ELSEVIER
Theoretical Computer Science 212 (1999) 29–75
Theoretical Computer Science

Infinite λ-calculus and types[1]

Alessandro Berarducci[a], Mariangiola Dezani-Ciancaglini[b,*]

[a] *Dipartimento di Matematica, Universitá di Pisa, v.Buonarroti 2, 56127 Pisa, Italy*
[b] *Dipartimento do Informatica, Universitá di Torino, c.Svizzera 185, 10149 Torino, Italy*

Abstract

Recent work on infinitary versions of the lambda calculus has shown that the infinite lambda calculus can be a useful tool to study the unsolvable terms of the classical lambda calculus. Working in the framework of the intersection type disciplines, we devise a type assignment system such that two terms are equal in the infinite lambda calculus iff they can be assigned the same types in any basis. A novel feature of the system is the presence of a type constant to denote the set of all terms of order zero, and the possibility of applying a type to another type. We prove a completeness and an approximation theorem for our system. Our results can be considered as a first step towards the goal of giving a denotational semantics for the lambda calculus which is suited for the study of the unsolvable terms. However, some noncontinuity phenomena of the infinite lambda calculus make a full realization of this idea (namely the construction of a filter model) a quite difficult task. © 1999 Published by Elsevier Science B.V. All rights reserved

Keywords: Infinite λ-calculus; Intersection types; λ-algebras

1. Introduction

An infinitary version of λ-calculus was presented by Berarducci at the meeting "Common foundations of logic and functional programming" held in Torino, Feb. 1994, and at the conference in honor of Roberto Magari, Siena, April 1994 (published in [7]). An infinite λ-calculus was independently developed at about the same time by Kennaway, Klop, Sleep, and de Vries (see [30]) with some differences reflecting the different motivations, as we will explain in the following. In October 1994, Berarducci met Klop in Pisa on occasion of a talk Klop gave on the infinite λ-calculus. Motivated by previous work with Intrigila [8], Berarducci was mainly interested in applications of infinite λ-calculus to the study of the properties of unsolvable terms in the classical λ-calculus. In particular he defined a special class of unsolvable terms which he called *mute* and

---

[*] Corresponding author. E-mail: berardu@dm.unipi.it, dezani@di.unito.it.

[1] Partly supported by ESPRIT BRA 7232 GENTZEN and by NATO Grant HTECH.LG960875.

PII: S0304-3975(98)00135-2

argued that the mute terms should be considered the terms which represent the notion of "completely undefined computation". To substantiate this claim he proved that it is consistent to simultaneously identify all the mute terms to an arbitrarily fixed term (not necessarily mute). This property is not shared by the class of unsolvable terms, and not even by the smaller class of easy terms as shown by [33]. Easy terms have been studied also in [4, 24–28, 32, 54]. Berarducci showed that the mute terms in his version of the infinite λ-calculus played the same role that the unsolvable terms played in the theory of Böhm trees [5]. The main result of [7] is that if we equate all the mute terms, then the infinite λ-calculus is Church-Rosser and every term has one and only one infinite normal form. It then follows that the infinite normal forms constitute a model of the λ-calculus which is similar to the model of Böhm trees but which does not equate all the unsolvable terms. To understand the idea behind infinite λ-calculus the reader can take a look at Definition 2.5 and Fig. 1 before continuing. Applications of the infinite λ-calculus to the study of the easy terms of the classical λ-calculus are given in [9].

Klop and his collaborators on the other hand were interested in generalizing their earlier work on transfinite reduction sequences in the context of term rewriting systems [29]. In [30] several versions of the infinite λ-calculus are defined but it is shown that only three of them have good properties. These three calculi can be distinguished by the behavior of the element $\perp$. In the version corresponding to the Böhm trees we get $\perp M = \perp = \lambda x.\perp$ and $\perp$ can be interpreted either as "lack of information" or as "unsolvable term". In the version corresponding to the lazy λ-calculus $\perp = \perp M \neq \lambda x.\perp$ and $\perp$ can be interpreted either as lack of information or as unsolvable term "of order zero", namely not reducible to an abstraction $\lambda x.M$. In the version corresponding to the one in [7] $\lambda x.\perp$, $\perp M$ and $\perp$ are pairwise distinct and $\perp$ is interpreted as a "mute term", namely an unsolvable term of order zero which cannot be further decomposed as the application of a term of order zero to some other term. In the latter version we cannot interpret $\perp$ as lack of information, since this would obviously imply $\perp M = \perp$ (where $=$ is β-conversion). To denote lack of information we will use instead Ω. The consistency of the infinite λ-calculus is guaranteed by a Church–Rosser theorem. In [30] this is proved for reduction sequences of every ordinal length, while in [7] one considers only reductions of length ω and proves a Church–Rosser theorem for the system consisting of the terms arising from finite terms.

An infinitary version of the λ-calculus was also studied in [35] but without a related notion of infinite β-reduction. Salibra and Goldblatt [44] consider an equational treatment of λ-calculus with an application to the infinite λ-calculus. They also point out the fact that a rigorous definition of the substitution operator for the infinite λ-calculus requires special care.

In this paper we deal with the version of [7]. In particular $\lambda x.\perp$, $\perp M$ and $\perp$ are all distinct, so not only we do not identify all the unsolvable terms, but we do not even identify all the unsolvable terms of order zero. The price to pay is the presence of some non-constructive features: there is no algorithm to test whether a term has a "top normal form" (the mute terms are those without top normal form). An analogous of mute terms for term rewriting systems (where no binding of variables is allowed) is given in [31].

In July 1994, Mariangiola Dezani proposed to investigate the infinite λ-calculus using the "intersection type disciplines". Fifteen years ago, Dana Scott introduced in [47] the information systems for solving domain equations. They have been recognized as a powerful tool for describing the denotational semantics of programming languages [52]. Abramsky's paper [1] is a mile-stone in this field, since the given formalism (domain prelocales) is quite powerful. In fact domain prelocales allow to represent SFP domains, so in particular Plotkin's powerdomain construction [37]. The main contribution is Abramsky's use of Stone duality to synthesize domain theory and logic of programs.

Intersection type disciplines are a simple case of information systems. They have been used for describing λ-models. The λ-theory of the model described in [6] is just the equality of Böhm trees [41]. In particular, every inverse limit construction can be very easily mimicked by a suitable type assignment system [10, 22, 42]. This technique has also allowed the construction of a Scott domain where all and only the terms with a normal form have an interpretation bigger than a given element [12], thus giving a denotational meaning to normal forms. Moreover, there are suitable filter models isomorphic to Plotkin's and Engeler's models, respectively [38]. Lastly, λ-models whose domain is a qualitative or quantitative domain and the functions are stable functions have corresponding filter models [23, 17]. All previous models are models for the classical λ-calculus, but intersection type disciplines are also suitable for describing models of the $\lambda\mathbf{I}$-calculus [51, 21], of the lazy λ-calculus [2], of the call-by-value [18, 40], of the lazy call-by-value λ-calculus [39], and of extensions including some parallel features (concurrent λ-calculus) [3, 14, 15].

In this approach, a λ-model is described, in a finitary way, by a system assigning types to terms, such that the interpretation of a term in the model is the set of types which can be deduced for it. This can be expressed through the slogan:

"*the meaning of a program is the set of all the propositions which are true of it*".

Logical presentations of domains are very simple, and yet useful to study theories of models. In fact, they are finitary descriptions, and this is essential to prove properties of the interpretations of terms. In particular, they allow standard techniques for proving approximation theorems, which are key steps in showing semantic equalities between terms, adequacy with respect to operational semantics, etc.

A natural question is whether the denotational approach can be used to analyze the infinite λ-calculus. To achieve this goal, our original idea was to extend the intersection type disciplines with a new feature: the application $\alpha\beta$ between two types α and β. In the intended interpretation a term has type $\alpha\beta$ if it can be expressed as the application of two terms, the first of type α and the second of type β. One must also introduce a new type constant ζ to be interpreted, in the closed term model, as the set of all terms of order zero (briefly *zero terms*). Note that the closed zero terms form a proper subset of the unsolvable terms. The resulting types are then rich enough to distinguish between various kinds of unsolvable terms of order zero in the closed term model. The mute ones will only have type ζ (besides the universal type ω), while the others will also have types of the form $\zeta\alpha$, $\zeta\alpha\beta$ etc., reflecting the fact that they can be expressed as an unsolvable term of order zero applied to some other term. We realized that in order to simplify our treatment it was better not to introduce a type for the mute terms: after all they are already characterized for *not* having other types besides ζ and ω. Our types can thus discriminate various classes of unsolvable terms, and this gave us hope to be able to find a filter model in which the model of [7] could be isomorphically embedded. We have partially succeeded in the sense that one of our results is that two terms have the same filter (i.e. they can be assigned the same types) iff they are equal in the infinite λ-calculus (Theorem 8.12). What is still lacking however is a good notion of application which turns the set of filters into a λ-model.

Finding the correct definition of the set of types, their semantic interpretation and also the type assignment system, was a much more difficult task than we expected. To see why, let us discuss in some detail the definition of the set of types and their interpretation. Usually, following [45], the type $\alpha \to \beta$ is interpreted as the set of all terms which applied to a term of type α yield a term of type β. This is called the *simple semantics* of types [19]. In this semantics the types deducible for an η-redex are also deducible for its contractum [19], and this disagrees with the fact that in the infinite λ-calculus they are in general unrelated. However, Scott in [46] proposes a second interpretation (the *F-semantics* according to [19]) where one requires in addition that a term of type $\alpha \to \beta$ must be reducible to a term beginning with a λ-abstraction, i.e. it should not be a zero term. So we chose this second alternative. In the F-semantics the universal type ω is different from $\omega \to \omega$: this is not exempt from complications. Indeed, since our system has a type ζ to represent the set of all zero terms, we are now in presence of two types with empty intersection: the type ζ itself, and the type $\omega \to \omega$. Hence, the type $\zeta \wedge (\omega \to \omega)$ is not inhabited (in any model) and the question arises whether we should consider it a legal type at all. We are thus lead to consider a set of "pretypes" and a smaller set of "types" where some obviously empty types like the one above are forbidden (Definition 3.6). The filter of a term will consist of types from the smaller set. The definition of the set of types is not immediate since after excluding the obviously empty type $\zeta \wedge (\omega \to \omega)$ we must still decide whether a finite intersection like $(\alpha_1 \to \beta_1) \wedge \cdots \wedge (\alpha_n \to \beta_n)$ is empty. The decisive idea comes from Scott theory of information systems [47]: consistent inputs should give consistent outputs. So, if we interpret the above intersection as the step function which gives an output in $\bigwedge_{i\in I} \beta_i$ whenever the input is in $\bigwedge_{i\in I} \alpha_i$ (where I is a subset of $\{1,\dots,n\}$), then we must require that if $\bigwedge_{i\in I} \beta_i$ is empty, so is $\bigwedge_{i\in I} \alpha_i$. The definition of the types is thus obtained by pruning the set of pretypes according to Scott's prescription. This excludes for instance $(\omega \to (\omega \to \omega)) \wedge (\omega \to \zeta)$, because given an input in ω we would get an output in $(\omega \to \omega) \wedge \zeta$, which is impossible since the latter is empty.

Scott's idea however does not solve all the problems. A type like $((\omega \to \omega) \to \zeta) \wedge (\zeta \to (\omega \to \omega))$ is not forbidden by the above considerations, so we consider it a legitimate type, and yet a moment of reflection shows that this type cannot be inhabited by a closed term. Indeed if M were an inhabitant of this type, then M would act like a function which permutes terms of order zero with the disjoint set of all abstraction terms, which is impossible since M would have no fixed points. Thus we face two alternatives. Either we restrict further the set of types by finding a syntactic characterization of the types which are inhabited by a term – but we do not know if this condition is decidable – or we admit, as we do, some types that, although not inhabited by a closed term, can possibly be inhabited in some model. So the need for a completeness theorem arises: if a term is not forbidden by the theory of information systems, it is inhabited in some model.

Before asking the question of inhabitation we must however decide how to interpret types in an arbitrary model. The main problem is how to interpret ζ. Remember that we must distinguish $\bot$ both from $\lambda x.\bot$ and from $\bot M$. The proposed solution is given in Definition 3.2 where we define the zero elements of a model so that the following holds: a zero element is not equal to a λ-abstraction; if M is a zero element, then the function $N \mapsto MN$ is injective. This choice assures us that in some models ζ is *not* the complement of $\omega \to \omega$. This is crucial, otherwise the same argument used for the closed term model Λ_0^β of the $\lambda\beta$-calculus will show that $(\zeta \to (\omega \to \omega)) \wedge ((\omega \to \omega) \to \zeta)$ is empty in every model. The above properties of zero elements are easily captured by our types, and they enter in the definition of the preorder between types (Definition 5.1). Note that, although a closed term of order zero is a zero element in the term model, its interpretation in another model of λ-calculus, may not be a zero element of that model. This happens, for instance, in a model where the term of order zero $\mathbf{\Omega}_2 \equiv (\lambda x.xx)(\lambda x.xx)$ satisfies the equality $\mathbf{\Omega}_2 = \lambda x.\mathbf{\Omega}_2$ or in a model where $\mathbf{\Omega}_2 N = \mathbf{\Omega}_2$ for every N (so $\mathbf{\Omega}_2$ is not injective). We will define a class of *adequate* models where these behaviors are forbidden. Moreover we require that in an adequate model the zero elements share the typical properties of the zero terms in the term model: a zero element applied to an arbitrary element remains a zero element, and $a \cdot b = a' \cdot b'$ implies $a = a'$ and $b = b'$, whenever a, a' are zero elements.

With our definitions ζ and $\omega \to \omega$ will not be always complementary: they will be complementary in the closed term model Λ_0^β but not in models having a "bottom element" (as it is always the case in filters models, where the filter generated by the less informative type ω is a bottom element).

Having discussed the syntax and semantics of types, let us now consider the problem of finding a good type assignment system. We can set ourselves various goals (we state the goals for closed terms, for open terms we need to consider basis and environments):

Goal 1. The system assigns the same types to M and N iff $M = N$ in the infinite λ-calculus.

Goal 2. The system assigns the type α to a closed term M iff M has that type in the closed term model Λ_0^β according to our semantics.

We can achieve Goal 1, but we had to replace Goal 2 with:

Goal 3. The system assigns a type α to a closed term M iff M has that type in all models of a certain class of models.

Any assignment system satisfying the first goal, must contain some non-constructive features, since the equality of two terms in the infinite λ-calculus is not semi-decidable.

We tried to keep the non-constructive features to a minimum by condensing them in a rule that permits to assign the type ζ to any *unsolvable* term of order zero and in a standard (Eq_β) rule . Our type assignment system (Definition 5.4) is sound in the sense that if a type α is assigned to a term M, then M will have semantically that type in any environment in the closed term model Λ_0^β, and more generally in every "adequate" model (Definitions (3.3)). It is important to stress that it would not be sound to assign ζ to a *solvable* zero term $xM_1 \dots M_k$, because in some environment the interpretation of $xM_1 \dots M_k$ is not a zero term.

We had to abandon any claim to capture exactly the closed term model (Goal 2) after we realized that Λ_0^β suffers from some non-continuity phenomena which are difficult to take into account unless one introduces some infinitary features in the type assignment system. (This would not be sensible since the point of this work is to analyze the infinitary notions of the infinite λ-calculus in terms of finitary notions like type, approximant, and deduction.) To illustrate the lack of continuity of the closed term model consider the fixed point combinator **Y**. It is easy to see that **YY** is an unsolvable term of order zero, however we cannot detect this fact by looking only at finitely many approximants of **Y**. In other words application is not continuous in the topology induced by the approximants (Theorem 4.1). Of course, the term **YY** itself does not bother us since we have audaciously introduced a rule to assign the type ζ to any unsolvable term of order zero. However, a slight variation on this theme shatters our hopes to reach Goal 2. Consider instead of **YY** the term $\lambda x.x\mathbf{YY}$. This is not an unsolvable term, but it becomes an unsolvable term of order zero if we apply it to the identity. Now it turns out that our types are so expressive that there is a type σ_I which in the closed term model Λ_0^β is inhabited only by the identity **I** (Lemma 4.3). In Λ_0^β the term $\lambda x.x\mathbf{YY}$ has type $\sigma_I \to \zeta$, but none of its approximants has this type. So there is little hope to detect the presence of this type in any reasonable proof system, and certainly not in a proof system satisfying an approximation theorem (Theorem 7.5). We therefore abandon Goal 2 and content ourselves with Goal 1, the characterization of equality in the infinite λ-calculus (Theorem 8.12), and Goal 3, the soundness and completeness of the type assignment system with respect to the adequate models (Theorem 9.21).

Ronchi della Rocca [41] proves a result similar to our Goal 1; more precisely, that two terms have the same Böhm tree iff they have the same set of types in the standard intersection type discipline [6]. The proof of [41] is based on the notion of principal type of an approximate normal form, which is a type completely describing the approximate normal form. Principal types (as defined in [11] and used in [41]) need an infinity of type variables and this agrees with the type syntax of [6]. Another related paper is [16], which proves that two terms have the same Lévy-Longo tree [34] iff they have the same set of types in the type discipline with union and intersection of [15]. Also [16] uses the notion of principal types, but it gets rid of type variables by replacing them with suitable constant types which depend on the involved terms. We follow a similar approach with one important difference: in order to prove that inclusion between sets of approximants corresponds to inclusion between deducible types we have to consider not just one, but two principal types. We will give an example illustrating the need for two different types (Example 8.23).

To prove the completeness theorem (Goal 3) one difficulty is that unsolvable terms of order zero can arise dynamically in rather unforeseeable ways, and a static type checking is unlikely to control this behavior. To settle this problem we are forced to introduce in the type assignment system a rule that says that β-convertible terms have the same types. To prove the consistency of the resulting system we use the approximation theorem (Theorem 7.5) to show by a purely proof theoretical argument that from a given basis Γ we cannot deduce two incompatible types for the same term (Theorem 7.6). So the deducible types form a filter. (This would be a consequence of the soundness of the assignment system if we already knew that the types of the basis Γ are inhabited in some model, but this is part of the completeness theorem, so we cannot use it yet.) Having proved that the deducible types form a filter, it would be natural to endow the set of filters with an application which makes them a model in which a filter has (semantically) exactly the types which are contained in the filter. But as we said, we did not manage to do this. Our proof of the completeness theorem relies instead on a term model. The idea is to enlarge the closed term model Λ_0^β in such a way that every type is inhabited. We do this by introducing a new constant for each type, and modifying the β-convertibility relation in such a way that the constant corresponding to the type α will indeed have that type, according to our semantics. The proof is inspired by [52, 19].

We have seen that two terms are equal in the infinite λ-calculus if and only if they have the same filter of deducible types in the assignment system. A natural question is whether this remains true if we replace the filter of deducible types with the larger filter of *semantically deducible* types, namely with the set of types which a given term has in the closed term model Λ_0^β according to the given semantics.

As we already observed, there are types which cannot be assigned to closed terms, for example $(\zeta \to (\omega \to \omega)) \wedge ((\omega \to \omega) \to \zeta)$. So we would like to investigate the decidability of type inhabitation for intersection-zero types. Urzyczyn proves in [50] that the inhabitation of intersection types with variables is undecidable.

It should be clear from this introduction that the present work leaves many questions unanswered or in wait for more satisfactory answers. We hope that this will stimulate further research towards the challenging goal of finding a flexible theory of filter models able to take into account the inner behavior of unsolvable terms.

2. Infinite λ-calculus and approximants

We assume familiarity with the λ-calculus: a standard reference is [6]. As usual Λ is the set of terms generated by the grammar

$$M ::= x \mid (\lambda x.M) \mid (MM),$$

where x ranges over a denumerable set Var of variables. We define $\Lambda(\bot)$ exactly as Λ but adding the clause that the constant $\bot$ is a term. For any $M \in \Lambda(\bot)$, $FV(M)$ denotes the set of free variables of M. A term M is closed iff $FV(M) = \emptyset$. We consider terms modulo α-conversion (renaming of bound variables).

Notation. We use $\equiv$ for syntactical equality up to renaming of bound variables, $=_\beta$ for β-conversion, $\rightarrow_\beta$ for (one step) β-reduction, and $\rightarrow_\beta^*$ for multistep β-reduction.

As usual for pure λ-calculus, we assume that application associates to the left and we write for instance MNP instead of $((MN)P)$. If $\vec{L} \equiv L_1 \cdots L_n$ is any (possibly empty) vector of terms, then $M\vec{L} \equiv ML_1 \ldots L_n$. The expression $\lambda x_1 \ldots x_n.M$ is short for $(\lambda x_1.(\ldots(\lambda x_n.M)\ldots))$.

We will abbreviate some terms as follows:

$$\begin{array}{ll}
\mathbf{I} \equiv \lambda x.x & \mathbf{1} \equiv \lambda xy.xy \\
\mathbf{K} \equiv \lambda xy.x & \mathbf{Y} \equiv \lambda y.(\lambda x.y(xx))(\lambda x.y(xx)) \\
\mathbf{\Delta}_2 \equiv \lambda x.xx & \mathbf{\Delta}_3 \equiv \lambda x.xxx \\
\mathbf{\Omega}_2 \equiv \mathbf{\Delta}_2\mathbf{\Delta}_2 & \mathbf{\Omega}_3 \equiv \mathbf{\Delta}_3\mathbf{\Delta}_3.
\end{array}$$

We introduce some notions following [7]. The following definitions apply both to $\Lambda(\bot)$ and to Λ.

Definition 2.1. A *zero term* is a term which cannot be β-reduced to an abstraction term, i.e. to a term of the form $\lambda x.M$.

