On the Semantics of Coinductive Types in Martin-Löf Type Theory

Federico De Marchi*

Department of Mathematics, University of Utrecht,
Utrecht, P.O. Box 80010, 3508 TA Utrecht, The Netherlands

Abstract. There are several approaches to the problem of giving a categorical semantics to Martin-Löf type theory with dependent sums and products and extensional equality types. The most established one relies on the notion of a type-category (or category with attributes) with Σ and Π types. We extend such a semantics by introducing coinductive types both on the syntactic level and in a type-category. Soundness of the semantics is preserved.

As an example of such a category, we prove that the type-category built over a locally cartesian closed category $\mathcal{C}$ admits coinductive types whenever $\mathcal{C}$ has final coalgebras for all polynomial functors.

1 Introduction

The problem of finding a categorical semantics to Martin-Löf type theory has given rise to a substantial amount of very interesting research, over the years. Most of it was inspired by Seely's first attempt to use locally cartesian closed categories [20], which however was proved slightly inaccurate. In fact, he had glossed over the need to have a choice of pullbacks that compose on the nose in the category, which is essential in order to interpret substitution. In order to fix the problem, Cartmell devised the notion of a contextual category [7], on which Streicher based his semantics for a dependent type theory with dependent products and sums, and extensional equality [21]. However, the axioms for a contextual category are very "uncategorical" in spirit, since they assume a well-founded order on objects; for this reason, this notion was later replaced by the more abstract one of category with attributes [11], or type-category, as Pitts calls them [19].

From a fibrational point of view, a type-category is just a split fibration. Adapting an argument of Bénabou [6], Hofmann could show that every locally cartesian closed category $\mathcal{C}$ gives rise to a split fibration which is equivalent (in a suitable higher-order sense) to the canonical indexing of $\mathcal{C}$. In this way, he could fix the bug in Seely's paper by interpreting type theory in the type-category built out of any locally cartesian closed one.

* This research is supported by NWO Grant n. 613.000.222.
Email: marchi@math.uu.nl

J.L. Fiadeiro et al. (Eds.): CALCO 2005, LNCS 3629, pp. 114–126, 2005.

In [23], Benno van den Berg and the present author suggested that such a semantics could be extended to one for a theory with coinductive types, provided the original locally cartesian closed category had M-types (i.e. final coalgebras of polynomial functors). It is the purpose of this paper to make that statement precise.

The problem of adding infinite objects to Martin-Löf type theory has been considered by several authors over the years [8,16,17]. The main source of trouble, in this case, is that infinite (non-well-founded) objects in a type might have infinitely long reductions, therefore making a full description of their normal form impossible to achieve. When in presence of well-founded types, one can give a description of an infinite object by its finite approximations [17,10,14,23,1]; however, in our setting we want to avoid using well-founded types. In Section 2 we shall introduce a system of rules for coinductive types. These are very close to the categorical formulation of the properties of the final coalgebra for a polynomial functor. In particular, our way of introducing terms of a coinductive type is by unfolding a coalgebra at a particular state. This is analogue to the concept of a productive definition of a term as discussed by Coquand in [8]. The guardedness he requires there is given in our context by the polynomial functor itself.

In Section 3 we recall the concept of a type-category and introduce that of a coinductive type therein, and we show how these categories provide a sound categorical semantics for the aforementioned type theory. Finally, in Section 4 we show that, following Hofmann's construction [11], any locally cartesian closed category with final coalgebras of polynomial functors gives rise to a type-category with coinductive types, thus providing a wide class of examples.

Acknowledgements. The author would like to thank Benno van den Berg for the several useful discussions, and to Dr. Thorsten Altenkirch for his valuable suggestions.

2 Coinductive Types in Martin-Löf Type Theory

We consider a version of Martin-Löf type theory with Σ-types, Π-types, and extensional equality, as presented in [15,18]. When one views the theory as a programming language, according to the "types-as-specification" paradigm, it may be desirable to allow for some programs to have an infinite computation. When using type theory to study constructive mathematics, coinductive types can help modelling some non-well-founded sets [2]. In presence of inductive types, infinite programs can be fully described by the collection of their finite (but arbitrarily long) approximations [17,16,9,14]. Alternatively, one can describe the elements of a non-well-founded set by means of a recursive definition, provided this is guarded, or productive [8,10]. In our axioms, we shall resort to the second method, since we do *not* assume to have W-types.

Categorically, it is clearly understood that non-well-founded terms over a signature can be collected into the final coalgebra for a polynomial functor associated to it [13,22,5,3]. In order to be closer to the categorical semantics we intend to present, we give axioms here, which closely resemble those of a final coalgebra.

