Wellfounded Trees and Dependent Polynomial Functors

Nicola Gambino* and Martin Hyland

Department of Pure Mathematics and Mathematical Statistics
University of Cambridge
{N.Gambino,M.Hyland}@dpmms.cam.ac.uk

Abstract. We set out to study the consequences of the assumption of types of wellfounded trees in dependent type theories. We do so by investigating the categorical notion of wellfounded tree introduced in [15]. Our main result shows that wellfounded trees allow us to define initial algebras for a wide class of endofunctors on locally cartesian closed categories.

1 Introduction

Types of wellfounded trees, or $\mathcal{W}$-types, are one of the most important components of Martin-Löf's dependent type theories. First, they allow us to define a wide class of inductive types [14]. Secondly, they play an essential role in the interpretation of constructive set theories in dependent type theories [3]. Finally, from the proof-theoretic point of view, they represent the paradigmatic example of a generalised inductive definition and contribute considerably to the proof-theoretic strength of dependent type theories [9].

In [15] a categorical counterpart of the notion of $\mathcal{W}$-type was introduced. In a locally cartesian closed category, $\mathcal{W}$-types are defined as the initial algebras for endofunctors of a special kind, the *polynomial functors*. The purpose of this paper is to study polynomial endofunctors and $\mathcal{W}$-types more closely. In particular, we set out to explore some of the consequences of the assumption that a locally cartesian closed category has $\mathcal{W}$-types, i.e. that every polynomial endofunctor has an initial algebra. We introduce *dependent polynomial functors*, that generalize polynomial functors, to explore these consequences.

Our main theorem then shows that the assumption of $\mathcal{W}$-types is sufficient to define explicitly initial algebras for dependent polynomial functors. We expect this result to lead to further insight into the interplay between dependent type theory and the theory of inductive definitions. In this paper, we will limit ourselves to giving only two applications of our main theorem. First, we show how the class of polynomial functors is closed under fixpoints. We haste to point out that related results appeared in [1, 2]. One of our original goals was indeed to put those results in a more general context and simplify their proofs.

* EPSRC Postdoctoral Research Fellow in Mathematics.

Secondly, we show how polynomial functors have free monads, and these free monads are themselves polynomial. The combination of these two facts leads to further observations concerning the categories of algebras of polynomial endofunctors. These results are particularly important for our ongoing research on 2-categorical models of the differential λ-calculus [6].

The interplay between dependent type theories and categories is here exploited twice. On the one hand, category theory provides a mathematically efficient setting to present results that apply not only to the categories arising from the syntax of dependent type theories, but also to the categories providing their models. On the other hand, dependent type theories provide a convenient language to manipulate and describe the objects and the arrows of locally cartesian closed categories via the internal language of such a category [16].

In order to set up the internal language for a locally cartesian closed category with $\mathcal{W}$-types, it is necessary to establish some technical results that ensure a correct interaction between the structural rules of the internal language and the rules for $\mathcal{W}$-types. Although these results are already contained in [15] we give new and simpler proofs of some of them. Once this is achieved, we can freely exploit the internal language to prove the consequences of the assumption of $\mathcal{W}$-types in a category.

Acknowledgements. We thank Thorsten Altenkirch, Marcelo Fiore and Erik Palmgren for stimulating discussions.

2 Polynomial Functors

2.1 Locally Cartesian Closed Categories

We say that a category $\mathcal{C}$ is a *locally cartesian closed category*, or a lccc for short, if for every object I of $\mathcal{C}$ the slice category $\mathcal{C}/I$ is cartesian closed[1]. Note that if $\mathcal{C}$ is a lccc then so are all its slices: for an object A in $\mathcal{C}$ and an object $f : B \to A$ in $\mathcal{C}/A$, there is indeed an isomorphism of categories $(\mathcal{C}/A)/f \cong \mathcal{C}/B$.

For an arrow $f : B \to A$ in a lccc $\mathcal{C}$ we write $\Delta_f : \mathcal{C}/A \to \mathcal{C}/B$ for the pullback functor. The key fact about locally cartesian closed categories is the following proposition [7].

Proposition 1. *Let $\mathcal{C}$ be a lccc. For any arrow $f : B \to A$ in $\mathcal{C}$, the pullback functor $\Delta_f : \mathcal{C}/A \to \mathcal{C}/B$ has both a left and a right adjoint.*

Given an arrow $f : B \to A$ in a lccc $\mathcal{C}$, we will write $\Sigma_f : \mathcal{C}/B \to \mathcal{C}/A$ and $\Pi_f : \mathcal{C}/B \to \mathcal{C}/A$ for the left and right adjoint to the pullback functor, respectively. We indicate the existing adjunctions as $\Sigma_f \dashv \Delta_f \dashv \Pi_f$.

[1] Here and in the following, when we require the existence of some structure in a category, we always mean that this structure is given to us by an explicitly defined operation.

An abuse of language. For an arrow $f : B \to A$, we write the image of $X \to A$ in $\mathcal{C}/A$ under Δ_f as $\Delta_f(X) \to B$: the arrows fit into the following pullback diagram

$$\begin{array}{ccc} \Delta_f X & \longrightarrow & X \\ \downarrow & & \downarrow \\ B & \xrightarrow{f} & A \end{array}$$

The Beck-Chevalley condition. The Beck-Chevalley condition, which holds in any lccc, expresses categorically that substitution behaves correctly with respect to type-formation rules. More precisely, it asserts that for a pullback diagram of the form

$$\begin{array}{ccc} D & \xrightarrow{k} & B \\ \downarrow{\scriptstyle g} & & \downarrow{\scriptstyle f} \\ C & \xrightarrow{h} & A \end{array}$$

the canonical natural transformations

$$\Sigma_g \; \Delta_k \Rightarrow \Delta_h \; \Sigma_f \,, \tag{1}$$
$$\Delta_h \; \Pi_f \Rightarrow \Pi_g \; \Delta_k \tag{2}$$

are part of natural isomorphisms. For details see [8].