Examples of such terms are $\mathbf{\Omega}_2$ and $\mathbf{\Omega}_3$. In [6, Definition 17.3.2] zero terms are called *terms of order zero*.

Definition 2.2. A *top normal form* (t.n.f.) is a term of one of the following three kinds:

(1) a variable;
(2) an abstraction term $\lambda x.M$;
(3) an application term of the form MN where M is a zero term.

It is easy to see that any β-reduct of a t.n.f. is again a t.n.f., and moreover it is a t.n.f. of the same kind. To see this it suffices to observe that if M is a zero term, then any β-reduct of MN has the form $M'N'$ with $M \rightarrow_\beta^* M'$ and $N \rightarrow_\beta^* N'$. We say that a term *has a top normal form* if it can be reduced to a term in top normal form.

Definition 2.3. A *mute term* is a term which has no top normal form.

For instance $\mathbf{\Omega}_2$ is mute. The β-reduction $\mathbf{\Omega}_3 \rightarrow_\beta^* \mathbf{\Omega}_3\mathbf{\Delta}_3$ shows that the zero term $\mathbf{\Omega}_3$ has a top normal form and therefore it is not mute. These notions are not constructive: there is no algorithm to test whether a term is a zero term.

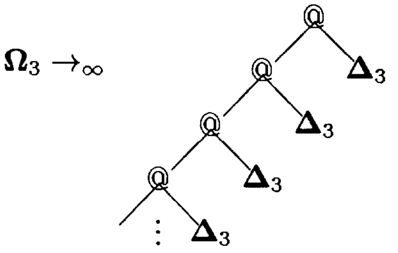


Fig. 1.

Definition 2.4. A *strong zero term* is a zero term M such that every substitution instance of M is again a zero term.

Clearly, a closed term is a zero term iff it is a strong zero term. A variable is a zero term but not a strong one. More generally it can be shown that M is a strong zero term iff M is a zero term and M cannot be β-reduced to a λ-free head normal form, i.e. to a term of the form $xM_1 \ldots M_n$ where x is a variable [7, Definition 11.7, Lemma 11.8]. Thus, M is a strong zero term iff M is a zero term and M is unsolvable. In [34] strong zero terms are called terms of proper order zero.

Definition 2.5. We identify terms with their parsing trees. So a term is a finite rooted tree with binary application nodes "@", unary abstraction nodes "λx" (where x is any variable), and leaves labeled by a variable.

An *infinite term* is defined in the same way except that trees are allowed to be finite or infinite, and leaves are labeled by a variable or by the constant $\perp$.

The infinite terms include as special cases the finite ones. We will see that the constant $\perp$ plays the role of a generic mute term to which all the others will be identified. Unlike what happens in the theory of Böhm trees, in the infinite λ-calculus, $\perp$, $\lambda x.\perp$ and $\perp M$ are not identified.

Infinite terms arise in a natural way as "limits" of infinite sequences of β-reductions if we try to compute top normal forms hereditarily. So if we start with $\mathbf{\Omega}_3$ we obtain the top normal form $\mathbf{\Omega}_3\mathbf{\Delta}_3$ and continuing in the same fashion we generate the infinite sequence of β-reductions $\mathbf{\Omega}_3 \rightarrow_\beta^* \mathbf{\Omega}_3\mathbf{\Delta}_3 \rightarrow_\beta^* \mathbf{\Omega}_3\mathbf{\Delta}_3\mathbf{\Delta}_3 \rightarrow_\beta^*$, etc. It is natural to take some kind of limit of this process and to set $\mathbf{\Omega}_3 \rightarrow_\infty \ldots \mathbf{\Delta}_3\mathbf{\Delta}_3\mathbf{\Delta}_3\mathbf{\Delta}_3$ (infinitely many $\mathbf{\Delta}_3$'s). In tree-form, it is represented as shown in Fig. 1.

The infinite term on the right is a normal form because it has no β-redexes (i.e. subterms of the shape $(\lambda x.M)N$). It is called the infinite normal form of $\mathbf{\Omega}_3$.

Among infinite terms we have both a notion of finite β-reduction $\rightarrow_\beta^*$ (defined in exactly the same way as for finite terms), and a notion of infinite β-reduction $\rightarrow_\infty$ defined as follows. Given two infinite terms M and N, we say $M \equiv_n N$ if the tree-representations of M and N coincide up to height n. This notion is slightly ambiguous since it is not invariant under α-conversion, however it can be made precise by identifying terms with their representation using de Brujin indexes [10]. Let $\langle M_i | i \geqslant 0\rangle$ be a sequence of infinite terms. We say $\lim\langle M_i\rangle = M$ if M is an infinite term, and $\forall n \exists i \forall m \geqslant i \;\; M_m \equiv_n M$.

Definition 2.6. Let $s: M_0 \to_\beta M_1 \to_\beta M_2 \to_\beta M_3 \to_\beta \cdots$ be an infinite sequence of β-reductions. We say that s *converges* to the infinite term M if
(1) $M = \lim\langle M_i\rangle$;
(2) the depth of the redex reduced in $M_i \to_\beta M_{i+1}$ tends to infinity with i.[2]

We now define $M \to_\infty N$ (*infinite β-reduction*) if and only if either $M \to_\beta^* N$ or there is an infinite sequence of reductions starting from M and converging to N.

The next example shows that the Church–Rosser property fails for infinite β-reductions.

Example 2.7. (Berarducci [7]) Let $Q \equiv \lambda x.\mathbf{I}(xx)$. We have the reductions

$$\begin{tikzcd}
QQ \arrow[r, "\beta"] \arrow[d, "\beta"', "*"] & \mathbf{I}(QQ) \arrow[r, "\beta"] \arrow[d, "\beta"', "*"] & \mathbf{I}(\mathbf{I}(QQ)) \arrow[r, "\infty"] \arrow[d, "\beta"', "*"] & \mathbf{I}(\mathbf{I}(\mathbf{I}(\mathbf{I}(\ldots)))) \\
\Delta_2\Delta_2 \arrow[r, "\beta"] & \Delta_2\Delta_2 \arrow[r, "\beta"] & \Delta_2\Delta_2 \quad \ldots & ?
\end{tikzcd}$$

but there is no reduction, whether finite or infinite, from $\mathbf{I}(\mathbf{I}(\mathbf{I}(\mathbf{I}(\ldots))))$ (infinitely many I's) to $\Delta_2\Delta_2$.

The term QQ responsible for the failure of the Church–Rosser property is mute. Mute terms are the only responsible for the failure of the Church–Rosser property in the sense that if we send all mute terms to $\perp$, then the Church-Rosser theorem is restored. To state this fact precisely, let us first observe that the notion of zero term, strong zero term and mute term extend to the infinite λ-calculus with the same definitions. So, for instance, a zero term is an infinite term M such that there is no *finite* β-reduction of the form $M \to_\beta^* \lambda x.P$. Actually it does not matter whether in the definitions we use finite or infinite β-reductions. This depends on the fact that if $M \to_\infty N$ and N is a zero term, then M is also a zero term. (When N is a normal form this is proved in [7, Lemma 9.3]. The general case follows easily.) According to the definitions $\perp$ is a mute term, since it can be reduced neither to an abstraction term, nor to a zero term applied to another term.

Following [5, Definition 3.1.1] we say that a *reduction relation* on the set of terms is a reflexive and transitive binary relation $\mathbf{R}$ which is *compatible* in the sense that $(F,F') \in \mathbf{R}$ implies $(FG,F'G) \in \mathbf{R}$, $(GF,GF') \in \mathbf{R}$ and $(\lambda x.F, \lambda x.F') \in \mathbf{R}$.

Definition 2.8. Define a *$\perp$-redex* as a mute term different from $\perp$. Define $\to_\perp$ as the least reduction relation on $\Lambda(\perp)$ which sends all the mute terms different from $\perp$ to $\perp$.

---

[2] The depth of an occurrence of a subterm P of M is the distance from the root of P to the root of M in the tree-representation of M.

Define $\rightarrow_{\beta\perp}$ as the least reduction relation on $\Lambda(\perp)$ which contains $\rightarrow_{\beta}$ (β-reduction) and $\rightarrow_{\perp}$.Now define $\rightarrow_{\infty\perp}$ exactly as $\rightarrow_{\infty}$ but starting from $\rightarrow_{\beta\perp}$ instead of $\rightarrow_{\beta}$.

We define $\rightarrow_{\beta\perp}{}^{*}$ as the transitive and reflexive closure of $\rightarrow_{\beta\perp}$, $=_{\beta\perp}$ as the equivalence relation generated by $\rightarrow_{\beta\perp}$, $=_{\infty\perp}$ as the equivalence relation generated by $\rightarrow_{\infty\perp}$, etc.

Theorem 2.9. *$\rightarrow_{\infty\perp}$ is transitive, it has the Church–Rosser property, and every term M has one and only one normal form with respect to $\rightarrow_{\infty\perp}$ reductions.*

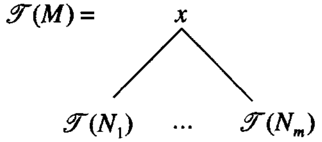
The above theorem was proved in [7] only for infinite terms M arising from finite ones (which is the form in which we need it). A proof for the general case can be found in [30]. We also need:

Theorem 2.10. (Berarducci [7]) *$\rightarrow_{\beta\perp}{}^{*}$ restricted to finite terms (i.e. to $\Lambda(\perp)$) has the Church-Rosser property.*

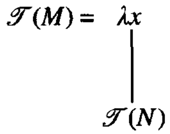
Definition 2.11. The *infinite normal form* ($\infty\perp$-normal form) of the term M is the normal form of M with respect to $\rightarrow_{\infty\perp}$.

In [7] the infinite tree of M is the parsing tree of the infinite normal form of M. We modify this representation in case of variable applications, following the convention of Böhm trees, since this simplifies our definition of the sets of approximants.

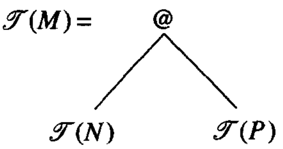
Definition 2.12. The *infinite tree* $\mathcal{T}(M)$ of the term M is defined by cases as follows:
if $M \rightarrow_{\beta}^{*} xN_1 \ldots N_m$ $(m \geqslant 0)$, then:



if $M \rightarrow_{\beta}^{*} \lambda x.N$, then:



if $M \rightarrow_{\beta}^{*} NP$, where N is a strong zero term, then:



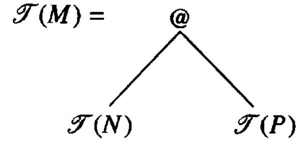
otherwise:

$$\mathscr{T}(M) = \bullet\ \bot$$

The fact that the infinite tree of a term is well defined, namely it does not depend on the choice of the reductions $\to_\beta^*$ involved in its definition, follows from the unicity of the infinite normal form of a term (and the obvious one to one correspondence between the infinite trees and the infinite normal forms).

Remark 2.13. $M =_{\infty\bot} N$ if and only if M and N have the same infinite normal form, or equivalently the same infinite tree.

Remark 2.14. From the equality



it does not follow that $M \to_\beta^* NP$. However it does follow that we have a reduction $M \to_\beta^* N'P'$ with $\mathscr{T}(N') = \mathscr{T}(N)$ and $\mathscr{T}(P') = \mathscr{T}(P)$. In fact, any reduction from M to a top normal form will work (by the unicity of the infinite normal form). A similar remark applies to the other cases in the definition of the infinite tree.

To study the infinite λ-calculus we introduce a notion of "approximate normal form". Approximate normal forms arise naturally by pruning infinite trees. Notice that the left son of a @-node must always be a tree representing a strong zero term, while in all other positions we can have trees representing arbitrary terms. So we add two constants to Λ, the constant Ω which approximates all terms and the constant $\bot$ which approximates all strong zero terms.

Definition 2.15. The set $\mathscr{A}$ of *approximate normal forms* is defined inductively as follows:
(1) $\Omega \in \mathscr{A}$,
(2) if $A \in \mathscr{A}$, then $\lambda x.A \in \mathscr{A}$,
(3) if $A_1,\ldots,A_n \in \mathscr{A}$, then $xA_1 \ldots A_n \in \mathscr{A}$ and $\bot A_1 \ldots A_n \in \mathscr{A}$ $(n \geqslant 0)$.

The definition of infinite tree easily extends to approximate normal forms by allowing terminal nodes labeled Ω.

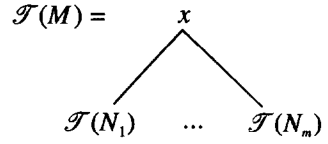
Definition 2.16. The relation $\preceq$ is the least preorder on $\mathscr{A}$, such that
(1) $\Omega \preceq A$;
(2) $\bot \preceq \bot A$;
(3) if $A \preceq A'$, then $\lambda x.A \preceq \lambda x.A'$;
(4) if $A_i \preceq A_i'$ for $i = 1,\ldots,n$, then $xA_1 \ldots A_n \preceq xA_1' \ldots A_n'$ and $\bot A_1 \ldots A_n \preceq \bot A_1' \ldots A_n'$.

We want to consider approximate normal forms obtained by cutting infinite trees at some level n. The first guess would be to replace all internal nodes at level n with Ω but this does not work because we may end up with objects of the form ΩA, which are not approximate normal forms. Roughly $\perp$ replaces @-nodes, while Ω replaces arbitrary nodes. The exact definition is the following.

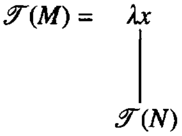
Definition 2.17. Let M be either an approximant or a term. The *n-th approximant* $(M)^n$ of M is the approximate normal form defined as follows.

$(M)^0 = \Omega$.

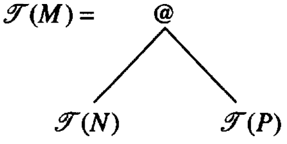
$(M)^{n+1} = x(N_1)^n \ldots (N_m)^n$ $(m \geqslant 0)$, whenever:



$(M)^{n+1} = \lambda x.(N)^n$, whenever:



$(M)^1 = \perp$ and $(M)^{n+2} = (N)^{n+1}(P)^{n+1}$, whenever:



$(M)^{n+1} = \perp$, whenever:

$\mathcal{T}(M) = \bullet \perp$

$(M)^{n+1} = \Omega$, whenever:

$\mathcal{T}(M) = \bullet \Omega$

We say that M and N *coincide on the first n levels* if $(M)^n \equiv (N)^n$.

It is easy to check that $(M)^n \equiv (N)^n$ implies $M \equiv_n N$, while the vice versa is not true, since for example $x\mathbf{II} \equiv_1 y\mathbf{II}$ (because in the parsing trees of $x\mathbf{II}$ and $y\mathbf{II}$ the variables x, y have depth 2) and $(x\mathbf{II})^1 \equiv x\Omega\Omega$, $(y\mathbf{II})^1 \equiv y\Omega\Omega$.

Example 2.18. The n-approximants of $\mathbf{\Omega}_3$ for $n = 0,\ldots,4$ are: $\Omega \preceq \bot \preceq \bot(\lambda x.\Omega) \preceq \bot(\lambda x.\Omega)(\lambda x.x\Omega\Omega) \preceq \bot(\lambda x.\Omega)(\lambda x.x\Omega\Omega)(\lambda x.xxx)$.

The set of approximants of M is the closure under $\preceq$ of the set of all terms of the form $(M)^n$ for all n.

Definition 2.19. The *set* $\mathscr{A}(M)$ *of approximants of* $M \in \Lambda$ is defined by

$$\mathscr{A}(M) = \{A \in \mathscr{A} \mid \exists n. A \preceq (M)^n\}.$$

For example, the term $x(\mathbf{\Omega}_3\mathbf{I})(\mathbf{II})$ has the following approximants: $\Omega \preceq x\Omega\Omega \preceq x(\bot\mathbf{I})\mathbf{I} \preceq x(\bot\mathbf{\Delta}_3\mathbf{I})\mathbf{I} \preceq x(\bot\mathbf{\Delta}_3\mathbf{\Delta}_3\mathbf{I})\mathbf{I} \preceq \cdots$.

Lemma 2.20. *The* set $\mathscr{A}(M)$ *is an ideal, i.e. it is downward closed and directed with respect to* $\preceq$.

Proof. $\mathscr{A}(M)$ is downward closed by definition. Notice that $(M)^n \preceq (M)^{n+1}$ for all n. The fact that $\mathscr{A}(M)$ is directed for all M follows easily. □

It is easy to verify that the sets of approximants characterize the mute, strong zero and zero terms.

Lemma 2.21. (1) *A term* M *is mute iff* $\mathscr{A}(M) = \{\bot, \Omega\}$.
(2) *A term* M *is strong zero iff* $\bot \in \mathscr{A}(M)$.
(3) *A term* M *is zero iff* $\lambda x.\Omega \notin \mathscr{A}(M)$.

Proof. Clearly $\Omega \in \mathscr{A}(M)$ for all M. Now if M is a mute term, i.e. a term without t.n.f., we have that its infinite tree is just a root labeled $\bot$. M is a strong zero term iff its infinite tree is just a root labeled $\bot$ or starts with a @-node. In both cases $(M)^1 \equiv \bot$. $\lambda x.\Omega \in \mathscr{A}(M)$ iff the root of $\mathscr{T}(M)$ is labeled λx, and so M is not a zero term. □

Lemma 2.22. *Let* $M \in \Lambda$. *If* $A \in \mathscr{A}(M)$ *and* $A \succeq (M)^n$, *then* $(A)^n \equiv (M)^n$.

Proof. By induction on n and by cases on Definition 2.16. The most interesting case is $(M)^n \equiv \bot$ and $A \equiv \bot B$. If $\bot B \in \mathscr{A}(M)$ then $M =_\beta PN$ for some P, N such that P is a strong zero term. $(PN)^n \equiv \bot$ implies $n = 1$ and $(\bot B)^1 \equiv \bot$. □

It is clear that the sets of approximants characterize the infinite normal forms of terms.

Theorem 2.23. $M =_{\infty\bot} N$ *if and only if* $\mathscr{A}(M) = \mathscr{A}(N)$.

3. Intersection-zero types and models of the λ-calculus

By model of λ-calculus we mean a λ-algebra in the sense of [5]. We quote Barendregt [5, pp. 86–87]:

> "It took some time after Scott gave his model construction for consensus to arise on the general notion of a model of the λ-calculus. See Koymans [1982] for the history. Presently one considers two kinds of models, viz. the λ-algebras and the λ-models. The λ-algebras satisfy all provable equations of the λ-calculus and form an equational class (axiomatized by $\mathsf{k}xy = x, \mathsf{s}xyz = xz(yz)$ and the five combinatory axioms of Curry). Therefore the λ-algebras are closed under substructures and homomorphic images. The λ-models on the other hand satisfy all provable equations and moreover the axiom of weak extensionality $\forall x(Mx = Nx) \to \lambda x.M = \lambda x.N$. It turns out that λ-models can be described by first-order axioms, but not by equations. Indeed λ-models are not closed under substructures nor under homomorphic images."

Then Barendregt continues:

> "In spite of not being weakly extensional, λ-algebras are worth studying; they are e.g. precomplete numbered sets in the sense of Ershov, see [Visser [1980]."

So we feel justified to use λ-algebras rather than λ-models in the proof of the completeness theorem (of course with a larger class of models it is easier to prove a completeness theorem). It should be remarked that the so called "closed term model of the $\lambda\beta$-calculus", is a λ-algebra and not a λ-model. The model that we will construct to prove the completeness theorem for our type assignment system is in fact very similar to the closed term model.

We give now the precise definition of λ-algebra using the approach of the "syntactical interpretations" of Hindley and Longo (see [5, Definitions 5.3.1 and 5.3.2]).[3]

Definition 3.1. A *λ-algebra* is a triple $(\mathscr{M}, \cdot, [\![\ ]\!]^{\mathscr{M}})$ where $\mathscr{M}$ is a non-empty set called the *domain* of the model, $\cdot$ is a binary operation on $\mathscr{M}$ called *application* and $[\![\ ]\!]^{\mathscr{M}}$ is an interpretation map which associates to every term $M \in \Lambda$ and every mapping $s : Var \to \mathscr{M}$, an element $[\![M]\!]_s^{\mathscr{M}} \in \mathscr{M}$, called the *interpretation* of the term M in the *environment* s, in such a way that the following axioms are satisfied:

(1) $[\![x]\!]_s^{\mathscr{M}} = s(x)$;

(2) $[\![MN]\!]_s^{\mathscr{M}} = [\![M]\!]_s^{\mathscr{M}} \cdot [\![N]\!]_s^{\mathscr{M}}$;

(3) $[\![\lambda x.M]\!]_s^{\mathscr{M}} \cdot a = [\![M]\!]_{s(a/x)}^{\mathscr{M}}$ where $s(a/x)$ is the environment which coincides with s except that it associates the value $a \in \mathscr{M}$ to the variable x;

(4) if s and s' coincide on the free variables of M, then $[\![M]\!]_s^{\mathscr{M}} = [\![M]\!]_{s'}^{\mathscr{M}}$;

(5) if $M =_\beta N$, then for every s we have $[\![M]\!]_s^{\mathscr{M}} = [\![N]\!]_s^{\mathscr{M}}$.

---

[3] There is a very simple definition of λ-model due to Meyer [36, 20, Definition 11.22], which however is not adequate for our purposes since we are interested in λ-algebras, not λ-models.

The first four axioms plus the invariance of the interpretation under α-conversion are the axioms of the *syntactic interpretation*, the last one is the characteristic axiom of the λ-algebras. It expresses the fact that a λ-algebra satisfies all the provable equalities of the $\lambda\beta$-calculus (for every interpretation of the free variables).