It is to be remarked that our definitions make equality undecidable in the system. However, the fact that we take extensional equality into account already breaks decidability, and we choose to accept this drawback. We are aware of recent work by Altenkirch on "tracking the proofs" of equality of various terms within a version of intensional type theory which he calls "observational type theory" [4]. This seems to be a promising area for further developments in the direction of actual implementations of the theory.

Now, we come to the axioms. The introduction rule takes the obvious form

$$[M\text{-}\textsc{form}] \qquad \frac{A \textit{ type} \qquad B(a) \textit{ type } [a : A]}{M(A,B) \textit{ type}}$$

We should think of the elements of type A as term constructors, the arity of $a : A$ being given by the type $B(a)$. Note that, following the conventions of [18], we are omitting those contexts which are not discharged by the rule, and we are omitting the obvious substitution rules that should come together with the introduction of a new type constructor. Given types A and $B(a)$ $[a : A]$ as above, we shall often write $P_b(X)$ for the type $(\Sigma a : A)(B(a) \to X)$.

Elements of the coinductive type $M(A,B)$ are defined corecursively. Whenever we have a type X and an element of the function type $f : X \to P_b(X)$, we can think of it as a way of describing the *evolution* of the elements of X according to the signature described by A and B. In particular, every element $x : X$ is *productive*, since $f(x).1$ is an element of type A (i.e. a term constructor of the specified signature) and $f(x).2$ takes any $b : B(f(x).1)$ to another element in X, from which we can reiterate the procedure indefinitely. In this way, we can associate to each element $x : X$ a (possibly non-well-founded) tree whose nodes are labelled by elements of type A and branches departing from a node labelled by a are labelled by elements of type $B(a)$. This is the *unfolding* of f at x, or the *behaviour* of x under the evolution f.

$$[M\text{-}\textsc{intro}] \qquad \frac{x : X \qquad f : X \to P_b(X)}{\mathsf{unfold}(f,x) : M(A,B)}$$

When we are given an element t of the coinductive type $M(A,B)$, which we think of as a tree, we can extract its *root*, which is an element of A, and its *branching function*, which has type $B(\mathsf{root}(t)) \to M(A,B)$.

$$[M\text{-}\textsc{elim}] \qquad \frac{t : M(A,B)}{\mathsf{root}(t) : A} \qquad\qquad \frac{t : M(A,B)}{\mathsf{br}(t) : B(\mathsf{root}(t)) \to M(A,B)}$$

In order to understand how root and br act on an element $t : M(A,B)$, suppose t is the tree representing the behaviour of some $x : X$ under $f : X \to P_b(X)$; that is, $t = \mathsf{unfold}(f,x)$. Then, $f(x)$ is an element of $P_b(X) \equiv (\Sigma a : A)(B(a) \to X)$, which is a pair (a,s). These are precisely the root and the branching function of the original t. This explanation, justifies the following equality rules:

$$[M\text{-EQ}] \quad \frac{x : X \qquad f : X \to P_b(X)}{\mathsf{root}(\mathsf{unfold}(f, x)) = f(x).1 : A}$$

$$\frac{x : X \qquad\qquad f : X \to P_b(X)}{\mathsf{br}(\mathsf{unfold}(f, x)) = (b)\mathsf{unfold}(f, (f(x).2)b) : B(f(x).1) \to M(A, B)}$$

$$\frac{t : M(A, B)}{t = \mathsf{unfold}((y)(\mathsf{root}(y), \mathsf{br}(y)), t) : M(A, B)}$$

By the definitions of root and br, it follows immediately that the abstraction $m{\equiv}(y)(\mathsf{root}(y), \mathsf{br}(y))$ has type $M(A, B) \to P_b(M(A, B))$. Therefore, for any $t : M(A, B)$ we can unfold m at t. The third equality rule above is stating precisely that $\mathsf{unfold}(m, t) = t$. This rule is essential in proving that, given any function $f : X \to P_b(X)$, the function $(x)\mathsf{unfold}(f, x)$ is the *unique* one to $M(A, B)$ which preserves the root and branching functions.

3 Type-Categories and Coinductive Types

Now that we have introduced the axioms of our type theory, we can approach the question of giving it a categorical semantics. It has already been mentioned that Seely's idea of using locally cartesian closed categories (lccc's) for modelling dependent types is very insightful, but not correct [20]. His idea was to interpret contexts by objects in the category, and a judgement of the form $A(x_1, \ldots, x_n)$ *type* $[x_1 : X_1, \ldots, x_n : X_n]$ by an arrow $\alpha : A \longrightarrow X$ over the object interpreting the context $[x_i : X_i]$. Given another context $[y_1 : Y_1, \ldots, y_m : Y_m]$ interpreted by an object Y, and an n-tuple of terms $f_i(y_1, \ldots, y_m) : X_i$, this is interpreted by an arrow $f : Y \longrightarrow X$, and the interpretation of the substituted term $A[f_i/x_i]$ is the pullback of α along f. Now, supposing we are given two composable substitutions, interpreted by maps

$$Z \xrightarrow{g} Y \xrightarrow{f} X,$$

the interpretation of the type $(A[f_i/x_i])[g_j/y_j]$ obtained by first performing the substitution f and then the substitution g, should be the same as the interpretation of the type $A[f_i[g_j/y_j]/x_i]$, obtained by performing the composite substitution on A. In other words, the pullback of α first along f and then along g ought to be the same as the pullback along the composite gf. Unfortunately, in general we cannot make a coherent choice of pullbacks in an lccc which is closed under pullback pasting. Hence, the need for a more refined model, in which substitution can be traced more accurately.

