrm{List}(X^{\mathcal{E}}/R^{\mathcal{E}}), c \in X^{\mathcal{E}}/R^{\mathcal{E}} \end{cases}$$

where $L' \equiv \mathrm{List}(X^{\mathcal{E}})/\,\mathrm{List}(R^{\mathcal{E}})$ and in turn

$$\mathcal{Q}(\mathsf{cons}) : \mathrm{List}(X^{\mathcal{E}})/\,\mathrm{List}(R^{\mathcal{E}}) \times X^{\mathcal{E}}/R^{\mathcal{E}} \longrightarrow \mathrm{List}(X^{\mathcal{E}})/\,\mathrm{List}(R^{\mathcal{E}})$$

is the unique map induced on the quotients $\mathrm{List}(X^{\mathcal{E}})/\mathrm{List}(R^{\mathcal{E}})$ and $X^{\mathcal{E}}/R^{\mathcal{E}}$ such that

$$\mathcal{Q}(\mathsf{cons})([s]_{L'},[x]_{X^{\mathcal{E}}/R^{\mathcal{E}}}) \equiv [\mathsf{cons}(s,x)]_{L'}$$

for $s \in \mathrm{List}(X^{\mathcal{E}})$ and $x \in X^{\mathcal{E}}$.

The inverse of $\mathrm{Un}_{\mathrm{list}}$, called $\mathcal{Q}(\mathrm{In}_{\mathrm{list}})$, can be defined as the unique map induced on the quotient $\mathrm{List}(X^{\mathcal{E}})/\mathrm{List}(R^{\mathcal{E}})$ by a map $\mathrm{In}_{\mathrm{list}} : \mathrm{List}(X^{\mathcal{E}}) \longrightarrow \mathrm{List}(X^{\mathcal{E}}/R^{\mathcal{E}})$ preserving the relation $\mathrm{List}(R^{\mathcal{E}})$ and defined by induction on $\mathrm{List}(X)^{\mathcal{E}}$ as follows: for $w \in \mathrm{List}(X^{\mathcal{E}})$

$$\mathrm{In}_{\mathrm{list}}(w) \equiv \begin{cases} \epsilon & \text{if } w = \epsilon \\ \mathsf{cons}(\mathrm{In}_{\mathrm{list}}(s),[x]_{X^{\mathcal{E}}/R^{\mathcal{E}}}) & \text{if } w = \mathsf{cons}(s,x) \text{ for } s \in \mathrm{List}(X^{\mathcal{E}}), x \in X^{\mathcal{E}} \end{cases}$$

In order to show that $\mathrm{In}_{\mathrm{list}}$ preserves the relation $\mathrm{List}(R^{\mathcal{E}})$, namely that if $\mathrm{List}(R^{\mathcal{E}})(w,w')$ holds then $\mathrm{In}_{\mathrm{list}}(w) =_{\mathrm{List}(X^{\mathcal{E}}/R^{\mathcal{E}})} \mathrm{In}_{\mathrm{list}}(w')$ holds, too, we prove by induction on natural numbers that $\mathrm{In}_{\mathrm{list}}(\mathsf{part_n}(w)) =_{\mathrm{List}(X^{\mathcal{E}}/R^{\mathcal{E}})} \mathrm{In}_{\mathrm{list}}(\mathsf{part_n}(w'))$ holds. ∎

Now, we are ready to prove:

6.3. THEOREM. *Both the syntactic categories $\mathcal{C}_{\mathcal{T}_{pn}}$ and $\mathcal{C}_{\mathcal{T}_{au}}$ are equivalent to $\mathcal{A}_{in}$.*

PROOF. Since by proposition 4.10 we know that $\mathcal{A}_{\mathrm{in}}$ is a list-arithmetic pretopos, by the initiality of $\mathcal{C}_{\mathcal{T}_{\mathrm{pn}}}$ and of $\mathcal{C}_{\mathcal{T}_{\mathrm{au}}}$, as stated in theorem 3.5, there exist an arithmetic pretopos functor $\mathcal{J}_{\mathrm{pn}} : \mathcal{C}_{\mathcal{T}_{\mathrm{pn}}} \longrightarrow \mathcal{A}_{\mathrm{in}}$ and a list-arithmetic pretopos functor $\mathcal{J}_{\mathrm{au}} : \mathcal{C}_{\mathcal{T}_{\mathrm{au}}} \longrightarrow \mathcal{A}_{\mathrm{in}}$ defined on objects and morphisms through the interpretation of $\mathcal{T}_{\mathrm{pn}}$ and of $\mathcal{T}_{\mathrm{au}}$ in $\mathcal{A}_{\mathrm{in}}$, respectively, as described in [Mai05a].

Then, note that $\mathcal{J}_{\mathrm{pn}} \cdot \mathcal{E}_{\mathrm{pn}}$ and $\mathcal{J}_{\mathrm{au}} \cdot \mathcal{E}_{\mathrm{au}}$ are both naturally isomorphic to $\mathcal{Y}$. Indeed, since the interpretation of a predicate P in the initial Skolem category turns out to be isomorphic to $\mathcal{Y}(P)$ thinking of P as a morphism, $eq(P,1) =_{\mathcal{S}_{\mathrm{in}}} P$ holds and $\mathcal{Y}$ preserves finite limits (see [CV98]), then the interpretation of $\mathcal{E}_{\mathrm{pn}}(P)$ is still isomorphic to $\mathcal{Y}(P)$ and the same happens for morphisms.

Hence, we get $\mathcal{Y} \simeq \mathcal{J}_{\mathrm{pn}} \cdot \mathcal{E}_{\mathrm{pn}} \simeq \mathcal{J}_{\mathrm{pn}} \cdot (\widehat{\mathcal{E}_{\mathrm{pn}}} \cdot \mathcal{Y}) \simeq (\mathcal{J}_{\mathrm{pn}} \cdot \widehat{\mathcal{E}_{\mathrm{pn}}}) \cdot \mathcal{Y}$ and by the uniqueness property of the exact completion we conclude that $\mathcal{J}_{\mathrm{pn}} \cdot \widehat{\mathcal{E}_{\mathrm{pn}}}$ is naturally isomorphic to the identity functor. The same argument lets us conclude that $\mathcal{J}_{\mathrm{au}} \cdot \widehat{\mathcal{E}_{\mathrm{au}}}$ is also isomorphic to the identity functor.

Moreover, being $\mathcal{C}_{\mathcal{T}_{\mathrm{pn}}}$ the initial up to iso arithmetic pretopos and $\mathcal{C}_{\mathcal{T}_{\mathrm{au}}}$ the initial up to iso list-arithmetic pretopos, thanks to theorem 3.5 we also get that $\widehat{\mathcal{E}_{\mathrm{pn}}} \cdot \mathcal{J}_{\mathrm{pn}}$ is naturally isomorphic to the identity functor and so is $\widehat{\mathcal{E}_{\mathrm{au}}} \cdot \mathcal{J}_{\mathrm{au}}$. Therefore, we conclude that both $\mathcal{C}_{\mathcal{T}_{\mathrm{pn}}}$ and $\mathcal{C}_{\mathcal{T}_{\mathrm{au}}}$ are equivalent to $\mathcal{A}_{\mathrm{in}}$. ∎

6.4. COROLLARY. *The initial arithmetic pretopos $\mathcal{C}_{\mathcal{T}_{pn}}$ is equivalent to the initial list-arithmetic pretopos $\mathcal{C}_{\mathcal{T}_{au}}$.*

From this corollary we deduce that parameterized list objects are definable in the initial arithmetic pretopos via the equivalences in theorem 6.3. But it is worth noticing also here, as after corollary 5.4, that, since such equivalences make use of interpretation functors defined by induction on type and term constructors, we can not directly deduce that any arithmetic pretopos is list-arithmetic.

This is not an exceptional fact: consider for example that the initial finite product category and the initial finite limit category coincide with the trivial category with one object and one arrow.

We end by leaving as an open problem whether a generic arithmetic pretopos is also list-arithmetic.

7. Free internal categories and diagrams in a list-arithmetic pretopos

In this section we show how to define free internal categories and diagrams within a list-arithmetic pretopos by means of its internal language (first shown in [Mai99b]). Recall that any list-arithmetic pretopos $\mathcal{U}$ is equivalent to the *syntactic category* $\mathcal{C}_{T(\mathcal{U})}$ built out of its internal type theory $T(\mathcal{U})$. This type theory is obtained from $\mathcal{T}_{\text{au}}$ by adding specific axioms regarding the proper dependent types and terms of $\mathcal{U}$ and their corresponding equalities valid in $\mathcal{U}$, as described in [Mai05a].