The notion of λ-model is obtained by adjoining the axiom (ξ) which says that if $[\![M]\!]^{\mathcal{M}}_{s(a/x)} = [\![N]\!]^{\mathcal{M}}_{s(a/x)}$ for every $a \in \mathcal{M}$, then $[\![\lambda x.M]\!]^{\mathcal{M}}_{s} = [\![\lambda x.N]\!]^{\mathcal{M}}_{s}$. In presence of the axiom (ξ) the characteristic axiom for λ-algebras becomes superfluous, since it can be derived. We follow Barendregt's convention [5, Convention 5.3.8] of writing equations valid in a λ-algebra informally, e.g. for $a \in \mathcal{M}$ one writes $(\lambda x.xx) \cdot a = a \cdot a$ rather than the formal $[\![(\lambda x.xx)y]\!]^{\mathcal{M}}_{s(a/y)} = [\![yy]\!]^{\mathcal{M}}_{s(a/y)}$ or $[\![\lambda x.xx]\!]^{\mathcal{M}} \cdot a = a \cdot a$. So in particular we have $\mathbf{1} \cdot a = \lambda y.a \cdot y$.

Notation. A λ-algebra whose domain is $\mathcal{M}$ will be denoted by $\mathcal{M}$.

We want to generalize the notion of zero term (Definition 2.1) to an arbitrary λ-algebra. An obvious generalization is: $a \in \mathcal{M}$ is a zero element of $\mathcal{M}$ iff $a \neq \mathbf{1} \cdot a$. However it turns out to be convenient to impose a further condition. Notice that for each zero term M the function $N \mapsto MN$ is injective. Therefore an equivalent definition of zero term is

$$M \text{ is a zero term iff } M \neq_\beta \mathbf{1}M \text{ and } MN =_\beta MN' \text{ implies } N =_\beta N'.$$

The above discussion leads us to the following definition.

Definition 3.2. Given a λ-algebra $\mathcal{M}$ and $a \in \mathcal{M}$, we say that a is a *zero element* of $\mathcal{M}$ if it satisfies
(1) $a \neq \mathbf{1} \cdot a$;
(2) for all $b, b' \in \mathcal{M}$, $a \cdot b = a \cdot b'$ implies $b = b'$.

Other properties of zero terms which can be easily verified are:
(1) if M is a zero term and P is any term, then MP is a zero term;
(2) if M, N are zero terms, P, Q are arbitrary terms, and $MP =_\beta NQ$, then $M =_\beta N$ and $P =_\beta Q$.
These properties involve the full set of zero terms, so we will require them in defining the adequacy of λ-algebras (Definition (3.3)).

Recall that a strong zero term is a zero term N such that every substitution instance of N is a zero term. Even this notion can be generalized to an arbitrary model of λ-calculus using homomorphisms instead of substitutions. However we do not need this generalization. What we need in connection with strong zero terms is part of the following definition.

Definition 3.3. A *λ-algebra $\mathcal{M}$* is *adequate* if the following conditions are satisfied:
(1) if a is a zero element and b is any element, then $a \cdot b$ is a zero element;

(2) if a, a' are zero elements, b, b' are arbitrary elements, and $a \cdot b = a' \cdot b'$, then $a = a'$ and $b = b'$.
(3) for all strong zero terms $N \in \Lambda$, and all s: $Var \to \mathcal{M}$, $[\![N]\!]_s^{\mathcal{M}}$ is a zero element of $\mathcal{M}$.

Notice that clause (2) of Definition (3.3) implies clause (2) of Definition 3.2.

In an adequate model the interpretation of a strong zero term is a zero element of the model. This is not true in a model satisfying the equality $\mathbf{\Omega}_2 = \lambda x.\mathbf{\Omega}_2$ or the equality $\mathbf{\Omega}_2 = \mathbf{\Omega}_2 M$ for all M. For instance the models of Böhm trees [5] and that of Lévy–Longo trees [34] are not adequate, while the (open or closed) term model of the $\lambda\beta$-calculus and that of the infinite λ-calculus [7] are adequate.

Our next goal is to introduce types and to discuss their interpretation in adequate λ-models.

Definition 3.4. The set **PTypes** of *pretypes* is the set of syntactic expressions inductively defined by
(1) $\omega, \zeta \in$ **PTypes** (type constants).
(2) If $\alpha, \beta \in$ **PTypes**, then $(\alpha \to \beta)$, $(\alpha\beta)$ and $(\alpha \wedge \beta)$ are in **PTypes**.

To spare parentheses, we assume that the arrow associates to the right and the following precedence between type constructors: application, intersection, arrow.

We say that $\alpha \to \beta$ is an arrow pretype, $\alpha\beta$ is an zero pretype, and $\alpha \wedge \beta$ is an intersection pretype.

The definition of **Types** $\subset$ **PTypes** can be better justified by first looking at the following notion of interpretation of **PTypes**. We interpret a zero pretype of the shape $\alpha\beta$ as the set of zero elements which are equal to the application of two elements belonging respectively to the interpretations of α and β.

Definition 3.5. Let $\mathcal{M}$ be a λ-algebra. The *interpretation of the pretype* α in $\mathcal{M}$ is the subset $[\![\alpha]\!]^{\mathcal{M}}$ of $\mathcal{M}$ inductively defined as follows:
(1) $[\![\omega]\!]^{\mathcal{M}} = \mathcal{M}$,
(2) $[\![\zeta]\!]^{\mathcal{M}} = \{d \in \mathcal{M} \mid d \text{ is a zero element}\}$,
(3) $[\![\alpha \to \beta]\!]^{\mathcal{M}} = \{d \in \mathcal{M} \mid d = \mathbf{1} \cdot d \text{ and } \forall e \in [\![\alpha]\!]^{\mathcal{M}}\ d \cdot e \in [\![\beta]\!]^{\mathcal{M}}\}$,
(4) $[\![\alpha\beta]\!]^{\mathcal{M}} = \{d \in \mathcal{M} \mid \exists e \in [\![\alpha]\!]^{\mathcal{M}} \exists f \in [\![\beta]\!]^{\mathcal{M}}\ e \cdot f = d\}$,
(5) $[\![\alpha \wedge \beta]\!]^{\mathcal{M}} = [\![\alpha]\!]^{\mathcal{M}} \cap [\![\beta]\!]^{\mathcal{M}}$.

We want to forbid pretypes whose interpretation is always empty, for example $\zeta \wedge (\omega \to \omega)$. We therefore restrict ourselves to a smaller set **Types** $\subset$ **PTypes**. The following definition of **Types** also assigns a kind to each type. There are three kinds: universal, arrow and zero. The types of kind universal are finite intersections of the constant ω. A finite intersection of arrow types is of arrow kind whenever it is a type. Analogously, a finite intersection of zero types is of zero kind whenever it is a type. Moreover, the intersection between ω and a type α is always a type of the same kind as α.

Notice that in the following definition we assume that intersection between pretypes is commutative and associative (this is justified by its interpretation, see clause (5) of Definition 3.5).

Notation. In $\bigwedge_{i\in I}\alpha_i$, I stands for a finite non-empty set of indexes.

Definition 3.6. Given $\alpha\in\mathbf{PTypes}$ we define two predicates $\alpha\in\mathbf{Types}$ and $\alpha\notin\mathbf{Types}$ by simultaneous induction on α by stipulating that $\alpha\in\mathbf{Types}$ iff one of the following conditions holds (and $\alpha\notin\mathbf{Types}$ iff all the conditions do not hold):

- (Universal kind) α is ω.
- (Arrow kind) α is a finite intersection of the form $\bigwedge_{i\in I}(\alpha_i\to\beta_i)$ where $\alpha_i,\beta_i\in\mathbf{Types}$ and for all $J\subseteq I$ either $\bigwedge_{j\in J}\beta_j\in\mathbf{Types}$ or $\bigwedge_{j\in J}\alpha_j\notin\mathbf{Types}$.
- (Zero kind) α is ζ or $\zeta\wedge\beta$, where β is of zero kind, or α is a finite intersection of the form $\bigwedge_{i\in I}\alpha_i\beta_i$ where $\bigwedge_{i\in I}\beta_i\in\mathbf{Types}$, and $\bigwedge_{i\in I}\alpha_i\in\mathbf{Types}$ is of zero kind.
- If $\alpha\in\mathbf{Types}$, then $\omega\wedge\alpha\in\mathbf{Types}$: the kind of $\omega\wedge\alpha$ is the kind of α.

In the above definition, the condition on types of arrow kind clearly corresponds to the existence of the sup of step functions (see [47]). The condition on types of zero kind implies for instance that if $\alpha\beta$ is a type, then α is of zero kind. Moreover, if $\alpha_1\beta_1\wedge\alpha_2\beta_2$ is a type, then $\alpha_1\wedge\alpha_2$ must be a type of zero kind and $\beta_1\wedge\beta_2$ must be a type (of any kind). We will see (Lemma 5.2) that clause (4) of Definition 3.5 implies that under any interpretation in an adequate model we have $[\![\alpha_1\beta_1\wedge\alpha_2\beta_2]\!]^{\mathscr{M}}\subseteq[\![(\alpha_1\wedge\alpha_2)(\beta_1\wedge\beta_2)]\!]^{\mathscr{M}}$. The monotonicity of the application and of the intersection implies also the reverse inclusion, so we have $[\![\alpha_1\beta_1\wedge\alpha_2\beta_2]\!]^{\mathscr{M}}=[\![(\alpha_1\wedge\alpha_2)(\beta_1\wedge\beta_2)]\!]^{\mathscr{M}}$.

In what follows, we will consider only types. α, β, γ will range over types of any kind, σ, τ, ρ will range over types of arrow kind (*arrow types*), π, μ, ν will range over types of zero kind (*zero types*).

As usual, we can connect term and type interpretations in λ-algebras obtaining the notion of semantic satisfiability.

Definition 3.7. A *basis* Γ is a (finite or infinite) set of statements of the shape $x:\alpha$, with distinct variables as subjects. In writing $\Gamma,x:\alpha$ we assume that x does not occur in Γ.

An *interpretation* is a pair $(\mathscr{M},s)$ where: $\mathscr{M}$ is a λ-algebra, and s is a map from the variables of the λ-calculus to $\mathscr{M}$.

An interpretation is *adequate* if $\mathscr{M}$ is an adequate λ-algebra.

Definition 3.8. (1) An interpretation $(\mathscr{M},s)$ satisfies Γ – written $(\mathscr{M},s)\models\Gamma$ – iff for all $x:\alpha\in\Gamma$, the element $s(x)$ belongs to the set $[\![\alpha]\!]^{\mathscr{M}}$.

(2) Given a basis Γ we define $\Gamma\models N:\alpha$ iff for every *adequate* interpretation $(\mathscr{M},s)$ such that $(\mathscr{M},s)\models\Gamma$, we have $[\![N]\!]^{\mathscr{M}}_s\in[\![\alpha]\!]^{\mathscr{M}}$.

(3) Given an adequate λ-algebra $\mathscr{M}$ we define $\mathscr{M}\models N:\alpha$ iff for every s we have $[\![N]\!]^{\mathscr{M}}_s\in[\![\alpha]\!]^{\mathscr{M}}$.

We extend the definition of $\models$ to approximants:

Definition 3.9. Let $A \in \mathscr{A}$.

(1) Given a basis Γ we define $\Gamma \models A:\alpha$ iff for all $M \in \Lambda$, if $A \in \mathscr{A}(M)$, then $\Gamma \models M:\alpha$.

(2) Given an adequate λ-algebra $\mathscr{M}$ we define $\mathscr{M} \models A:\alpha$ iff for all $M \in \Lambda$, if $A \in \mathscr{A}(M)$, then $\mathscr{M} \models M:\alpha$.

4. Non-continuity results

The equivalence classes of terms modulo $=_{\infty\perp}$ form a model of the $\lambda\beta$-calculus which is discussed in [7]. We can put a topology on this model using the approximate normal forms. A basic open set is a set of the form $\{M|A \in \mathscr{A}(M)\}$ where $A \in \mathscr{A}$. The next theorem shows that unlike what happens in the model of Böhm trees, application is non-continuous in this model in fact we show that it is not continuous in the first argument. To determine whether a given $A \in \mathscr{A}$ is an approximant of MN, it is in general not sufficient to consider a large enough finite portion of $\mathscr{A}(M)$.

Theorem 4.1. *Application is non-continuous in the topology induced by the approximants.*

Proof. We show that $\perp \in \mathscr{A}(\mathbf{YY})$, but for every approximant $A \in \mathscr{A}(\mathbf{Y})$ there is a term M with this approximant and yet $\perp \notin \mathscr{A}(M\mathbf{Y})$.

It is easy to see that $\mathbf{YY}$ is a strong zero term because of the sequence of head reductions $\mathbf{YY} \to_\beta^* \mathbf{Y}(\mathbf{YY}) \to_\beta^* \mathbf{YY}(\mathbf{Y}(\mathbf{YY}))$.

Let A be an approximant of $\mathbf{Y}$. Then A is of the form $\lambda x.x(x(x(\dots x\Omega)))$. Replacing Ω with $\mathbf{YK}$, we obtain a term M with this approximant and such that $M\mathbf{Y}$ is not even a zero term. □

We show that our types are so expressive that there are types which, interpreted in the closed term model Λ_0^β of the $\lambda\beta$-calculus, are inhabited by only one term.

Definition 4.2. Let $\sigma_I = (\zeta \to \zeta) \wedge ((\omega \to \zeta) \to \omega \to \zeta)$.

Lemma 4.3. $\Lambda_0^\beta \models M:\sigma_I$ *iff* $M =_\beta \mathbf{I}$.

Proof. The "if" part is trivial. To prove the converse assume $\Lambda_0^\beta \models M:\sigma_I$. Then in particular $\Lambda_0^\beta \models M:\omega \to \omega$, so $M =_\beta \lambda x.P$ for some P.

If P is a strong zero term, then $\Lambda_0^\beta \not\models M:(\omega \to \zeta) \to \omega \to \zeta$ because taking an arbitrary closed term T of type $\omega \to \zeta$ (e.g. $T \equiv \lambda y.\mathbf{\Omega}_2$), we have $\Lambda_0^\beta \not\models MT : \omega \to \zeta$.

If P reduces to an abstraction $\lambda y.P'$, then $\Lambda_0^\beta \not\models M : \zeta \to \zeta$.

So P must be β-convertible either to x or to a term of the form $xP_1 \dots P_n$ with $n \geqslant 1$. To exclude the latter case we observe that if $n \geqslant 1$, $\Lambda_0^\beta \not\models \lambda x.xP_1 \dots P_n : (\omega \to \zeta) \to \omega \to \zeta$. □

We now prove the failure of the approximation theorem for the closed term model.

Theorem 4.4. *There is a closed term M and a type α such that $\Lambda_0^\beta \models M:\alpha$, but for no approximant A of M we have $\Lambda_0^\beta \models A:\alpha$.*

Proof. We show that $\Lambda_0^\beta \models \lambda x.x\mathbf{YY}:\sigma_I \to \zeta$ but for no approximant A of $\lambda x.x\mathbf{YY}$ we have $\Lambda_0^\beta \models A:\sigma_I \to \zeta$. Since $\mathbf{YY}$ is a zero term and $\mathbf{I}$ is the only term of type σ_I, $\Lambda_0^\beta \models \lambda x.x\mathbf{YY}:\sigma_I \to \zeta$.

Consider now an approximant A of $\lambda x.x\mathbf{YY}$. To avoid trivial cases we can assume that $A \equiv \lambda x.xA_1A_2$, where A_1 and A_2 are approximants of $\mathbf{Y}$. By the proof of Theorem 4.1 there are terms M,N with these approximants and such that MN is not a zero term. It then follows that $\Lambda_0^\beta \not\models \lambda x.xMN:\sigma_I \to \zeta$ and by Definition 3.9 $\Lambda_0^\beta \not\models A:\sigma_I \to \zeta$. □

5. The type assignment systems for terms

We will define two type assignment systems, one for terms in the present section and one for approximate normal forms in the following section. We will then prove in Section 7 the approximation theorem: a type can be assigned to a term iff it can be assigned to one of its approximants. Since the approximation theorem does not hold for the closed term model Λ_0^β, the type assignment system for terms will not be complete with respect to Λ_0^β. However, this system will be useful to characterize the $=_{\infty\perp}$-convertibility of terms (see Section 8) and it turns out to be complete with respect to adequate λ-algebras (see Section 9). We begin by defining a preorder between types.

Definition 5.1. Let $\leqslant$ be the smallest binary relation over types such that:

(1) $\leqslant$ is a preorder in which $\wedge$ is the meet and ω is the top:
  (a) $\alpha \leqslant \alpha$;
  (b) $\alpha \leqslant \beta$ and $\beta \leqslant \gamma$ imply $\alpha \leqslant \gamma$;
  (c) $\alpha \leqslant \omega$;
  (d) $\alpha \wedge \beta \leqslant \alpha$ and $\alpha \wedge \beta \leqslant \beta$;
  (e) $\gamma \leqslant \alpha$ and $\gamma \leqslant \beta$ imply $\gamma \leqslant \alpha \wedge \beta$.
(2) the arrow satisfies:
  (a) $\alpha \to \omega \leqslant \omega \to \omega$;
  (b) $(\alpha \to \beta) \wedge (\alpha \to \gamma) \leqslant \alpha \to \beta \wedge \gamma$;
  (c) $\alpha \geqslant \alpha'$ and $\beta \leqslant \beta'$ imply $\alpha \to \beta \leqslant \alpha' \to \beta'$.
(3) the zero types satisfy:
  (a) $\zeta\alpha \leqslant \zeta$;
  (b) $\pi\alpha \wedge \pi'\alpha' \leqslant (\pi \wedge \pi')(\alpha \wedge \alpha')$;
  (c) $\pi \leqslant \pi'$ and $\alpha \leqslant \alpha'$ imply $\pi\alpha \leqslant \pi'\alpha'$.

$\alpha = \beta$ is short for $\alpha \leqslant \beta$ and $\beta \leqslant \alpha$.

Notice that we have $\omega \wedge \alpha = \alpha$ for every type α, $(\omega \to \omega) \wedge \sigma = \sigma$ for every arrow type σ, and $\zeta \wedge \pi = \pi$ for every zero type π. Moreover, relations (2)(a), (2)(b), and (3)(b) in the above definition are equalities, the last one is Lemma 5.3(5).
In the following we will identify equivalent types, in particular in Lemma 6.2 we will omit intersections with ω.

The rules of Definition 5.1 are sound for adequate interpretations.

Lemma 5.2. (1) *If* $d \in [\![\pi]\!]^{\mathcal{M}}$, $\mathcal{M}$ *is an adequate* λ*-algebra, and* π *is a zero type, then* d *is a zero element of* $\mathcal{M}$.
(2) *If* $\alpha \leqslant \beta$, *then* $[\![\alpha]\!]^{\mathcal{M}} \subseteq [\![\beta]\!]^{\mathcal{M}}$ *under every adequate interpretation.*

Proof. (1) The proof is by induction on the definition of zero types observing that, by Definition 3.3, in an adequate λ-algebra $e \cdot f$ is a zero element whenever e is a zero element.

(2) By induction on $\leqslant$. The only interesting case is clause $\pi\alpha \wedge \pi'\alpha' \leqslant (\pi \wedge \pi')(\alpha \wedge \alpha')$. We have from $d \in [\![\pi\alpha \wedge \pi'\alpha']\!]^{\mathcal{M}}$ that $e \cdot f = d$ and $e' \cdot f' = d$ for some $e \in [\![\pi]\!]^{\mathcal{M}}$, $f \in [\![\alpha]\!]^{\mathcal{M}}$, $e' \in [\![\pi']\!]^{\mathcal{M}}$, $f' \in [\![\alpha']\!]^{\mathcal{M}}$. Notice that e and e' are zero elements by (1), therefore $e \cdot f = e' \cdot f'$ implies by Definition (3.3) $e = e'$ and $f = f'$. We get $e \in [\![\pi \wedge \pi']\!]^{\mathcal{M}}$ and $f \in [\![\alpha \wedge \alpha']\!]^{\mathcal{M}}$, so we conclude $d \in [\![(\pi \wedge \pi')(\alpha \wedge \alpha')]\!]^{\mathcal{M}}$. □

The relation $\leqslant$ between types has the following properties, which will be useful in the sequel.

Lemma 5.3. (1) *If* $\bigwedge_{i \in I}(\alpha_i \to \beta_i) \leqslant \gamma < \omega$, *or* $\gamma \leqslant \bigwedge_{i \in I}(\alpha_i \to \beta_i)$, *then* γ *is an arrow type.*
(2) *If* $\bigwedge_{i \in I}(\alpha_i \to \beta_i) \leqslant \alpha \to \beta$, *where* $\beta \neq \omega$, *then for some* $J \subseteq I$ *it holds that* $\alpha \leqslant \bigwedge_{j \in J} \alpha_j$ *and* $\bigwedge_{j \in J} \beta_j \leqslant \beta$.
(3) *If* $\pi \leqslant \alpha < \omega$, *or* $\alpha \leqslant \pi$, *then* α *is a zero type.*
(4) $\pi\alpha \leqslant \pi'\alpha'$ *implies* $\pi \leqslant \pi'$ *and* $\alpha \leqslant \alpha'$.
(5) $\pi\alpha \wedge \pi'\alpha' = (\pi \wedge \pi')(\alpha \wedge \alpha')$ *for all zero types* π, π'.
(6) *If* π *is a zero type, then either* $\pi = \zeta$ *or* $\pi = \mu\alpha$ *for some* μ, α.
(7) *Assume* $\beta \leqslant \alpha_1$ *and* $\beta \leqslant \alpha_2$. *If* $\beta \in$ **Types**, *then* $\alpha_1 \wedge \alpha_2 \in$ **Types**.

