Non-trivial Power Types can't be Subtypes of Polymorphic Types

Andrew M. Pitts*
Computer Laboratory, University of Cambridge, England

Abstract

This paper establishes a new, limitative relation between the polymorphic lambda calculus and the kind of higher-order type theory which is embodied in the logic of toposes. It is shown that any embedding in a topos of the cartesian closed category of (closed) types of a model of the polymorphic lambda calculus must place the polymorphic types well away from the powertypes $\sigma \to \Omega$ of the topos, in the sense that $\sigma \to \Omega$ is a subtype of a polymorphic type only in the case that σ is empty (and hence $\sigma \to \Omega$ is terminal). As corollaries, we obtain strengthenings of Reynolds' result on the non-existence of set-theoretic models of polymorphism.

Introduction

The results reported in this paper have their origin in Reynolds' discovery that the standard set-theoretic model of the simply typed lambda calulus cannot be extended to model the polymorphic, or second-order, typed lambda calculus. In [9] Reynolds speculated that there might be a model of polymorphism in which the types σ are interpreted (in an environment) as *sets* $[\![\sigma]\!]$, in such a way that a function type $\sigma \to \sigma'$ is interpreted standardly as the set of *all* functions from $[\![\sigma]\!]$ to $[\![\sigma']\!]$. (Second-order product types $\Pi\alpha.\,\sigma[\alpha]$ were to be interpreted in some non-standard way—the thought being that simple cardinality considerations preclude the possibility of also interpreting $\Pi\alpha.\,\sigma[\alpha]$ standardly via an indexed cartesian product of sets). In [10] Reynolds formulated a precise definition of what constitutes such a model and then proved that no such structure exists.

This result soon became well known, but perhaps not so well understood (by this author, at least). Shortly afterwards Plotkin gave a version of the proof which clarified Reynolds' original proof in two ways. Firstly, Plotkin took Reynolds' notion of 'set-theoretic model of polymorphism' and generalized it to a notion of a *$\mathcal{K}$-model*, where $\mathcal{K}$ is a cartesian closed category (ccc) whose objects are used for the denotations of the closed polymorphic types. Secondly, Plotkin isolated the key step in Reynolds' proof as a special case of a proposition about functors $T : \mathcal{K} \longrightarrow \mathcal{K}$ which are *expressible* in a $\mathcal{K}$-model via expressions in the polymorphic lambda calculus. The proposition is that every such functor has a *weakly initial algebra*: see [11].

Reynolds' notion of model in [10] corresponds to the special case of a $\mathcal{K}$-model with the ccc $\mathcal{K}$ equal to $\mathcal{S}et$, the category of sets and functions. So in the terminology of [11], the result in [10] is that no $\mathcal{S}et$-model exists. However, one can interpret Reynolds' original question about the possibility of giving a set-theoretic model of polymorphism in a slightly more general way:

Question 1 (Mitchell) *Is there a $\mathcal{K}$-model with $\mathcal{K}$ a full sub-ccc of $\mathcal{S}et$?*

Specifying a full sub-ccc of $\mathcal{S}et$ amounts to giving a collection of sets which is closed under taking finite cartesian products and under set exponentiation; and then a $\mathcal{K}$-model for such a $\mathcal{K}$ does

*Research supported by a Royal Society University Research Fellowship.

indeed provide a semantics for polymorphism in which types σ are interpreted as sets $[\![\sigma]\!]$ (the sets which are objects of $\mathcal{K}$) and function types $\sigma \to \sigma'$ are interpreted as the sets $[\![\sigma]\!] \to [\![\sigma']\!]$ of all functions from $[\![\sigma]\!]$ to $[\![\sigma']\!]$. So this would appear to meet the criteria for a set-theoretic semantics of polymorphism whose function types are standard. Unfortunately, as it stands, the proof in [11] that there is no $\mathcal{S}et$-model does not extend to resolve the above question. However, it is an immediate corollary of the main result of this paper (Theorem 1) that *the only $\mathcal{K}$-models with $\mathcal{K}$ a full sub-ccc of $\mathcal{S}et$ are degenerate, in the sense that all the objects of $\mathcal{K}$ are sets with at most one element.*