The use of the internal language is helpful here to perform proofs by induction. In particular, in building free internal categories and diagrams a key difficulty is to define operations on proper subtypes of list types with no specific recursion principle available. We overcome this difficulty by defining such operations by recursion on natural numbers through iteration of suitable operations. And here the use of a logical/type theoretic language in which to perform inductions as usual is more helpful than the categorical reasoning via universal properties.

This technique has already been applied to prove that the slice categories of a locos enjoy list objects in section 3.8. Moreover, it is also the same technique that one can adopt to prove that in a Skolem category the natural numbers object N is a list object over itself as stated in proposition 4.4. Actually it is enough to work in a regular locos to build free internal categories and diagrams as shown here.

7.1. Free internal categories. Given a graph $\mathcal{G} \equiv (\mathcal{G}_0, \mathcal{G}_1, \mathsf{dm}_{\mathcal{G}}, \mathsf{cd}_{\mathcal{G}})$ internal to a list-arithmetic pretopos $\mathcal{U}$

$$\begin{tikzcd} \mathcal{G}_1 \arrow[r, shift left, "\mathsf{dm}_{\mathcal{G}}"] \arrow[r, shift right, "\mathsf{cd}_{\mathcal{G}}"'] & \mathcal{G}_0 \end{tikzcd}$$

we build the free internal category generated from it. For a general account on these concepts about internal category theory we refer to [Joh77] or [MM92] or [Joh02a].

7.2. Definition. Given a graph $\mathcal{G} \equiv (\mathcal{G}_0, \mathcal{G}_1, \mathsf{dm}_{\mathcal{G}}, \mathsf{cd}_{\mathcal{G}})$ internal to $\mathcal{U}$, we define the category $\mathcal{C}^{\mathcal{G}}$ as a candidate to be the free internal category generated from $\mathcal{G}$ as follows.

The *objects* of $\mathcal{C}^{\mathcal{G}}$ are defined as the graph objects ${\mathcal{C}^{\mathcal{G}}}_0 \equiv \mathcal{G}_0$.

The *morphisms* of $\mathcal{C}^{\mathcal{G}}$ are defined as the coproduct of the graph objects $\mathcal{G}_0$ representing the identity maps with lists of composable graph arrows represented by $\widehat{{\mathcal{C}^{\mathcal{G}}}_1}$:

$${\mathcal{C}^{\mathcal{G}}}_1 \equiv \mathcal{G}_0 \oplus \widehat{{\mathcal{C}^{\mathcal{G}}}_1}$$

where $\widehat{\mathcal{C}^{\mathcal{G}}_1}$ is formally defined as the equalizer of the set of non-empty lists of composable arrows, namely lists of arrows $\lfloor f_1,\ldots,f_n \rfloor$ such that $\lfloor \mathsf{dm}_{\mathcal{G}}(f_2),\ldots,\mathsf{dm}_{\mathcal{G}}(f_n)\rfloor$, that is $\mathsf{frt}\cdot\mathsf{Lst}(\mathsf{dm}_G)([f_1,\ldots,f_n])$, is equal to the list $\lfloor \mathsf{cd}_{\mathcal{G}}(f_1),\ldots,\mathsf{cd}_{\mathcal{G}}(f_{n-1})\rfloor$, that is $\mathsf{bck}\cdot\mathsf{Lst}(\mathsf{cd}_G)(\lfloor f_1,\ldots,f_n\rfloor)$ (see the definition of such operations in the appendix):

$$\widehat{\mathcal{C}^{\mathcal{G}}_1} \hookrightarrow \mathrm{List}^*(\mathcal{G}_1) \underset{\mathsf{bck}\cdot\mathsf{Lst}(\mathsf{cd}_{\mathcal{G}})}{\overset{\mathsf{frt}\cdot\mathsf{Lst}(\mathsf{dm}_{\mathcal{G}})}{\rightrightarrows}} \mathrm{List}(\mathcal{G}_0)$$

where

$$\widehat{\mathcal{C}^{\mathcal{G}}_1} \equiv \quad \Sigma_{w\in \mathrm{List}^*(\mathcal{G}_1)}\, \mathsf{frt}\cdot\mathsf{Lst}(\mathsf{dm}_{\mathcal{G}})(w) =_{\mathrm{List}(\mathcal{G}_0)} \mathsf{bck}\cdot\mathsf{Lst}(\mathsf{cd}_{\mathcal{G}})(w)$$

The *domain* morphism $\mathsf{dm}_{\mathcal{C}^{\mathcal{G}}} : \mathcal{C}^{\mathcal{G}}{}_1 \longrightarrow \mathcal{C}^{\mathcal{G}}{}_0$ is defined as the coproduct morphism of the identity with the domain of the first morphism of the list

$$\mathsf{dm}_{\mathcal{C}^{\mathcal{G}}} \equiv \mathrm{id} \oplus (\mathsf{dm}_{\mathcal{G}}\cdot(\mathsf{fst}\cdot\pi_1))$$

while the *codomain* morphism $\mathsf{cd}_{\mathcal{C}^{\mathcal{G}}} : \mathcal{C}^{\mathcal{G}}{}_1 \longrightarrow \mathcal{C}^{\mathcal{G}}{}_0$ is defined as the coproduct morphism of the identity with the codomain of the last morphism of the list

$$\mathsf{cd}_{\mathcal{C}^{\mathcal{G}}} \equiv \mathrm{id} \oplus (\mathsf{cd}_{\mathcal{G}}\cdot(\mathsf{las}\cdot\pi_1))$$

The *unit* $e_{\mathcal{C}^{\mathcal{G}}} : \mathcal{C}^{\mathcal{G}}{}_0 \to \mathcal{C}^{\mathcal{G}}{}_1$ is defined as the first injection

$$\mathsf{e}_{\mathcal{C}^{\mathcal{G}}}(x) \equiv \mathsf{inl}(x) \qquad \text{for } x \in \mathcal{C}^{\mathcal{G}}{}_0$$

It follows immediately that

$$\mathsf{dm}_{\mathcal{C}^{\mathcal{G}}}\cdot \mathsf{e}_{\mathcal{C}^{\mathcal{G}}} = \mathrm{id} \qquad \mathsf{cd}_{\mathcal{C}^{\mathcal{G}}}\cdot \mathsf{e}_{\mathcal{C}^{\mathcal{G}}} = \mathrm{id}$$

The *composition* of morphisms

$$\mathsf{Cmp}_{\mathcal{C}^{\mathcal{G}}}(w) \in \mathcal{C}^{\mathcal{G}}{}_1 \ [w \in \mathcal{C}^{\mathcal{G}}{}_1 \times_{\mathcal{C}^{\mathcal{G}}{}_0} \mathcal{C}^{\mathcal{G}}{}_1]$$

where $\mathcal{C}^{\mathcal{G}}{}_1 \times_{\mathcal{C}^{\mathcal{G}}{}_0} \mathcal{C}^{\mathcal{G}}{}_1$ is the vertex of the pullback of $\mathsf{dm}_{\mathcal{C}^{\mathcal{G}}}$ along $\mathsf{cd}_{\mathcal{C}^{\mathcal{G}}}$

$$\mathcal{C}^{\mathcal{G}}{}_1 \times_{\mathcal{C}^{\mathcal{G}}{}_0} \mathcal{C}^{\mathcal{G}}{}_1 \equiv \Sigma_{f\in \mathcal{C}^{\mathcal{G}}{}_1}\, \Sigma_{g\in \mathcal{C}^{\mathcal{G}}{}_1}\, \mathsf{cd}_{\mathcal{C}^{\mathcal{G}}}(f) =_{\mathcal{C}^{\mathcal{G}}{}_0} \mathsf{dm}_{\mathcal{C}^{\mathcal{G}}}(g)$$

is defined by cases after noting that by distributivity of coproducts with respect to pullbacks we have

$$\begin{aligned}&(\mathcal{G}_0 \oplus \widehat{\mathcal{C}^{\mathcal{G}}_1}) \times_{\mathcal{C}^{\mathcal{G}}{}_0} (\mathcal{G}_0 \oplus \widehat{\mathcal{C}^{\mathcal{G}}_1})\\ &\qquad \cong (\mathcal{G}_0 \times_{\mathcal{C}^{\mathcal{G}}{}_0} \mathcal{G}_0) \oplus (\mathcal{G}_0 \times_{\mathcal{C}^{\mathcal{G}}{}_0} \widehat{\mathcal{C}^{\mathcal{G}}_1}) \oplus (\widehat{\mathcal{C}^{\mathcal{G}}_1} \times_{\mathcal{C}^{\mathcal{G}}{}_0} \mathcal{G}_0) \oplus (\widehat{\mathcal{C}^{\mathcal{G}}_1} \times_{\mathcal{C}^{\mathcal{G}}{}_0} \widehat{\mathcal{C}^{\mathcal{G}}_1})\end{aligned}$$

In the next we simply write $\langle z_1,_{\mathcal{C}^{\mathcal{G}}{}_0} z_2\rangle$ for $\langle z_1, \langle z_2, \mathsf{eq}\rangle\rangle$ given $z_1, z_2 \in \mathcal{C}^{\mathcal{G}}{}_1$ such that $\mathsf{cd}_{\mathcal{C}^{\mathcal{G}}}(z_1) =_{\mathcal{C}^{\mathcal{G}}{}_0} \mathsf{dm}_{\mathcal{C}^{\mathcal{G}}}(z_2)$ holds.