Proof. (1)–(4). By induction on the definition of $\leqslant$.

(5) In fact $\pi\alpha \wedge \pi'\alpha' \leqslant (\pi \wedge \pi')(\alpha \wedge \alpha')$ by clause (3)(b) in Definition 5.1. The converse follows from clauses (1)(d), (3)(c), and (1)(e) of the same definition.

(6) First observe that $\zeta \wedge \pi = \pi$ for all zero types π. So by (5) each intersection of zero types is either equal to ζ or to a zero type of the shape $\pi\alpha$.

(7) By cases using (1)–(4). □

The first six rules of our type system for terms are standard in the intersection type disciplines. Rules (ζ) and (*app*) take into account zero types. They oblige us to have rule (Eq_β), as otherwise the zero types would not be invariant under β-expansion of subjects. For example, without (Eq_β) we have $\vdash \mathbf{\Omega}_2\mathbf{I} : \zeta(\omega \to \omega)$, but $\nvdash (\lambda xy.y\mathbf{\Delta}_2 x)\mathbf{I}\mathbf{\Delta}_2 : \zeta(\omega \to \omega)$.

Definition 5.4. The type assignment system for terms is defined by the following axioms and rules (where M, N are terms):

$$(\mathrm{Ax})\,\Gamma, x:\alpha \vdash x:\alpha \qquad\qquad (\omega)\,\Gamma \vdash M:\omega$$

$$(\to \mathrm{I})\frac{\Gamma, x:\alpha \vdash M:\beta}{\Gamma \vdash \lambda x.M:\alpha \to \beta} \qquad\qquad (\to \mathrm{E})\frac{\Gamma \vdash M:\alpha \to \beta \quad \Gamma \vdash N:\alpha}{\Gamma \vdash MN:\beta}$$

$$(\wedge \mathrm{I})\frac{\Gamma \vdash M:\alpha \quad \Gamma \vdash M:\beta}{\Gamma \vdash M:\alpha \wedge \beta\text{[4]}} \qquad\qquad (\leqslant)\frac{\Gamma \vdash M:\alpha \quad \alpha \leqslant \beta}{\Gamma \vdash M:\beta}$$

$$(\zeta)\frac{M \textit{ is a strong zero term}}{\Gamma \vdash M:\zeta} \qquad\qquad (app)\frac{\Gamma \vdash M:\pi \quad \Gamma \vdash N:\alpha}{\Gamma \vdash MN:\pi\alpha}$$

$$(Eq_\beta)\quad \frac{\Gamma \vdash N:\alpha \quad M =_\beta N}{\Gamma \vdash M:\alpha}$$

Remark 5.5. As remarked in the introduction the following formulation of rule (ζ) would not be sound because the interpretation of a solvable zero term in some environment may not be a zero term:

$$\frac{M \text{ is a zero term}}{\Gamma \vdash M:\zeta}$$

We discarded this rule also because it would allow to deduce both ζ and $\omega \to \omega$ for x from the premise $x : \omega \to \omega$, but $\zeta \wedge (\omega \to \omega)$ is not a type.

It is easy to verify that the above system is sound.

Lemma 5.6. *If* $\Gamma \vdash M:\alpha$, *then* $\Gamma \models M:\alpha$ *according to Definition* 3.8.

Proof. By induction on deductions. All rules but (ζ) and (app) are standard. The soundness of rules (ζ) and (app) follows directly from the definition of type interpretation (Definition 3.5). □

Since terms are considered modulo α-conversion, the weakening rule is admissible. Moreover we have $\Gamma_{|M} \vdash M{:}\alpha$ whenever $\Gamma \vdash M{:}\alpha$, where $\Gamma_{|M} = \{x{:}\beta \in \Gamma \mid x \in FV(M)\}$.

We define $Dom(\Gamma) = \{x \mid x : \alpha \in \Gamma \text{ and } \alpha \neq \omega\}$ and we assume $x{:}\omega \in \Gamma$ whenever $x \notin Dom(\Gamma)$. This is sound in view of rule (ω).

In the following we shall sometimes refer to the "best" basis which can possibly be formed out of two given bases. This is done by taking the intersection of the types which are predicates of the same variable, when they are compatible.

---

[4] Lemma 7.6(1) will show that $\alpha \wedge \beta$ is always a type.

Definition 5.7. We say that two bases Γ and Γ' are *compatible* if and only if $x : \alpha \in \Gamma$ and $x : \beta \in \Gamma'$ imply $\alpha \wedge \beta \in \mathbf{Types}$. If Γ and Γ' are compatible bases we define their union $\uplus$ as

$$\Gamma \uplus \Gamma' = \{x : \alpha \wedge \beta \mid x : \alpha \in \Gamma \text{ and } x : \beta \in \Gamma'\}.$$

Accordingly we define

$$\Gamma \Subset \Gamma' \Leftrightarrow \exists \Gamma''. \Gamma \uplus \Gamma'' = \Gamma'.$$

Notice that $x : \alpha \wedge \omega \in \Gamma \uplus \Gamma'$ whenever $x : \alpha \in \Gamma$ and $x \notin Dom(\Gamma')$, since by convention we get $x{:}\omega \in \Gamma'$. Similarly when $x : \beta \in \Gamma'$ and $x \notin Dom(\Gamma)$.

The intersection type assignment systems have been used in the literature to build filter models, as discussed in the introduction. Also here we have a notion of filter and of term interpretation.

Definition 5.8. (1) A filter over **Types** is a set $f \subseteq \mathbf{Types}$ such that
- $\omega \in f$;
- if $\alpha \leqslant \beta$ and $\alpha \in f$, then $\beta \in f$;
- if $\alpha, \beta \in f$, then $\alpha \wedge \beta \in f$.

(2) Let $\mathscr{F} \subseteq \wp(\mathbf{Types})$ be the set of filters over **Types**.
(3) If $h \subseteq \mathbf{Types}$, $\uparrow h$ denotes the filter generated by h.

By Lemma 7.6(2) for each term M and basis Γ the set $\{\alpha \mid \Gamma \vdash M{:}\alpha\}$ is a filter. If s is a mapping from term variables to $\mathscr{F}$, and $s \models \Gamma$ is short for $x : \alpha \in \Gamma$ implies $\alpha \in s(x)$, we can define a term interpretation by

$$[\![M]\!]_s^{\mathscr{F}} = \{\alpha \mid \exists \Gamma \text{ such that } s \models \Gamma \text{ and } \Gamma \vdash M : \alpha\}.$$

It is easy to verify that $[\![M]\!]_s^{\mathscr{F}} \in \mathscr{F}$ for all M,s. Clearly β-convertible terms have the same interpretation. But we cannot say that $\mathscr{F}$ is a λ-model, since we don't have an appropriate notion of application. Consider the set of types of Δ_3, which contains for example $\pi \to \pi\pi\pi$ for all zero types π, and contains also all the types of the form $(\alpha_1 \to \alpha_1) \wedge (\alpha_2 \to \alpha_2) \wedge ((\alpha_1 \to \alpha_1) \to (\alpha_2 \to \alpha_2) \to \alpha_3) \to \alpha_3$ for suitable types $\alpha_i, \ldots$. It is not clear how starting from this set applied to itself one could obtain the type $\zeta(\zeta \to \zeta\zeta\zeta)$, which can be derived for $\Delta_3\Delta_3$, without essentially mimicking the reduction $\to_{\infty\bot}$.

6. The type assignment systems for approximate normal forms

We define a type assignment system for approximate normal forms $A \in \mathscr{A}$ which is sound with respect to the semantics given in Definition 3.9, namely $\Gamma \vdash A{:}\alpha$ implies $\Gamma \models A{:}\alpha$. This soundness follows from Lemma 5.6 and Theorem 7.5.

Definition 6.1. The type assignment system for approximate normal forms is defined by the same axioms and rules of Definition 5.4 except that rule (Eq_β) is omitted and rule (ζ) becomes

$$(\zeta) \Gamma \vdash \bot : \zeta.$$

The following lemma shows that for every basis Γ and every $A \in \mathscr{A}$, there are types which cannot be derived for A using Γ. We cannot use a semantic argument to prove this result since we do not know yet that every basis is semantically consistent (in the sense that it is satisfied by some adequate interpretation). This will be shown in Section 9.

Lemma 6.2 (Generation Lemma). (1) *$\Gamma \vdash \Omega : \alpha$ implies $\alpha = \omega$;*

(2) *If $\Gamma \vdash A : \alpha$, $\alpha \neq \omega$, and*
  (a) *$A \equiv x$, then $\Gamma = \Gamma', x : \beta$ for some $\beta \leqslant \alpha$;*
  (b) *$A \equiv \bot$, then $\zeta \leqslant \alpha$;*
  (c) *$A \equiv \lambda x.A'$, then $\alpha = \wedge_{i \in I}(\alpha_i \to \beta_i)$ and $\forall i \in I\ \Gamma, x : \alpha_i \vdash A' : \beta_i$;*
  (d) *$A \equiv x\vec{A}A'$, then there is β such that $\Gamma \vdash A' : \beta$, and either $\Gamma \vdash x\vec{A} : \beta \to \alpha$, or $\alpha \geqslant \pi\beta$ and $\Gamma \vdash x\vec{A} : \pi$ for some π;*
  (e) *$A \equiv \bot\vec{A}A'$, then there is β such that $\Gamma \vdash A' : \beta$, $\alpha \geqslant \pi\beta$ and $\Gamma \vdash \bot\vec{A} : \pi$ for some π.*

(3) *If $\Gamma \vdash A : \alpha$ and $\Gamma \vdash A : \beta$, then $\alpha \wedge \beta \in$* **Types**.

Proof. (1) is immediate. We prove (2) and (3) by a simultaneous induction on A, showing each time first (2) by a secondary induction on deductions.

- $A \equiv x$. Point (2)(a) follows easily by induction on deductions, using the transitivity of $\leqslant$.

  Part (3) follows from (2)(a) and Lemma 5.3(7).
- $A \equiv \bot$. Point (2)(b) follows easily by induction on deductions, using the fact that if $\zeta \leqslant \alpha$ and $\zeta \leqslant \beta$, then $\zeta \leqslant \alpha \wedge \beta$.

  Part (3) follows from (2)(b) and Lemma 5.3(7).
- $A \equiv \lambda x.A'$. Point (2)(c) is proved by induction on deductions. If the last applied rule is $(\leqslant)$, $\alpha = \bigwedge_{i \in I}(\alpha_i \to \beta_i)$ follows from Lemma 5.3(1) and $\Gamma, x : \alpha_i \vdash A' : \beta_i$ follows from Lemma 5.3(2).

  For (3) let by (2)(c) $\alpha = \bigwedge_{i \in I}(\alpha_i \to \beta_i)$, $\beta = \bigwedge_{j \in J}(\alpha_j \to \beta_j)$. Consider $K \subseteq I \cup J$: if $\bigwedge_{k \in K} \alpha_k \notin$ **Types** there is no problem. If $\bigwedge_{k \in K} \alpha_k \in$ **Types** we have by (2)(c) that $\Gamma, x : \alpha_k \vdash A' : \beta_k$ for all $k \in K$, therefore using rule $(\leqslant)$ we have $\Gamma, x : \bigwedge_{k \in K} \alpha_k \vdash A' : \beta_k$ for all $k \in K$. This implies by induction $\bigwedge_{k \in K} \beta_k \in$ **Types**, so we conclude $\alpha \wedge \beta \in$ **Types**.
- $A \equiv x\vec{A}A'$. The proof of (2)(d) is by induction on deductions. The only interesting case is when the last applied rule is $(\wedge \mathrm{I})$

$$(\wedge \mathrm{I}) \frac{\Gamma \vdash A : \alpha_1 \qquad \Gamma \vdash A : \alpha_2}{\Gamma \vdash A : \alpha_1 \wedge \alpha_2}$$

By the secondary induction there are, for $i = 1,2$, β_i such that $\Gamma \vdash A' : \beta_i$, and either $\Gamma \vdash x\vec{A} : \beta_i \to \alpha_i$, or $\alpha_i \geqslant \pi_i \beta_i$ and $\Gamma \vdash x\vec{A} : \pi_i$ for some π_i. By the induction with respect to (3) we cannot have $\Gamma \vdash x\vec{A} : \beta_1 \to \alpha_1$ and $\Gamma \vdash x\vec{A} : \pi_2$, or $\Gamma \vdash x\vec{A} : \beta_2 \to \alpha_2$ and $\Gamma \vdash x\vec{A} : \pi_1$. So we get $\beta_1 \wedge \beta_2 \in$ **Types** and either $(\beta_1 \to \alpha_1) \wedge (\beta_2 \to \alpha_2) \in$ **Types** or $\pi_1 \wedge \pi_2 \in$ **Types**. Therefore using rules $(\leqslant)$ and $(\wedge \mathrm{I})$ we can deduce $\Gamma \vdash A' : \beta_1 \wedge \beta_2$ and either $\Gamma \vdash x\vec{A} : \beta_1 \wedge \beta_2 \to \alpha_1 \wedge \alpha_2$ or $\Gamma \vdash x\vec{A} : \pi_1 \wedge \pi_2$ and $\alpha_1 \wedge \alpha_2 \geqslant (\pi_1 \wedge \pi_2)(\beta_1 \wedge \beta_2)$.

For (3) let $\Gamma \vdash A : \alpha_1$ and $\Gamma \vdash A : \alpha_2$. By (2)(d) there are for $i = 1,2$ β_i such that $\Gamma \vdash A' : \beta_i$, and either $\Gamma \vdash x\vec{A} : \beta_i \to \alpha_i$, or $\alpha_i \geqslant \pi_i \beta_i$ and $\Gamma \vdash x\vec{A} : \pi_i$ for some π_i. So we can conclude as above that $\beta_1 \wedge \beta_2 \in$ **Types** and either $(\beta_1 \to \alpha_1) \wedge (\beta_2 \to \alpha_2) \in$ **Types** or $\pi_1 \wedge \pi_2 \in$ **Types**. This implies either $\alpha_1 \wedge \alpha_2 \in$ **Types** or $(\pi_1 \wedge \pi_2)(\beta_1 \wedge \beta_2) \in$ **Types**. In the second case we get $(\pi_1 \wedge \pi_2)(\beta_1 \wedge \beta_2) \leqslant \pi_1 \beta_1 \wedge \pi_2 \beta_2$ by Lemma 5.3(5). This implies $(\pi_1 \wedge \pi_2)(\beta_1 \wedge \beta_2) \leqslant \alpha_1 \wedge \alpha_2$, so we can conclude $\alpha_1 \wedge \alpha_2 \in$ **Types** by Lemma 5.3(7).

- $A \equiv \bot\vec{A}A'$. The proof of this case is similar to and simpler than that of the previous case. □

Points (1)(d) and (1)(e) of the above lemma are the key tools to prove the approximation theorem. The existence of the type β expresses a kind of continuity which would be difficult to prove semantically.

The set of types of approximate normal forms is not decreasing wrt the order relation $\preceq$ between approximate normal forms.

Lemma 6.3. (1) *If* $\Gamma \vdash A : \alpha$ *and* $A \preceq A'$ *then* $\Gamma \vdash A' : \alpha$.

(2) *If A and A' are approximants of the same term M, then a basis Γ cannot assign to A an arrow type and to A' a zero type.*

Proof. (1). By induction on the definition of $\preceq$.

(2). Since $\mathscr{A}(M)$ is directed (Lemma 2.20) reasoning for a contradiction we would get a single approximate normal form which has both an arrow and a zero type (by (1)). This is impossible by Lemma 6.2(3) because the intersection of a zero type and an arrow type is not a type. □

We now establish a relation between the assignment system for terms and the assignment system for approximate normal form which will be used in the approximation theorem.

Definition 6.4. Let $\vdash^-$ be the assignment system of Definition 5.4 with rule (Eq_β) omitted. Let us say that a term M *has strictly all the types* of the approximant A if for each Γ, α such that $\Gamma \vdash A : \alpha$ we have $\Gamma \vdash^- M : \alpha$.

Lemma 6.5. *If $A \in \mathscr{A}(M)$ and $\Gamma \vdash A : \alpha$, then there is a β-reduct M' of M which has strictly all the types of A.*

Proof. If $\alpha = \omega$ there is nothing to prove. When $\alpha \neq \omega$ the proof is by induction on the length of $A \in \mathscr{A}(M)$. There are four cases to consider according to the possible shapes of A.

Case 1. If $A \equiv \Omega$ then $\alpha = \omega$ by the generation lemma.

Case 2. Suppose $A \equiv \lambda x.B \in \mathscr{A}(M)$. Then $M \rightarrow_\beta^* \lambda x.N$ with $B \in \mathscr{A}(N)$. By induction there is a β-reduct N' of N which has strictly all the types of B. Now assume $\Gamma \vdash \lambda x.B{:}\,\alpha$. We will show that $\Gamma \vdash^- \lambda x.N'{:}\,\alpha$. By the generation lemma $\alpha = \bigwedge_{i\in I}(\alpha_i \rightarrow \beta_i)$ with $\Gamma, x{:}\,\alpha_i \vdash B{:}\,\beta_i$ for each i. Since N' has strictly all the types of B we have $\Gamma, x{:}\,\alpha_i \vdash^- N'{:}\,\beta_i$ for each i. It follows that $\Gamma \vdash^- \lambda x.N'{:}\,\alpha_i \rightarrow \beta_i$ for each i, and therefore $\Gamma \vdash^- \lambda x.N'{:}\,\alpha$.

Case 3. Suppose $A \equiv xA_1 \ldots A_m \in \mathscr{A}(M)$. If $m = 0$, then $M \rightarrow_\beta^* x$ and there is nothing to prove. If $m > 0$, then $M \rightarrow_\beta^* xN_1 \ldots N_m$ with $A_i \in \mathscr{A}(N_i)$ for each i. By induction there is a β-reduct $xN_1' \ldots N_{m-1}'$ of $xN_1 \ldots N_{m-1}$ which has strictly all the types of $xA_1 \ldots A_{m-1}$ and a β-reduct N_m' of N_m which has strictly all the types of A_m. Assume now $\Gamma \vdash xA_1 \ldots A_m{:}\,\alpha$ and let us show that $\Gamma \vdash^- xN_1' \ldots N_m'{:}\,\alpha$. By the generation lemma there is β such that $\Gamma \vdash A_m{:}\,\beta$ and either $\Gamma \vdash xA_1 \ldots A_{m-1}{:}\,\beta \rightarrow \alpha$ or $\alpha \geqslant \pi\beta$ and $\Gamma \vdash xA_1 \ldots A_{m-1}{:}\,\pi$ for some π. So we have $\Gamma \vdash^- N_m'{:}\,\beta$ and either $\Gamma \vdash^- xN_1' \ldots N_{m-1}'{:}\,\beta \rightarrow \alpha$ or $\alpha \geqslant \pi\beta$ and $\Gamma \vdash^- xN_1' \ldots N_{m-1}'{:}\,\pi$ for some π. In both cases $\Gamma \vdash^- xN_1' \ldots N_m'{:}\,\alpha$.

Case 4. Suppose $A \equiv \perp A_1 \ldots A_m \in \mathscr{A}(M)$. If $m = 0$, then by the generation lemma $\zeta \leqslant \alpha$ and $\Gamma \vdash^- M{:}\,\zeta$ because M is a strong zero term. It then follows $\Gamma \vdash^- M{:}\,\alpha$ and we are done. Finally assume $m > 0$. Then $M \rightarrow_\beta^* NP$ with N a strong zero term, $\perp A_1 \ldots A_{m-1} \in \mathscr{A}(N)$ and $A_m \in \mathscr{A}(P)$. By induction there is a β-reduct N' of N which has strictly all the types of $\perp A_1 \ldots A_{m-1}$ and a β-reduct P' of P which has strictly all the types of A_m. We show that $N'P'$ has strictly all the types of $\perp A_1 \ldots A_m$. So assume $\Gamma \vdash \perp A_1 \ldots A_m{:}\,\alpha$ and let us show $\Gamma \vdash^- N'P'{:}\,\alpha$. Then by the generation lemma there is β such that $\Gamma \vdash A_m{:}\,\beta$ and $\alpha \geqslant \pi\beta$ for some π such that $\Gamma \vdash \perp A_1 \ldots A_{m-1}{:}\,\pi$. We then have $\Gamma \vdash^- P'{:}\,\beta$ and $\Gamma \vdash^- N'{:}\,\pi$. Hence $\Gamma \vdash^- N'P'{:}\,\pi\beta$ and since $\alpha \geqslant \pi\beta$ we are done. □

7. Approximation theorem

We prove the approximation theorem by means of a variant of Tait's "computability" technique [48]. We define sets of "approximable" and "computable" terms. The computable terms are defined by induction on types, and every computable term is shown to be approximable (Lemma 7.3(2)). Using induction on type derivations, we show that every term is computable for the appropriate type (Lemma 7.4).