In order to state the main result, we must consider another extension of the simply typed lambda calculus somewhat different from the polymorphic calculus, namely the *Higher-Order Logic of Toposes*, or HOLT for short. In HOLT, the usual apparatus of the simply typed lambda calculus (function types, application and lambda abstraction) is extended by finite product types (with associated projection and pairing operations) and by a ground type Ω of 'truth values' equipped with an equality test $=_\sigma : \sigma \to \sigma \to \Omega$ for each type σ. As a logic, HOLT can be formulated as a system for deriving equations between terms of equal type using the usual rules of equational logic augmented by certain axioms (such as the β and η axioms for λ-abstraction and extensionality axioms for the equality tests). The close correspondence between theories in the simply typed lambda calculus and cartesian closed categories extends to a similar correspondence between theories in HOLT and toposes (which are those ccc's which also possess a subobject classifier). We refer the reader to Part II of [6] for a detailed account of this correspondence and for other, equivalent formulations of the higher-order logic of toposes.

One of these equivalent formulations, and probably the most convenient one, is as a predicate logic. Singling out the terms of type Ω and calling them *formulas*, then all the usual propositional operations on formulas (conjunction $\wedge$, disjunction $\vee$, implication $\Rightarrow$, and so on) are definable, as are quantified formulas ($\forall x \in \sigma.\, \phi$, $\exists x \in \sigma.\, \phi$). A type of the form $\sigma \to \Omega$ acts as a *powertype* for the type σ, because the terms of type $\sigma \to \Omega$ act like the characteristic functions of subtypes of σ—given a formula $\phi : \Omega$ possibly involving a variable $x : \sigma$, we can separate out the subtype '$\{x \in \sigma \mid \phi\}$' via the lambda abstraction $\lambda x \in \sigma.\, \phi : \sigma \to \Omega$. From this viewpoint, HOLT is a kind of *intuitionistic* set theory. Intuitionistic, because although Ω contains constants t and f for 'truth' and 'falsity', in general a topos does not satisfy the Law of Excluded Middle, which says that every element of Ω is either t or f: $(\forall p \in \Omega.\, (p =_\Omega \mathsf{t}) \vee (p =_\Omega \mathsf{f})) =_\Omega \mathsf{t}$.

Experience with toposes over the last 15 or so years shows that it is possible to encode a lot of mathematical constructs within the language of HOLT. Moreover, because the higher-order logic of toposes is intuitionistic, many possibilites for the particular topos $\mathcal{S}et$ which are ruled out by the non-constructive nature of classical set theory, become feasible for a more general topos. This is precisely the case for models of polymorphism. In [8] it was shown how to fully embed any categorical-style model of second-order typed lambda calculus in a topos in such a way that the original model appears in the corresponding internal logic of the topos as a 'set of sets, $\mathcal{U}$, closed under exponentiation and $\mathcal{U}$-indexed cartesian products'. Such a structure in a topos models not only the function types of the polymorphic calculus in a standard way, but also the second-order product types. As well as the examples manufactured in [8], one 'naturally occuring' example is the much-studied *modest sets* model of polymorphism, for which the enveloping topos is Hyland's *effective topos* : see [3] and [2]. But a non-trivial example of this kind of structure is not possible in the topos $\mathcal{S}et$: simple cardinality considerations show that any such $\mathcal{U}$ would have to contain only sets with at most one element.

The categorical-style models, $\mathcal{P}$, of polymorphism considered in [8] (and before that in [12]) are in particular $\mathcal{K}$-models in the sense of Reynolds and Plotkin where $\mathcal{K} = \mathcal{P}(1, \mathcal{U})$ is the ccc of (denotations of) closed types and terms in the model $\mathcal{P}$. The construction of [8] results in a certain topos $\mathcal{E}$ derived from $\mathcal{P}$, containing

$\mathcal{K}$ as a full sub-ccc (and with other properties besides: one should note that *any* ccc $\mathcal{K}$ can be embedded as a sub-ccc of a topos, for example via the Yoneda embedding of $\mathcal{K}$ into the category of presheaves on $\mathcal{K}$ valued in a category of suitably large sets). In such a situation, it is natural to ask how the polymorphic types (i.e. the objects of $\mathcal{K}$) relate to the larger collection of HOLT types (i.e. the objects of $\mathcal{E}$). In particular, is it possible for $\mathcal{K}$ to be the whole of $\mathcal{E}$? In other words, after Reynolds we cannot hope for a $\mathcal{S}et$-model, but perhaps it is possible to have an $\mathcal{E}$-model with $\mathcal{E}$ a non-trivial topos ? Unfortunately even this is not possible, since we will prove:

Theorem 1 *Suppose that $\mathcal{E}$ is a topos and $\mathcal{K}$ is a full sub-ccc of $\mathcal{E}$ for which there is a $\mathcal{K}$-model (in the sense of [11]) of the polymorphic typed lambda calculus . If X is an object of $\mathcal{E}$ and the powerobject $X \to \Omega$ is a subobject of an object in $\mathcal{K}$, then X is empty, that is, X is isomorphic to the initial object* 0.