In detail the composition is defined by cases as follows

$$\mathsf{Cmp}_{\mathcal{C}^{\mathcal{G}}}(\langle z_1,_{\mathcal{C}^{\mathcal{G}}_0}\ z_2\rangle) \equiv \begin{cases} \mathsf{inl}(x) & \text{if } z_1 = \mathsf{inl}(x) = z_2 \\ & \quad \text{for } x \in \mathcal{G}_0 \\ \mathsf{inr}(s) & \text{if } z_1 = \mathsf{inl}(x) \text{ and } z_2 = \mathsf{inr}(s) \\ & \quad \text{for } x \in \mathcal{G}_0 \text{ and } s \in \widehat{\mathcal{C}^{\mathcal{G}}_1} \\ \mathsf{inr}(s) & \text{if } z_2 = \mathsf{inl}(y) \text{ and } z_1 = \mathsf{inr}(s) \\ & \quad \text{for } y \in \mathcal{G}_0 \text{ and } s \in \widehat{\mathcal{C}^{\mathcal{G}}_1} \\ \mathsf{inr}(\langle \lfloor \pi_1(s_1), \pi_1(s_2) \rfloor, \mathsf{eq}\rangle) & \text{if } z_1 = \mathsf{inr}(s_1) \text{ and } z_2 = \mathsf{inr}(s_2) \\ & \quad \text{for } s_1 \in \widehat{\mathcal{C}^{\mathcal{G}}_1} \text{ and } s_2 \in \widehat{\mathcal{C}^{\mathcal{G}}_1} \end{cases}$$

It is easy to check that this is well defined and that the composed morphism has the right domain and codomain:

$$\mathsf{dm}_{\mathcal{C}^{\mathcal{G}}} \cdot \pi_1' = \mathsf{dm}_{\mathcal{C}^{\mathcal{G}}} \cdot \mathsf{Cmp}_{\mathcal{C}^{\mathcal{G}}} \qquad \mathsf{cd}_{\mathcal{C}^{\mathcal{G}}} \cdot \pi_2' = \mathsf{cd}_{\mathcal{C}^{\mathcal{G}}} \cdot \mathsf{Cmp}_{\mathcal{C}^{\mathcal{G}}}$$

where π_1' and π_2' are respectively the first and second projection of the pullback of $\mathsf{dm}_{\mathcal{C}^{\mathcal{G}}}$ along $\mathsf{cd}_{\mathcal{C}^{\mathcal{G}}}$.

Moreover, given that the operation of appending a list to another is associative, it follows that the composition is associative, too

$$\mathsf{Cmp}_{\mathcal{C}^{\mathcal{G}}} \cdot (\mathsf{Cmp}_{\mathcal{C}^{\mathcal{G}}} \times \mathrm{id}) \cdot \sigma = \mathsf{Cmp}_{\mathcal{C}^{\mathcal{G}}} \cdot (\mathrm{id} \times \mathsf{Cmp}_{\mathcal{C}^{\mathcal{G}}})$$

where σ is isomorphism from $\mathcal{C}^{\mathcal{G}}_1 \times_{\mathcal{C}^{\mathcal{G}}_0} (\mathcal{C}^{\mathcal{G}}_1 \times_{\mathcal{C}^{\mathcal{G}}_0} \mathcal{C}^{\mathcal{G}}_1)$ to $(\mathcal{C}^{\mathcal{G}}_1 \times_{\mathcal{C}^{\mathcal{G}}_0} \mathcal{C}^{\mathcal{G}}_1) \times_{\mathcal{C}^{\mathcal{G}}_0} \mathcal{C}^{\mathcal{G}}_1$.

Finally it is easy to show that the defined composition admits the morphism $\mathsf{e}_{\mathcal{C}^{\mathcal{G}}}$ as identity:

$$\mathsf{Cmp}_{\mathcal{C}^{\mathcal{G}}} \cdot (\mathsf{e}_{\mathcal{C}^{\mathcal{G}}} \times \mathrm{id}) = \pi_2' \qquad \mathsf{Cmp}_{\mathcal{C}^{\mathcal{G}}} \cdot (\mathrm{id} \times \mathsf{e}_{\mathcal{C}^{\mathcal{G}}}) = \pi_1'$$

Then, we are ready to prove:

7.3. PROPOSITION. *Internally to any list-arithmetic pretopos $\mathcal{U}$, given a graph $\mathcal{G} \equiv (\mathcal{G}_0, \mathcal{G}_1, \mathsf{dm}_{\mathcal{G}}, \mathsf{cd}_{\mathcal{G}})$*

$$\begin{tikzcd} \mathcal{G}_1 \arrow[r, shift left, "\mathsf{dm}_{\mathcal{G}}"] \arrow[r, shift right, "\mathsf{cd}_{\mathcal{G}}"'] & \mathcal{G}_0 \end{tikzcd}$$

then the category $\mathcal{C}^{\mathcal{G}} \equiv (\mathcal{C}^{\mathcal{G}}_0, \mathcal{C}^{\mathcal{G}}_1, \mathsf{dm}_{\mathcal{C}^{\mathcal{G}}}, \mathsf{cd}_{\mathcal{C}^{\mathcal{G}}}, \mathsf{e}_{\mathcal{C}^{\mathcal{G}}}, \mathsf{Cmp}_{\mathcal{C}^{\mathcal{G}}})$ defined above in definition 7.2

$$\begin{tikzcd} \mathcal{C}^{\mathcal{G}}_1 \arrow[r, shift left=2, "\mathsf{dm}_{\mathcal{C}^{\mathcal{G}}}"] \arrow[r, shift right=2, "\mathsf{cd}_{\mathcal{C}^{\mathcal{G}}}"'] & \mathcal{C}^{\mathcal{G}}_0 \arrow[l, "\mathsf{e}_{\mathcal{C}^{\mathcal{G}}}" description] \end{tikzcd} \qquad \mathsf{Cmp}_{\mathcal{C}^{\mathcal{G}}} : \mathcal{C}^{\mathcal{G}}_1 \times_{\mathcal{C}^{\mathcal{G}}_0} \mathcal{C}^{\mathcal{G}}_1 \longrightarrow \mathcal{C}^{\mathcal{G}}_1$$

is the free internal category generated by it.