Definition 7.1. We define two predicates $\mathsf{App}(\Gamma,\alpha,M)$ and $\mathsf{Comp}(\Gamma,\alpha,M)$ as follows:

(1) $\mathsf{App}(\Gamma,\alpha,M) \Leftrightarrow \exists A \in \mathscr{A}(M).\ \Gamma \vdash A{:}\,\alpha$;

(2) (a) $\mathsf{Comp}(\Gamma,\omega,M)$ is always true;

(b) $\mathsf{Comp}(\Gamma,\pi,M) \Leftrightarrow \mathsf{App}(\Gamma,\pi,M)$ for every type π of zero kind;

(c) $\mathsf{Comp}(\Gamma,\alpha\to\beta,M)$ holds iff $\mathsf{App}(\Gamma,\omega\to\omega,M)$, and $\mathsf{Comp}(\Gamma',\alpha,N)$ implies $\mathsf{Comp}(\Gamma\uplus\Gamma',\beta,MN)$, for each Γ' and N such that Γ and Γ' are compatible bases;

(d) $\mathsf{Comp}(\Gamma,\alpha\wedge\beta,M)\Leftrightarrow\mathsf{Comp}(\Gamma,\alpha,M)$ and $\mathsf{Comp}(\Gamma,\beta,M)$.

These notions agree with the typing rule ($\leqslant$) and depend only on the equivalence classes of terms modulo β-conversion.

Lemma 7.2. (1) *If* $\alpha\leqslant\beta$ *and* $\mathsf{Comp}(\Gamma,\alpha,M)$ *then* $\mathsf{Comp}(\Gamma,\beta,M)$.[5]

(2) *If* $M =_\beta M'$, *then* $\mathsf{App}(\Gamma,\alpha,M)$ *iff* $\mathsf{App}(\Gamma,\alpha,M')$ *and* $\mathsf{Comp}(\Gamma,\alpha,M)$ *iff* $\mathsf{Comp}(\Gamma,\alpha,M')$.

(3) *Let* $z\notin FV(M)$ *and* $\Gamma'=\Gamma,z{:}\alpha$. *If* $\mathsf{App}(\Gamma',\beta,Mz)$ *and* $\mathsf{App}(\Gamma,\omega\to\omega,M)$, *then* $\mathsf{App}(\Gamma,\alpha\to\beta,M)$.

Proof. (1) Easy induction with respect to the definition of $\leqslant$.

(2) For App it suffices to observe that two β-convertible terms have the same approximants. For Comp we reason by induction on types.

(3) Assume that $A\in\mathscr{A}(Mz)$ and $\Gamma'\vdash A{:}\beta$. Moreover, assume that $\Gamma\vdash A'{:}\,\omega\to\omega$ for some $A'\in\mathscr{A}(M)$. We must prove that there exists $\widehat{A}\in\mathscr{A}(M)$ such that $\Gamma\vdash\widehat{A}{:}\,\alpha\to\beta$. If $\beta=\omega$, one has $\Gamma\vdash A'{:}\alpha\to\omega$, since $\alpha\leqslant\omega$; hence we can take $\widehat{A}\equiv A'$. If $A\equiv\Omega$ we get $\beta=\omega$ by Lemma 6.2(1).
If M is beta equal to an abstraction, then $\lambda z.Mz =_\beta M$ and we can choose $\widehat{A}\equiv\lambda z.A$. If $A\equiv xA_1\ldots A_nZ$ we can choose $\widehat{A}\equiv xA_1\ldots A_n$. Indeed, since $\Gamma'\vdash A{:}\,\beta$, we have by Lemma 6.2(2)(d) that either $\beta\geqslant\pi\gamma$ and $\Gamma'\vdash\widehat{A}{:}\,\pi$, or $\Gamma'\vdash\widehat{A}{:}\,\gamma\to\beta$, for some γ with $\Gamma'\vdash Z{:}\,\gamma$. Notice that $z\notin FV(M)$ implies $z\notin FV(\widehat{A})$, so we get either $\Gamma\vdash\widehat{A}{:}\,\pi$, or $\Gamma\vdash\widehat{A}{:}\,\gamma\to\beta$. The condition $\Gamma\vdash A'{:}\,\omega\to\omega$ forbids $\Gamma\vdash\widehat{A}{:}\,\pi$ by Lemma 6.3(2). As an approximant of z, the term Z is either z or Ω, and in both cases we must have $\alpha\leqslant\gamma$. Thus $\Gamma\vdash\widehat{A}{:}\,\alpha\to\beta$, as desired.
The case $A\equiv\bot A_1\ldots A_nZ$ is excluded by Lemma 6.3(2). □

We can now show that computability implies approximability.

Lemma 7.3. *For all* Γ, α, $\vec{L}$ *and* M:

(1) $\mathsf{App}(\Gamma,\alpha,x\vec{L})\Rightarrow\mathsf{Comp}(\Gamma,\alpha,x\vec{L})$;

(2) $\mathsf{Comp}(\Gamma,\alpha,M)\Rightarrow\mathsf{App}(\Gamma,\alpha,M)$.

Proof. We prove (1) and (2) by a simultaneous induction on α.

- α is a zero type. Both parts are clear by definition.
- $\alpha\equiv\alpha_1\to\alpha_2$. For part (1) assume $\mathsf{App}(\Gamma,\alpha,x\vec{L})$. This means that there is an approximant $A\in\mathscr{A}(x\vec{L})$ with $\Gamma\vdash A:\alpha_1\to\alpha_2$. By Lemma 6.3(1) A can be taken of the form $x\vec{A}$, where $\vec{A}$ is a vector of approximants of $\vec{L}$.

---

[5] The same property trivially holds for $\mathsf{App}(\ ,\ ,\ )$.

We show that $\mathsf{Comp}(\Gamma,\alpha_1 \to \alpha_2,x\vec{L})$. Assume $\mathsf{Comp}(\Gamma',\alpha_1,N)$, where Γ,Γ' are compatible. By the induction hypothesis, part (2), we have $\mathsf{App}(\Gamma',\alpha_1,N)$, that is, there is an approximant $B \in \mathscr{A}(N)$ of type α_1 in the context Γ'. The term $AB \equiv x\vec{A}B$ is an approximant of $x\vec{L}N$, and we have $\Gamma \uplus \Gamma' \vdash AB : \alpha_2$. Thus $\mathsf{Comp}(\Gamma\uplus\Gamma',\alpha_2,x\vec{L}N)$ follows from part (1) of the induction hypothesis.

For part (2) suppose $\mathsf{Comp}(\Gamma,\alpha_1 \to \alpha_2,M)$. We get by definition $\mathsf{App}(\Gamma,\omega \to \omega,M)$. Let $\Gamma' = \Gamma,z:\alpha_1$, where z is fresh. Then we have

$$\begin{aligned} &\mathsf{Comp}(\{z:\alpha_1\},\alpha_1,z) && \text{by part (1)}\\ &\Rightarrow \mathsf{Comp}(\Gamma',\alpha_2,Mz) && \text{by definition}\\ &\Rightarrow \mathsf{App}(\Gamma',\alpha_2,Mz) && \text{by induction}\\ &\Rightarrow \mathsf{App}(\Gamma,\alpha_1 \to \alpha_2,M) && \text{by Lemma 7.2(3).}\end{aligned}$$

- $\alpha \equiv \alpha_1 \wedge \alpha_2$. For part (1) we need that if $\Gamma \vdash A:\alpha_1 \wedge \alpha_2$, then $\Gamma \vdash A:\alpha_1$ and $\Gamma \vdash A:\alpha_2$ (rule $(\leqslant)$). For part (2) we need that any two approximations have a common refinement (Lemma 2.20), and that if A' refines A, A' has all the types of A in any basis (Lemma 6.3(1)). □

Lemma 7.4. *Let* $\Gamma = \{x_1:\beta_1,\ldots,x_n:\beta_n\}$ *and let* $\Gamma \vdash M:\alpha$. *Assume that* $\mathsf{Comp}(\Gamma_i,\beta_i,N_i)$ *holds for each* $i \leqslant n$ *and take* $\Gamma' = \biguplus_{i=1}^{n} \Gamma_i$. *Then*

$$\mathsf{Comp}(\Gamma',\alpha,M[N_1/x_1,\ldots,N_n/x_n]).$$

Proof. By induction on the derivation of $\Gamma \vdash M:\alpha$. Cases (Ax) and (ω) are immediate. Cases $(\to \mathrm{E})$ and $(\wedge \mathrm{I})$ follow from the induction hypothesis. Case $(\leqslant)$ follows from the induction hypothesis and Lemma 7.2(1). Case (Eq_β) follows from the induction hypothesis and Lemma 7.2(2).

For the proof in case $(\to \mathrm{I})$, let $M \equiv \lambda y.P$ and $\alpha \equiv \alpha_1 \to \alpha_2$. Suppose that $\Gamma,y:\alpha_1 \vdash P:\alpha_2$ has been derived. Since $\lambda y.\Omega$ is an approximant of $(\lambda y.P)[\vec{N}/\vec{x}]$ of type $\omega \to \omega$, we have that $\mathsf{App}(\Gamma,\omega \to \omega,M[\vec{N}/\vec{x}])$ holds.

If $\mathsf{Comp}(\Gamma'',\alpha_1,Q)$ then from the induction hypothesis

$$\mathsf{Comp}(\Gamma' \uplus \Gamma'',\alpha_2,P[Q/y,\vec{N}/\vec{x}]).$$

There is no loss of generality in assuming that $y \notin FV(\vec{N})$, so that

$$P[Q/y,\vec{N}/\vec{x}] \equiv P[\vec{N}/\vec{x}][Q/y] \text{ and } (\lambda y.P[\vec{N}/\vec{x}])Q \equiv ((\lambda y.P)[\vec{N}/\vec{x}])Q.$$

By the invariance of computability under β-conversion (Lemma 7.2(2)), it follows that $\mathsf{Comp}(\Gamma' \uplus \Gamma'',\alpha_2,((\lambda y.P)[\vec{N}/\vec{x}])Q)$, and hence

$$\mathsf{Comp}(\Gamma',\alpha_1 \to \alpha_2,(\lambda y.P)[\vec{N}/\vec{x}]),$$

since the computable term Q was arbitrary.

For rule (app) assume $M \equiv PQ$ and by induction $\mathsf{Comp}(\Gamma',\pi,P[\vec{N}/\vec{x}])$ and $\mathsf{Comp}(\Gamma',\alpha,Q[\vec{N}/\vec{x}])$. Then $\mathsf{App}(\Gamma',\pi,P[\vec{N}/\vec{x}])$ and $\mathsf{App}(\Gamma',\alpha,Q[\vec{N}/\vec{x}])$ by Lemma 7.3(2).

This means $\Gamma' \vdash A : \pi$ for some $A \in \mathscr{A}(P[\vec{N}/\vec{x}])$ and $\Gamma' \vdash A' : \alpha$ for some $A' \in \mathscr{A}(Q[\vec{N}/\vec{x}])$. Notice that by Lemma 6.2(2)(c) A cannot be an abstraction, therefore $AA' \in \mathscr{A}(M[\vec{N}/\vec{x}])$. Moreover we derive $\Gamma' \vdash AA' : \pi\alpha$, so we conclude $\mathsf{Comp}(\Gamma', \pi\alpha, M[\vec{N}/\vec{x}])$. □

We can now prove the approximation theorem.

Theorem 7.5 (Approximation Theorem). *$\Gamma \vdash M : \alpha$ iff there is $A \in \mathscr{A}(M)$ such that $\Gamma \vdash A : \alpha$.*

Proof. ($\Rightarrow$) Since $\mathsf{App}(\{x{:}\beta\}, \beta, x)$ holds for any variable x and type β, then by Lemma 7.3(1), we have $\mathsf{Comp}(\{x{:}\beta\}, \beta, x)$. Thus we can apply Lemma 7.4 for the identity substitution, to obtain $\mathsf{Comp}(\Gamma, \alpha, M)$. We conclude using Lemma 7.3(2).

($\Leftarrow$) By Lemma 6.5 $\Gamma \vdash A : \alpha$ implies $\Gamma \vdash^{-} M' {:} \alpha$ for some β-reduct M' of M. By an application of the rule (Eq_β) we conclude $\Gamma \vdash M {:} \alpha$. □

As an immediate consequence we have that only consistent types can be derived for the same term. So the set of types of any term in a given basis is a filter.

Lemma 7.6. (1) *If $\Gamma \vdash M{:}\alpha$ and $\Gamma \vdash M{:}\beta$, then $\alpha \wedge \beta \in$* **Types**.

(2) *For each term M and basis Γ the set $\{\alpha \mid \Gamma \vdash M{:}\alpha\}$ is a filter.*

Proof. (1) By Theorem 7.5 there are $A, A' \in \mathscr{A}(M)$ such that $\Gamma \vdash A{:}\alpha$ and $\Gamma \vdash A'{:}\beta$. Then by Lemma 6.3(1) we have both $\Gamma \vdash A'' : \alpha$ and $\Gamma \vdash A'' : \beta$, where A'' is the sup of A and A' (A'' exists by Lemma 2.20). So by Lemma 6.2(3) we conclude $\alpha \wedge \beta \in$ **Types**.
(2) By (1) and rules (ω), $(\wedge \mathrm{I})$, $(\leqslant)$ of Definition 5.4. □

Remark 7.7. Let $\vdash^{-}$ the type assignment system of Definition 6.4, i.e. the system without rule rule (Eq_β). Lemma 7.6(1) holds a fortiori for $\vdash^{-}$, and it implies that we cannot derive a zero type for a term which is an abstraction. In particular we cannot type a β-redex with rule (app). So we can show that subject reduction holds for $\vdash^{-}$ in the standard way.

From Lemma 2.21, and Theorem 7.5, we get that types characterize mute terms, strong zero terms and zero terms.

Lemma 7.8. (1) *A term M is mute iff $\vdash M : \alpha$ implies $\zeta \leqslant \alpha$.*

(2) *A term M is strong zero iff $\vdash M : \zeta$.*

(3) *A term M is zero iff $\nvdash M : \omega \to \omega$.*

8. Principal types

In this section we show that two terms have the same types in every basis iff they are equal in the infinite λ-calculus (Theorem 8.12). We will prove also that inclusion of approximants coincide with inclusion of deducible types (Theorem 8.22).

Recall that in our system we do not have type variables. The next definition gives us a sequence of zero types which in some respect behave as variables.

Definition 8.1. Let $\theta = (\zeta \to \omega \to \omega) \wedge ((\omega \to \omega) \to \zeta)$. We define ϕ_0 as the type $\zeta(\zeta \to \theta)$ and inductively $\phi_{i+1} = \phi_i\theta$.

Lemma 8.2. *For each finite set of integers I the expression $\bigwedge_{i\in I} \phi_i$ is a zero type and if $\bigwedge_{i\in I} \phi_i \leqslant \phi_j$, then $j \in I$.*

Proof. Given I, the expression $\bigwedge_{i\in I} \phi_i$ has the form $\bigwedge_{i\in I} \pi_i\alpha_i$ where each π_i is either ζ (if $i = 0$) or ϕ_{i-1} and each α_i is either $\zeta \to \theta$ (if $i = 0$) or θ. We must prove that this expression is a type. By induction on the length of the expressions (and observing that the intersection of ζ with any zero type is a zero type) $\bigwedge_{i\in I} \pi_i$ is a zero type. We would be done if we show that $\bigwedge_{i\in I} \alpha_i$ is a type. For this it suffices to prove that $\theta \wedge (\zeta \to \theta)$ is a type, namely the intersection of $(\zeta \to \omega \to \omega) \wedge ((\omega \to \omega) \to \zeta)$ with $\zeta \to \theta$ is a type. This is clear by definition of the set of types since $(\omega \to \omega) \wedge \theta = \theta$ is a type.

To prove the second part we need the following:

Claim. *$\theta \not\leqslant \zeta \to \theta$ and $\zeta \to \theta \not\leqslant \theta$.*

Suppose, for the sake of a contradiction, that $\theta \leqslant \zeta \to \theta$. Then by Lemma 5.3(2) either $\omega \to \omega \leqslant \theta$ or $\zeta \leqslant \theta$. Both are impossible since a zero type is not comparable with an arrow type. Similarly, if $\zeta \to \theta \leqslant \theta$, then $\theta \leqslant \zeta$, which is again impossible.

Now assume $\bigwedge_{i\in I} \phi_i \leqslant \phi_j$ and we prove $j \in I$ by induction on the length of the types involved.

Case 1. If 0 is not in I and $0 \neq j$, then we can write the given inequality as $\bigwedge_{i\in I} \phi_{i-1}\theta \leqslant \phi_{j-1}\theta$. By Lemma 5.3(4) it follows $\bigwedge_{i\in I} \phi_{i-1} \leqslant \phi_{j-1}$, hence by induction $j - 1 = i - 1$ for some $i \in I$, and therefore $j \in I$.

Case 2. If $0 \in I$ and $0 \neq j$, we consider $H = I\backslash\{0\}$ and we write the given inequality as $\zeta(\zeta \to \theta) \wedge \bigwedge_{h\in H} \phi_{h-1}\theta \leqslant \phi_{j-1}\theta$. This entails $\zeta \wedge \bigwedge_{h\in H} \phi_{h-1} \leqslant \phi_{j-1}$ by Lemma 5.3(4). If H is non-empty ζ can be dropped from the left-hand-side and induction applies. If H is empty we have $\zeta \leqslant \phi_{j-1}$, which is impossible.

Case 3. If $0 \in I$ and $0 = j$ there is nothing to prove. □

Definition 8.3. (1) We define Γ_∞ as the basis $\{x_n{:}\, \phi_n \mid n \in \mathbb{N}\}$.

(2) Let $A \in \mathscr{A}$. The *zero characteristic type* $ct(A)$ of A is defined as follows.

- (a) $ct(\Omega) = \omega$,
- (b) $ct(\lambda x_i.A) = \phi_i \to ct(A)$,
- (c) $ct(x_iA_1 \ldots A_n) = \phi_i ct(A_1) \ldots ct(A_n)$,
- (d) $ct(\bot A_1 \ldots A_n) = \zeta ct(A_1) \ldots ct(A_n)$.

Zero characteristic types of α-convertible approximate normal forms coincide up to a permutation of the ϕ_i.

By induction on the length of A we have:

Lemma 8.4. $\Gamma_\infty \vdash A: ct(A)$.

Each zero characteristic type can be deduced only for approximate normal forms of a particular shape, as proved in the following Lemma 8.6. First we need one technical lemma.

Lemma 8.5. (1) *If* $\Gamma_\infty \vdash x_i : \alpha$ *then* α *is a zero type.*

(2) *For no approximate normal form* A *we have* $\Gamma_\infty \vdash A : \theta$ *or* $\Gamma_\infty \vdash A : \zeta \to \theta$.

Proof. (1) By Lemmas 6.2(2)(a) and 5.3(3) since all types in Γ_∞ are zero types.

(2) Suppose $\Gamma_\infty \vdash A : \theta$. Lemma 6.2(2) implies either $A \equiv \lambda x.A'$ or $A \equiv x\vec{A}$, for some x, A', $\vec{A}$.

Recalling the definition of θ, in the first case we get by Lemma 6.2(2)(c) $\Gamma_\infty, x:\zeta \vdash A' : \omega \to \omega$ and $\Gamma_\infty, x : \omega \to \omega \vdash A' : \zeta$. Then Lemma 6.2(2) implies $A' \equiv y\vec{B}$, for some y, $\vec{B}$. From $\Gamma_\infty, x:\zeta \vdash A' : \omega \to \omega$ we have $\Gamma_\infty, x:\zeta \vdash y : \vec{\alpha} \to \omega \to \omega$ for some $\vec{\alpha}$. This is impossible by (1) being also ζ a zero type.

In the second case we get by Lemma 6.2(2)(d) $\Gamma_\infty \vdash x : \vec{\alpha} \to \theta$ for some $\vec{\alpha}$, which is again impossible by (1).

The proof that for no approximate normal form A we have $\Gamma_\infty \vdash A : \zeta \to \theta$ is similar. □

Lemma 8.6. (1) *If* $\Gamma_\infty \vdash A : \phi_i$, *then* $A \equiv x_i$.

(2) *If* $\Gamma_\infty \vdash A : \pi\alpha$ *and* $\pi\alpha \neq \phi_i$ *for all* i, *then* $A \equiv A_1 A_2$ *with* $\Gamma_\infty \vdash A_1 : \pi$ *and* $\Gamma_\infty \vdash A_2 : \alpha$.

(3) *If* $\Gamma_\infty \vdash A : \phi_i \to \alpha$, *then* $A \equiv \lambda x_i.A'$ *and* $\Gamma_\infty \vdash A' : \alpha$.

Proof. We prove point (1), the others follow easily from Lemmas 6.2(2) and 8.5(1). If $\Gamma_\infty \vdash A : \phi_i$ Lemma 6.2(2) implies either $A \equiv \bot\vec{A}A'$ or $A \equiv x\vec{A}$, for some $\vec{A}$, A', x.

In the first case we get by Lemma 6.2(2)(e) $\Gamma_\infty \vdash A' : \zeta \to \theta$, if $i = 0$, and $\Gamma_\infty \vdash A' : \theta$, if $i > 0$. Both cases are impossible by Lemma 8.5.

In the second case, if $\vec{A}$ is not empty, then by Lemma 6.2(2)(d) either we must deduce an arrow type for x from Γ_∞ – which is impossible by Lemma 8.5(1) – or some of the component of $\vec{A}$ must have type $\zeta \to \theta$ or θ. We conclude that $\vec{A}$ is empty and $A \equiv x_i$, because if $A \equiv x_j$ we have $\phi_i \leqslant \phi_j$ by Lemma 6.2(2)(a), which implies $i = j$ by Lemma 8.2. □

Notice that the condition $\pi\alpha \neq \phi_i$ in Lemma 8.6(2) is necessary, as Lemma 8.6(1) shows.