The first attempt in this direction was that of Cartmell [7], who proposed in his PhD thesis the notion of a contextual category. This has been further studied by Streicher [21]; however, Cartmell himself, and later other authors, found that contextual categories have a rather technical and cumbersome definition, that could be left aside, in favour of what have been called *categories with attributes*, or *type-categories* [19,11].

A type-category, in the notation of Pitts, is specified by a category $\mathcal{C}$ with a terminal object 1, together with the following extra structure:

– for each object X in $\mathcal{C}$, a set $Type_{\mathcal{C}}(X)$ of *X-indexed types*;
– for each X in $\mathcal{C}$, a map $p : Type_{\mathcal{C}}(X) \longrightarrow \mathrm{Ob}(\mathcal{C}/X)$ which takes an X-indexed type A to the *canonical projection*

$$\pi_A : X \ltimes A \longrightarrow X$$

from the *total object* $X \ltimes A$ of A to X itself;
– for each map $f : Y \longrightarrow X$ in $\mathcal{C}$, an operation assigning to each X-indexed type A a Y-indexed type f^*A, called the *pullback of A along f*, together with a morphism

$$f \ltimes A : Y \ltimes f^*A \longrightarrow X \ltimes A$$

making the following into a pullback:

$$\begin{tikzcd} Y \ltimes f^*A \arrow[r, "f \ltimes A"] \arrow[d, "\pi_{f^*A}"'] & X \ltimes A \arrow[d, "\pi_A"] \\ Y \arrow[r, "f"'] & X. \end{tikzcd} \tag{1}$$

These data are subject to the following coherence conditions, for $A \in Type_{\mathcal{C}}(X)$, $f : Y \longrightarrow X$ and $g : Z \longrightarrow Y$:

$$\begin{array}{ccc} \mathsf{id}_X^* A = A & \text{and} & \mathsf{id}_X \ltimes A = \mathsf{id}_{X \ltimes A}; \\ g^*(f^*A) = (fg)^*A & \text{and} & (f \ltimes A)(g \ltimes f^*A) = (fg) \ltimes A. \end{array}$$

Example (The syntactic category of a theory). We shall provide a wide class of examples of type-categories in the next section. For the time, it is useful to notice that any dependent type theory $\mathbb{T}$ gives rise to a type-category $\mathcal{T}$. Objects in $\mathcal{T}$ are equivalence classes of well-formed contexts in the theory, modulo the relation determined by provable equality of two contexts. An arrow f from (the equivalence class of) a context $Y = [y_j : Y_j]$ $(j = 1, \ldots, m)$ to $X = [x_i : X_i]$ $(i = 1, \ldots, n)$ consists of the equivalence class (again, modulo provable equality) of an n-tuple of terms $f_i(y_1, \ldots, y_m) : X_i$. The final object in $\mathcal{T}$ is clearly given by the empty context. The family $Type_{\mathcal{T}}(X)$ consists of all those types A for which the judgement A *type* $[x_i : X_i]$ is derivable in $\mathbb{T}$, the canonical projection of such a type being the projection

$$(x_i)_{i=1,\ldots,n} : [x_1 : X_1, \ldots, x_n : X_n, x : A] \longrightarrow [x_1 : X_1, \ldots, x_n : X_n].$$

The reindexing along a context morphism $f : Y \longrightarrow X$ of a type $A(x_1, \ldots, x_n)$ depending on the context X, is the type obtained by substituting the x_i's by the terms specified by f:

$$f^*A(y_1, \ldots, y_m) = A[f_1/x_1, \ldots, f_n/x_n].$$

It is clear that these data satisfy the conditions for a type-category, which is called the *syntactic category* built over $\mathbb{T}$.