The axiom of choice. The type-theoretic axiom of choice [14] is expressed by the fact that, for two arrows $g : C \to B$ and $f : B \to A$, the canonical natural transformation

$$\Sigma_h \; \Pi_p \; \Delta_{\varepsilon_C} \Rightarrow \Pi_f \; \Sigma_g \,,$$

is part of a natural isomorphism, where

$$\begin{array}{ccc} \Delta_f \Pi_f C & \xrightarrow{p} & \Pi_f C \\ {\scriptstyle q}\downarrow & & \downarrow{\scriptstyle h} \\ B & \xrightarrow{f} & A \end{array}$$

is a pullback diagram and $\varepsilon_C : \Delta_f \; \Pi_f \, C \to C$ is a component of the counit of the adjunction $\Delta_f \dashv \Pi_f$.

2.2 Internal Language

Associated to a lccc $\mathcal{C}$ there is a dependent type theory $\mathsf{Th}(\mathcal{C})$ to which we shall refer as the *internal language* of $\mathcal{C}$. A complete presentation of such a dependent

type theory can be found in [16]. See also [13] for more information. We limit ourselves to recalling only those aspects that are most relevant for the remainder of this work. We use the judgement forms

$$(B_a \mid a \in A)\,,\ (B_a = B'_a \mid a \in A)\,,\ (b_a \in B_a \mid a \in A)\,,\ (b_a = b'_a \in B_a \mid a \in A)$$

to express, for $a \in A$, that B_a is a type, that B_a and B'_a are equal types, that b is a term of type B_a, and that b_a and b'_a are equal terms of type B_a, respectively. The dependent type theory $\mathsf{Th}(\mathcal{C})$ has the following primitive forms of type:

$$1\,,\quad Id_A(a,a')\,,\quad \sum_{a\in A} B_a\,,\quad \prod_{a\in A} B_a\ .$$

We refer to these as the unit, identity, dependent sum and dependent product types, respectively. As usual, these primitive forms of type allow us to define the forms of type $A \times B$ and B^A, to which we refer as the product and function types. The dependent type theory $\mathsf{Th}(\mathcal{C})$ has a straightforward interpretation in $\mathcal{C}$ and thus provides a convenient language to define objects and arrows in $\mathcal{C}$. An example: to define an arrow $g : C \to B$ in $\mathcal{C}/A$ with domain $C \to A$ and codomain $B \to A$ in $\mathcal{C}/A$ it is sufficient to derive in $\mathsf{Th}(\mathcal{C})$ a judgement of form

$$(\, g_a(c) \in B_a \mid a \in A\,,\ c \in C_a\,)\,.$$

The required arrow is then defined as the interpretation of this judgement in $\mathcal{C}$.

2.3 Polynomial Functors

Let $\mathcal{C}$ be a lccc. For an object X we write X also for the unique arrow $X \to 1$ into the terminal object 1 of $\mathcal{C}$. Observe that arrows $B : B \to 1$ and $A : A \to 1$ determine functors $\Delta_B : \mathcal{C} \to \mathcal{C}/B$ and $\Sigma_A : \mathcal{C}/A \to \mathcal{C}$. We are now ready to introduce polynomial functors. For an arrow $f : B \to A$ in $\mathcal{C}$ we define an endofunctor $\mathcal{P}_f : \mathcal{C} \to \mathcal{C}$, called the *polynomial functor associated to* f, as the composite

$$\mathcal{C} \xrightarrow{\Delta_B} \mathcal{C}/B \xrightarrow{\Pi_f} \mathcal{C}/A \xrightarrow{\Sigma_A} \mathcal{C}\,. \tag{3}$$

Definition 2. We say that an endofunctor on $\mathcal{C}$ is *polynomial* if it is naturally isomorphic to a functor $\mathcal{P}_f : \mathcal{C} \to \mathcal{C}$ explicitly defined as

$$\mathcal{P}_f =_{\mathrm{df}} \Sigma_A\, \Pi_f\, \Delta_B$$

for some arrow $f : B \to A$ of $\mathcal{C}$.

Let us look more closely at the definition of polynomial endofunctors. Given an arrow $f : B \to A$, let us consider $\Sigma_A : \mathcal{C}/A \to \mathcal{C}$ and $\Delta_B : \mathcal{C} \to \mathcal{C}/B$. The functor Δ_B takes an object X of $\mathcal{C}$ to the left side of the pullback diagram:

$$\begin{array}{ccc} X \times B & \longrightarrow & X \\ \downarrow & & \downarrow{\scriptstyle X} \\ B & \xrightarrow{\ B\ } & 1 \end{array}$$

We can therefore write $\Delta_B X = X \times B$. The action of Σ_A is very simple: given an object $Y \to A$ of $\mathcal{C}/A$ let $\Sigma_A(Y \to A) =_{\mathrm{df}} Y$. These observations lead to a description of polynomial functors in the internal language, which we shall exploit. The object $f : B \to A$ of $\mathcal{C}/A$ determines the judgement $(B_a \mid a \in A)$ of $\mathsf{Th}(\mathcal{C})$. We can then explicitly define in $\mathsf{Th}(\mathcal{C})$

$$\mathcal{P}_f(X) =_{\mathrm{df}} \sum_{a \in A} X^{B_a},$$

for a type X. The interpretation in $\mathcal{C}$ of the right-hand side of the definition is indeed $\mathcal{P}_f(X)$, as defined in (3).