In particular, if $\mathcal{K}$ were the whole of $\mathcal{E}$, then every object of $\mathcal{E}$ would be empty and hence $\mathcal{E}$ would be *trivial*, in the sense of being equivalent to the one-object-one-morphism category:

Corollary 1 *There is no $\mathcal{K}$-model of polymorphism for which $\mathcal{K}$ is a non-trivial topos.*

Another special case of the theorem is when $\mathcal{E} = \mathcal{S}et$. Then $\Omega = \{\mathsf{t}, \mathsf{f}\}$ is a two element set and so an object X contains a powerobject as a subobject just in case X is a set with at least two different elements. Consequently we obtain the result which was mentioned above:

Corollary 2 *All $\mathcal{K}$-models of polymorphism with $\mathcal{K}$ a full sub-ccc of $\mathcal{S}et$ are degenerate, in the sense that all the objects of $\mathcal{K}$ are sets with at most one element.*

Our proof of Theorem 1 builds on the argument given in [11] for the non-existence of a $\mathcal{S}et$-model. In section 1, we briefly recall the Reynolds-Plotkin result on polymorphically expressible functors. In section 2 we sketch the main new argument, which produces from the hypotheses of the theorem an object I in $\mathcal{E}$ equipped with an isomorphism $(I \to P) \to P \cong I$, where P is the powerobject $X \to \Omega$. Finally in section 3 we recall the fact that a suitable form of Cantor's Theorem is provable in the higher-order logic of toposes, and then deduce from the above isomorphism that X is isomorphic to 0.

Acknowledgement The first version I obtained of Theorem 1 was weaker than the one presented here, in that it contained the additional assumption that the ccc $\mathcal{K}$ has equalizers (of parallel pairs of morphisms); this weaker version is still sufficient to deduce Corollary 1, since toposes are in particular ccc's with equalizers. I am grateful to John Mitchell for spurring me on to remove the unecessary assumption of equalizers and in particular for raising the Question 1 which is answered here in the negative.

1 Polymorphic Expressibility

We need to consider not just the *pure* polymorphic typed lambda calculus, but that defined over some signature containing type constants κ, type operators F of various arities $n \geq 1$ (which can be applied to an n-tuple of types $\sigma_1, \ldots, \sigma_n$ to produce another type $F(\sigma_1, \ldots, \sigma_n)$) and individual constants k^σ of various types σ. So the polymorphic types σ are built up from type variables $\alpha_1, \alpha_2, \ldots$ using the grammar

$$\sigma ::= \alpha \mid \kappa \mid F(\sigma, \ldots, \sigma) \mid \sigma \to \sigma \mid \Pi\alpha.\,\sigma$$

and then the terms t of each type σ are built up from individual variables $x_1^\sigma, x_2^\sigma, \ldots$ using the following rules (where '$t : \sigma$' means that t is a well-formed term of type σ):

- if c^σ is a variable or constant, then $c^\sigma : \sigma$;
- if $t : \sigma \to \tau$ and $s : \sigma$, then $ts : \tau$;
- if $t : \tau$, then $\lambda x^\sigma.\, t : \sigma \to \tau$;
- if $t : \Pi\alpha.\,\sigma$, then $t_\tau : \sigma[\tau/\alpha]$
(the type is the result of substituting τ for α in σ);

- if $t : \sigma$, then $\Lambda\alpha.\, t : \Pi\alpha.\, \sigma$ provided that α is not free in any type which is the type of an individual variable occurring freely in t.

The last clause refers to the freeness of variables: the type variable α is bound in $\Pi\alpha.\, \sigma$ and $\Lambda\alpha.\, t$ as is the individual variable x^σ in $\lambda x^\sigma.\, t$; all other occurences of variables are free. A type or term with no free type variables will be called *(type-)closed.*

A description of a *categorical* semantics of these polymorphic types and terms based upon Lawvere's notion of 'hyperdoctrine' is given in [12] (for the higher-order calculus) and in some detail in [8]. In this semantics β and η conversion hold for both kinds of abstraction (λ and Λ). In [11] an environment-style semantics is given, which is intentionally quite weak (it satisfies β and η conversion for λ-abstraction and a limited form of β-conversion for Λ-abstraction) and is tailored to obtaining the results of that paper and no more. (See also [1] for a semantics in a similar style; and see [7] for a detailed comparison between the categorical- and the environment-style models in the case of the simply typed lambda calculus.)