PROOF. Let ${\mathcal{C}^{\mathcal{G}}}_0$ and ${\mathcal{C}^{\mathcal{G}}}_1$ be defined as above. Then we define the injection from the graph into its claimed free internal category as follows: $j_0 : \mathcal{G}_0 \longrightarrow {\mathcal{C}^{\mathcal{G}}}_0$ is the identity map, that is $j_0 \equiv \mathrm{id}_{\mathcal{G}_0}$ in $\mathcal{U}$, and $j_1 : \mathcal{G}_1 \to {\mathcal{C}^{\mathcal{G}}}_1$ is defined as $j_1(f) \equiv \mathsf{inr}(\langle \lfloor f \rfloor, \mathsf{eq} \rangle)$ for $f \in \mathcal{G}_1$. Then, (j_0, j_1) is a graph morphism from $\mathcal{G}$ to $\mathcal{C}^{\mathcal{G}}$ since it satisfies

$$\mathsf{dm}_{\mathcal{C}^{\mathcal{G}}} \cdot j_1 = j_0 \cdot \mathsf{dm}_{\mathcal{G}} \qquad \mathsf{cd}_{\mathcal{C}^{\mathcal{G}}} \cdot j_1 = j_0 \cdot \mathsf{cd}_{\mathcal{G}}$$

Now, we prove that (j_0, j_1) has the *universal property* of lifting a graph morphism (ψ_0, ψ_1) from $\mathcal{G}$ to an internal category $\mathcal{D}$ to a unique internal functor $(\psi_0^{\mathsf{lf}}, \psi_1^{\mathsf{lf}})$ from $\mathcal{C}^{\mathcal{G}}$ to $\mathcal{D}$ such that $\psi_0^{\mathsf{lf}} \cdot j_0 = \psi_0$ and $\psi_1^{\mathsf{lf}} \cdot j_1 = \psi_1$

$$\begin{tikzcd}
\mathcal{G} \arrow[r, "(j_0, j_1)"] \arrow[dr, "(\psi_0, \psi_1)"'] & \mathcal{C}^{\mathcal{G}} \arrow[d, "({\psi_0}^{\mathsf{lf}}, {\psi_1}^{\mathsf{lf}})"] \\
& \mathcal{D}
\end{tikzcd}$$

Given an internal category $\mathcal{D} \equiv (\mathcal{D}_0, \mathcal{D}_1, \mathsf{dm}_{\mathcal{D}}, \mathsf{cd}_{\mathcal{D}}, \mathsf{e}_{\mathcal{D}}, \mathsf{Cmp}_{\mathcal{D}})$

$$\begin{tikzcd}
\mathcal{D}_1 \arrow[r, shift left=2, "\mathsf{dm}_{\mathcal{D}}"] \arrow[r, shift right=2, "\mathsf{cd}_{\mathcal{D}}"'] & \mathcal{D}_0 \arrow[l, "\mathsf{e}_{\mathcal{D}}" description]
\end{tikzcd} \qquad \mathsf{Cmp}_{\mathcal{D}} : \mathcal{D}_1 \times_{\mathcal{D}_0} \mathcal{D}_1 \longrightarrow \mathcal{D}_1$$

and a graph morphism (ψ_0, ψ_1) where $\psi_0 : \mathcal{G}_0 \to \mathcal{D}_0$ and $\psi_1 : \mathcal{G}_1 \to \mathcal{D}_1$ are maps in $\mathcal{U}$ satisfying

$$\mathsf{dm}_{\mathcal{D}} \cdot \psi_1 = \psi_0 \cdot \mathsf{dm}_{\mathcal{G}} \qquad \mathsf{cd}_{\mathcal{D}} \cdot \psi_1 = \psi_0 \cdot \mathsf{cd}_{\mathcal{G}}$$

we define an internal functor

$$\psi_0^{\mathsf{lf}} : {\mathcal{C}^{\mathcal{G}}}_0 \longrightarrow \mathcal{D}_0 \qquad\qquad \psi_1^{\mathsf{lf}} : {\mathcal{C}^{\mathcal{G}}}_1 \longrightarrow \mathcal{D}_1$$

such that $\psi_0^{\mathsf{lf}} \cdot j_0 = \psi_0$ and $\psi_1^{\mathsf{lf}} \cdot j_1 = \psi_1$ as follows. Being $\mathcal{C}_0 = \mathcal{G}_0$, we obviously define

$$\psi_0^{\mathsf{lf}} \equiv \psi_0$$

Then, in order to define $\psi_1^{\mathsf{lf}} : {\mathcal{C}^{\mathcal{G}}}_1 \to \mathcal{D}_1$, recall that ${\mathcal{C}^{\mathcal{G}}}_1 = \mathcal{G}_0 \oplus \widehat{{\mathcal{C}^{\mathcal{G}}}_1}$. Hence we define ψ_1^{lf} as a coproduct morphism of the action on identity morphisms and on lists of composable graph morphisms. To identity morphisms we associate the corresponding identity ones by using ψ_0. Instead to a list of composable $\mathcal{G}$-graph morphisms we associate the composition of their values as $\mathcal{D}$-morphisms via ψ_1. Hence, we define

$$\psi_1^{\mathsf{lf}} \equiv (\mathsf{e}_{\mathcal{D}} \cdot \psi_0) \oplus (\mathsf{app} \cdot \mathcal{C}(\psi_1))$$

where

$$\mathcal{C}(\psi_1) : \widehat{{\mathcal{C}^{\mathcal{G}}}_1} \longrightarrow \widehat{{\mathcal{C}^{\mathcal{D}}}_1}$$

is just the lifting of ψ_1 between the corresponding list parts defined as follows: for $z \in \widehat{\mathcal{C}^{\mathcal{G}}_1}$

$$\mathcal{C}(\psi_1)(z) \equiv \langle \mathsf{Lst}(\psi_1)(z_1), \mathsf{eq}\rangle \in \widehat{\mathcal{C}^{\mathcal{D}}_1}$$

where $z_1 \equiv \pi_1(z)$ and

$$\mathsf{app} : \widehat{\mathcal{C}^{\mathcal{D}}_1} \longrightarrow \mathcal{D}_1$$

is a map sending a list of composable morphisms of $\mathcal{D}_1$ into their composition. At this point note that *we can not define* app *by using an inductive elimination rule on* $\widehat{\mathcal{C}^{\mathcal{D}}_1}$ *as for* $\mathrm{List}^*(\mathcal{G}_1)$ *in proposition A.4 of the appendix.* But we define app by iteration: we first fix a list s obtained by first projection from $\widehat{\mathcal{C}^{\mathcal{D}}_1}$, then we compose the first element of the list with the second and so on, namely to define the value of $\mathsf{app}(s)$ we iterate the described operation for a number of times equal to the length of s minus one (we start to apply from zero!). Formally, we define app as follows: for $s \in \widehat{\mathcal{C}^{\mathcal{D}}_1}$

$$\mathsf{app}(s) \equiv \widetilde{\mathsf{app}}(s, \mathsf{lh}(s_1) - 1)$$

by using the operation

$$\widetilde{\mathsf{app}}(s, n) \in \mathcal{D}_1 \ [s \in \widehat{\mathcal{C}^{\mathcal{D}}_1}, n \in N]$$

defined in turn by induction on natural numbers as follows:

$$\widetilde{\mathsf{app}}(s, 0) \equiv \mathsf{p}_1(s_1)$$

$$\widetilde{\mathsf{app}}(s, n+1) \equiv \begin{cases} \mathsf{Cmp}_{\mathcal{D}}(\langle \widetilde{\mathsf{app}}(s, n),_{\mathcal{D}_0} \mathsf{p}_{\mathsf{n+2}}(s_1)\rangle) & \text{if } n+2 \leq \mathsf{lh}(s_1) \\ \widetilde{\mathsf{app}}(s, n) & \text{if } n+2 > \mathsf{lh}(s_1) \end{cases}$$

where $s_1 \equiv \pi_1(s)$.

Actually, in order to guarantee that $\widetilde{\mathsf{app}}$ is well-defined we need to know that $\widetilde{\mathsf{app}}(s, n)$ has the same codomain as the domain of $\mathsf{p}_{\mathsf{n+2}}(s_1)$ and we have to add this information when defining it by induction. Therefore, formally $\widetilde{\mathsf{app}}$ is obtained by applying the first projection on the corresponding term of type

$$\Sigma_{x\in\mathcal{D}_1}\ ((\mathsf{cd}_{\mathcal{D}}(x) =_{\mathcal{D}_0} \mathsf{dm}_{\mathcal{D}}(\mathsf{p}_{\mathsf{n+2}}(s_1)) \ \wedge \ n+2 \leq \mathsf{lh}(s_1)) \ \vee \ n+2 > \mathsf{lh}(s_1)) \ [s \in \widehat{\mathcal{C}^{\mathcal{D}}_1}, n \in N]$$

defined by induction on natural numbers essentially as $\widetilde{\mathsf{app}}$ together with a proof of the needed extra information.