2.4 Basic Properties of Polynomial Functors

Proposition 3. *The composition of two polynomial functors is polynomial.*

Proof. See [2] for a proof using the internal language and [8] for a proof using diagrammatic reasoning. Note that the proof uses crucially the axiom of choice.

We now assume that the lccc $\mathcal{C}$ has finite coproducts. As pullback functors are left adjoints, finite coproducts are preserved under pullbacks and so in particular disjoint. Hence they can be represented in a familiar way in the internal language $\mathsf{Th}(\mathcal{C})$, which now has also the primitive forms of type 0 and $A + B$ called the empty and disjoint sum types, respectively.

The class of polynomial functors is closed under a further operation, that will be very important in the following. Recall from Appendix A that for an endofunctor $P : \mathcal{C} \to \mathcal{C}$ and an object X of $\mathcal{C}$ there is a functor $P_X : \mathcal{C} \to \mathcal{C}$ whose action on an object Y is defined as $P_X(Y) =_{\mathrm{df}} X + P(Y)$.

Proposition 4. *Let $P : \mathcal{C} \to \mathcal{C}$ be a functor and X be an object of $\mathcal{C}$. If P is polynomial then so is P_X.*

Proof. We give a proof using the internal language. Let $f : B \to A$ and consider the polynomial functor $\mathcal{P}_f$ associated to f. For X and Y in $\mathcal{C}$ we then have the following chain of equalities and isomorphisms:

$$X + \mathcal{P}_f(Y) = X + \sum_{a \in A} Y^{B_a} \cong \sum_{z \in X + A} Y^{B_z}$$

where $(B_z \mid z \in X + A)$ is defined so that the judgements $(B_{\iota_1(x)} = 0 \mid x \in X)$ and $(B_{\iota_2(a)} = B_a \mid a \in A)$ are derivable.

The notion of strength for a functor is recalled in Definition A.7 in Appendix A.

Proposition 5. *Every polynomial functor has a strength.*

Proof. Let us use the internal language to define the arrow

$$\sigma_{X,Y} : X \times \mathcal{P}_f Y \to \mathcal{P}_f(X \times Y)$$

which gives us one of the components of the required strength σ for a polynomial functor $\mathcal{P}_f$. First, observe that the domain and the codomain of $\sigma_{X,Y}$ can be described in $\mathsf{Th}(\mathcal{C})$ as $X \times \sum_{a \in A} Y^{B_a}$ and $\sum_{a \in A}(X \times Y)^{B_a}$ respectively. We can then define $\sigma_{X,Y}$ by letting

$$\sigma_{X,Y}(x, a, t) =_{\mathrm{df}} (a, (\lambda b \in B_a)(x, t(b))$$

for $(x, a, t) \in X \times \sum_{a \in A} Y^{B_a}$.

3 Change of Base

In the following, we shall be interested in the effect that pullback functors have on the algebras of polynomial endofunctors. More precisely, for an arrow $u : I \to J$ in $\mathcal{C}$, we will show that the algebras for the polynomial functor $\mathcal{P}_f$ on $\mathcal{C}/J$ associated to an arrow f of $\mathcal{C}/J$ can be mapped functorially into algebras for the polynomial endofunctor $\mathcal{P}_{\Delta_u(f)}$ on $\mathcal{C}/I$ associated to the arrow $\Delta_u(f)$ of $\mathcal{C}/I$. This fact is a purely formal consequence of some general observations concerning the 2-category of polynomial functors, that we define below. The treatment is inspired by the formal theory of monads [10, 17].

Note that without loss of generality we can assume that J is the terminal object of $\mathcal{C}$ and thus consider only the pullback functors $\Delta_I : \mathcal{C} \to \mathcal{C}/I$ determined by arrows $I : I \to 1$.

3.1 The 2-category of Polynomial Functors

Let us define the 2-category Poly . An object of Poly is a pair $(\mathcal{C}, \mathcal{P}_f)$ where $\mathcal{C}$ is a lccc and $\mathcal{P}_f$ is the polynomial endofunctor on $\mathcal{C}$ associated to an arrow $f : B \to A$ in $\mathcal{C}$. A 1-cell with domain $(\mathcal{C}, \mathcal{P}_f)$ and codomain $(\mathcal{D}, \mathcal{P}_g)$ is given by a pair (F, ϕ) where $F : \mathcal{C} \to \mathcal{D}$ is a functor and $\phi : \mathcal{P}_g F \Rightarrow F \mathcal{P}_f$ is a natural transformation. If (F, ϕ) and (G, ψ) are 1-cells, then a 2-cell with source (F, ϕ) and target (G, ψ) is a natural transformation $\sigma : F \Rightarrow G$ together with a commuting diagram

$$\begin{array}{ccc} \mathcal{P}_g\, F & \xrightarrow{\mathcal{P}_g\, \sigma} & \mathcal{P}_g\, G \\ {\scriptstyle \phi}\downarrow & & \downarrow{\scriptstyle \psi} \\ F\, \mathcal{P}_f & \xrightarrow{\sigma\, \mathcal{P}_f} & G\, \mathcal{P}_f \end{array}$$

For our purposes, it will be sufficient to recall the definition of the action of the 2-functor $\mathsf{Alg} : \mathsf{Poly} \to \mathsf{Cat}$ on objects and 1-cells. For an object $(\mathcal{C}, \mathcal{P}_f)$ of Poly we define

$$\mathsf{Alg}(\mathcal{C}, \mathcal{P}_f) =_{\mathrm{df}} \mathcal{P}_f\text{-}alg\,,$$

where $\mathcal{P}_f$-*alg* is the category of $\mathcal{P}_f$-algebras, as defined in Appendix A. Given a 1-cell $(F, \phi) : (\mathcal{C}, \mathcal{P}_f) \to (\mathcal{D}, \mathcal{P}_g)$ in Poly, the functor $\mathsf{Alg}(F, \phi)$ is defined by mapping a $\mathcal{P}_f$-algebra $x : \mathcal{P}_f\, X \to X$ into the $\mathcal{P}_g$-algebra

$$\mathcal{P}_g\, F\, X \xrightarrow{\phi_X} F\, \mathcal{P}_f\, X \xrightarrow{Fx} F\, X\,.$$