Intuitively $ct(A)$ is the "best" type deducible for A from the basis Γ_∞. However this does not mean that all other deducible types are $\geqslant ct(A)$, for example $ct(\mathbf{I}) = \phi_1 \to \phi_1$, but $\Gamma_\infty \vdash \mathbf{I} : (\omega \to \omega) \to \omega \to \omega$. We will prove that if $\Gamma_\infty \vdash B : ct(A)$, then B will have all the types of A (in Γ_∞).

Definition 8.7. We define $A \preceq_\infty B$ iff $\Gamma_\infty \vdash B{:}\, ct(A)$.

By the approximation theorem we have easily the following lemma.

Lemma 8.8. *Given two terms M and N with $\{\alpha \mid \Gamma_\infty \vdash M{:}\,\alpha\} \subseteq \{\alpha \mid \Gamma_\infty \vdash N{:}\,\alpha\}$ and given $A \in \mathscr{A}(M)$, there is some $B \in \mathscr{A}(N)$ with $A \preceq_\infty B$.*

Proof. We have $\Gamma_\infty \vdash A{:}\, ct(A)$, hence $\Gamma_\infty \vdash M{:}\, ct(A)$ by the easy part of the approximation theorem. By our hypothesis $\Gamma_\infty \vdash N{:}\, ct(A)$. By Theorem 7.5 there is some $B \in \mathscr{A}(N)$ with $\Gamma_\infty \vdash B{:}\, ct(A)$. □

Our next goal is to give a geometric meaning to $\preceq_\infty$.

Definition 8.9. We say that B is a *zero-expansion* of A iff B is obtained from A by replacing some occurrences of Ω by arbitrary approximate normal forms, and some occurrences of $\bot$ by approximate normal forms of the shape $\bot\vec{C}$ or $x\vec{C}$.

It is important to remark that free variables can be captured and become bound in this replacement.

Example 8.10. $x\bot x(\lambda y.y)$ is a zero-expansion of $\bot x(\lambda y.\bot)$.

Note that if $A \preceq B$, then B is a zero-expansion of A. More generally it is easy to see that B is a zero-expansion of A if B is obtained from some $B' \succeq A$ by replacing some occurrences of $\bot$ with some variables. Clearly if A is a zero-expansion of B and B is a zero expansion of A, then $A \equiv B$.

The following lemma explains how the relation $\preceq_\infty$ binds the structure of approximate normal forms.

Lemma 8.11. *$A \preceq_\infty B$ iff B is a zero-expansion of A.*

Proof. (*Only if*): Suppose $\Gamma_\infty \vdash B{:}\, ct(A)$. We prove that B is a zero-expansion of A by induction on the length of A.

The crucial case is when $A \equiv \bot A_1 \ldots A_n$ $(n \geqslant 0)$. Then $ct(A)$ has the form $\zeta\alpha_1 \ldots \alpha_n$. If B has type $ct(A)$ in the basis Γ_∞, then by Lemma 8.6(2) and Lemma 6.2(2) B must be of the form $x\vec{B}B_1 \ldots B_n$ or $\bot\vec{B}B_1 \ldots B_n$ where $\vec{B}$ may be empty and for $i = 1, \ldots, n$, B_i has type α_i in Γ_∞. By induction for $i = 1, \ldots, n$, B_i is a zero-expansion of A_i. Since $x\vec{B}$ and $\bot\vec{B}$ are zero-expansion of $\bot$, we conclude that B is a zero-expansion of A.

If $A \equiv x_i A_1 \ldots A_n$ $(n \geqslant 0)$, then $ct(A)$ has the form $\phi_i \alpha_1 \ldots \alpha_n$. If B has type $ct(A)$ in the basis Γ_∞, then by Lemma 8.6, B must be of the form $x_i B_1 \ldots B_n$ with B_i of type α_i and induction applies.

If $A \equiv \lambda x_i.A'$, then $ct(A)$ has the form $\phi_i \to \alpha$. If B has type $ct(A)$ in Γ_∞ then B must be an abstraction $\lambda y.B'$ and by a renaming of bound variables we can assume $y \equiv x_i$. Then $\Gamma_\infty \vdash B'{:}\,\alpha$ and induction applies.

If $A \equiv \Omega$ there is nothing to prove.

(*If*) For the implication from right to left assume that B is a zero-expansion of A. Then B is obtained from some $C \succeq A$ by replacing some occurrences of $\bot$ by some variables. We have $\Gamma_\infty \vdash C{:}\,ct(A)$ by Lemma 6.3(1). Moreover in the basis Γ_∞ each type deducible for $\bot$ is also deducible for a variable. We conclude that $\Gamma_\infty \vdash B{:}\,ct(A)$. □

Now we can prove that types deducible from Γ_∞ characterize infinite trees.

Theorem 8.12. $\{\alpha \mid \Gamma_\infty \vdash M{:}\,\alpha\} = \{\alpha \mid \Gamma_\infty \vdash N{:}\,\alpha\}$ *if and only if* $M =_{\infty\bot} N$.

Proof. (*Only if*) By Theorem 2.23 it suffices to show that the infinite trees of M and N coincide for the first n levels for every n (Definition 2.17). Let $A \in \mathscr{A}(M)$ be the n-th approximant of M (Definition 2.17), i.e. $A \equiv (M)^n$. By Lemma 8.8 there is $B \in \mathscr{A}(N)$ with $A \preceq_\infty B$ and we can assume that $B \succeq (N)^n$. By symmetry there is $A' \in \mathscr{A}(M)$ with $B \preceq_\infty A'$ and we can assume $A' \succeq (M)^n$. So we get $A \equiv (M)^n \preceq A'$, which implies by Lemma 2.22 $(A)^n \equiv (A')^n$. This together with $A \preceq_\infty B \preceq_\infty A'$ gives us $(A)^n \equiv (B)^n \equiv (A')^n$. So we conclude $(M)^n \equiv (N)^n$ using Lemma 2.22.

(*If*) By Theorem 2.23 and Theorem 7.5. □

Our next goal is to show that if $\{\alpha \mid \Gamma \vdash M{:}\,\alpha\} \subseteq \{\alpha \mid \Gamma \vdash N{:}\,\alpha\}$ for every Γ, then $\mathscr{A}(M) \subseteq \mathscr{A}(N)$. It will not be enough to take the basis Γ_∞ or any other single basis Γ. In fact, if Γ assigns to x_i a zero type (like the basis Γ_∞), then all the types assigned to $\mathbf{\Omega}_2$ (namely ζ and ω) will also be assigned to x_i. On the other hand if Γ assigns to x_i an arrow type then all the types assigned to x_i can also be assigned to its η-expansion $\lambda y.x_i y$. In both cases we do not have the inclusion of the set of approximants. Our strategy will be to consider both the basis Γ_∞ and a suitable set of other basis which assign to the variables arrow types.

In order to find the basis that we need let us first discuss a technical point. Let $\Gamma = \Gamma' \uplus \{x{:}\,\alpha_1 \to \cdots \to \alpha_k \to \beta\}$. In general given a valid judgment of the form $\Gamma \vdash xA_1 \ldots A_k{:}\,\beta$ it is not necessarily the case that $\Gamma \vdash A_i{:}\,\alpha_i$ for $i = 1,\ldots,k$ (for instance β might be ω, or Γ' may contain other declarations involving x). We will restrict to a special kind of judgments where in the above situation we can indeed deduce $\Gamma \vdash A_i{:}\,\alpha_i$.

Lemma 8.13. (1) *Let* $n \in \mathbb{N}$. *Given a finite set of integers* I, *if for each* $i \in I$ *the expression* $\alpha_1^{(i)} \to \cdots \to \alpha_n^{(i)} \to \phi_i$ *is a type, then the intersection* $\bigwedge_{i\in I}(\alpha_1^{(i)} \to \cdots \to \alpha_n^{(i)} \to \phi_i)$ *is a type.*

(2) *Suppose that* $\bigwedge_{i\in I}(\alpha_1^{(i)} \to \cdots \to \alpha_n^{(i)} \to \phi_i) \leqslant \beta_1 \to \cdots \to \beta_n \to \phi_j$. *Then* $j \in I$ *and* $\alpha_1^{(j)} \geqslant \beta_1, \cdots, \alpha_n^{(j)} \geqslant \beta_n$.

Proof. (1) By induction on n. The case $n = 0$ holds since $\bigwedge_{i\in I} \phi_i$ is a type by Lemma 8.2. The induction step follows immediately by induction from Definition 3.6.

(2) By induction on n. The case $n = 0$ holds by Lemma 8.2. For $n > 0$ we use Lemma 5.3(2) to reduce to the case $n - 1$. □

Definition 8.14. Γ is a *special basis of order n* if each type declaration in Γ has the form $x: \bigwedge_{i\in I}(\alpha_1^{(i)} \to \cdots \to \alpha_n^{(i)} \to \omega \to \phi_i)$.

Note that each member of the above intersection ends with a different ϕ_i.

Notation. $\omega^h \to \sigma$ is inductively defined by $\omega^0 \to \sigma = \sigma$ and $\omega^{h+1} \to \sigma = \omega \to \omega^h \to \sigma$.

Lemma 8.15. *Let* $\Gamma = \Gamma' \uplus \{x: \alpha_1 \to \cdots \to \alpha_k \to \omega^{n-k} \to \omega \to \phi_j\}$ *be a special basis of order n and let* $k \leqslant n$. *If* $\Gamma \vdash xB_1 \ldots B_k : \omega^{n-k} \to \omega \to \phi_j$, *then* $\Gamma \vdash B_i : \alpha_i$ *for* $i = 1, \ldots, k$.

Proof. By Lemma 6.2(2)(d) there are types $\beta_1, \ldots, \beta_k$ such that $\Gamma \vdash B_i : \beta_i$ and $\Gamma \vdash x: \beta_1 \to \cdots \to \beta_k \to \omega^{n-k} \to \omega \to \phi_j$. By definition of special basis of order n, Γ assigns to x a finite intersection of arrow types such that exactly one of the types of the intersection ends with ϕ_j, so this type must be $\alpha_1 \to \cdots \to \alpha_k \to \omega^{n-k} \to \omega \to \phi_j$. By Lemma 8.13(2) we have $\alpha_i \geqslant \beta_i$. Hence $\Gamma \vdash B_i : \alpha_i$. □

The special basis of order n give us useful information about approximate normal forms of order n, defined below.

Definition 8.16. $\mathscr{A}_n$ is the subset of $\mathscr{A}$ consisting of all those $A \in \mathscr{A}$ such that, if $BA_1 \ldots A_k$ is a subterm of A, then $k \leqslant n$.

Clearly, $\mathscr{A} = \bigcup_n \mathscr{A}_n$. Note that $\mathscr{A}_n$ is closed under subterms.

Remark 8.17. If the length of $\vec{B}$ is $\leqslant n$, then from a special basis of order n we cannot deduce a zero type for $x\vec{B}$.

Now we define for each n and for each $A \in \mathscr{A}_n$ a special basis and a type which intuitively give the "best" information about A. In the next definition we write $pp_n(A) = \langle \Gamma : \alpha \rangle$ although to be precise we should write $\langle \Gamma : \alpha \rangle \in pp_n(A)$ since the principal pair $pp_n(A)$ is unique only up to a permutaion of the ϕ_i and different ϕ_i are associated to different occurrences of the same variable.

Definition 8.18. Let $A \in \mathscr{A}_n$. The *n-th arrow principal pair* $pp_n(A)$ of A is inductively defined as follows.

(1) $pp_n(\Omega) = \langle \emptyset; \omega \rangle$.

(2) If $pp_n(A) = \langle \Gamma, x: \beta; \alpha \rangle$, then $pp_n(\lambda x.A) = \langle \Gamma; \beta \to \alpha \rangle$.

(3) If $pp_n(A) = \langle \Gamma; \alpha \rangle$ and x does not occur in Γ, then $pp_n(\lambda x.A) = \langle \Gamma; \omega \to \alpha \rangle$.

(4) If $pp_n(A_i) = \langle \Gamma_i; \alpha_i \rangle$ $(1 \leqslant i \leqslant k)$ where $k \leqslant n$ and $\biguplus_{i=1}^k \Gamma_i$ is a special basis of order n,[6] then $pp_n(\bot A_1 \ldots A_k) = \langle \biguplus_{i=1}^k \Gamma_i; \zeta \alpha_1 \ldots \alpha_k \rangle$.

(5) If $pp_n(A_i) = \langle \Gamma_i; \alpha_i \rangle$ $(1 \leqslant i \leqslant k)$ where $k \leqslant n$ and $\Gamma = \biguplus_{i=1}^k \Gamma_i \uplus \{x: \alpha_1 \to \ldots \to \alpha_k \to \omega^{n-k} \to \omega \to \phi_j\}$ is a special basis of order n,[7] then $pp_n(xA_1 \ldots A_k) =$

---

[6] This condition can always be satisfied by choosing Γ_i which do not contain the same ϕ_l.

[7] This condition can always be satisfied by choosing Γ_i which do not contain the same ϕ_l, and ϕ_j new.

$\langle \Gamma; \omega^{n-k} \to \omega \to \phi_j \rangle$. (In particular when $k = 0$ we obtain $pp_n(x) = \langle \{x: \omega^n \to \omega \to \phi_j\}; \omega^n \to \omega \to \phi_j \rangle$).

It is easy to verify that $\Gamma \vdash A: \alpha$ whenever $pp_n(A) = \langle \Gamma; \alpha \rangle$.

The basis of $pp_n(A)$ assigns to each free variable x in A a finite intersection of arrow types, where each member of the intersection corresponds to a different occurrence of x in A and ends with a different ϕ_i. Contrary to what happens in the basis Γ_∞, where the ϕ_i correspond to the variables, here the ϕ_i are associated to the various occurrences of the variables. If we already know that B is a zero expansion of A, $pp_n(A)$ can give us the lacking information to conclude $A \preceq B$.

Lemma 8.20. *Let $A,B \in \mathscr{A}_n$ with B a zero-expansion of A. Suppose that $pp_n(A) = \langle \Gamma; \alpha \rangle$ and $\Gamma \vdash B: \alpha$. Then $A \preceq B$.*

We actually need a stronger result in which we allow to extend Γ and B is not required to be in $\mathscr{A}_n$: we only require that a sufficiently large h-th approximant of B is in $\mathscr{A}_n$.

Lemma 8.21. *Let $A \in \mathscr{A}_n$ and let B a zero-expansion of A. Suppose that $pp_n(A) = \langle \Gamma; \alpha \rangle$ and $\Gamma' \vdash B: \alpha$ for some special basis $\Gamma' \supseteq \Gamma$ of order n. Suppose further that some h-th approximant $(B)^h$ of B is in $\mathscr{A}_n$ where h is so big that $(A)^h \equiv A$. Then $A \preceq B$.*

Proof. The proof is by induction on A.

If $A \equiv \bot A_1 \ldots A_k$ $(k \geqslant 0)$, then $pp_n(A) = \langle \biguplus_{i=1}^k \Gamma_i; \zeta\alpha_1 \ldots \alpha_k \rangle$ with $pp_n(A_i) = \langle \Gamma_i; \alpha_i \rangle$ $(i = 1,\ldots,k)$. Since B is a zero expansion of A, B must be either of the form $\bot \vec{B} B_1 \ldots B_k$ or $x \vec{B} B_1 \ldots B_k$ with B_i a zero expansion of A_i. In both the second case by our hypothesis the length of $\vec{B} B_1 \ldots B_k$ is $\leqslant n$. By assumption $\Gamma' \vdash B: \zeta\alpha_1 \ldots \alpha_k$ for some special basis $\Gamma' \supseteq \biguplus_{i=1}^k \Gamma_i$ of order n. In the first case $\Gamma' \vdash B_i: \alpha_i$ $(i = 1,\ldots,k)$ by Lemma 6.2. It is easy to see that induction applies and gives us $A_i \preceq B_i$ $(i = 1,\ldots,k)$. If $B \equiv \bot \vec{B} B_1 \ldots B_k$, then $A \preceq B$ and we are done. If $B \equiv x \vec{B} B_1 \ldots B_k$ we reach a contradiction because a special basis of order n cannot assign to such a term a zero type (Remark 8.17).

If $A \equiv x A_1 \ldots A_k$ $(k \geqslant 0)$, then $pp_n(A) = \langle \biguplus_{i=1}^k \Gamma_i \uplus \{x: \alpha_1 \to \ldots \to \alpha_k \to \omega^{n-k} \to \omega \to \phi_j\}; \omega^{n-k} \to \omega \to \phi_j \rangle$ where $pp_n(A_i) = \langle \Gamma_i; \alpha_i \rangle$ $(i = 1,\ldots,k)$. Since B is a zero-expansion of A, B must be of the form $x B_1 \ldots B_k$ with B_i a zero-expansion of A_i $(i = 1,\ldots,k)$. By assumption $\Gamma' \vdash B: \omega^{n-k} \to \omega \to \phi_j$ for some special basis $\Gamma' \supseteq \biguplus_{i=1}^k \Gamma_i$ of order n. This implies $\Gamma' \vdash B_i: \alpha_i$ $(i = 1,\ldots,k)$ by Lemma 8.15. It is easy to see that induction applies and gives us $A_i \preceq B_i$ $(i = 1,\ldots,k)$. Hence $A \preceq B$.

The case $A \equiv \lambda x.A'$ is easy. □

Theorem 8.22. *$\{\alpha | \Gamma \vdash M: \alpha\} \subseteq \{\alpha | \Gamma \vdash N: \alpha\}$ for every Γ if and only if $\mathscr{A}(M) \subseteq \mathscr{A}(N)$.*

Proof. Assume the antecedent. Then by the approximation theorem for every $A \in \mathscr{A}(M)$ and every valid judgment $\Gamma \vdash A: \alpha$, there is $B \in \mathscr{A}(N)$ with $\Gamma \vdash B: \alpha$. Take

$A \in \mathscr{A}(M)$ and let us show that $A \in \mathscr{A}(N)$. Let h be so big that $(A)^h \equiv A$. Let n be so big that A and $(N)^h$ belong to $\mathscr{A}_n$ and let $pp_n(A) = \langle \Gamma; \alpha \rangle$. Then there is $B \in \mathscr{A}(N)$ with $\Gamma \vdash B{:}\,\alpha$. Since $\Gamma_\infty \vdash A{:}\,ct(A)$, there is a $B' \in \mathscr{A}(N)$ with $\Gamma \vdash B'{:}\,ct(A)$. So B' is a zero expansion of A. Since the set of approximants of $\mathscr{A}(N)$ is directed we can assume that $B' \equiv B$. Moreover we can assume that $B \succeq (N)^h$. By Lemma 8.21 we conclude that $A \preceq B$. Since $\mathscr{A}(N)$ is downward closed, $A \in \mathscr{A}(N)$.

The converse follows immediately from the approximation theorem (Theorem 7.5). □

The proof of Theorem 8.22 gives us a discrimination algorithm, i.e. for two arbitrary terms M, N with different infinite trees, we can always find an integer n, two types α, β and a special basis Γ of order n such that $\Gamma_\infty \vdash M : \alpha$, $\Gamma \vdash M : \beta$, and either $\Gamma_\infty \not\vdash N : \alpha$, or $\Gamma \not\vdash N : \beta$. We simply take an approximate normal form A such that $A \in \mathscr{A}(M)$ and $A \notin \mathscr{A}(N)$. Now we compute:

- $\alpha = ct(A)$;
- h as the least i such that $(A)^i \equiv A$;
- n as the least j such that both A and $(N)^h$ are in $\mathscr{A}_j$;
- $\langle \Gamma; \beta \rangle = pp_n(A)$.

Example 8.23. Let $M \equiv \mathbf{\Omega}_2$, and $N \equiv x_1$: then $\bot \in \mathscr{A}(M)$ and $\bot \notin \mathscr{A}(N)$. We have $ct(\bot) = \zeta$, $h = 1$, $n = 0$, $pp_0(\bot) = \langle \emptyset; \zeta \rangle$. Now $\Gamma_\infty \vdash x_1{:}\,\zeta$, but $\not\vdash x_1{:}\,\zeta$. If instead we consider $M \equiv \lambda x_2.x_1x_2$, and $N \equiv x_1$ we get $M \in \mathscr{A}(M)$ and $M \notin \mathscr{A}(N)$. In this case $ct(M) = \phi_2 \to \phi_1\phi_2$, $h = 3$, $n = 1$, $pp_1(M) = \langle \{x_1{:}\,(\omega^2 \to \phi_2) \to \omega \to \phi_1\}; (\omega^2 \to \phi_2) \to \omega \to \phi_1 \rangle$. It is easy to verify that $\Gamma_\infty \not\vdash x_1{:}\,\phi_2 \to \phi_1\phi_2$, but $\{x_1{:}\,(\omega^2 \to \phi_2) \to \omega \to \phi_1\} \vdash x_1{:}{:}\,(\omega^2 \to \phi_2) \to \omega \to \phi_1$.

This example shows why two basis and two types are necessary to distinguish terms with different sets of approximants.

9. Completeness

Our completeness proof uses essentially a term model. Following the approach of [52] (which is inspired by [19]), we consider the λ-calculus enriched with a new constant c^α for each $\alpha \in$ **Types** (more precisely c^α is associated to the equivalence class of α modulo the equivalence of types induced by the preorder $\leqslant$). We consider a type assignment system which assigns to c^α type α. We borrow from [52] the idea of modifying the conversion between terms taking into account typing information. In the term model modulo the modified conversion c^α will have type α semantically. So in this model every type is inhabited. We write c, c' etc. to denote one of the c^α.