For both kinds of model, part of the structure is a cartesian closed category $\mathcal{K}$ which is used in particular to give denotations to the *closed* types and terms. Since this part of a model plays the principle role in [11], Reynolds and Plotkin call their models of polymorphism $\mathcal{K}$-models. We will not recall here the details of the definitions of either the categorical or the Reynolds-Plotkin notions of model of polymorphism. Instead we note that the first kind can be regarded as a particular instance of the second, but that all the 'naturally occuring' models (known to the author) satisfy the more stringent requirements of the categorical semantics.

Now let $\mathcal{K}$ be a fixed ccc for which there is some $\mathcal{K}$-model. We recall the result in [11] on polymorphic expressibility of a functor $T : \mathcal{K} \longrightarrow \mathcal{K}$ (see below for an explanation of this notion):

Proposition 1 (Reynolds-Plotkin) *If $T : \mathcal{K} \longrightarrow \mathcal{K}$ is expressible in a $\mathcal{K}$-model, then there is a weakly initial T-algebra, that is, an object W of $\mathcal{K}$ equipped with a morphism $w : T(W) \longrightarrow W$ with the property that for any similar data $f : T(K) \longrightarrow K$ there is some (not necessarily unique) morphism $\bar{f} : W \longrightarrow K$ satisfying $\bar{f} \circ w = f \circ T(\bar{f})$.*

For our purposes here it is sufficient to use a slightly stronger notion of polymorphic expressibility than that which is given in [11]. So we will say that $T : \mathcal{K} \longrightarrow \mathcal{K}$ is *expressible* in a $\mathcal{K}$-model if there is a type with at most one free type variable, $\tau[\alpha]$, and a term $t : \Pi\alpha.\, \Pi\beta.\, (\alpha \to \beta) \to (\tau[\alpha] \to \tau[\beta])$ which together 'induce the action of T on $\mathcal{K}$'. This means that if K is an object of $\mathcal{K}$, evaluating $\tau[\alpha]$ in the environment which assigns K to α yields another object of $\mathcal{K}$, which is to be $T(K)$; and similarly, evaluating $t_{\alpha\beta} x^{\alpha\to\beta}$ in a suitable environment determined by $f : K \longrightarrow K'$ will yield $T(f) : T(K) \longrightarrow T(K')$.

The only example of a polymorphically expressible functor we need to consider is that given by double exponentiation by an object. For any object K of $\mathcal{K}$, let $T_K : \mathcal{K} \longrightarrow \mathcal{K}$ be the functor

$$T_K(-) \equiv ((-) \to K) \to K.$$

If K is the denotation of some closed type κ in a $\mathcal{K}$-model, then T_K is expressible in that model: for we can take

$$\begin{aligned} \tau[\alpha] &\equiv (\alpha \to \kappa) \to \kappa \\ t &\equiv \Lambda\alpha.\, \Lambda\beta.\, \lambda y.\, \lambda u.\, \lambda z.\, u(z \circ y) \end{aligned}$$

where $y : \alpha \to \beta, u : (\alpha \to \kappa) \to \kappa, z : \beta \to \kappa$ and $z \circ y \equiv \lambda x^\alpha.\, z(yx)$. (See [11], Proposition 2.) By changing model we can remove the restriction on K, and obtain:

Corollary 3 *Let $\mathcal{K}$ be a ccc for which there is a $\mathcal{K}$-model and let K be any object of $\mathcal{K}$. Then the functor $T_K : \mathcal{K} \longrightarrow \mathcal{K}$ possesses a weakly initial algebra.*

Proof From the remarks above, to apply Proposition 1 it is sufficient to find a $\mathcal{K}$-model for a signature of type and individual constants for which K is the denotation of some closed type over the signature. This may not be the case for the signature and $\mathcal{K}$-model of it which are given at first.

However, we can expand the signature by adding a new type constant naming the object K; and it is then possible to extend the original $\mathcal{K}$-model to a new $\mathcal{K}$-model of the bigger signature. Then T_K is expressible in this new model and so by Proposition 1, it has a weakly initial algebra.