Observe that if $\mathcal{C}$ and $\mathcal{D}$ are lccc's and $F : \mathcal{C} \to \mathcal{D}$ is a locally cartesian closed functor, i.e. a functor preserving the locally cartesian closed structure up to isomorphism, then there is an obvious 1-cell $\chi_F : (\mathcal{C}, \mathcal{P}_f) \to (\mathcal{D}, \mathcal{P}_{Ff})$. The next lemma is a simple but very useful fact, whose proof follows by a direct calculation.

Lemma 6. *Let $\mathcal{C}$ and $\mathcal{D}$ be lccc's, and let $F : \mathcal{C} \to \mathcal{D}$ be a locally cartesian closed functor. For any arrow $f : B \to A$ in $\mathcal{C}$ the diagram*

$$\begin{array}{ccc} \mathcal{P}_f\text{-}alg & \longrightarrow & \mathcal{C} \\ {\scriptstyle \mathsf{Alg}(F,\chi_F)}\downarrow & & \downarrow{\scriptstyle F} \\ \mathcal{P}_{Ff}\text{-}alg & \longrightarrow & \mathcal{D} \end{array}$$

where the horizontal arrows are the obvious forgetful functors, commutes.

3.2 Pullback of Algebras

Let $\mathcal{C}$ be a lccc and let I be an object of $\mathcal{C}$. Recalling from [7] that the pullback functor $\Delta_I : \mathcal{C} \to \mathcal{C}/I$ is locally cartesian closed, we can apply Lemma 6 and obtain, for any arrow $f : B \to A$ in $\mathcal{C}$, a commuting diagram of form

$$\begin{array}{ccc} \mathcal{P}_f\text{-}alg & \longrightarrow & \mathcal{C} \\ {\scriptstyle F_I}\downarrow & & \downarrow{\scriptstyle \Delta_I} \\ \mathcal{P}_g\text{-}alg & \longrightarrow & \mathcal{C}/I \end{array}$$

where $g =_{\mathrm{df}} \Delta_I(f)$ and $F_I =_{\mathrm{df}} \mathsf{Alg}(\Delta_I, \chi_{\Delta_I})$. Let us describe this diagram more explicitly. First, observe that g fits into the pullback diagram

$$\begin{array}{ccc} B \times I & \longrightarrow & B \\ {\scriptstyle g}\downarrow & & \downarrow{\scriptstyle f} \\ A \times I & \longrightarrow & A \end{array}$$

and hence $g = f \times 1_I$. The action of the functor F_I on objects can now be easily described. A $\mathcal{P}_f$-algebra in $\mathcal{C}$

$$\mathcal{P}_f X \xrightarrow{x} X \tag{4}$$

is mapped into the $\mathcal{P}_g$-algebra in $\mathcal{C}/I$ given by the arrow in $\mathcal{C}/I$

$$\mathcal{P}_g\ \Delta_I\, X \xrightarrow{\ \phi_X\ } \Delta_I\ \mathcal{P}_f\, X \xrightarrow{\ \Delta_I(x)\ } \Delta_I\, X\,, \tag{5}$$

where ϕ is part of the isomorphism described in the next proposition.

Proposition 7. *Let $\mathcal{C}$ be a lccc and let I be an object of $\mathcal{C}$. For any arrow $f : B \to A$ in $\mathcal{C}$ there is an isomorphism $\Delta_I\ \mathcal{P}_f \cong \mathcal{P}_g\ \Delta_I$, where $g =_{\mathrm{df}} \Delta_I f$.*

Proof. Observe that we have the following chain of equalities and isomorphisms:

$$\begin{aligned}
\Delta_I\ \mathcal{P}_f &= \Delta_I\ \Sigma_A\ \Pi_f\ \Delta_B && \text{by definition of } \mathcal{P}_f\,,\\
&\cong \Sigma_{\Delta_I A}\ \Pi_{\Delta_I(f)}\ \Delta_{\Delta_I B}\ \Delta_I && \text{since } \Delta_I \text{ is cartesian closed},\\
&= \Sigma_{A\times I}\ \Pi_g\ \Delta_{B\times I}\ \Delta_I && \text{by definition of } \Delta_I \text{ and of } g\,,\\
&= \mathcal{P}_g\ \Delta_I && \text{by definition of } \mathcal{P}_g\,,
\end{aligned}$$

which proves the claim.

We can also describe the action of F_I in the internal language of $\mathcal{C}$. First of all observe that for $((a,i),t) \in \sum_{(a,i)\in A\times I} X^{B_a}$ we can define

$$\phi_X((a,i),t) =_{\mathrm{df}} ((a,t),i) \in \sum_{a\in A} X^{B_a} \times I\,.$$

An algebra as in (4) determines a judgement

$$\big(\ x(a,t) \in X \mid (a,t) \in \sum_{a\in A} X^{B_a}\ \big)\,.$$

and therefore, for $((a,i),t) \in \sum_{(a,i)\in A\times I} X^{B_a}$, we can define

$$\Delta_I(x)((a,t),i) =_{\mathrm{df}} (x(a,t),i) \in X \times I\,.$$

It is now simple to observe that the derivable judgements

$$\big(\ \phi_X((a,i),t) \in \sum_{a\in A} X^{B_a} \times I \quad\mid\quad ((a,i),t) \in \sum_{(a,i)\in A\times I} X^{B_a}\ \big)$$

and

$$\big(\ \Delta_I(x)((a,t),i) \in X \times I \mid ((a,t),i) \in \sum_{a\in A} X^{B_a} \times I\ \big)\,.$$

express the arrows in (5), as required.