Definition 9.1. $\Lambda(\mathscr{C})$ is the set of terms X generated by

$$X := x \mid c \mid XX \mid \lambda x.X, \quad \text{where } c \in \mathscr{C}.$$

Let $\Lambda(\mathscr{C})_0$ be the set of closed terms of $\Lambda(\mathscr{C})$. Notice that the subterms of terms in $\Lambda(\mathscr{C})_0$ generally belong to $\Lambda(\mathscr{C})$, not to $\Lambda(\mathscr{C})_0$. The definition of zero term and strong zero term in $\Lambda(\mathscr{C})_0$ is the same as for the ordinary λ-calculus considering the $c's$ as free variables. So a term $X \in \Lambda(\mathscr{C})_0$ is *strong zero* iff X is a zero term and X cannot be β-reduced to a λ-free head normal form, i.e. to a term of the shape $c\vec{Y}$.

Definition 9.2. The type assignment $\vdash^+$ is obtained from $\vdash$ by adding an axiom

$$(c^\alpha)\Gamma \vdash c^\alpha : \alpha$$

for each constant c^α.

Remark 9.3. An alternative definition is: given $X \in \Lambda(\mathscr{C})_0$ we define $\vdash^+ X:\alpha$ iff $\Gamma \vdash X:\alpha$ where Γ is the basis which assigns to each c^α type α. This makes sense since the constants c^α can also be considered as free variables.

Given $M \in \Lambda$ with free variables $x_1,\ldots,x_n$, we have

$$\{x_1:\beta_1,\ldots,x_n:\beta_n\} \vdash M:\alpha \text{ iff } \vdash^+ M[c^{\beta_1}/x_1,\ldots,c^{\beta_n}/x_n]:\alpha.$$

So $\vdash^+$ can be defined in terms of the old derivability $\vdash$. This implies that all the properties of the type assignment system still hold, modulo the natural mapping between free variables in Λ and constants in $\Lambda(\mathscr{C})_0$. In particular the approximation theorem is true for $\Lambda(\mathscr{C})_0$, by extending the definitions of infinite trees and n-th approximants in the obvious way, and the definition of $\mathscr{A}(\ )$ accordingly.

The model we will use to prove the completeness theorem will be an adequate λ-algebra given by a quotient of $\Lambda(\mathscr{C})_0$ with respect to a suitable equivalence relation $\sim$. We cannot take $\sim$ to be the β-conversion because otherwise all the constants c^α will be zero elements of the model according to Definition 3.2 while – in order to make all types inhabited – we would like that c^α belongs to the interpretation of the type α according to our semantics (Definition 3.5). More generally we would like that if $\vdash^+ : X:\alpha$, then (the equivalence class of) X belongs to the interpretation of the type α. In particular we need that $c\vec{X}$ belongs to the interpretation of the type $\pi\alpha$ whenever $\vdash^+ c\vec{X} : \pi\alpha$, i.e. we need to find Y, Z such that $c\vec{X} \sim YZ$ and Y belongs to the interpretation of π, Z belongs to the interpretation of α. The original idea was to define $X \sim Y$ iff X and Y have the same types in the assignment system $\vdash^+$. This might work, but unfortunately with this definition we are not able to prove that $X \sim X'$ and $Y \sim Y'$ imply $XY \sim X'Y'$.[8] So we take a weaker definition of $\sim$ which includes the β-conversion and is included in the relation "having the same types". The definition is quite technical. First we ensure that whenever we cannot deduce ζ for a term X,

---

[8] To solve this problem we may define $X \sim Y$ iff for every context $C[\ ]$ the terms $C[X]$ and $C[Y]$ have the same types. With this definition $\sim$ will respect application but we are not able to prove that if $\vdash^+ X:\zeta$, $\vdash^+ X':\zeta$ and $XY \sim X'Y'$, then $X \sim X'$ and $Y \sim Y'$. This is needed to ensure that when the equivalence classes of X and X' belong to the interpretation of ζ, then X and X' are zero terms which satisfy condition (2) of the definition of adequate λ-algebra (Definition 3.3).

then the interpretation of X (i.e. its equivalence class) will not be a zero element of the model. We have this for free when X reduces to an abstraction, so we need to consider only the case $X =_\beta c\vec{X}$. Observe that if we cannot deduce ζ for a term X, then either we can deduce only types in $\uparrow \{\omega\}$, or we can deduce $\omega \rightarrow \omega$. If we can deduce $\omega \rightarrow \omega$ we will arrange so that $X{\sim}\mathbf{1}X$. If we can deduce only types in $\uparrow \{\omega\}$ we will arrange so that X is in relation $\sim$ with c^ω. So in particular X will fail to be a zero element because we put $c^\omega Y \sim c^\omega$ for all Y. It is important to keep in mind that all the terms that we will identify will have the same types. To improve proof readability, we will define $\rightarrow^*$ as the least reduction relation containing both $\rightarrow_\beta$ and an ad hoc binary relation that we call $\rightarrow_c$. So we start with the definition of $\rightarrow_c$.

Definition 9.4. Let $\rightarrow_c$ be the binary relation on $\Lambda(\mathscr{C})_0$ defined by the following rules:
$(c - \omega)$ $c^\omega X \rightarrow_c c^\omega$;
$(c - \mathit{zero})$ $c^{\pi\alpha} \rightarrow_c c^\pi c^\alpha$ for all zero types π;
$(c - \eta)$ $\lambda x.c^\sigma x \rightarrow_c c^\sigma$, for all arrow types σ;
$(c - \mathit{arrow})$ $c^{\bigwedge_{i\in I}(\alpha_i \rightarrow \beta_i)}X \rightarrow_c c^{\bigwedge_{j\in J}\beta_j}$ where $J = \{i \in I \mid \vdash^+ X : \alpha_i\}$.
(If J is empty we define $\bigwedge_{j\in J} \beta_j = \omega$.)

Notice that $\rightarrow_c$ is not closed under contexts. Closure under contexts will be taken later in the definition of $\rightarrow^*$. The following lemma motivates our definition.

Lemma 9.5. *If X, $Y \in \Lambda(\mathscr{C})_0$, and $X \rightarrow_c Y$, then* $\{\alpha \mid \vdash^+ X : \alpha\} = \{\alpha \mid \vdash^+ Y : \alpha\}$.

Proof. We check only that $c^{\bigwedge_{i\in I}(\alpha_i \rightarrow \beta_i)}X$ has the same types of $c^{\bigwedge_{j\in J}\beta_j}$ where $J = \{i \in I \mid \vdash^+ X : \alpha_i\}$. So suppose $\vdash^+ c^{\bigwedge_{i\in I}(\alpha_i \rightarrow \beta_i)}X : \alpha$. We prove that $\vdash^+ c^{\bigwedge_{j\in J}\beta_j} : \alpha$. This amounts to showing that $\alpha \geqslant \bigwedge_{j\in J} \beta_j$. By the approximation theorem and the generation lemma there is γ such that $\vdash^+ c^{\bigwedge_{i\in I}(\alpha_i \rightarrow \beta_i)} : \gamma \rightarrow \alpha$ and $\vdash^+ X : \gamma$. By the generation lemma $\bigwedge_{i\in I}(\alpha_i \rightarrow \beta_i) \leqslant \gamma \rightarrow \alpha$. So by Lemma 5.3(2) there is $H \subseteq I$ with $\bigwedge_{h\in H} \alpha_h \geqslant \gamma$ and $\bigwedge_{h\in H} \beta_h \leqslant \alpha$. The former inequality implies $\vdash^+ X : \bigwedge_{h\in H} \alpha_h$ and therefore $H \subseteq J$. Hence $\alpha \geqslant \bigwedge_{h\in H} \beta_h \geqslant \bigwedge_{j\in J} \beta_j$.

The other direction is easy. □

We are interested in the reduction on $\Lambda(\mathscr{C})$ generated by $\rightarrow_c$ and $\rightarrow_\beta$.

Definition 9.6. $\rightarrow^*$ is the least reduction relation containing $\rightarrow_c$ and $\rightarrow_\beta$.

Also $\rightarrow^*$ preserves typing on terms of $\Lambda(\mathscr{C})_0$, i.e. we have the following subject conversion theorem.

Theorem 9.7. *If X, $Y \in \Lambda(\mathscr{C})_0$, and $X \rightarrow^* Y$, then* $\{\alpha \mid \vdash^+ X : \alpha\} = \{\alpha \mid \vdash^+ Y : \alpha\}$.

Proof. We can assume $X \equiv C[T]$ and $Y \equiv C[Z]$ for some context $C[\ ]$ and terms T, Z such that either T is a β-redex and Z is its contractum, or $T \rightarrow_c Z$. In the first case we can apply rule (Eq_β). In the second case suppose that X has type α in the system

$\vdash^+$. If we suppose instead that Y has type α in the system $\vdash^+$ the proof is similar. By the approximation theorem and Lemma 6.5 there is a β-reduct X' of X such that one can deduce $X'\colon \alpha$ in the system $\vdash^+$ without using the rule (Eq_β). Since every T begins with a constant or has the form $\lambda x.cx$, we have a good control of the residues of the T in X' (note that T is a closed term). Indeed if T does not have the form $\lambda x.cx$, then X' must be of the form $C'[T']$ with $C[y] \rightarrow_\beta^* C'[y]$ and $T \rightarrow_\beta^* T'$ (y is a fresh variable). If T has the form $\lambda x.cx$, then X' must be of the form $C'[T']$ with $C[y] \rightarrow_\beta^* C'[y]$ and either $T' \equiv T$ or $T' \equiv c$ (y is a fresh variable). Since T, T' and Z have the same types (by rule (Eq_β) and Lemmas 9.5) and we did not use rule (Eq_β) to assign α to X', we have $\vdash^+ C'[Z]\colon \alpha$. Since $Y =_\beta C'[Z]$ we conclude $\vdash^+ Y\colon \alpha$. ☐

We need to prove that $\rightarrow^*$ is Church–Rosser. We essentially use the following result due to Takahashi [49]:

a semi-orthogonal conditional λ-calculus is confluent.

The rules allowed in conditional λ-calculi are rule schemes, but we don't need such generality. For our purpose it is sufficient to borrow from [49] the following:

the β-rule $(\lambda x.M)N \rightarrow_\beta M[N/x]$ is a rule scheme;
the generalized η-rule $\lambda x.Mx \rightarrow_\eta M$ if $x \notin FV(M)$ and $\mathbf{p}(M)$,
where $\mathbf{p}$ is a predicate closed under substitution and reduction, is a rule scheme;
the Mitschke δ-rules [5, Theorem 15.3.3]

$$\begin{array}{ll} \delta M_1 \ldots M_n \rightarrow_\delta N_1 & \text{if } \mathbf{p}_1(M_1,\ldots,M_n) \\ \ldots & \ldots \\ \delta M_1 \ldots M_n \rightarrow_\delta N_k & \text{if } \mathbf{p}_k(M_1,\ldots,M_n) \end{array}$$

where $N_1,\ldots,N_k$ are closed terms and $\mathbf{p}_1,\ldots,\mathbf{p}_k$ are mutually disjoint predicates, closed under substitution and reduction, are rule schemes. (If $\mathbf{p}_i(M_1,\ldots,M_n)$ implies that $M_1,\ldots,M_n$ are closed, then $\mathbf{p}_i$ is obviously closed under substitutions and we need only require closure under reduction.)

We give now the definition of conditional λ-calculus.

Definition 9.8. A conditional λ-calculus is a reduction system $\langle \Lambda_\Xi; \rightarrow_\Xi \rangle$, where:

(1) the terms in Λ_Ξ are constructed from a countable set of variables and function symbols (we consider the abstraction operator λx[9] as a unary function symbol);
(2) $\rightarrow_\Xi$ is a reduction relation induced by a set of rule schemes providing:
  (a) different rule schemes share no redexes;
  (b) each rule scheme is closed under substitution;
  (c) the predicate of each rule scheme is closed under substitution and reduction.[10]

---

[9] In addition to λx there may be other operators which bind variables.

[10] The third condition of [49] is more general, but the present one is sufficient for us.

To define semi-orthogonality, we need the notion of *critical containment* between redexes, which in turn would require the introduction of rule schemes in their full generality. Roughly, a redex R contains critically another redex R' if R' is a subterm of R and they share some function symbols which characterize them as redexes. We hope the following examples be enlightening. The β-redex $(\lambda x.Mx)N$ with $x \notin FV(M)$ contains critically the η-redex $\lambda x.Mx$. Notice that they share the same λx. Instead, the β-redex $(\lambda x.M)(\lambda y.Ny)$ with $y \notin FV(N)$ contains non-critically the η-redex $\lambda y.Ny$.

Definition 9.9. A conditional λ-calculus where:

(1) each redex contains critically at most another redex;

(2) if redex R contains critically R', then the term obtained from R by contracting R' is identical with the contractum of R;

is semi-orthogonal.[11]

In the light of these definitions we examine the rules of $\rightarrow_c$. Clearly, rules $(c-\omega)$ and $(c-zero)$ are Mitschke's δ-rules. Also rules $(c-arrow)$ are Mitschke's δ-rules, thanks to Theorem 9.7, which guarantees the closure under reduction. To better understand why, let us consider the case $I=\{1,2\}$. Then we can rewrite

$$c^{\wedge_{i\in I}(\alpha_i\rightarrow\beta_i)}X \rightarrow_c c^{\wedge_{j\in J}\beta_j} \text{ where } J=\{i\in I \mid \vdash^+ X:\alpha_i\}$$

as

$$\begin{array}{lll}
c^{\wedge_{i\in I}(\alpha_i\rightarrow\beta_i)}X \rightarrow_c c^{\beta_1\wedge\beta_2} & \text{if } \vdash^+ X:\alpha_1\wedge\alpha_2 & \\
c^{\wedge_{i\in I}(\alpha_i\rightarrow\beta_i)}X \rightarrow_c c^{\beta_1} & \text{if } \vdash^+ X:\alpha_1 \text{ and } & \nvdash^+ X:\alpha_2 \\
c^{\wedge_{i\in I}(\alpha_i\rightarrow\beta_i)}X \rightarrow_c c^{\beta_2} & \text{if } \nvdash^+ X:\alpha_1 \text{ and } & \vdash^+ X:\alpha_2 \\
c^{\wedge_{i\in I}(\alpha_i\rightarrow\beta_i)}X \rightarrow_c c^{\omega} & \text{if } \nvdash^+ X:\alpha_1 \text{ and } & \nvdash^+ X:\alpha_2.
\end{array}$$

Rule $(c-\eta)$ is a generalized η-rule, so it is a rule scheme in the sense of [49].

It is easy to verify that $\langle\Lambda(\mathscr{C});\rightarrow^*\rangle$ is a conditional λ-calculus in the sense of Definition 9.8. Moreover $\langle\Lambda(\mathscr{C});\rightarrow^*\rangle$ is semi-orthogonal: the only case of critical containment between redexes is $(\lambda x.cx)F$, but we have $(\lambda x.cx)F \rightarrow_\beta cF$ and $(\lambda x.cx)F \rightarrow_c cF$. So we can conclude:

Theorem 9.10. *The reduction relation* $\rightarrow^*$ *is confluent on* $\Lambda(\mathscr{C})$.

Appendix A contains a direct proof of the above theorem which follows the well-known proof scheme of Tait and Martin-Löf as in [5, Section 3.2].

Definition 9.11. $\sim$ is the least congruence relation on $\Lambda(\mathscr{C})$ containing $\rightarrow_c$ and $\rightarrow_\beta$.

Since $\rightarrow^*$ preserves types also $\sim$ preserves type.

To define our model we restrict $\sim$ to $\Lambda(\mathscr{C})_0$.

---

[11] The definition of semi-orthogonality given in [49] properly includes the present one.

Definition 9.12. For $X \in \Lambda(\mathscr{C})_0$ let $[X] = \{Y \in \Lambda(\mathscr{C})_0 \mid Y \sim X\}$. Define
(1) $\Lambda(\mathscr{C})_0/\sim = \{[X] \mid X \in \Lambda(\mathscr{C})_0\}$;
(2) $[X] \cdot [Y] = [XY]$.

Notice that $\cdot$ is well defined since $\sim$ is a congruence.

Definition 9.13. Given an environment $s: Var \to \Lambda(\mathscr{C})_0/\sim$, $s(x_i) = [X_i]$, and a term $M \in \Lambda$ with free variables $x_1, \dots, x_n$, we define $[\![M]\!]_s^{\Lambda(\mathscr{C})_0}$ as the equivalence class modulo $\sim$ of $M[X_1/x_n, \dots, X_n/x_n]$.

Note that $[\![\ ]\!]_s^{\Lambda(\mathscr{C})_0}$ is well defined since $\sim$ is a compatible relation on $\Lambda(\mathscr{C})$, so the given definition does not depend on the choice of the representatives X_i of the equivalence classes $s(x_i) \in \Lambda(\mathscr{C})_0/\sim$.

Theorem 9.14. *$\langle \Lambda(\mathscr{C})_0/\sim, \cdot, [\![\ ]\!]^{\Lambda(\mathscr{C})_0} \rangle$ is a λ-algebra.*

Proof. Obvious from the fact that $\sim$ is the restriction to the closed terms of a compatible congruence containing $=_\beta$. ☐

Notice that $\langle \Lambda(\mathscr{C})_0/\sim, \cdot, [\![\ ]\!]^{\Lambda(\mathscr{C})_0} \rangle$ is not extensional, and it does not even satisfy the weak form of extensionality required for a λ-model (axiom (ξ)). For example let $\theta = (\zeta \to \omega \to \omega) \wedge ((\omega \to \omega) \to \zeta)$ and $\psi = ((\omega \to \omega) \to \omega \to \omega) \wedge (\zeta \to \zeta)$. The following two terms are extensionally equivalent: $\lambda x.c^\theta(c^\theta x)$ and c^ψ. In fact both terms give c^ω if the input has only type ω, give $c^{\omega \to \omega}$ if the input has an arrow type, and give c^ζ if the input has a zero type. But we cannot show $\lambda x.c^\theta(c^\theta x) \sim c^\psi$. The same example shows that the relation $X \sim X'$ is strictly weaker than the relation "having the same types".

We use the above model to prove the completeness of $\vdash$. We are going to use the fact that, by Lemma 7.6(2), for every $X \in \Lambda(\mathscr{C})_0$ the set $\{\alpha \mid \vdash^+ X : \alpha\}$ is a filter. So in particular we cannot derive both a type of arrow kind and a type of zero kind for X.

The following three lemmas characterize equivalence classes of terms according to their typings.

Lemma 9.15. *$X \sim \mathbf{1}X$ iff $\vdash^+ X : \omega \to \omega$.*

Proof. If $X \sim \mathbf{1}X$, then clearly $\vdash^+ X : \omega \to \omega$. Conversely if $\vdash^+ X : \omega \to \omega$, then by the approximation theorem and the generation lemma either $X =_\beta \lambda x.X'$ or $X =_\beta c^\sigma \vec{X}$ for some arrow type σ and some $\vec{X} = X_1, \dots, X_n$ with $n \geqslant 0$. We have $c^\sigma X_1 X_2 \dots X_n \sim c^\rho X_2 \dots X_n$ where $c^\sigma X_1 \to_c c^\rho$. Also the type ρ must be of arrow kind since $\vdash^+ X : \omega \to \omega$. So we can iterate until we get $X \sim c^\tau$ for some τ of arrow kind. But then $c^\tau \sim \lambda x.c^\tau x =_\beta \mathbf{1}X$ and we are done. ☐

Lemma 9.16. *$X \sim c^\omega$ iff $\{\alpha \mid \vdash^+ X : \alpha\} = \uparrow \{\omega\}$.*

Proof. The "only if" part is easy. For the converse suppose $\{\alpha \mid \vdash^+ X:\alpha\} =\uparrow \{\omega\}$. Then X cannot be β-reduced to a λ-abstraction or to a strong zero term (since otherwise X would have type $\omega \to \omega$ or ζ respectively). So X must have a β-reduct of the form $c\vec{X}$ for some $\vec{X} = X_1,\ldots,X_n$. If $c \equiv c^{\pi}$ with π of zero kind, then $\vdash^+ X:\zeta$, contrary to the assumptions. If $c \equiv c^{\omega}$ we get $X \sim c^{\omega}$ by repeated applications of rule $c^{\omega}Y \to_c c^{\omega}$ and we are done. If $c \equiv c^{\bigwedge_{i\in I}(\alpha_i \to \beta_i)}$ then $\vec{X}$ cannot be empty, since otherwise $\vdash^+ X:\bigwedge_{i\in I}(\alpha_i \to \beta_i)$. We conclude by induction on the length of $\vec{X}$ observing that $cX_1\ldots X_k \to^* c^{\bigwedge_{j\in J}\beta_j}X_2\ldots X_k$ for some $J \subseteq I$. □

Lemma 9.17. (1) *Suppose* $\vdash^+ X:\zeta$ *and* $XY \to^* Z$. *Then* $Z \equiv X'Y'$ *with* $X \to^* X'$ *and* $Y \to^* Y'$.
(2) $\vdash^+ X:\pi\alpha$ *iff* $X \sim YZ$ *for some* Y,Z *with* $\vdash^+ Y:\pi$ *and* $\vdash^+ Z:\alpha$.
(3) $\vdash^+ X:\zeta$ *iff* $[X]$ *is a zero element of* $\Lambda(\mathscr{C})_0/\sim$.

Proof. (1) Since $\vdash^+ X:\zeta$, X cannot reduce to a λ-abstraction otherwise it would also have type $\omega \to \omega$, contradicting the fact that the types of X form a filter. So by definition of $\to^*$ the only possibility is $Z \equiv X'Y'$ with $X \to^* X'$ and $Y \to^* Y'$.