When interpreting a type theory in a type-category $\mathcal{C}$, objects of $\mathcal{C}$ are used to represent well-formed contexts of the theory, whereas arrows are used to interpret substitution; that is, tuples of terms of the appropriate types, which depend on the variables defined in the domain. If we interpret a context X by an object (which we denote again by X, abusing the notation), then a judgement of the form A *type* $[X]$ is interpreted by an element $A \in Type_{\mathcal{C}}(X)$. If $f : Y \longrightarrow X$ is a substitution, then the pullback f^*A will interpret the type A with the variables substituted according to f. The coherence conditions expressed above ensure that composite substitutions are correctly interpreted. The total object of a type A depending on a context X interprets the context $[X, a : A]$, with the obvious projection onto X (see [19,11] for further details). Finally, terms of a given type are interpreted by sections of the canonical projection; that is, a judgement $t : A\ [X]$ is interpreted by a morphism $t : X \longrightarrow X \ltimes A$ in $\mathcal{C}$ such that $\pi_A t = \mathsf{id}_X$. Given a substitution $f : Y \longrightarrow X$ between contexts, and a term t as above, the term obtained by substituting all the variables in t according to f is interpreted by the unique section f^*t of π_{f^*A} such that $(f \ltimes A)f^*t = tf$, which is determined by the pullback (1).

Note that the map $p : Type_{\mathcal{C}}(X) \longrightarrow \mathrm{Ob}(\mathcal{C}/X)$ induces the structure of a category on the collection of types over X, in an obvious way: maps between two elements A and B are maps in the slice category $\mathcal{C}/X$ between the canonical projections π_A and π_B. The map p then becomes a full and faithful functor from $Type_{\mathcal{C}}(X)$ to $\mathcal{C}/X$. Moreover, for a $\mathcal{C}$-morphism $f : Y \longrightarrow X$ the pullback functor $f^* : \mathcal{C}/X \longrightarrow \mathcal{C}/Y$ restricts to a functor $f^* : Type_{\mathcal{C}}(X) \longrightarrow Type_{\mathcal{C}}(Y)$, whose action on objects is precisely the one specified by the type-category structure. The association $X \mapsto Type_{\mathcal{C}}(X)$ and $f \mapsto f^*$ defines a functor $\mathcal{C} \to \mathcal{C}at$. Functoriality is ensured by the coherence conditions for the pullback functors, and it is precisely the condition needed in order for the substitution to be correctly interpreted. We could not use the slice categories $\mathcal{C}/X$ because they give rise to a *pseudo*-functor, and the action of the pullback functors composes only up to isomorphism.

Now, suppose the left adjoint Σ_f to the pullback functor $f^* : \mathcal{C}/X \longrightarrow \mathcal{C}/Y$ restricts to categories of types as well. Then, it is possible to interpret dependent sums in our model. If A *type* $[X]$ and $B(a)$ *type* $[X, a : A]$ are judgements in the theory, interpreted by objects A in $Type_{\mathcal{C}}(X)$ and B in $Type_{\mathcal{C}}(X \ltimes A)$, then the composite $\Sigma_{\pi_A}\pi_B = \pi_A\pi_B$ is the canonical projection of an object in $Type_{\mathcal{C}}(X)$, which we define to be the interpretation of the type $\Sigma(A, B)$ in the context X.

Likewise, we can interpret dependent products in $\mathcal{C}$ provided the pullback functors have a right adjoint. More specifically, we need that for A in $Type_{\mathcal{C}}(X)$ and B in $Type_{\mathcal{C}}(X \ltimes A)$ there is an indexed type $\Pi(A, B)$ in $Type_{\mathcal{C}}(X)$ and a morphism $ap_{A,B} : \pi_A^*\Pi(A, B) \longrightarrow B$ in $Type_{\mathcal{C}}(X \ltimes A)$ with the obvious universal property. Stability of the interpretation under substitution is ensured by the further requirements that, for any $f : Y \longrightarrow X$ in $\mathcal{C}$,

$$f^*\Pi(A, B) = \Pi(f^*A, f^*B) \quad \text{and} \quad (f \ltimes A)^* ap_{A,B} = ap_{f^*A,(f \ltimes A)^*B}.$$

We refer the reader to Streicher's monograph [21] for a treatment of extensional equality types.

Once we have defined an interpretation of the type-valued and term-valued function symbols of a theory into a type-category, we can use the rules for dependent products and coproducts in order to inductively define an interpretation of all well-formed contexts, of types depending on a context, and of their terms. We shall then say that a judgement of the form A *type* $[X]$ is satisfied by the model if A is interpreted by an element of $Type_{\mathcal{C}}(X)$, and similarly for one of the form $a : A\ [X]$. Equality judgements will be satisfied when the two sides of the equality have the same interpretation in $\mathcal{C}$. The properties of dependent products and coproducts ensure that this model is sound, in the sense that any judgement derivable in the type theory is satisfied by any interpretation (see [19] for more details).