4 Wellfounded Trees

Definition 8. We say that a lccc $\mathcal{C}$ *has* $\mathcal{W}$*-types* if for every arrow $f : B \to A$ in $\mathcal{C}$ there is a diagram $\mathcal{P}_f(\mathcal{W}_f) \to \mathcal{W}_f$ which is an initial algebra for $\mathcal{P}_f : \mathcal{C} \to \mathcal{C}$.

Recall that, by a theorem of Lambek, the arrow $\mathcal{P}_f(\mathcal{W}_f) \to \mathcal{W}_f$ is an isomorphism. Once the internal language of a lccc with $\mathcal{W}$-types is set up, we will therefore be allowed to write

$$W \cong \sum_{a \in A} W^{B_a}$$

where $f : B \to A$ and $W =_{\mathrm{df}} \mathcal{W}_f$. The next subsection is devoted to justify the use of the internal language in connection to $\mathcal{W}$-types.

4.1 Pullback of Wellfounded Trees

In [15] it is proved that if $\mathcal{C}$ has $\mathcal{W}$-types then so do all its slices. A proof of this fact can be obtained by defining explicitly initial algebras for polynomial endofunctors on the slice categories. It is also observed there that the pullback functors preserve $\mathcal{W}$-types. Although in [15] it is suggested to prove this second fact using the explicit definition of $\mathcal{W}$-types in slice categories, we now show that it is possible to give a proof that does not involve any reference to the explicit definition of $\mathcal{W}$-types in slice categories.

Let $\mathcal{C}$ be a lccc and let I be an object in $\mathcal{C}$. Recall from Proposition 7 that for an arrow $f : B \to A$ there is a natural isomorphism $\Delta_I \; \mathcal{P}_f \cong \mathcal{P}_g \; \Delta_I$, where $g =_{\mathrm{df}} \Delta_I(f)$. In Subsection 3.2 we used one part of this isomorphism, namely the natural transformation $\phi : \mathcal{P}_g \; \Delta_I \Rightarrow \Delta_I \; \mathcal{P}_f$, to define a functor $F_I : \mathcal{P}_f\text{-}alg \to \mathcal{P}_g\text{-}alg$. We now use the other part of the isomorphism, namely the natural transformation $\psi : \Delta_I \; \mathcal{P}_f \Rightarrow \mathcal{P}_g \; \Delta_I$ that is inverse to ϕ, to define a functor $G_I : \mathcal{P}_g\text{-}alg \to \mathcal{P}_f\text{-}alg$ that is right adjoint to F_I. First of all, observe that ψ gives us a natural transformation $\xi : \mathcal{P}_f \; \Pi_I \Rightarrow \Pi_I \; \mathcal{P}_g$ that is defined as the composite

$$\mathcal{P}_f \; \Pi_I \xRightarrow{\eta} \Pi_I \; \Delta_I \; \mathcal{P}_f \; \Pi_I \xRightarrow{\Pi_I \, \psi \, \Pi_I} \Pi_I \; \mathcal{P}_g \; \Delta_I \; \Pi_I \xRightarrow{\Pi_I \, \mathcal{P}_g \, \varepsilon} \Pi_I \; \mathcal{P}_g$$

where η and ε are the unit and the counit of the adjunction $\Delta_I \dashv \Pi_I$, respectively. Hence we have that $(\Pi_I, \xi) : (\mathcal{C}/I, \mathcal{P}_g) \to (\mathcal{C}, \mathcal{P}_f)$ is a 1-cell in Poly and thus we can simply define

$$G_I =_{\mathrm{df}} \mathsf{Alg}(\Pi_I, \xi) \, .$$

Like we did for the functor F_I in Subsection 3.2, we can describe G_I in the internal language of $\mathcal{C}$. Let us consider a $\mathcal{P}_g$-algebra, i.e. an arrow $z : \mathcal{P}_g Z \to Z$ in $\mathcal{C}/I$. This arrow determines the judgement

$$\Big(\; z(i,(a,s)) \in Z_i \mid i \in I \, , (a,s) \in \sum_{a \in A} {Z_i}^{B_a} \; \Big)$$

where $(Z_i \mid i \in I)$ is the judgement associated to the object $Z \to I$ of $\mathcal{C}/I$. We can then derive the judgement

$$\big((\lambda i \in I)\, z(i, (a, (\lambda b \in B_a)\, t(b, i))) \in \prod_{i \in I} Z_i \mid (a, t) \in \sum_{a \in A} \prod_{i \in I} {Z_i}^{B_a} \big)$$

which gives us a $\mathcal{P}_f$-algebra $\mathcal{P}_f \Pi_I Z \to \Pi_I Z$. This is exactly the image under G_I of the $\mathcal{P}_f$-algebra $\mathcal{P}_g Z \to Z$. The following result is the key component to prove the desired pullback stability property. A proof can be easily be obtained either reasoning with diagrams or with the internal language.

Theorem 9. *Let $\mathcal{C}$ be a lccc and let $f : B \to A$ be an arrow in $\mathcal{C}$. For any object I of $\mathcal{C}$ the adjunction $\Delta_I \dashv \Pi_I$ lifts to an adjunction $F_I \dashv G_I$, i.e. in the diagram*

$$\begin{array}{ccc} \mathcal{P}_g\text{-}alg & \longrightarrow & \mathcal{C}/I \\ {\scriptstyle F_I}\uparrow\downarrow{\scriptstyle G_I} & & {\scriptstyle \Delta_I}\uparrow\downarrow{\scriptstyle \Pi_I} \\ \mathcal{P}_f\text{-}alg & \longrightarrow & \mathcal{C} \end{array}$$

where $g =_{\mathrm{df}} \Delta_I(f)$, the inner and outer squares commute.