(*End of Proof*)

The proof of Corollary 3 highlights an important difference between the style of model in [11] and the categorical notion of model [8]: a $\mathcal{K}$-model is given relative to a *particular* signature, whereas a categorical model is not. Instead, a categorical model is capable of giving a semantics for any signature once a structure for that signature has been specified in the model. Thus the change of model in the above proof would be unnecessary if we restricted attention just to the categorical style of model.

2 Initial T_P-Algebras

In this section we suppose given a cartesian closed category $\mathcal{K}$ for which there is a $\mathcal{K}$-model of polymorphism. Suppose also that $\mathcal{E}$ is a topos containing $\mathcal{K}$ as a full sub-ccc: in other words, we can regard the objects of $\mathcal{K}$ as a subcollection of the objects of $\mathcal{E}$ which is closed under the operations of taking finite products and exponentials in $\mathcal{E}$. Suppose further that X is an object of $\mathcal{E}$ and that the powerobject $P \equiv (X \to \Omega)$ is a subobject of an object in $\mathcal{K}$, so that there is a monomorphism $m : P \rightarrowtail K$ with K in $\mathcal{K}$. The aim in this section is to show how to construct an object I in $\mathcal{E}$ together with an isomorphism $i : (I \to P) \to P \cong I$.

By definition, an object I together with a morphism $i : ((I \to P) \to P) \longrightarrow I$ constitutes an *algebra* (I, i) for the functor

$$T_P(-) \equiv ((-) \to P) \to P : \mathcal{E} \longrightarrow \mathcal{E}.$$

These algebras are the objects of a category $T_P\text{-}\mathcal{A}lg$, whose morphisms $(I, i) \longrightarrow (J, j)$ are morphisms $f : I \longrightarrow J$ in $\mathcal{E}$ satisfying that $j \circ T_P(f) = f \circ i$. It is well known that if (I, i) is an initial object in this category, then i is necessarily an isomorphism (see [5], [13], [11]). (Recall that an object 0 in a category is *initial* if for every object X there is a unique morphism $0 \longrightarrow X$; a *weakly initial* object satisfies the same condition except for the uniqueness requirement on the morphism.)

So to fulfil our aim of constructing an object I and isomorphism $i : ((I \to P) \to P) \cong I$, we must construct an initial algebra for T_P. In fact we only construct an initial algebra for the restriction of T_P to an endofunctor $\mathcal{S} \longrightarrow \mathcal{S}$, where $\mathcal{S}$ is a certain full subcategory of $\mathcal{E}$ to be defined below—but this is sufficient. The construction is in two steps:

Step 1 It is the case that T_P is a natural retract of T_K, that is, there are natural transformations $\iota : T_P \longrightarrow T_K$ and $\rho : T_K \longrightarrow T_P$ with $\rho \circ \iota = \mathrm{id}$. (This is because $P \equiv (X \to \Omega)$, being a powerobject in a topos, is injective and hence the monomorphism $m : P \rightarrowtail K$ has a left inverse, i.e. there is $\ell : K \longrightarrow P$ with $\ell \circ m = \mathrm{id}$; indeed, one suitable ℓ is defined in the internal higher-order logic of the topos $\mathcal{E}$ by $\lambda k \in K. \{x \in X \mid \forall S \in P. m(S) = k \Rightarrow x \in S\}$. So we can take $\iota \equiv ((-) \to \ell) \to m$ and $\rho \equiv ((-) \to m) \to \ell$.)

Recall that by hypothesis, the objects of $\mathcal{K}$ are a subcollection of those of $\mathcal{E}$ and are closed under exponentiation in $\mathcal{E}$. Thus T_K is an endofunctor of both $\mathcal{K}$ and the enveloping topos $\mathcal{E}$. By Corollary 3 we have an algebra (W, w) for T_K which is weakly initial for the collection of T_K-algebras (E, e) whose underlying object E is in $\mathcal{K}$. Then by composing with the component of ι at W, $\iota_W : T_P(W) \longrightarrow T_K(W)$, we can turn the T_K-algebra (W, w) into a T_P-algebra $(W, w \circ \iota_W)$.