Example (Interpretation in the syntactic category). Given a type theory $\mathbb{T}$, this has an obvious interpretation into its syntactic category $\mathcal{T}$. A context is interpreted by its equivalence class, a judgement of the form A *type* $[X]$ is interpreted by the element A in $Type_{\mathcal{T}}(X)$, and a term $t : A\ [X]$ is interpreted by the section

$$(x_1, \ldots, x_n, t(x_1, \ldots, x_n)) : [x_1 : X_1, \ldots, x_n : X_n] \longrightarrow [x_1 : X_1, \ldots, x_n : X_n, x : A]$$

of the canonical projection of A. It is clear that a judgement is provable in $\mathbb{T}$ if an only if it is satisfied by the model.

We now proceed to specify the amount of structure needed in order to interpret coinductive types. Unsurprisingly, the properties closely resemble the type theoretic rules described in Section 2.

Definition 1. A type-category $\mathcal{C}$ with dependent sums, dependent products and extensional equality (in the sense of [11]), *has coinductive types* if for any A in $Type_{\mathcal{C}}(X)$ and B in $Type_{\mathcal{C}}(X \ltimes A)$ there is a type

$$M(A, B) \text{ in } Type_{\mathcal{C}}(X)$$

with the following properties:

– for any type Y in $Type_{\mathcal{C}}(X)$ and sections y of π_Y and g of $\pi_{Y \to P_b(Y)}$ there is a section
$$unfold(g, y) \quad \text{of} \quad \pi_{M(A,B)};$$
– for any section t of $\pi_{M(A,B)}$ there are sections
$$root(t) \quad \text{of} \quad \pi_A \quad \text{and} \quad br(t) \quad \text{of} \quad \pi_{root(t)^* B \to M(A,B)}$$
– and the following are equal sections of π_A, $\pi_{(g(y).1)^* B \to M(A,B)}$ and $\pi_{M(A,B)}$, respectively:

$$\begin{aligned} root(unfold(g, y)) &= g(y).1 \\ br(unfold(g, y)) &= (b)unfold(g, (g(y).2)b) \\ unfold((s)(root(s), br(s)), t) &= t; \end{aligned}$$

– for a morphism $f : X' \longrightarrow X$ in $\mathcal{C}$, the following coherence condition holds:

$$f^*M(A, B) = M(f^*A, f^*B),$$

together with the analogous conditions for the aforementioned sections.

It is immediate from the definition how to interpret a Martin-Löf type theory with coinductive types in any type-category which has coinductive types, and soundness readily extends.

Theorem 2 (soundness). *With the notion of satisfaction described above, the collection of judgements satisfied by a model in a type-category with coinductive types is closed under the rules of Σ-types, Π-types, extensional equality as well as those given in Section 2 for coinductive types.*

Moreover, it is an immediate consequence of the definitions that the syntactic category of a type theory $\mathbb{T}$ with coinductive types has the structure of a type-category with coinductive types, which provides a complete semantics for $\mathbb{T}$, in the sense that a judgement is provable in the theory if and only if it is satisfied by the model.

4 From M-Types to Coinductive Types

As we mentioned in the introduction, the reason for introducing type-categories was that of overcoming the problem of not being able to make a coherent choice of pullbacks in a locally cartesian closed category, in such a way that the pullback functors between the slice categories would compose on the nose (instead of up to isomorphism). Another way to phrase the problem is in terms of fibrations. The association $X \mapsto \mathcal{C}/X$ defines a pseudofunctor $\mathcal{C}^{\mathrm{op}} \to \mathcal{C}at$ (it is a pseudofunctor precisely because composition of the pullback functors is possible only up to coherent isomorphisms), which is also called the *canonical indexing* of $\mathcal{C}$. It is a way to view $\mathcal{C}$ as an indexed category, or, equivalently, as a cloven fibration (for definitions, see for example [12]). Moreover, the strict functoriality on the composition of the pullback amounts exactly to saying that this fibration is split. So, locally cartesian closed categories fail to support an interpretation of type theory because their canonical indexing is not a split fibration.

However, Bénabou had described a method of associating to any fibration an equivalent split one (i.e. a split fibration on the same base category, whose fibres are pointwise equivalent to those of the given one) [6]. Hofmann found an application of his work in the present context [11], and described an explicit way of associating a type-category $\widehat{\mathcal{C}}$ to any lccc $\mathcal{C}$, in such a way that $Type_{\widehat{\mathcal{C}}}(X)$ is equivalent to $\mathcal{C}/X$ for every object X in $\mathcal{C}$.

The underlying category of $\widehat{\mathcal{C}}$ is again $\mathcal{C}$. For an object X in $\mathcal{C}$, the collection $Type_{\widehat{\mathcal{C}}}(X)$ consists of those functors $F : \mathcal{C}/X \longrightarrow \mathcal{C}^{\to}$ which take any morphism in $\mathcal{C}/X$ to a pullback square, and such that the codomain of $F(f)$ is the domain of f for any object f in $\mathcal{C}/X$. Here, the notation $\mathcal{C}^{\to}$ indicates the category whose objects are arrows in $\mathcal{C}$ and morphisms are commuting squares.