Corollary 10. *Let $f : B \to A$ be an arrow of $\mathcal{C}$. For any object I there is an isomorphisms $\mathcal{W}_{\Delta_I} \cong \Delta_I \mathcal{W}_f$.*

Proof. The functor F_I is a left adjoint and hence preserves initial objects.

4.2 Free Monads for Polynomial Functors

We begin exploring the consequences of the assumption that a lccc has $\mathcal{W}$-types. The following theorem, which makes use of the notion of pointwise free monad introduced in Definition A.1 will have a crucial role in the following. As we did in the discussion leading to Proposition 4, we assume that our lccc has coproducts.

Theorem 11. *If $\mathcal{C}$ has $\mathcal{W}$-types then every polynomial endofunctor on $\mathcal{C}$ has a pointwise free monad.*

Proof. Let $P : \mathcal{C} \to \mathcal{C}$ be a polynomial functor. If we knew that for every object X of $\mathcal{C}$ the functor $P_X : \mathcal{C} \to \mathcal{C}$ had an initial algebra, then we could invoke Proposition A.6 and conclude the desired claim. By Proposition 4, however, it is actually the case that the functors $P_X : \mathcal{C} \to \mathcal{C}$, for X in $\mathcal{C}$, are polynomial, and therefore they have an initial algebra by the assumption that $\mathcal{C}$ has $\mathcal{W}$-types.

From now on, we assume that $\mathcal{C}$ has $\mathcal{W}$-types and refer to $\mathcal{W}_f$ as the $\mathcal{W}$-type for the arrow f. Theorem 11 ensures that free monads for polynomial functors exist. We begin by deriving some information on these free monads.

Proposition 12. *Free monads for polynomial functors are strong monads.*

Proof. The claim is a consequence of Proposition 5 and Proposition A.9.

The next corollary allows us to observe the existence of structure on the categories of algebras for polynomial functors.

Corollary 13. *For every polynomial functor P on $\mathcal{C}$, the category P-alg is isomorphic to the category T-Alg, where T is free monad on P.*

Proof. The claim follows by Proposition A.2 and by Proposition A.4.

5 Dependent Polynomial Functors

We can now pick up the fruits of the work done in the last section and exploit freely the internal language to prove further consequences of the assumption of the existence of $\mathcal{W}$-types in a lccc. Here we show how $\mathcal{W}$-types can be used to define initial algebras for a class of functors that is wider than the one of polynomial functors. For categorical counterparts of the arguments sketched here we invite the reader to consult [8].

5.1 Initial Algebras

Let $f : B \to A$ be an arrow in a lccc $\mathcal{C}$ with $\mathcal{W}$-types, and define $W =_{\mathrm{df}} \mathcal{W}_f$. For an arrow $g : C \to B$ we define an endofunctor $\mathcal{D}_g : \mathcal{C}/W \to \mathcal{C}/W$, called the *dependent polynomial functor associated to* g by letting

$$\mathcal{D}_g\left(X_{(a,t)} \mid (a,t) \in W\right) =_{\mathrm{df}} \Big(\sum_{b \in B_a} X_{t(b)}^{C_b} \mid (a,t) \in W \Big)$$

for $(X_{(a,t)} \mid (a,t) \in W)$ in $\mathcal{C}/W$. Observe that every polynomial functor is a dependent polynomial functor. The next theorem is our main result.

Theorem 14. *Every dependent polynomial functor has an initial algebra.*

Proof. We only illustrate the construction of the desired initial algebra, and leave the verification of the appropriate properties to the reader. Let us continue to use the notation adopted to introduce dependent polynomial functor and consider an arrow $g : C \to B$. First of all, define D and E as the objects fitting in the pullback diagram

$$\begin{array}{ccccc} E & \xrightarrow{h} & D & \xrightarrow{p_W} & W \\ \downarrow & & \downarrow{\scriptstyle p_B} & & \downarrow{\scriptstyle p_A} \\ C & \xrightarrow{g} & B & \xrightarrow{f} & A \end{array}$$

where $p_A : W \to A$ is the arrow mapping $(a,t) \in W$ into $a \in A$. In the internal logic of $\mathcal{C}$ we have $D \cong \sum_{a \in A} B_a \times W^{B_a}$. Define $V =_{\mathrm{df}} \mathcal{W}_h$ and observe

$$V \cong \{(a,b,t,v) \mid (a,b,t) \in D\,,\ v \in V^{C_b}\}\,.$$

since $E_{(a,b,t)} \cong C_b$ for $(a,b,t) \in D$. We will define X as the object fitting in the equaliser diagram

$$X \longrightarrow V \underset{n}{\overset{m}{\rightrightarrows}} \sum_{d \in D} W^{E_d}$$

The object $X \to W$ of $\mathcal{C}/W$ that is an initial algebra for $\mathcal{D}_g$ will then be given as the composite

$$X \xrightarrow{e} V \xrightarrow{p_D} D \xrightarrow{j} W \ ,$$

where $p_D : V \to D$ maps $(a,b,t,v) \in V$ into $(a,b,t) \in D$. It remains to define the arrows m and n necessary to isolate X. For $(a,t,b,v) \in V$ define $m(a,b,t,v) =_{\mathrm{df}} (a,b,t,j\ p\ v)$ and observe m has the correct target. For $(a,b,t,v) \in V$ define $n(a,b,t,v) =_{\mathrm{df}} (a,b,t,(\lambda\_ \in C_b)t(b))$ and observe that, since $t(b) \in W$ for $(a,t) \in W$ and $b \in B_a$, $n(a,b,t,v)$ lies in the appropriate object.