Step 2 Form the T_P-algebra (I, i) which internally to the topos $\mathcal{E}$ is the intersection of all the T_P-subalgebras of $(W, w \circ \iota_W)$. (In general, such a construction is possible because T_P is an '$\mathcal{E}$-indexed functor'; but given the specific form of P as $X \to \Omega$ and T_P as $((-) \to P) \to P$, one can also give an explicit description of I using the internal higher-order logic of $\mathcal{E}$: I is the subobject of W described by

$$\{y \in W \mid \forall S \in (W \to \Omega). (\forall u \in T_P(W). \\ \text{'}u \in T_P(S)\text{'} \Rightarrow w(\iota_W(u)) \in S) \Rightarrow y \in S\},$$

where '$u \in T_P(S)$' stands for '$\forall x, x' \in X. \forall y \in W. \forall z \in W \to P. (x \in u(z) \wedge x' \in z(y) \Rightarrow y \in S)$'.)

Definition Let $\mathcal{S}$ be the full subcategory of $\mathcal{E}$ whose objects are subobjects of objects in $\mathcal{K}$.

Lemma 1 *If E is in $\mathcal{S}$, then so is $T_P(E)$.*

Proof Since P is a powerobject, T_P maps monomorphisms to split monomorphisms. (For if $a : E \rightarrowtail A$ is a monomorphism, then $T_P(a)$ can be described in the internal higher-order logic of $\mathcal{E}$ as $\lambda u \in T_P(E). \lambda z \in (A \to P). u(z \circ a)$; and then a left inverse $r : T_P(A) \longrightarrow T_P(E)$ for $T_P(a)$ is described by $\lambda u \in T_P(A). \lambda z \in (E \to P). u(\exists a(z))$, where $\exists a(z) \in (A \to P)$ is $\lambda y \in A. \{x \in X \mid \exists v \in E. x \in z(v) \wedge a(v) = y\}$, where we are using the fact that P is $X \to \Omega$.)

Thus given E in $\mathcal{S}$, witnessed by some monomorphism $a : E \rightarrowtail A$ with A in $\mathcal{K}$, then the composition of the monomorphism $T_P(a) : T_P(E) \rightarrowtail T_P(A)$ with the monomorphism $\iota_A : T_P(A) \rightarrowtail T_K(A)$ constructed in Step 1 above, witnesses that $T_P(E)$ is also in $\mathcal{S}$.

(*End of Proof*)

Thus T_P restricts to an endofunctor of $\mathcal{S}$. We claim that (I, i) constructed in Step 2 is an initial algebra for $T_P : \mathcal{S} \longrightarrow \mathcal{S}$. To see this, we use the following consequence of the weak initiality property of (W, w):

Lemma 2 *For every T_P-algebra (E, e) with E in $\mathcal{S}$, there is a morphism in T_P-$\mathcal{A}lg$ from a T_P-subalgebra of $(W, w \circ \iota_W)$ to (E, e).*

Proof Since E is in $\mathcal{S}$, there is a monomorphism $a : E \rightarrowtail A$ with A in $\mathcal{K}$. Then as we noted in the proof of Lemma 1, $T_P(a)$ is a split monomorphism, with left inverse $r : T_P(A) \longrightarrow T_P(E)$ say. Using r and $\rho : T_K \longrightarrow T_P$ from Step 1, we get a T_K-algebra $(A, a \circ e \circ r \circ \rho_A)$. Since A is in $\mathcal{K}$, the weak initiality property of (W, w) furnishes a T_K-algebra morphism $f : (W, w) \longrightarrow (A, a \circ e \circ r \circ \rho_A)$ and hence a T_P-algebra morphism $f : (W, w \circ \iota_W) \longrightarrow (A, a \circ e \circ r)$. But (E, e) is a T_P-subalgebra of $(A, a \circ e \circ r)$ via the monomorphism a; so forming the pullback in T_P-$\mathcal{A}lg$ of this subalgebra along f, we obtain a subalgebra of $(W, w \circ \iota_W)$ equipped with a morphism to (E, e).

(*End of Proof*)

The following two properties of (I, i) with regard to (external) T_P-subalgebras of $(W, w \circ \iota_W)$ are both straightforward consequences of the definition of (I, i).

Lemma 3 *(i) (I, i) is a subalgebra of any T_P-subalgebra of $(W, w \circ \iota_W)$; and*
(ii) (I, i) contains no proper T_P-subalgebra.