Given an object X in $\mathcal{C}$ and a functor $F \in Type_{\widehat{\mathcal{C}}}(X)$, we define its canonical projection $p(F)$ to be $\pi_F = F(\mathsf{id}_X)$, which is a map over X. Its domain will be the total object $X \ltimes F$. Finally, given a morphism $f : Y \longrightarrow X$ in $\mathcal{C}$, the pullback of F along f is given by the functor f^*F which takes an object g over Y to the arrow $F(fg)$. The map $f \ltimes F : Y \ltimes f^*F \longrightarrow X \ltimes F$ is the top arrow in the following square

$$\begin{tikzcd} Y \ltimes f^*F \arrow[r, "f \ltimes F"] \arrow[d, "F(f)=\pi_{f^*F}"'] & X \ltimes F \arrow[d, "\pi_F = F(\mathsf{id}_X)"] \\ Y \arrow[r, "f"'] & X, \end{tikzcd}$$

which is a pullback because it is the action of the functor F on the arrow f in $\mathcal{C}/X$. It is not hard to show that these data satisfy the necessary coherence conditions, therefore they define a type-category, which has the universal property that $Type_{\widehat{\mathcal{C}}}(X)$ is equivalent to $\mathcal{C}/X$ for every object X in $\mathcal{C}$, one direction of the equivalence being the canonical projection functor $p : Type_{\widehat{\mathcal{C}}}(X) \longrightarrow \mathcal{C}/X$.

Furthermore, any extra structure on $\mathcal{C}$ induces some structure in $\widehat{\mathcal{C}}$, with the exception of impredicative universes, as Hofmann explains. In particular, the presence of left and right adjoints to the pullback functors in $\mathcal{C}$ ensures that $\widehat{\mathcal{C}}$ has dependent sums and products, respectively, whereas the existence of equalisers determines the existence of extensional equality types. For example, for elements A in $Type_{\widehat{\mathcal{C}}}(X)$ and B in $Type_{\widehat{\mathcal{C}}}(X \ltimes A)$, the total objects for $\Sigma(A, B)$ and $\Pi(A, B)$ are

$$\pi_{\Sigma(A,B)} = \Sigma_{\pi_B}\pi_A = X \ltimes A \ltimes B \xrightarrow{\pi_B} X \ltimes A \xrightarrow{\pi_A} X$$

and

$$\pi_{\Pi(A,B)} = \Pi_{\pi_A}\pi_B : X \ltimes \Pi(A, B) \longrightarrow X.$$

Having dependent products and coproducts, we can form, for any Y in $Type_{\widehat{\mathcal{C}}}(X)$ and A and B as above, the element

$$P_b(Y) = \Sigma(A, \Pi(B, \pi_B^* \pi_A^* Y))$$

which sits again in $Type_{\widehat{\mathcal{C}}}(X)$.

Lemma 3. *Given A and B as above, the association $Y \mapsto P_b(Y)$ defines a polynomial functor on $Type_{\widehat{\mathcal{C}}}(X)$.*

Proof. Let us first remind that, given a map $f : B \longrightarrow A$ in a locally cartesian closed category $\mathcal{C}$, the polynomial endofunctor on $\mathcal{C}$ associated to f is defined by

$$P(X) = \Sigma_A (A \times X \xrightarrow{\pi_A} A)^{(B \xrightarrow{f} A)},$$

where the exponential is taken in the slice category $\mathcal{C}/A$. We have already mentioned that the functor $p : Type_{\widehat{\mathcal{C}}}(X) \longrightarrow \mathcal{C}/X$ defines an equivalence. In particular, this means that $Type_{\widehat{\mathcal{C}}}(X)$ is locally cartesian closed; hence, it makes sense to talk about polynomial functors on it. Moreover, if we show that the canonical projection of $P_b(Y)$ is a polynomial expression over π_Y, the same will hold in $Type_{\widehat{\mathcal{C}}}(X)$, because of the equivalence, and the result will be proved.

Using the descriptions above for dependent products and coproducts, we get the following chain of equalities in $\mathcal{C}/X$:

$$\begin{aligned} p(\Sigma(A, \Pi(B, \pi_B^*\pi_A^*Y))) &= \Sigma_{\pi_A}(p(\Pi(B, \pi_B^*\pi_A^*Y))) \\ &= \Sigma_{\pi_A}\Pi_{\pi_B}(\pi_B^*\pi_A^*p(Y)) \\ &= \Sigma_{\pi_A}((\pi_A^*p(Y))^{\pi_B}). \end{aligned}$$

Now, note that π_A is the unique map from π_A to the terminal object id_X in $\mathcal{C}/X$, and $\pi_A^*p(Y)$ is the first projection from $\pi_A \times p(Y)$; hence, we can rewrite $p(\Sigma(A, \Pi(B, \pi_B^*\pi_A^*Y)))$ as

$$\Sigma_{\pi_A}((\pi_A \times p(Y) \to \pi_A)^{\pi_B}) = P_{\pi_B}(p(Y)).$$

Functoriality of this expression in Y is an easy check, which closes the proof. □

Using the previous result, we can now deduce the existence of coinductive types in the type category associated to an lccc with *M-types* (i.e. final coalgebras of polynomial functors, [23]).