5.2 Applications

Let us consider two arrows $f : B \to A$ and $g : D \to C$ in a lccc $\mathcal{C}$ with $\mathcal{W}$-types. We can then define a functor $F : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ whose action on an object (X,Y) is defined as

$$F(X,Y) =_{\mathrm{df}} \mathcal{P}_f(X) \times \mathcal{P}_g(Y) \, .$$

For a fixed object X of $\mathcal{C}$, the functor $F^X : \mathcal{C} \to \mathcal{C}$ that maps Y into $F(X,Y)$ can easily be seen to be polynomial. It therefore has an initial algebra, that we denote as

$$F^X\big(\mu Y.F(X,Y)\big) \longrightarrow \mu Y.F(X,Y)$$

The assignment of $\mu Y.F(X,Y)$ to X can then be extended to a functor $\mathcal{C} \to \mathcal{C}$. We refer to these functors as *fixpoint functors*.

Theorem 15. *Fixpoint functors are polynomial.*

Proof. We limit ourselves to sketch the main idea of the argument. Let us actually suppose that the fixpoint functor is polynomial, and let $Q \to P$ be an arrow in $\mathcal{C}$ such that

$$\mu Y.F(X,Y) \cong \sum_{p \in P} X^{Q_p}$$

Direct calculations imply that there must be isomorphisms

$$P \cong A \times \sum_{c \in C} P^{D_c}$$
$$Q_p \cong B_a + \sum_{d \in D_c} Q_{t(d)}$$

for $(a, c, t) \in A \times \sum_{c \in C} P^{D_c}$. The first isomorphism certainly holds if we define P as the $\mathcal{W}$-type of the arrow $g \times 1_A : D \times A \to C \times A$ and use Corollary 10. Theorem 14 shows that it is also possible to satisfy the second isomorphism by defining $Q \to P$ as the initial algebra for an appropriate dependent polynomial functor.

Theorem 16. *The free monad on a polynomial functor is polynomial.*

Proof. The reasoning used to prove the claim is completely analogous to the one used to prove Proposition 15.

A Algebras and Monads

In this appendix we review some facts concerning endofunctors and monads, and some results concerning free monads. For more details the reader is invited to refer to [4, 5, 11]. Let $P : \mathcal{C} \to \mathcal{C}$ be an endofunctor on a category $\mathcal{C}$. An *algebra* for P, or a P-algebra, is a diagram of the form $a : PA \to A$ in $\mathcal{C}$. An arrow of P-algebras from $a : PA \to A$ to $b : PB \to B$ is given by a commuting diagram

$$\begin{array}{ccc} PA & \xrightarrow{Pf} & PB \\ {\scriptstyle a}\downarrow & & \downarrow{\scriptstyle b} \\ A & \xrightarrow{f} & B \end{array}$$

There is then a manifest category $P\text{-}alg$ of P-algebras and P-algebra arrows. We write $U : P\text{-}alg \to \mathcal{C}$. for the obvious forgetful functor.

Definition A.1. Let P be an endofunctor on $\mathcal{C}$. We say that P *has a pointwise free monad* if the forgetful functor $U : P\text{-}alg \to \mathcal{C}$ has a left adjoint.

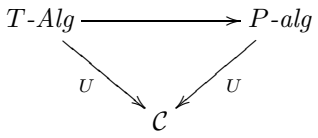
The next proposition shows that the existence of a pointwise free monad for an endofunctor is a necessary and sufficient condition for the category of algebras for the endofunctor to be isomorphic to a category of algebras for a monad.

Proposition A.2. *The forgetful functor $U : P\text{-}alg \to \mathcal{C}$ has a left adjoint if and only if it is monadic over $\mathcal{C}$.*

Proof. The proof is an application of Beck's theorem [12] characterising monadic adjunctions. One should observe that the functor U satisfies all the hypothesis of Beck's theorem except for the existence of a left adjoint.

When (T, η, μ) is a monad on a category $\mathcal{C}$ we write $T\text{-}Alg$ for the usual category of T-algebras. Note that we follow a suggestion of Peter Freyd in using $P\text{-}alg$ for the algebras of an endofunctor P and $T\text{-}Alg$ for the algebras for a monad T. Again, we write $U : T\text{-}Alg \to \mathcal{C}$ for the obvious forgetful functor. We can then restate Proposition A.2 as follows.

Proposition A.3. *(T, η, μ) is a pointwise free monad for P if and only if there is an equivalence T-Alg $\to$ P-alg such that the following diagram commutes*



We wish to give a more concrete description of the pointwise free monad for an endofunctor on a locally cartesian closed category with coproducts. We now introduce a family of functors $P_X : \mathcal{C} \to \mathcal{C}$, for X in $\mathcal{C}$, necessary to state the next proposition. First of all, observe that the function mapping a pair (X, Y) of objects of $\mathcal{C}$ into $X + PY$ extends easily to a functor from $\mathcal{C} \times \mathcal{C}$ to $\mathcal{C}$. This determines a functor $\mathcal{C} \to \mathsf{End}(\mathcal{C})$ mapping an object X of $\mathcal{C}$ into an endofunctor P_X on $\mathcal{C}$. More explicitly, the action of P_X on an object Y of $\mathcal{C}$ is such that $P_X(Y) = X + PY$.

Proposition A.4. *Let P be an endofunctor on $\mathcal{C}$. The following are equivalent:*

(i) *The endofunctor P has a pointwise free monad,*
(ii) *The comma category $X \downarrow U$ has an initial object, for all X in $\mathcal{C}$.*
(iii) *The endofunctor P_X has an initial algebra, for all X in $\mathcal{C}$.*

Proof. The equivalence (i) $\Leftrightarrow$ (ii) follows by Definition A.1 and by the possibility of determining a left adjoint via initial objects in comma categories [12]. The equivalence (ii) $\Leftrightarrow$ (iii) follows from the existence of an isomorphism of categories $X \downarrow \mathcal{C} \cong P_X$-*alg*.