Then Lemma 2 and Lemma 3(i) together imply that (I, i) is weakly initial for $T_P : \mathcal{S} \longrightarrow \mathcal{S}$. (Note that since I is a subobject of W, it is in $\mathcal{S}$.) But then Lemma 3(ii) shows that (I, i) is actually initial: given two T_P-algebra morphisms $f, g : (I, i) \longrightarrow (E, e)$ with E in $\mathcal{S}$, forming the equalizer of f and g we get a subalgebra of (I, i), which by (ii) must be the whole of I—so that the equalizer of f and g is an isomorphism and thus $f = g$.

Thus (I, i) is an initial algebra for $T_P : \mathcal{S} \longrightarrow \mathcal{S}$ and as we remarked above, this implies that $i : ((I \to P) \to P) \longrightarrow I$ is an isomorphism, as required.

3 Cantor's Theorem in a Topos

Classically, Cantor's Theorem says that the cardinality of a set I is less than that of its powerset $\mathsf{P}I$. Specifically, the existence of a surjective mapping from a subset of I onto $\mathsf{P}I$ leads to a contradiction via the well known diagonal argument. The hypothesis amounts to asserting the existence of a relation $R \subseteq I \times \mathsf{P}I$ satisfying $M(R)$ and $E(R)$ where

$$\begin{aligned} M(R) \quad &\equiv \quad \forall u \in I. \forall U, U' \in \mathsf{P}I. \\ &\qquad R(u, U) \wedge R(u, U') \Rightarrow U = U' \\ E(R) \quad &\equiv \quad \forall U \in \mathsf{P}I. \exists u \in I. R(u, U). \end{aligned}$$

Then forming

$$D \equiv \{u \in I \mid \exists U \in \mathsf{P}I. R(u, U) \wedge \neg u \in U\},$$

since $E(R)$ holds we can find $d \in I$ satisfying $R(d, D)$ and hence

$$\begin{aligned} d \in D \quad \Leftrightarrow \quad & \exists U \in \mathsf{P}I.\, R(d,U) \wedge \neg d \in U \\ & \text{(by definition of } D\text{)} \\ \Leftrightarrow \quad & \exists U \in \mathsf{P}I.\, R(d,U) \wedge R(d,D) \\ & \wedge \neg d \in U \\ & \text{(since } R(d,D) \text{ holds)} \\ \Leftrightarrow \quad & \exists U \in \mathsf{P}I.\, U = D \wedge \neg d \in U \\ & \text{(since } M(R) \text{ holds)} \\ \Leftrightarrow \quad & \neg d \in U \end{aligned}$$

But $(d \in D \;\Leftrightarrow\; \neg d \in D)$ is always false, so we have a contradiction.

The above expressions and argument translate directly into the higher order logic of toposes with $I \to \Omega$ for $\mathsf{P}I$, R of type $(I \times (I \to \Omega)) \to \Omega$, etc. (In particular, the Law of Excluded Middle is not needed for the diagonal argument.) In other words, for any object I in a topos, the following sentence in the internal language of the topos is satisfied:

$$\forall R \in (I \times (I \to \Omega)) \to \Omega.\, \neg(M(R) \wedge E(R)) \quad (1)$$

where $M(R)$ and $E(R)$ are as above. Here we need the following corollary of this fact:

Lemma 4 *Suppose that $\mathcal{E}$ is a topos and that X and I are objects of $\mathcal{E}$ for which there is an isomorphism $i : ((I \to P) \to P) \cong I$, where P is the powerobject $X \to \Omega$. Then $X \cong 0$.*

Proof To see that $X \cong 0$ it is sufficient to show that in the internal higher-order logic of $\mathcal{E}$ the sentence $\forall x \in X.\, \mathsf{f}$ is satisfied. Arguing informally, given $x \in X$ we have $\lambda S \in P.\,(Sx) : P \to \Omega$ which provides a left inverse for $\lambda \omega \in \Omega.\, \{x' \in X \mid \omega\} : \Omega \to P$. Then (as in section 2) T_Ω becomes a natural retract of T_P and in particular, there is a monomorphism $m_x : ((I \to \Omega) \to \Omega) \rightarrowtail ((I \to P) \to P)$. But then defining $R_x \in (I \times (I \to \Omega)) \to \Omega$ by

$$\begin{aligned} & R_x(u, U) \;\Leftrightarrow \\ & \qquad u = i\,(m_x(\{U' \in I \to \Omega \mid U' = U\})) \end{aligned}$$

one has $M(R_x) \wedge E(R_x)$, which by (1) is equal to f. So f can be derived from the assumption $x \in X$; in other words $\forall x \in X.\, \mathsf{f}$ holds, as required.