(2) By Theorem 7.5 there is $A \in \mathscr{A}(X)$ such that $\vdash^+ A : \pi\alpha$. By Lemma 6.2(2) the only possibilities are $A \equiv \bot\vec{A}A'$ or $A \equiv c\vec{A}$. In the first case we have $X =_\beta YZ$ with $\bot\vec{A} \in \mathscr{A}(Y)$ and $A' \in \mathscr{A}(Z)$. So by Lemma 6.2(2)(*e*), Lemma 5.3(4) and Theorem 7.5 we get $\vdash^+ Y : \pi$, $\vdash^+ Z : \alpha$. In the second case we have $X =_\beta c\vec{X}$ for some $\vec{X} = X_1\ldots X_n$. If $n > 0$ we get by the same argument $\vdash^+ cX_1\ldots X_{n-1} : \pi$ and $\vdash^+ X_n : \alpha$. If $n = 0$ then by the generation lemma $c \equiv c^{\beta}$ with $\beta \leqslant \pi\alpha$. But then we must have by Lemma 5.3(4),(6) $\beta = \pi'\alpha'$ with $\pi' \leqslant \pi$ and $\alpha' \leqslant \alpha$, and we can take $Y \equiv c^{\pi'}$ and $Z \equiv c^{\alpha'}$.

(3) ($\Rightarrow$) $\vdash^+ X:\zeta$ implies $X \not\sim \mathbf{1}X$ otherwise X would have a type of arrow kind. Moreover $XY \sim XY'$ gives us $Y \sim Y'$ by (1) and the confluence of $\to^*$. So $[X]$ is a zero element.

($\Leftarrow$) Suppose $[X]$ is a zero element and $\nvdash^+ X:\zeta$. Then either $\vdash^+ X:\omega \to \omega$ or $\{\alpha \mid \vdash^+ X:\alpha\} =\uparrow \{\omega\}$. If $\vdash^+ X:\omega \to \omega$, then $X \sim \mathbf{1}X$ by Lemma 9.15. If $\{\alpha \mid \vdash^+ X:\alpha\} =\uparrow \{\omega\}$ then $X \sim c^{\omega}$ by Lemma 9.16. In both cases we have a contradiction since neither $\mathbf{1}X$ nor c^{ω} are zero elements (the latter because $c^{\omega}Y \sim c^{\omega}$ for all Y). □

Now we can prove the adequacy of our model.

Theorem 9.18. *The* λ-*algebra* $\langle \Lambda(\mathscr{C})_0/\sim, \cdot, [\![\ ]\!]^{\Lambda(\mathscr{C})_0}\rangle$ *is adequate.*

Proof. We need to check the conditions of Definition (3.3). Condition (1) follows from Lemma 9.17(3), since $\vdash^+ X : \zeta$ implies $\vdash^+ XY : \zeta$ for all Y. For condition (2) assume $\vdash^+ X : \zeta$, $\vdash^+ X' : \zeta$, and $XY \sim X'Y'$. By the Church-Rosser property and Lemma 9.17(1) there are X'', Y'' such that $X \to^* X''$, $X' \to^* X''$, $Y \to^* Y''$, and $Y' \to^* Y''$. So we conclude $X \sim X'$ and $Y \sim Y'$. Condition (3) follows from rule (ζ) and Lemma 9.17(3). □

Lemma 9.19. $[X] \in [\![\alpha]\!]$ *iff* $\vdash^+ X : \alpha$.

Proof. We prove ($\Leftarrow$) and ($\Rightarrow$) simultaneously by induction on α.

- $\alpha \equiv \zeta$. By Lemma 9.17(3).
- $\alpha \equiv \omega \to \omega$. ($\Leftarrow$). By Lemma 9.15 we get $X \sim \mathbf{1}X$.
  ($\Rightarrow$). Let ad absurdum $\not\vdash^+ X : \omega \to \omega$. By Lemma 9.16 this implies either $\vdash^+ X : \zeta$ or $X \sim c^\omega$. The first case is impossible, since by Lemma 9.17(3) we would have $[X] \in [\![\zeta]\!]$. The second case is also impossible, since $c^\omega \not\sim \mathbf{1}c^\omega$.
- $\alpha \equiv \beta \to \gamma$. ($\Leftarrow$). $\vdash^+ X : \beta \to \gamma$ implies $\vdash^+ X : \omega \to \omega$, so by Lemma 9.15 we get $X \sim \mathbf{1}X$:

$$\begin{array}{lll} [Y] \in [\![\beta]\!] & \Rightarrow \vdash^+ Y : \beta & \text{by induction wrt } (\Rightarrow) \\ & \Rightarrow \vdash^+ XY : \gamma & \text{by rule } (\to \mathrm{E}) \\ & \Rightarrow [X] \cdot [Y] \in [\![\gamma]\!] & \text{by induction} \\ & \Rightarrow [X] \in [\![\beta \to \gamma]\!] & \text{by definition.} \end{array}$$

($\Rightarrow$) Notice that $[X] \in [\![\beta \to \gamma]\!]$ implies $[X] \in [\![\omega \to \omega]\!]$, so we have $\vdash^+ X : \omega \to \omega$ as in the previous case. Moreover:

$$\begin{array}{lll} [X] \in [\![\beta \to \gamma]\!] & \Rightarrow \forall [Y] \in [\![\beta]\!].\ [X] \cdot [Y] \in [\![\gamma]\!] & \text{by definition} \\ & \Rightarrow [X] \cdot [c^\beta] \in [\![\gamma]\!] & \text{since } [c^\beta] \in [\![\beta]\!] \\ & & \text{by induction}(\Leftarrow) \\ & \Rightarrow \vdash^+ Xc^\beta : \gamma & \text{by induction} \\ & \Rightarrow \vdash^+ X : \beta \to \gamma & \text{by Lemma 7.2(3), and} \\ & & \text{Theorem 7.5.} \end{array}$$

- $\alpha \equiv \pi\beta$. ($\Leftarrow$) By Lemma 9.17(2) we have $X \sim YZ$ and $\Gamma \vdash^+ Y : \pi$, $\Gamma \vdash^+ Z : \beta$ for some Y, Z. By induction $[Y] \in [\![\pi]\!]$ and $[Z] \in [\![\beta]\!]$, so we conclude $[X] \in [\![\pi\beta]\!]$.

  For ($\Rightarrow$), notice that $[X] \in [\![\pi\beta]\!]$ implies $YZ \sim X$, for some $Y \in [\![\pi]\!]$ and $Z \in [\![\beta]\!]$. By induction wrt ($\Rightarrow$) $\vdash^+ Y : \pi$ and $\vdash^+ Z : \beta$, so $\vdash^+ YZ : \pi\beta$ by rule (*app*), and we can conclude $\vdash^+ X : \pi\beta$ by Theorem 9.7.
- $\alpha \equiv \beta \wedge \gamma$. Easy by induction. □

Definition 9.20. $s_\Gamma(\ )$ is the term environment defined by $s_\Gamma(x) = [c^\alpha]$ iff $x : \alpha \in \Gamma$.[12]

Theorem 9.21. $\Gamma \models M : \alpha$ *implies* $\Gamma \vdash M : \alpha$.

Proof. We use the model $\langle A(\mathscr{C})_0/\sim, \cdot, [\![\ ]\!]^{A(\mathscr{C})_0}\rangle$, and the environment s_Γ. Let $\Gamma = \{\vec{x}:\vec{\beta}\}$. We get $[\![M]\!]_{s_\Gamma} = [M[\vec{c^\beta}/\vec{x}]]$. $[\![M]\!]_{s_\Gamma} \in [\![\alpha]\!]$ implies $\vdash^+ M[\vec{c^\beta}/\vec{x}] : \alpha$ by Lemma 9.19. So we conclude $\Gamma \vdash M : \alpha$. □

---

[12] By the assumption on bases (see p. 25) we get $s_\Gamma(x) = [c^\omega]$ whenever $x \notin Dom(\Gamma)$.

Appendix A

We introduce a notion of parallel reduction generalizing Definition 3.2.3 of [5].

Definition A.1. We define a binary relation $\rightarrow_p$ on $\Lambda(\mathscr{C})$ called *parallel reduction* as follows.

(1) $F \rightarrow_p F$;
(2) if $F \rightarrow_p F'$, then $\lambda x.F \rightarrow_p \lambda x.F'$;
(3) if $F \rightarrow_p F'$ and $G \rightarrow_p G'$, then $FG \rightarrow_p F'G'$;
(4) if $F \rightarrow_p F'$ and $G \rightarrow_p G'$, then $(\lambda x.F)G \rightarrow_p F'[G'/x]$;
(5) if $F \rightarrow_c F'$, then $F \rightarrow_p F'$
(in this case F and F' are necessarily closed, i.e. they belong to $\Lambda(\mathscr{C})_0$).

The idea is that if $F \rightarrow_p F'$ then F can be reduced to F' in parallel, i.e. without reducing new redexes created by previous reductions. Keeping this in mind, the following lemma should not be surprising.

Lemma A.2. *If $F \rightarrow_p F'$ and $G \rightarrow_p G'$, then $F[G/x] \rightarrow_p F'[G'/x]$.*

Proof. This is proved in [5] for parallel β-reductions by induction on the definition of $F \rightarrow_p F'$. The only new case to take into account is when $F \rightarrow_p F'$ is $F \rightarrow_c F'$. In this case there is nothing to prove since F and F' are closed. □

Lemma A.3. *$\rightarrow_p$ satisfies the diamond property, i.e. if $F \rightarrow_p F_1$ and $F \rightarrow_p F_2$, then there is an F_3 with $F_1 \rightarrow_p F_3$ and $F_2 \rightarrow_p F_3$.*

Proof. By induction on the definition of $F \rightarrow_p F_1$. We adapt the proof of [5, Lemma 3.2.6] for the parallel β-reduction. So we consider only the new cases.

The most interesting case is when $F \equiv c^{\bigwedge_{i\in I}(\alpha_i \rightarrow \beta_i)}X$, $F_1 \equiv c^{\bigwedge_{j\in J}\beta_j}$ with $J = \{i \in I \mid \vdash^+ X : \alpha_i\}$, and $F_2 \equiv c^{\bigwedge_{i\in I}(\alpha_i \rightarrow \beta_i)}X'$ with $X \rightarrow_p X'$. Notice that by Theorem 9.7 we have that X and X' have the same types. Then we can take $F_3 \equiv c^{\bigwedge_{j\in J}\beta_j}$.

If $F \rightarrow_p F_1$ is $c^{\omega}X \rightarrow_c c^{\omega}$, and $F \rightarrow_p F_2$ is $c^{\omega}X \rightarrow_p c^{\omega}X'$, then one takes $F_3 \equiv c^{\omega}$.

If $F \rightarrow_p F_1$ is $(\lambda x.c^{\sigma}x)G \rightarrow_p c^{\sigma}G'$ (by applying either the β-rule or the rule for c^{σ}), and F_2 is either $(\lambda x.c^{\sigma}x)G''$ or $c^{\sigma}G''$, then one can find a common reduct F_3 of the form $c^{\sigma}G'''$.

The other cases are trivial. □

Let $\rightarrow_p^*$ be the transitive closure of $\rightarrow_p$.

Lemma A.4. *$\rightarrow_p^*$ is the least reduction relation containing $\rightarrow_c$ and $\rightarrow_\beta$.*

Proof. Let $\rightarrow$ be the least compatible reflexive relation containing $\rightarrow_\beta$ and $\rightarrow_c$ and let $\rightarrow^*$ be its transitive closure (i.e. $\rightarrow^*$ is defined in Definition 9.6). We have the inclusions $\rightarrow \ \subseteq \ \rightarrow_p \ \subseteq \ \rightarrow_p^* \ \subseteq \ \rightarrow^*$. Since $\rightarrow_p^*$ is the transitive closure of $\rightarrow_p$ we conclude that $\rightarrow_p^*$ coincides with $\rightarrow^*$, and therefore it is the least reduction relation containing $\rightarrow_c$ and $\rightarrow_\beta$. □

Since $\rightarrow_p$ has the diamond property also $\rightarrow_p^*$ has the diamond property and we have thus proved that the least reduction relation containing $\rightarrow_c$ and $\rightarrow_p$ is confluent.

Acknowledgements

Corrado Böhm introduced both authors to the world of research: he was an exceptional guide for his enthusiasm and his encouragement. The authors are very grateful to Chantal Berline for reading an uncountable number of different version of this paper pointing out a long list of mistakes. The authors would like to thank Masako Takahashi-Horai for helpful discussions concerning the confluence proof in Section 9. This paper has strongly benefited from comments and remarks by the Anonymous Referees. During the elaboration of the final version of this paper Mariangiola Dezani-Ciancaglini was visiting the Tokyo Institute of Technology. She would like to thank her host Masako Takahashi-Horai and the whole Department of Mathematical and Computing Sciences for the ideal working conditions they provided.

References

[1] S. Abramsky, Domain theory in logical form, Ann. Pure Appl. Logics 51 (1991) 1–77.

[2] S. Abramsky, C.-H.L. Ong, Full abstraction in the lazy lambda calculus, Inform. and Comput. 105 (1993) 159–267.

[3] F. Alessi, Type Preorders, CAAP'94, Lecture Notes in Computer Science, vol. 787, Springer, Berlin, 1994, pp. 37–51.

[4] J. Baeten, B. Boerboom, **Ω** can be anything it should not be, Indag. Math. 41 (1979) 111–120.

[5] H. Barendregt, The Lambda Calculus Its Syntax and Semantics, Studies in Logic, vol. 103, North-Holland, Amsterdam, 1984.

[6] H. Barendregt, M. Coppo, M. Dezani-Ciancaglini, A filter lambda model and the completeness of type assignment, J. Symbolic Logic 48 (1983) 931–940.

[7] A. Berarducci, Infinite lambda-calculus and non-sensible models, Logic and Algebra, Lecture Notes in Pure and Applied Mathematics, vol. 180, Marcel Dekker, New York, 1996, pp. 339–378.

[8] A. Berarducci, B. Intrigila, Some new results on easy lambda-terms, Theoret. Comput. Sci. 121 (1993) 71–88.

[9] A. Berarducci, B. Intrigila, Church-rosser λ-theories, infinite λ-terms and consistency problems, Logic: from Foundations to Applications, European Logic Colloquium, Clarendon Press, Oxford, 1996, pp. 33–58.

[10] M. Coppo, M. Dezani-Ciancaglini, F. Honsell, G. Longo, Extended type structures and filter lambda models, Logic Colloquium '82, North-Holland, Amsterdam, 1983, pp. 241–262.

[11] M. Coppo, M. Dezani-Ciancaglini, B. Venneri, Principal type schemes and λ-calculus semantics, To H.B.Curry, Essays on Combinatory Logic, Lambda Calculus and Formalism, Academic Press, New York, 1980, pp. 535–560.

[12] M. Coppo, M. Dezani-Ciancaglini, M. Zacchi, Type theories, normal forms and $\mathscr{D}_\infty$ lambda models, Inform. and Control 72(2) (1987) 85–116.

[13] N.G. de Bruijn, Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, Indag. Math. 34 (1972) 381–392.

[14] M. Dezani-Ciancaglini, U. de'Liguoro, A. Piperno, Filter models for conjunctive-disjunctive λ-calculus, Theoret. Comput. Sci. 170(1–2) (1996) 83–128.

[15] M. Dezani-Ciancaglini, U. de'Liguoro, A. Piperno, A filter model for concurrent λ-calculi, SIAM J. Comput. 27(5) (1998) 1376–1419.

[16] M. Dezani-Ciancaglini, J. Tiuryn, P. Urzyczyn, Discrimination by parallel observers, LICS '97, IEEE Comput. Soc. Press, Los Alamitos, 1997, pp. 396–407.

[17] P. Di Gianantonio, F. Honsell, An Abstract Notion of Application, TLCA'93, Lecture Notes in Computer Science, vol. 664, Springer, Berlin, 1993, pp. 124–138.

[18] L. Egidi, F. Honsell, S. Ronchi Della Rocca, Operational, denotational and logical descriptions: a case study, Fund. Inform. 16(2) (1992) 149–170.

[19] J.R. Hindley, The completeness theorem for typing λ-terms, Theoret. Comput. Sci. 22 (1983) 1–17.

[20] J.R. Hindley, J.P. Seldin, Introduction to Combinators and λ-calculus, Cambridge Univ. Press., Cambridge, 1986.

[21] F. Honsell, M. Lenisa, Some Results on the Full Abstraction Problem for Restricted Lambda Calculi, MFCS'93, Lecture Notes in Computer Science, vol. 711, Springer, Berlin, 1993, pp. 84–104.

[22] F. Honsell, S. Ronchi Della Rocca, An approximation theorem for topological lambda models and the topological incompleteness of lambda calculus, J. Comput. System. Sci. 45 (1992) 49–75.

[23] F. Honsell, S. Ronchi Della Rocca, Reasoning about interpretations in qualitative lambda models, Programming Concept and Methods, North-Holland, Amsterdam, 1990, pp. 505–521.

[24] G. Jacopini, A condition for identifying two elements of whatever model of combinatory logic, LCCS, Lecture Notes in Computer Science, vol. 37, Springer, Berlin, 1975, pp. 213–219.

[25] G. Jacopini, M. Venturini Zilli, Equating for recurrent terms of λ-calculus and combinatory logic, Istituto per le applicazioni del calcolo "Mauro Picone", Quaderni serie 3, no. 85, Roma 1978, 3–14.

[26] G. Jacopini, M. Venturini Zilli, Easy terms in the lambda calculus, Fundam. Inform. 8 (2) (1985) 225–233.

[27] Y. Jiang, Consistance et inconsistance de théories de Lambda-calculus étendus, Thése de Doctorat d'État, Université Paris VII, 1993.

[28] Y. Jiang, Consistency of a λ-theory with n-tuples and easy terms, Archive Math. Logic 34 (1995) 79–96.

[29] J.R. Kennaway, J. W. Klop, R. Sleep, F.-J. de Vries, Transfinite reductions in orthogonal term rewriting systems, Inform. and Comput. 119(1) (1995) 18–38.

[30] R. Kennaway, J.W. Klop, R. Sleep, F.-J. de Vries, Infinitary lambda calculus, Theoret. Comput. Sci. 175(1) (1997) 93–126.

[31] J.R. Kennaway, V. van Oostrom, F.J. de Vries, Meaningless Terms in Rewriting, ALP'96, Lecture Notes in Computer Science, vol. 1139, Springer, Berlin, 1996, pp. 254–268.

[32] R. Kerth, Isomorphisme et équivalence équationnelle entre modèles du λ-calcul, Thése de Doctorat d'État, Université Paris VII, 1995.

[33] B. Intrigila, A problem on easy terms in λ-calculus, Fundam. Inform. 15(1) (1991) 99–106.

[34] G. Longo, Set theoretical models of lambda calculus: theory, expansions and isomorphisms, Ann. Pure Appl. Logic 24 (1983) 153–188.

[35] R. Nakajima, Infinite normal forms of the λ-calculus, LCCS, Lecture Notes in Computer Science, vol. 37, Springer, Berlin, 1975, pp. 62–82.

[36] A. Meyer, What is a model of the lambda calculus?, Inform. and Comput. 52 (1982) 87–122.

[37] G. Plotkin, A Powerdomain construction, SIAM J. of Comput. 5 (1976) 457–487.

[38] G. Plotkin, A set-theoretical definition of application, Theoret. Comput. Sci. 121 (1994) 351–410.

[39] A. Pravato, Categorical models of untyped lambda calculi: a monoidal approach, Ph.D. Thesis, Torino University, 1997.

[40] A. Pravato, S. Ronchi Della Rocca, L. Roversi, Categorical semantics of the call-by-value λ-calculus, TLCA'95, Lecture Notes in Computer Science, vol. 902, Springer, Berlin, 1995, pp. 381–396.

[41] S. Ronchi Della Rocca, Characterization theorems for a filter lambda model, Inform. and Control 54 (1982) 201–216.

[42] S. Ronchi Della Rocca, Basic lambda calculus, Notes of the advanced school on typed lambda calculus and functional programming, Udine University, 1994.

[43] S. Ronchi Della Rocca, B. Venneri, Principal type schemes for an extended type theory, Theoret. Comput. Sci. 28 (1984) 151–169.

[44] A. Salibra, R. Goldblatt, A finite equational axiomatization of the functional algebras for the lambda calculus, Inform. and Comput. (to appear).

[45] D.S. Scott, Lambda calculus: some models, some philosophy, The Kleene Symp., North-Holland, Amsterdam, 1980, pp. 223–266.

[46] D.S. Scott, Letter to Albert Meyer, 1980.

[47] D.S. Scott, Domains for denotational semantics, ICALP'82, Lecture Notes in Computer Science, vol. 140, Springer, Berlin, 1982, pp. 577–613.

[48] W.W. Tait, Intensional interpretation of functionals of finite types I, J. Symbolic Logic 32 (1967) 198–212.

[49] M. Takahashi, λ-calculi with conditional rules, TLCA'93, Lecture Notes in Computer Science, vol. 664, Springer, Berlin, 1993, pp. 405–417.

[50] P. Urzyczyn, The emptiness problem for intersection types, LICS'94, IEEE Comput. Soc. Press, Los Alamitos, 1994, pp. 300–309.

[51] S. van Bakel, Complete restrictions of the intersection type discipline, Theoret. Comput. Sci. 102 (1992) 135–163.

[52] G. Winskell, The Formal Semantics of Programming Languages: an Introduction, The MIT Press, Cambridge, 1994.

[53] H. Yokouchi, F-semantics for type assignment systems, Theoret. Comput. Sci. 129 (1994) 39–77.

[54] C. Zylberajch, Syntaxe ed sémantique de la facilité en λ-calcul, Thése de Doctorat d'État, Université Paris VII, 1991.