Recall from [5, 17] that the category of monads on $\mathcal{C}$, $\mathsf{Mnd}(\mathcal{C})$, has monads as objects and monad natural transformations as arrows. In the following definition we make use of the forgetful functor $U : \mathsf{Mnd}(\mathcal{C}) \to \mathsf{End}(\mathcal{C})$ mapping a monad to its functor part.

Definition A.5. Let $P : \mathcal{C} \to \mathcal{C}$. By a *uniform free monad* on P we shall mean an initial object of the comma category $P \downarrow U$.

Let us make this definition more explicit. A uniform free monad for an endofunctor $P : \mathcal{C} \to \mathcal{C}$ is determined by a monad (T, η, μ) on $\mathcal{C}$ and by a natural transformation $\alpha : P \Rightarrow T$ with the following universal property: for any monad (T', η', μ') and any natural transformation $\alpha' : P \Rightarrow T'$ there exists a unique monad transformation $\lambda : T \Rightarrow T'$ such that $\lambda\, \alpha = \alpha'$. The next proposition connects the notions of pointwise and free monad. It can be proved using either Proposition A.3 or Proposition A.4. Details are contained in [8].

Proposition A.6. *A pointwise free monad determines a uniform free monad.*

We conclude this appendix by recalling the notions of strong functor and strong monad and a simple fact about them. Let $(\mathcal{C}, \otimes, I, a, l, r)$ be a monoidal category, where I is the unit object and a, l, r are the isomorphisms expressing the associativity, left and right unit laws. For diagrammatic counterparts of the conditions expressed in the next definitions, we invite the reader to refer to [8].

Definition A.7. Let $P : \mathcal{C} \to \mathcal{C}$ be a functor. By a *strength* for P we mean a dinatural transformation σ with components $\sigma_{X,Y} : X \otimes PY \to P(X \otimes Y)$, for X and Y in $\mathcal{C}$, such that the following equations hold:

$$P(l) \circ \sigma_{X,I} = l\,, \qquad P(r) \circ \sigma_{I,Y} = r\,,$$
$$\sigma_{X,Y\otimes Z} \circ 1_X \otimes \sigma_{Y,Z} = \sigma_{X\times Y,Z}\,,$$

for all X, Y, Z in $\mathcal{C}$.

Definition A.8. Let (T, η, μ) be a monad on $\mathcal{C}$. By a *strength* for (T, η, μ) we mean a strength σ for the functor T such that the following equations hold:

$$\sigma_{X,Y} \circ 1_A \otimes \eta_Y = \eta_{X\otimes Y}$$
$$\mu_{X\otimes Y} \circ T(\sigma_{X,Y}) \circ \sigma_{X,TY} = \sigma_{X,Y} \circ 1_X \otimes \mu_Y$$

We shall now consider a cartesian closed category $\mathcal{C}$ and refer to the closed monoidal structure associated to it. The following result is proved in [8].

Proposition A.9. *Let P be an endofunctor on $\mathcal{C}$ and (T, η, μ) be the free monad on P. A strength for the functor P determines a strength for the monad (T, η, μ).*

References

1. Abbott, M.: *Categories of Containers*. Ph.D. thesis, University of Leicester (2003).
2. Abbott, M., Altenkirch, T., Ghani, N.: Categories of Containers. in *Foundations of Software Science and Computation Structures* LNCS 2620, Springer (2003) 23 – 38.
3. Aczel, P.: The Type Theoretic Interpretation of Constructive Set Theory: Inductive Definitions, in: R.B. Barcan Marcus et al. (eds.) *Logic, Methodology and Philosophy of Science, VII*, North-Holland (1986).
4. Barr, M.: Coequalizers and free triples. *Math. Z.* 116 (1970) 307 – 322.
5. Barr, M., Wells, C.: *Toposes, triples and theories*. Springer-Verlag (1985).
6. Ehrhard, T., Regnier, L.: The differential lambda-calculus. *Theoretical Computer Science*. To appear.
7. Freyd, P.: Aspects of topoi. *Bull. Austral. Math. Soc.* 7 (1972) 1 – 76.
8. Gambino, N., Hyland M.: Wellfounded trees and free monads. In preparation (2003).
9. Griffor, E., Rathjen, M.: The Strength of Some Martin-Löf's Type Theories, *Archiv for Mathematical Logic* 33 (1994) 347 – 385.
10. Kelly, G.M., Street, R.: Review of the elements of 2-categories. *Proc. Sydney Category Theory Seminar 1972/73* LNM 420, Springer (1974) 75 – 103.
11. Kelly, G.M.: A unified treatment of transfinite constructions for free algebras, free monoids, colimit, associated sheaves and so on. *Bull. Austral. Math. Soc.* 22 (1980) 1 – 83.
12. Mac Lane, S.: *Categories for the working mathematician*. Springer-Verlag (1998).
13. Maietti, M.E.: *The type theory of categorical universes*. Ph.D. thesis, Università di Padova (1998).
14. Martin-Löf, P. *Intuitionistic Type Theory*. Bibliopolis (1984).
15. Moerdijk, I., Palmgren, E.: Wellfounded trees in categories. *Annals of Pure and Applied Logic* 104 (2000) 189 – 218.
16. Seely, R.A.G.: Locally cartesian closed categories and type theory. *Math. Proc. Camb. Phil. Soc.* 95 (1984) 33 – 48.
17. Street, R.: The formal theory of monads. *J. of Pure and Appl. Algebra* 2 (1972) 149 – 168.