Clearly, it follows that $\psi_1^{\mathsf{lf}} \cdot j_1 = \psi_1$. Moreover, we can prove that ψ_1^{lf} is functorial, i.e. it satisfies the following equations:

$$\mathsf{dm}_{\mathcal{D}} \cdot \psi_1^{\mathsf{lf}} = \psi_0^{\mathsf{lf}} \cdot \mathsf{dm}_{\mathcal{C}^{\mathcal{G}}} \qquad \mathsf{cd}_{\mathcal{D}} \cdot \psi_1^{\mathsf{lf}} = \psi_0^{\mathsf{lf}} \cdot \mathsf{cd}_{\mathcal{C}^{\mathcal{G}}}$$
$$\psi_1^{\mathsf{lf}} \cdot \mathsf{e}_{\mathcal{C}^{\mathcal{G}}} = \mathsf{e}_{\mathcal{D}} \cdot \psi_0^{\mathsf{lf}}$$
$$\mathsf{Cmp}_{\mathcal{D}} \cdot (\psi_1^{\mathsf{lf}} \times \psi_1^{\mathsf{lf}}) = \psi_1^{\mathsf{lf}} \cdot \mathsf{Cmp}_{\mathcal{C}^{\mathcal{G}}}$$

In order to prove the first two equations, we show that the following equations hold, for $s \in \widehat{\mathcal{C}^{\mathcal{D}}}_1$,

$$\mathsf{dm}_{\mathcal{D}}(\widetilde{\mathsf{app}}(s,n)) =_{\mathcal{D}_0} \mathsf{dm}_{\mathcal{D}}(\mathsf{p_1}(s_1)) \qquad \mathsf{cd}_{D}(\widetilde{\mathsf{app}}(s,n)) =_{\mathcal{D}_0} \mathsf{cd}_{\mathcal{D}}(\mathsf{p_{n+1}}(s_1))$$

by induction on $n \in N$.

The third equation follows easily. Instead in order to prove the last equation, for $z \in \widehat{\mathcal{C}^{\mathcal{D}}}_1$ and $w \in \widehat{\mathcal{C}^{\mathcal{D}}}_1$ such that $\mathsf{cd}_{\mathcal{C}^{\mathcal{D}}}(\mathsf{inr}(z)) =_{\mathcal{D}_0} \mathsf{dm}_{\mathcal{C}^{\mathcal{D}}}(\mathsf{inr}(w))$ we derive the validity of the following: for $n \leq \mathsf{lh}(z_1) - 1$

$$\widetilde{\mathsf{app}}(\langle \lfloor z_1, w_1 \rfloor, \mathsf{eq} \rangle, n) =_{\mathcal{D}_1} \widetilde{\mathsf{app}}(z_1, n)$$

and for $n \leq \mathsf{lh}(w_1) - 1$

$$\mathsf{Cmp}_{\mathcal{D}}(\widetilde{\mathsf{app}}(z, l_1),_{\mathcal{D}_0} \widetilde{\mathsf{app}}(w, n)) =_{\mathcal{D}_1} \widetilde{\mathsf{app}}(\langle \lfloor z_1, w_1 \rfloor, \mathsf{eq} \rangle, l_1 + n + 1)$$

where $l_1 = \mathsf{lh}(z_1) - 1$ and $z_1 \equiv \pi_1(z)$ and $w_1 \equiv \pi_1(w)$. The actual proof is derived by induction on $n \in N$ towards the type obtained by enriching the above equations with the constrain on n in disjunctive form, as done to derive $\widetilde{\mathsf{app}}$.

Now we end by proving the *uniqueness* of $(\psi_0^{\mathsf{lf}}, \psi_1^{\mathsf{lf}})$. Suppose that there exists an internal functor given by $\rho_0 : \mathcal{C}^{\mathcal{G}}{}_0 \to \mathcal{D}_0$ and $\rho_1 : \mathcal{C}^{\mathcal{G}}{}_1 \to \mathcal{D}_1$ such that $\rho_0 \cdot j_0 = \psi_0$ and $\rho_1 \cdot j_1 = \psi_1$. Then, since $j_0 = \mathrm{id}$ we get that $\rho_0 = \psi_0^{\mathsf{lf}}$ follows trivially. It remains to prove that also $\rho_1 = \psi_1^{\mathsf{lf}}$. To this purpose we prove by induction on $n \in N$ the following: for $z \in \widehat{\mathcal{C}^{\mathcal{G}}}_1$ and $n \geq 1$

$$\rho_1(\mathsf{inr}(\widehat{\mathsf{part_n}}(z))) =_{\mathcal{D}_1} \psi_1^{\mathsf{lf}}(\mathsf{inr}(\widehat{\mathsf{part_n}}(z)))$$

where $\widehat{\mathsf{part_n}}(z) \equiv \langle \mathsf{part_n}(z_1), \mathsf{eq} \rangle$ and $z_1 \equiv \pi_1(z)$.

Then, by elimination on the sum type $\mathcal{C}^{\mathcal{G}}{}_1$ and knowing that $\mathsf{part}_{\mathsf{lh(s)}}(s) =_{\mathrm{List}^*(\mathcal{G}_1)} s$ holds for $s \in \mathrm{List}^*(\mathcal{G}_1)$, we get that $\psi_1^{\mathsf{lf}}(z) =_{\mathcal{D}_1} \rho_1(z)$ holds for $z \in \mathcal{C}^{\mathcal{G}}{}_1$.

Therefore, we conclude that $(\psi_0^{\mathsf{lf}}, \psi_1^{\mathsf{lf}})$ is the unique internal functor obtained by lifting (ψ_0, ψ_1). This ends the proof that $\mathcal{C}^{\mathcal{G}}$ is the free category internally to a list-arithmetic pretopos. ∎

7.4. Free internal diagrams. Applying the same technique adopted to build free internal categories, we can also prove that in any list-arithmetic pretopos free internal categorical diagrams exist. Also here, the key point is to show the universal property of the free internal diagram by defining an operation on a proper subtype of a list type by iteration on natural numbers.

7.5. Proposition. *In any list-arithmetic pretopos $\mathcal{U}$, given the internal graph $\mathcal{G} \equiv (\mathcal{G}_0, \mathcal{G}_1, \mathsf{dm}_{\mathcal{G}}, \mathsf{cd}_{\mathcal{G}})$*

$$\begin{tikzcd} \mathcal{G}_1 \arrow[r, shift left, "\mathsf{dm}_{\mathcal{G}}"] \arrow[r, shift right, "\mathsf{cd}_{\mathcal{G}}"'] & \mathcal{G}_0 \end{tikzcd}$$

and the internal diagram $\pi_0 : F \longrightarrow \mathcal{G}_0$ with the action

$$\mu : F \times_{\mathcal{G}_0} \mathcal{G}_1 \longrightarrow F$$

such that the following diagram commutes

$$\begin{tikzcd} F \times_{\mathcal{G}_0} \mathcal{G}_1 \arrow[r, "\pi_2"] \arrow[d, "\mu"] & \mathcal{G}_1 \arrow[d, "\mathsf{cd}_{\mathcal{G}}"'] \\ F \arrow[r, "\pi_0"'] & \mathcal{G}_0 \end{tikzcd}$$

where $F \times_{\mathcal{G}_0} \mathcal{G}_1$ is the vertex of the pullback of $\mathsf{dm}_{\mathcal{G}}$ along π_0

$$F \times_{\mathcal{G}_0} \mathcal{G}_1 \equiv \Sigma_{f\in F}\Sigma_{z\in\mathcal{G}_1}\pi_0(f) =_{\mathcal{G}_0} \mathsf{dm}_{\mathcal{G}}(z)$$

and π_2 its second projection,

we can lift μ to an unique action on ${\mathcal{C}^{\mathcal{G}}}_1$

$$\mu^{\mathsf{lf}} : F \times_{{\mathcal{C}^{\mathcal{G}}}_0} {\mathcal{C}^{\mathcal{G}}}_1 \longrightarrow F$$

where $F \times_{{\mathcal{C}^{\mathcal{G}}}_0} {\mathcal{C}^{\mathcal{G}}}_1$ is the vertex of the pullback of $\mathsf{dm}_{\mathcal{C}^{\mathcal{G}}}$ along π_0

$$F \times_{{\mathcal{C}^{\mathcal{G}}}_0} {\mathcal{C}^{\mathcal{G}}}_1 \equiv \quad \Sigma_{f\in F}\Sigma_{z\in{\mathcal{C}^{\mathcal{G}}}_1}\pi_0(f) =_{\mathcal{G}_0} \mathsf{dm}_{\mathcal{C}^{\mathcal{G}}}(z)$$