Theorem 4. *The type category $\widehat{\mathcal{C}}$ associated to a locally cartesian closed category $\mathcal{C}$ with M-types has coinductive types.*

Proof. Let $A \in Type_{\widehat{\mathcal{C}}}(X)$ and $B \in Type_{\widehat{\mathcal{C}}}(X \ltimes A)$. Then, by Lemma 3, the mapping

$$Y \mapsto P_b(Y)$$

defines a polynomial functor over $Type_{\widehat{\mathcal{C}}}(X) \simeq \mathcal{C}/X$. By assumption, this functor has a final coalgebra

$$m : M(A, B) \longrightarrow \Sigma(A, B \to \pi_A^*(M(A, B))), \tag{2}$$

whose domain is to interpret our coinductive type. In fact, here we are implicitly using the fact that locally cartesian closed pretoposes with M-types are closed under slicing, as proved in [23].

In order to give a concrete description of the functor $M(A, B) : \mathcal{C}/X \longrightarrow \mathcal{C}^{\to}$, we can reason as follows. For the functor to satisfy the coherence condition of Definition 1, it is necessary that the equation $f^*M(A, B) = M(f^*A, f^*B)$ holds for any $f : X' \longrightarrow X$ in $\mathcal{C}$. In particular, this means that, for an arrow $g : Y \longrightarrow X'$, one must have

$$M(A, B)(f \circ g) = M(f^*A, f^*B)(g)$$

and choosing $g = \mathsf{id}_{X'}$ we get that $M(A, B)(f) = p(M(f^*A, f^*B))$ for any f in $\mathcal{C}/X$. Therefore, in order to describe $M(A, B)$, we first make a choice of final coalgebras for the functor $P_{\pi_{f^*B}}$ in each slice category $\mathcal{C}/X'$ (for $f : X' \longrightarrow X$), and then define $M(A,B)(f)$ to be the carrier of the chosen coalgebra for $P_{\pi_{f^*B}}$. The coherence condition is then automatically fulfilled.

Given sections $y : X \longrightarrow X \ltimes Y$ of π_Y and $g : X \longrightarrow X \ltimes (Y \to P_b(Y))$, g determines a map $\widetilde{g}$ over X:

$$\begin{tikzcd} X \ltimes Y \arrow[rr, "\widetilde{g}"] \arrow[dr, bend right, "\pi_Y"'] & & X \ltimes P_b(Y) \arrow[dl, bend left, "\pi_{P_b(Y)}"] \\ & X. & \end{tikzcd}$$

This is a P_b-coalgebra in $Type_{\widehat{\mathcal{C}}}(X)$, hence there is a unique coalgebra morphism

$$\begin{tikzcd} X \ltimes Y \arrow[rr, "\overline{g}"] \arrow[dr, bend right, "\pi_Y"'] & & X \ltimes M(A,B) \arrow[dl, bend left, "\pi_{M(A,B)}"] \\ & X. & \end{tikzcd}$$

Post-composition with $\overline{g}$ takes y to a section of $\pi_{M(A,B)}$:

$$unfold(g, y) = \overline{g}y.$$

Given a section t of $\pi_{M(A,B)}$, post-composition with the map m of (2) gives a section of $p(P_b(M(A,B))) = \Sigma_{\pi_A} p(B \to \pi_A^* M(A,B))$. Projection on A then determines a section

$$root(t) = (mt).1 \quad \text{of} \quad \pi_A,$$

whereas the second projection is a section

$$br(t) = (mt).2 \quad \text{of} \quad \pi_{root(t)^*B \to M(A,B)}.$$

The various equations of Definition 1 are obviously satisfied by the data we have just defined, because of the finality of $M(A,B)$. □

Remark. The choice of a collection of final coalgebras made in the proof is needed in order to ensure coherence of coinductive types under pullback. The situation is analogous to that of Hofmann [11], where he needs a choice of pullbacks and equalisers in order to describe the identity types.

5 Conclusions

We have extended the extensional type theory of Martin-Löf with Σ-types and Π-types, by adding coinductive types. These allow for the specification of infinite programs, and for the study of non-well-founded structures.