(*End of Proof*)

We can now complete the proof of Theorem 1. Suppose $\mathcal{E}$ is a topos containing a full sub-ccc $\mathcal{K}$ and that a $\mathcal{K}$-model of polymorphism exists. If X is an object of $\mathcal{E}$ for which the powerobject $P \equiv (X \to \Omega)$ is a subobject of some object in $\mathcal{K}$, we saw in section 2 that there is an object I in $\mathcal{E}$ together with an isomorphism $i : ((I \to P) \to P) \cong I$. Then by the above lemma, we have that $X \cong 0$.

Conclusion

The setting we have considered is one in which a model of the polymorphic lambda calculus is embedded in a model of a certain kind of constructive set theory—the higher-order logic of toposes. In view of the results of [8], we can say that such a situation is the norm rather than the exception; and for at least one important model of polymorphism (the modest sets, where the enveloping topos is Hyland's effective topos), it is the natural setting.

In this setting, we have seen that the properties which the object of truth values, Ω, and the equality tests, $=_\sigma : \sigma \to \sigma \to \Omega$, possess in a topos, imply that a powertype $\sigma \to \Omega$ can be contained in a type arising from the model of polymorphism only in the trivial case that σ is empty. This result puts limitations on the kind of cartesian closed category $\mathcal{K}$ for which there is a $\mathcal{K}$-model of polymorphism (Corollaries 1 and 2). In particular it shows, possibly surprisingly, that Reynolds' result on the non-existence of $\mathcal{S}et$-models has nothing to do with the non-constructive nature of classical set theory and everything to do with the fact that the category of sets is a topos.

References

[1] K. B. Bruce and A. R. Meyer, *The Semantics of Second Order Polymorphic Lambda Calculus.* In [4], pp. 131–144.

[2] A. Carboni, P. J. Freyd and A. Scedrov, *A categorical approach to realizability and*

polymorphic types. In M. Main *et al* (eds), *Proc. 3rd ACM Workshop on the Mathematical Foundations of Programming Semantics (New Orleans, 1987)*, Lecture Notes in Computer Science No.298 (Springer-Verlag, Berlin, 1988), pp. 23–42.

[3] J. M. E. Hyland, E. P. Robinson and G. Rosolini, *The discrete objects in the effective topos*, Proc. London Math. Soc., to appear.

[4] G. Kahn *et al* (eds), *Semantics of Data Types*, Lecture Notes in Computer Science No. 173 (Springer, Berlin, 1984).

[5] J. Lambek, *A Fixpoint Theorem for Complete Categories*, Math. Zeitschr. 103(1968) 151–161.

[6] J. Lambek and P. J. Scott, *Introduction to Higher Order Categorical Logic*, Cambridge Studies in Advanced Mathematics 7 (Cambridge University Press, 1986).

[7] J. C. Mitchell and P. J. Scott, *Typed Lambda Models and Cartesian Closed Categories*. In J. W. Gray (ed.), *Categories in Computer Science and Logic (Proc. Boulder, 1987)*, Contemporary Math. (Amer. Math. Soc., Providence RI), to appear.

[8] A. M. Pitts, *Polymorphism is set theoretic, constructively*. In D. H. Pitt *et al* (eds), *Category Theory and Computer Science*, Lecture Notes in Computer Science No. 283 (Springer-Verlag, Berlin, 1987), pp. 12–39.

[9] J. C. Reynolds, *Types, Abstraction and Parametric Polymorphism*. In R. E. A. Mason (ed.), *Information Processing '83* (Elsevier Science Publishers B.V., North-Holland, 1983), pp. 513–523.

[10] J. C. Reynolds, *Polymorphism is not Set-Theoretic*. In [4], pp. 145–156.

[11] J. C. Reynolds and G. D. Plotkin, *On Functors Expressible in the Polymorphic Typed Lambda Calculus*. In G. Huet (ed.), *Logical Foundations of Functional Programming* (Addison-Wesley), to appear.

[12] R. A. G. Seely, *Categorical semantics for higher order polymorphic lambda calculus*, Jour. Symbolic Logic 52(1987) 969–989.

[13] M. B. Smyth and G. D. Plotkin , *The category-theoretic solution of recursive domain equations*, SIAM Jour. Computing 11(1982) 761–783.