that is μ^{lf} is an internal diagram on the underlying graph of the free category $\mathcal{C}^{\mathcal{G}} \equiv ({\mathcal{C}^{\mathcal{G}}}_0, {\mathcal{C}^{\mathcal{G}}}_1, \mathsf{dm}_{\mathcal{C}^{\mathcal{G}}}, \mathsf{cd}_{\mathcal{C}^{\mathcal{G}}}, \mathsf{e}_{\mathcal{C}^{\mathcal{G}}}, \mathsf{Cmp}_{\mathcal{C}^{\mathcal{G}}})$ such that

$$\begin{tikzcd} F \times_{\mathcal{G}_0} {\mathcal{C}^{\mathcal{G}}}_1 \arrow[r, "\pi'_2"] \arrow[d, "\mu^{\mathsf{lf}}"] & {\mathcal{C}^{\mathcal{G}}}_1 \arrow[d, "\mathsf{cd}_{\mathcal{C}^{\mathcal{G}}}"'] \\ F \arrow[r, "\pi_0"'] & {\mathcal{C}^{\mathcal{G}}}_0 \end{tikzcd}$$

$$\pi_0 \cdot \mu^{\mathsf{lf}} = \mathsf{cd}_{\mathcal{C}^{\mathcal{G}}} \cdot \pi'_2$$

where π'_1 and π'_2 are respectively the first and the second projections of the pullback of $\mathsf{dm}_{\mathcal{C}^{\mathcal{G}}}$ along π_0, and in addition also the following diagrams commute

$$\begin{tikzcd} F \times_{{\mathcal{C}^{\mathcal{G}}}_0} {\mathcal{C}^{\mathcal{G}}}_0 \arrow[r, "id\times\mathsf{e}_{\mathcal{C}^{\mathcal{G}}}"] \arrow[dr, "\pi'_1"'] & F \times_{{\mathcal{C}^{\mathcal{G}}}_0} {\mathcal{C}^{\mathcal{G}}}_1 \arrow[d, "\mu^{\mathsf{lf}}"] \\ & F \end{tikzcd} \qquad \begin{tikzcd} F \times_{{\mathcal{C}^{\mathcal{G}}}_0} ({\mathcal{C}^{\mathcal{G}}}_1 \times_{{\mathcal{C}^{\mathcal{G}}}_0} {\mathcal{C}^{\mathcal{G}}}_1) \arrow[r, "(\mu^{\mathsf{lf}}\times id)\cdot\sigma"] \arrow[d, "id\times\mathsf{Cmp}_{\mathcal{C}^{\mathcal{G}}}"'] & F \times_{{\mathcal{C}^{\mathcal{G}}}_0} {\mathcal{C}^{\mathcal{G}}}_1 \arrow[d, "\mu^{\mathsf{lf}}"] \\ F \times_{{\mathcal{C}^{\mathcal{G}}}_0} {\mathcal{C}^{\mathcal{G}}}_1 \arrow[r, "\mu^{\mathsf{lf}}"'] & F \end{tikzcd}$$

where

$$\begin{aligned} & F \times_{{\mathcal{C}^{\mathcal{G}}}_0} ({\mathcal{C}^{\mathcal{G}}}_1 \times_{{\mathcal{C}^{\mathcal{G}}}_0} {\mathcal{C}^{\mathcal{G}}}_1) \\ & \qquad \equiv \ \Sigma_{f\in F}\Sigma_{z_1\in{\mathcal{C}^{\mathcal{G}}}_1}\Sigma_{z_2\in{\mathcal{C}^{\mathcal{G}}}_1}\pi_0(f) =_{\mathcal{G}_0} \mathsf{dm}_{\mathcal{C}^{\mathcal{G}}}(z_1) \wedge \mathsf{cd}_{\mathcal{C}^{\mathcal{G}}}(z_1) =_{\mathcal{G}_0} \mathsf{dm}_{\mathcal{C}^{\mathcal{G}}}(z_2) \end{aligned}$$

is the vertex of the pullback along π'_2 of the first projection of the pullback of $\mathsf{dm}_{\mathcal{C}^{\mathcal{G}}}$ along $\mathsf{cd}_{\mathcal{C}^{\mathcal{G}}}$ and σ is the isomorphism from $F \times_{{\mathcal{C}^{\mathcal{G}}}_0} ({\mathcal{C}^{\mathcal{G}}}_1 \times_{{\mathcal{C}^{\mathcal{G}}}_0} {\mathcal{C}^{\mathcal{G}}}_1)$ to $(F \times_{{\mathcal{C}^{\mathcal{G}}}_0} {\mathcal{C}^{\mathcal{G}}}_1) \times_{{\mathcal{C}^{\mathcal{G}}}_0} {\mathcal{C}^{\mathcal{G}}}_1$.

PROOF. In order to define the action μ^{lf}, note that by distributivity of coproducts with respect to pullbacks we get

$$F \times_{\mathcal{C}^{\mathcal{G}}{}_0} \mathcal{C}^{\mathcal{G}}{}_1 = F \times_{\mathcal{C}^{\mathcal{G}}{}_0} (\mathcal{G}_0 \ \oplus \ \widehat{\mathcal{C}^{\mathcal{G}}{}_1}) \simeq (F \times_{\mathcal{G}_0} \mathcal{G}_0) \oplus (F \times_{\mathcal{G}_0} \widehat{\mathcal{C}^{\mathcal{G}}{}_1})$$

Hence, by the elimination rule on the sum type we define $\mu^{\mathsf{lf}} : F \times_{\mathcal{C}^{\mathcal{G}}{}_0} \mathcal{C}^{\mathcal{G}}{}_1 \longrightarrow F$ as follows: for $f \in F$, $z \in \mathcal{C}^{\mathcal{G}}{}_1$ such that $\langle f,_{\mathcal{C}^{\mathcal{G}}{}_0} z \rangle \in F \times_{\mathcal{C}^{\mathcal{G}}{}_0} \mathcal{C}^{\mathcal{G}}{}_1$

$$\widetilde{\mu^{\mathsf{lf}}}(\langle f,_{\mathcal{C}^{\mathcal{G}}{}_0} z \rangle) \equiv \begin{cases} f & \text{if } z = \mathsf{inl}(x) \text{ for } x \in \mathcal{G}_0 \\ \widetilde{\mu^{\mathsf{lf}}}(f, s, \mathsf{lh}(s_1) - 1) & \text{if } z = \mathsf{inr}(s) \text{ for } s \in \widehat{\mathcal{C}^{\mathcal{G}}{}_1} \\ & \qquad \text{such that } \pi_0(f) =_{\mathcal{G}_0} \mathsf{dm}_{\mathcal{C}^{\mathcal{G}}}(\mathsf{inr}(s)) \end{cases}$$

where $s_1 \equiv \pi_1(s)$ and in turn

$$\widetilde{\mu^{\mathsf{lf}}}(f, s, n) \in F \ [f \in F, s \in \widehat{\mathcal{C}^{\mathcal{G}}{}_1}, d \in \pi_0(f) =_{\mathcal{G}_0} \mathsf{dm}_{\mathcal{C}^{\mathcal{G}}}(\mathsf{inr}(s))]$$

is defined as follows:

$$\widetilde{\mu^{\mathsf{lf}}}(f, s, 0) \equiv \mu(\langle f,_{\mathcal{G}_0} \mathsf{p}_1(s_1) \rangle)$$

$$\widetilde{\mu^{\mathsf{lf}}}(f, s, n+1) \equiv \begin{cases} \mu(\langle \widetilde{\mu^{\mathsf{lf}}}(f, s, n),_{\mathcal{G}_0} \mathsf{p}_{\mathsf{n+2}}(s_1) \rangle) & \text{if } n+2 \le \mathsf{lh}(s_1) \\ \widetilde{\mu^{\mathsf{lf}}}(f, s, n) & \text{if } n+2 > \mathsf{lh}(s_1) \end{cases}$$

But in order to assure the well definedness of the above definition we need to know the validity of $\pi_0(\widetilde{\mu^{\mathsf{lf}}}(f, s, n)) =_{\mathcal{G}_0} \mathsf{dm}_{\mathcal{G}}(\mathsf{p}_{\mathsf{n+2}}(s_1))$. Therefore, $\widetilde{\mu^{\mathsf{lf}}}$ is formally obtained by applying the first projection on the corresponding term of type

$$\Sigma_{x \in F}((\pi_0(x) =_{\mathcal{G}_0} \mathsf{dm}_{\mathcal{G}}(\mathsf{p}_{\mathsf{n+2}}(s_1)) \ \wedge \ n+2 \le \mathsf{lh}(s_1)) \ \vee \ n+2 > \mathsf{lh}(s_1)) \ [s \in \widehat{\mathcal{C}^{\mathcal{G}}{}_1}, n \in N]$$

defined by induction on natural numbers essentially as $\widetilde{\mathsf{app}}$ in the previous section together with a proof of the needed extra information.