A sound categorical semantics has been given, in terms of type-categories with coinductive types, which extends the well-known semantics presented in [19,11]. We also prove that any locally cartesian closed category with M-types gives rise to a type-category with coinductive types. This formalised the statement in [23] saying that coinductive pretoposes give a semantics of type theories with coinductive types.

From a computational point of view, the choice of working with extensional equality (as well as the axioms we adopted for M-types in the type theory) are rather unpleasant, since they make equality undecidable. We heard from Altenkirch that the so-called "observational types" he introduced in [4] might help handling infinite objects in an intensional setting, and that some research in this direction has been undertaken.

In [23], the structure of a coinductive pretopos is studied, and that is not only a locally cartesian closed category with M-types, but also exact (and with distributive sums). These further properties have been ignored in the present setting. They might be of use in giving a semantics to the study of setoids built out of a type theory with conductive types, and this topic is clearly related to the study of non-well-founded set theory [2].

References

1. Michael Abbott, Thorsten Altenkirch, and Neil Ghani. Representing strictly positive types. Presented at APPSEM annual meeting, invited for submission to Theoretical Computer Science, 2004.
2. Peter Aczel. *Non-Well-Founded Sets*. Center for the Study of Language and Information, Stanford University, 1988. CSLI Lecture Notes, Volume 14.
3. Peter Aczel, Jiří Adámek, Stefan Milius, and Jiří Velebil. Infinite trees and completely iterative theories: a coalgebraic view. *Theoretical Computer Science*, 300:1–45, 2003.
4. T. Altenkirch. Extensional equality in intensional type theory. In *14th Symposium on Logic in Computer Science (LICS'99)*, pages 412–421. IEEE, 1999.
5. Michael Barr. Terminal coalgebras for endofunctors on sets. Available from `ftp://www.math.mcgill.ca/pub/barr/trmclgps.zip`, 1999.
6. Jean Bénabou. Fibered categories and the foundations of naive category theory. *J. Symbolic Logic*, 50(1):10–37, 1985.
7. John Cartmell. Generalised algebraic theories and contextual categories. *Ann. Pure Appl. Logic*, 32(3):209–243, 1986.
8. Thierry Coquand. Infinite objects in type theory. In *Types for proofs and programs (Nijmegen, 1993)*, volume 806 of *Lecture Notes in Comput. Sci.*, pages 62–78. Springer, 1994.
9. Veronica Gaspes. Infinite objects in type theory, 1997.
10. Lars Hallnäs. On the syntax of infinite objects: an extension of Martin-Löf's theory of expressions. In *COLOG-88 (Tallinn, 1988)*, volume 417 of *Lecture Notes in Comput. Sci.*, pages 94–104. Springer, 1990.
11. Martin 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. Springer, 1995.
12. Bart Jacobs. *Categorical logic and type theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1999.

13. Bart Jacobs and Jan Rutten. A tutorial on (co)algebras and (co)induction. *Bulletin of the EATCS*, 62:222–259, 1996.
14. Ingrid Lindström. A construction of non-well-founded sets within Martin-Löf's type theory. *Journal of Symbolic Logic*, 54(1):57–64, 1989.
15. Per Martin-Löf. *Intuitionistic type theory*, volume 1 of *Studies in Proof Theory. Lecture Notes*. Bibliopolis, Naples, 1984.
16. Per Martin-Löf. Mathematics of infinity. In *COLOG-88 (Tallinn, 1988)*, volume 417 of *Lecture Notes in Comput. Sci.*, pages 146–197. Springer, 1990.
17. N. P. Mendler, P. Panangaden, and R. L. Constable. Infinite objects in type theory. In *Symposium on Logic in Computer Science (LICS '86)*, pages 249–257. IEEE Computer Society Press, 1986.
18. Bengt Nordström, Kent Petersson, and Jan M. Smith. *Programming in Martin-Löf's type theory*, volume 7 of *International Series of Monographs on Computer Science*. The Clarendon Press Oxford University Press, 1990.
19. Andrew M. Pitts. Categorical logic. In *Handbook of logic in computer science, Vol. 5*, Oxford Sci. Publ., pages 39–128. Oxford Univ. Press, 2000.
20. R. A. G. Seely. Locally cartesian closed categories and type theory. *Math. Proc. Cambridge Philos. Soc.*, 95(1):33–48, 1984.
21. Thomas Streicher. *Semantics of type theory*. Progress in Theoretical Computer Science. Birkhäuser, 1991. Correctness, completeness and independence results.
22. Daniele Turi and Jan Rutten. On the foundations of final coalgebra semantics: non-well-founded sets, partial orders, metric spaces. *Mathematical Structures in Computer Science*, 8(5):481–540, October 1998.
23. Benno van den Berg and Federico De Marchi. Non-well-founded trees in categories. (submitted). Available online at `http://arxiv.org/abs/math.CT/0409158`.