It follows that μ^{lf} is an internal diagram, namely that

$$\pi_0 \cdot \mu^{\mathsf{lf}} = \mathsf{cd}_{\mathcal{C}^{\mathcal{G}}} \cdot \pi_2' \qquad \mu^{\mathsf{lf}} \cdot (\mathrm{id} \times \mathsf{e}_{\mathcal{C}^{\mathcal{G}}}) = \pi_1' \qquad \mu^{\mathsf{lf}} \cdot (\mathrm{id} \times \mathsf{Cmp}_{\mathcal{C}^{\mathcal{G}}}) = \mu^{\mathsf{lf}} \cdot (\mu^{\mathsf{lf}} \times \mathrm{id}) \cdot \sigma$$

and that μ^{lf} is unique with arguments similar to those used for ψ_1^{lf} in theorem 7.3. ∎

8. Conclusions

As promised we have shown here the following facts:

- arithmetic universes built by Joyal are list-arithmetic pretopoi;
- the initial arithmetic universe among Joyal's constructions is equivalent to the initial list-arithmetic pretopos;
- any list-arithmetic pretopos enjoys free categories and diagrams generated from graphs.

We think that these three facts provide the claimed justification for the identification of the general notion of arithmetic universe with that of list-arithmetic pretopos:

8.1. Definition. An *arithmetic universe* is a list-arithmetic pretopos.

We leave as an open problem whether the notion of arithmetic pretopos, namely a pretopos with a parameterized natural numbers object, can be taken as a notion of arithmetic universe. This suspicion is supported by the fact that the initial arithmetic pretopos is equivalent to the initial arithmetic universe built by Joyal. However we doubt that any arithmetic pretopos supports free internal categories and diagrams or is list-arithmetic.

Finally, our general definition of arithmetic universe as list-arithmetic pretopos is equivalent to that used in a recent talk by André Joyal [Joy05].

Acknowledgements. My acknowledgements go first to Martin Hyland, who proposed me to work on arithmetic universes and helped me with many fruitful discussions during my staying in Cambridge. Many thanks also to Steve Vickers for providing me Gavin Wraith's unpublished notes [Wra85] with the master's thesis of his student [Mor96] and to Robin Cockett for sending Roland's master's thesis [Rol76]. Then, I wish also to thank Pino Rosolini, Giovanni Sambin, Paul Taylor and Silvio Valentini for their generous promptness in discussing my research work. Finally, I am very grateful to André Joyal for his interest in this work and for letting me know his draft about Gödel incompleteness.

A. Appendix: Useful operations on list types

Here, we describe some very useful operations on list types following the notation in remark 3.3.

We warn the reader that when defining an operation by elimination on the list type we simply write the base and inductive steps necessary to define it instead of writing the whole resulting proof-term.

A.1. Definition. We define the *length* of a list $\mathsf{lh}(s) \in \mathrm{List}(A)$ $[s \in \mathrm{List}(A)]$ by the elimination rule on the list type as follows

$$\mathsf{lh}(\epsilon) \equiv 0 \qquad\qquad \mathsf{lh}(\lfloor s, a \rfloor) \equiv \mathsf{lh}(s) + 1$$

A.2. Definition. Given a term $\phi(x) \in B$ $[x \in A]$ we lift this term on lists by defining

$$\mathsf{Lst}(\phi)(s) \in \mathrm{List}(B)\ [s \in \mathrm{List}(A)]$$

by elimination on the list type as follows

$$\mathsf{Lst}(\phi)(\epsilon) \equiv \epsilon \qquad\qquad \mathsf{Lst}(\phi)(\lfloor s, a \rfloor) \equiv \lfloor \mathsf{Lst}(\phi)(s), \phi(a) \rfloor$$

Now, we define the type of non-empty lists which also enjoys a specific induction.

A.3. Definition. The *type of non-empty lists* is defined as

$$\mathrm{List}^*(A) \equiv \ \Sigma_{t\in \mathrm{List}(A)} \mathsf{lh}(t) \geq 1$$

where $n \geq m$ may be defined as $\exists_{y\in N}\ n =_N m + y$.

In the next, given a non-empty list $l \in \mathrm{List}(A)$ such that $p \in \mathsf{lh}(l) \geq 1$ we simply write

$$l^* \in \mathrm{List}^*(A) \qquad\qquad \text{instead of } \langle l, p\rangle \in \mathrm{List}^*(A)$$

Observe that we can think of $\mathrm{List}^*(A)$ as the inductive type with the following constructors:

- the basic constructors of $\mathrm{List}^*(A)$ are the one element lists of A: for $a \in A$

$$\lfloor a \rfloor^* \in \mathrm{List}^*(A)$$

- the list constructor cons of $\mathrm{List}(A)$ produces a constructor

$$\mathsf{cons}_{\mathrm{List}^*}(z, a) \in \mathrm{List}^*(A)\ [z \in \mathrm{List}^*(A), a \in A]$$

  defined as $\mathsf{cons}_{\mathrm{List}^*}(z, a) \equiv \mathsf{cons}(z_1, a)^*$ where $z_1 \equiv \pi_1(z)$.

In order to define operations on the type $\mathrm{List}^*(A)$ of non-empty lists, we can use an elimination rule analogous to that one for $\mathrm{List}(A)$. This elimination says that in order to define an operation on the type $\mathrm{List}^*(A)$ we first define the operation on the one element lists, which is the base case, and then we specify how to define it on the lists obtained by adding a new element to a list on which the operation is supposed to be already defined, which is the inductive case.

A.4. Proposition. [Induction on non-empty lists] *Given the type $B(s)$ $[s \in \mathrm{List}^*(A)]$ and the following terms*

$$\begin{array}{ll} \textit{(base case)} & d_1(x) \in B(\lfloor x \rfloor^*)\ [x \in A] \\ \textit{(inductive case)} & d_2(s, y, w) \in B(\mathsf{cons}_{\mathrm{List}^*}(s, y))\ [s \in \mathrm{List}^*(A), y \in A, w \in B(s)] \end{array}$$

there exists a term $El_{\mathrm{List}^*}(z) \in B(z)$ $[z \in \mathrm{List}^*(A)]$ such that for $s \in \mathrm{List}^*(A)$, $x \in A$, $y \in A$

$$\begin{array}{l} El_{\mathrm{List}^*}(\lfloor x \rfloor^*) = d_1(x) \\ El_{\mathrm{List}^*}(\mathsf{cons}_{\mathrm{List}^*}(s, y)) = d_2(s, y, El_{\mathrm{List}^*}(s)) \end{array}$$

Proof. By using the hypothesis of our statement we can derive a proof of the type

$$\Sigma_{z'\in \mathrm{List}(A)^*}(B(z') \times w =_{\mathrm{List}(A)} \pi_1(z')) + w =_{\mathrm{List}(A)} \epsilon$$

by induction on $w \in \mathrm{List}(A)$. Then, observe that for $z \in \mathrm{List}^*(A)$, since $\pi_1(z) =_{\mathrm{List}(A)} \epsilon$ is false by sum disjointness (with a proof similar to that of proposition 5.1), and since $z = z' \in \mathrm{List}^*(A)$ if $\pi_1(z) =_{\mathrm{List}(A)} \pi_1(z')$ holds, we then conclude a proof of $B(z)$ as claimed. ∎

A.5. REMARK. In order to facilitate the definition of operations on non-empty lists, here and in the main body of this paper we simply write

$$l \in \mathrm{List}^*(A) \text{ if } l \in \mathrm{List}(A) \text{ with a proof of } \mathsf{lh}(l) \geq 1$$

and conversely also simply

$$l \in \mathrm{List}(A) \text{ if } l \in \mathrm{List}^*(A)$$

A.6. DEFINITION. On $\mathrm{List}^*(A)$ we define the operation $\mathsf{frt}(s) \in \mathrm{List}(A)$ $[s \in \mathrm{List}^*(A)]$ taking the front of a non-empty list: for $s \in \mathrm{List}^*(A)$ and $a \in A$

$$\mathsf{frt}(\lfloor a \rfloor) \equiv \epsilon \qquad \mathsf{frt}(\lfloor s, a \rfloor) \equiv \lfloor \mathsf{frt}(s), a \rfloor$$

and the operation $\mathsf{bck}(s) \in \mathrm{List}(A)$ $[s \in \mathrm{List}^*(A)]$ taking the back of a non-empty list: for $s \in \mathrm{List}^*(A)$ and $a \in A$

$$\mathsf{bck}(\lfloor a \rfloor) \equiv \epsilon \qquad \mathsf{bck}(\lfloor s, a \rfloor) \equiv s$$

Moreover, we define the operation $\mathsf{fst}(s) \in A$ $[s \in \mathrm{List}^*(A)]$ selecting the first element of a list as follows: for $s \in \mathrm{List}^*(A)$ and $a \in A$

$$\mathsf{fst}(\lfloor a \rfloor) \equiv a \qquad \mathsf{fst}(\lfloor s, a \rfloor) \equiv \mathsf{fst}(s)$$

and the operation $\mathsf{las}(s) \in A$ $[s \in \mathrm{List}^*(A)]$ selecting the last element of a list as follows: for $s \in \mathrm{List}^*(A)$ and $a \in A$

$$\mathsf{las}(\lfloor a \rfloor) \equiv a \qquad \mathsf{las}(\lfloor s, a \rfloor) \equiv a$$

Then, we define projections on a non-empty list. For every $n \in N$ we define the n-th projection $\mathsf{p_n}(s) \in A$ $[s \in \mathrm{List}^*(A), n \in N]$ by the elimination rule on non-empty lists as follows

$$\mathsf{p_n}(s) \equiv \begin{cases} a & \text{if } s \equiv \lfloor a \rfloor \\ \mathsf{p_n}(s') & \text{if } s \equiv \lfloor s', a \rfloor \text{ and } n \leq \mathsf{lh}(s') \\ a & \text{if } s \equiv \lfloor s', a \rfloor \text{ and } n > \mathsf{lh}(s') \end{cases}$$

Finally, we use projections to define the n-th part of a list

$$\mathsf{part_n}(s) \in \mathrm{List}(A) \; [s \in \mathrm{List}(A), n \in N]$$

by induction on natural numbers as follows: given a list $s \in \mathrm{List}(A)$

$$\mathsf{part_0}(s) \equiv \epsilon$$

$$\mathsf{part_{n+1}}(s) \equiv \begin{cases} \lfloor \mathsf{part_n}(s), \mathsf{p_{n+1}}(s) \rfloor & \text{if } n+1 \leq \mathsf{lh}(s) \\ \mathsf{part_n}(s) & \text{if } n+1 > \mathsf{lh}(s) \end{cases}$$

References

[BCRS98] L. Birkedal, A. Carboni, G. Rosolini, and D. Scott. Type theory via exact categories (extended abstract). In *Thirteenth Annual IEEE Symposium on Logic in Computer Science (Indianapolis, IN, 1998)*, IEEE Computer Soc., pages 188–198, 1998.

[Car95] A. Carboni. Some free constructions in realizability and proof theory. *Journal of Pure and Applied Algebra*, 103:117–148, 1995.

[CLW93] A. Carboni, S. Lack, and R.F.C. Walters. Introduction to extensive and distributive category. *Journal of Pure and Applied Algebra*, 84:145–158, 1993.

[CV98] A. Carboni and E.M. Vitale. Regular and exact completions. *Journal of Pure and Applied Algebra*, 125:79–116, 1998.

[Coc90] J.R.B. Cockett. List-arithmetic distributive categories: locoi. *Journal of Pure and Applied Algebra*, 66:1–29, 1990.

[dB91] N.G. de Bruijn. Telescopic mapping in typed lambda calculus. *Information and Computation*, 91:189–204, 1991.

[Hof95] M. Hofmann. 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, 1995.

[Hyl82] J.M.E. Hyland. The effective topos. In *The L.E.J. Brouwer Centenary Symposium (Noordwijkerhout, 1981)*, volume 110 of *Stud. Logic Foundations Math.*, pages 165–216. North-Holland, Amsterdam-New York, 1982.

[JM95] A. Joyal and I. Moerdijk. *Algebraic set theory*, volume 220 of *Lecture Note Series*. Cambridge University Press, 1995.

[Joh77] P.T. Johnstone. *Topos theory* Academic Press, 1977.

[Joh02a] P.T. Johnstone. *Sketches of an elephant: a topos theory compendium. Vol. 1*, volume 43 of *Oxford Logic Guides.* The Clarendon Press, Oxford University Press, New York,, 2002.

[Joh02b] P. T. Johnstone. *Sketches of an elephant: a topos theory compendium. Vol. .*, volume 44 of *Oxford Logic Guides.* The Clarendon Press, Oxford University Press, New York,, 2002.

[Joy05] A. Joyal. The Gödel incompleteness theorem, a categorical approach. *Cahiers de topologie et géometrie différentielle categoriques*, 16(3), 2005. Short abstract of the talk given at the International conference *Charles Ehresmann: 100 ans*, Amiens, 7-9 October, 2005.

[LR03] F.W. Lawvere and R. Rosebrugh. *Sets for mathematics.* Cambridge University Press, 2003.

[Mai99a] M.E. Maietti. 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, 1999.

[Mai99b] M.E. Maietti. The typed calculus of arithmetic universes. Technical report, University of Birmingham, CSR-99-14, December 1999. Also Technical Report-University of Padova n.5 Dec. 1999.

[Mai03] M.E. Maietti. Joyal's arithmetic universes via type theory. In *Category Theory in Computer Science, 2002*, volume 69 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2003.

[Mai05a] M.E. Maietti. Modular correspondence between dependent type theories and categories including pretopoi and topoi. *Mathematical Structures in Computer Science*, 15(6), 2005.

[Mai05b] M.E. Maietti. 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*, pages 105–126. Elsevier, 2005.

[Mar84] P. Martin-Löf. *Intuitionistic Type Theory, notes by G. Sambin of a series of lectures given in Padua, June 1980.* Bibliopolis, Naples, 1984.

[MM92] S. MacLane and I. Moerdijk. *Sheaves in Geometry and Logic. A first introduction to Topos theory.* Springer Verlag, 1992.

[MMdPR05] M.E. Maietti, P. Maneggia, V. de Paiva, and E. Ritter. Relating categorical semantics for intuitionistic linear logic. *Applied Categorical Structures*, 13(1):1–36, 2005.

[Mor96] A. Morrison. Reasoning in arithmetic universes. Master's thesis, University of London - Imperial College of Science, Technology and Medicine, advisor: S. Vickers, September 1996.

[MR77] M. Makkai and G. Reyes. *First order categorical logic*, volume 611 of *Lecture Notes in Mathematics*. Springer Verlag, 1977.

[NPS90] B. Nordström, K. Peterson, and J. Smith. *Programming in Martin-Löf's Type Theory.* Clarendon Press, Oxford, 1990.

[Odi89] P. Odifreddi. *Classical recursion theory*, volume 125 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., 1989.

[Pit00] A.M. Pitts. Categorical logic. In Oxford University Press, editor, *Handbook of Logic in Computer Science*, volume 5 of *Oxford Sci. Publ.*, pages 39–128, 2000.

[Rol76] S. Roland. Essai sur les mathématiques algorithmiques. Master's thesis, Université du Québec, Montréal - Maîtrise ès sciences (mathématiques), advisor: A. Joyal, January 1976.

[See84] R. Seely. Locally cartesian closed categories and type theory. *Math. Proc. Cambr. Phyl. Soc.*, 95:33–48, 1984.

[Smi88] J. Smith. The independence of Peano's fourth axiom from Martin Löf's type theory without universes. *Journal of Symbolic Logic*, 53, 1988.

[Str91] Th. Streicher. *Semantics of type theory.* Birkhäuser, 1991.

[Tay05] P. Taylor. Inside every model of abstract Stone duality lies an arithmetic universe. In *Category Theory in Computer Science, 2004*, volume 122 of *Electronic Notes in Theoretical Computer Science*, pages 247–296. Elsevier, 2005.

[Wra85] G.C. Wraith. Notes on arithmetic universes and Gödel incompleteness theorems. Unpublished manuscript, 1985.

Dipartimento di Matematica Pura ed Applicata
University of Padova
via Belzoni n.7, 35100 Padova, Italy
Email: `maietti@math.unipd.it`

This article may be accessed at `http://www.tac.mta.ca/tac/` or by anonymous ftp at `ftp://ftp.tac.mta.ca/pub/tac/html/volumes/24/3/24-03.{dvi,ps,pdf}`