An Alpha-Corecursion Principle for the Infinitary Lambda Calculus

Alexander Kurz, Daniela Petrişan, Paula Severi, Fer-Jan Vries

An Alpha-Corecursion Principle for the Infinitary Lambda Calculus

Alexander Kurz, Daniela Petrişan, Paula Severi and Fer-Jan de Vries

Department of Computer Science, University of Leicester, UK

Abstract. Gabbay and Pitts proved that lambda-terms up to alpha-equivalence constitute an initial algebra for a certain endofunctor on the category of nominal sets. We show that the terms of the infinitary lambda-calculus form the final coalgebra for the same functor. This allows us to give a corecursion principle for alpha-equivalence classes of finite and infinite terms. As an application, we give corecursive definitions of substitution and of infinite normal forms (Böhm, Lévy-Longo and Berarducci trees).

1 Introduction

Classical λ-calculus considers only finite terms [5]. Infinitary λ-calculus aims to treat finite and infinite terms in one notational framework together with finite and infinite reductions [16,17,15]. In the calculus one can express that certain terms have an infinite normal form. We illustrate this idea with the example of streams. We choose the standard format of the Church numerals [5] to represent natural numbers by λ-terms. In this format $\mathbf{n}$ abbreviates the λ-term $\lambda fx.f^n x$ which represents the natural number n; the successor function is written as $\mathbf{succ} = \lambda yfx.f(yfx)$ and the pairing function as $\mathbf{cons} = \lambda yxz.zxy$. Using the fixed point combinator $\mathbf{fix} = \lambda f.(\lambda x.f(xx))\lambda x.f(xx)$, the function **from** on streams given by

$$\mathbf{from}\ x = \mathbf{cons}\ x\ (\mathbf{from}(\mathbf{succ}\ x))$$

can now be defined in λ-calculus as follows:

$$\mathbf{from} = \mathbf{fix}(\lambda f.\mathbf{cons}\ x\ (f(\mathbf{succ}\ x))$$

The infinite β-reduction sequence

$$\mathbf{from\ 0} \to_\beta \mathbf{cons\ 0}\ (\mathbf{from\ 1}) \to_\beta \mathbf{cons\ 0}\ (\mathbf{cons\ 1}\ (\mathbf{from\ 2}))\ldots \qquad (1)$$

has the infinite λ-term $\mathbf{cons\ 0}\ (\mathbf{cons\ 1}\ (\mathbf{cons\ 2}(\ldots)))$ as a limit.

How are such infinite terms defined? Let Λ be the set of λ-terms over some fixed set $\mathcal{V}$ of variables extended with a constant $\bot$ [5]. The set Λ^∞ of finite and infinite λ-terms is defined by coinduction from the grammar:

$$M ::= \bot \mid x \mid (\lambda x.M) \mid (MM) \tag{2}$$

where x ranges over a given set $\mathcal{V}$ of variables.

A typical corecursive definition on the set Λ^∞ is the one of substitution. Substitution on Λ^∞ is defined as for the set Λ of finite terms but using corecursion instead of recursion. An attempt to define $M[x := N]$ as the result of substituting x by N in M by corecursion is as follows.

$$\begin{array}{rcll} y[x := N] & = & \begin{cases} N & \text{if } y = x \\ y & \text{otherwise} \end{cases} & \\ (PQ)[x := N] & = & (P[x := N]Q[x := N]) & \\ (\lambda y.P)[x := N] & = & \lambda y.(P[x := N]) & \text{if } y \notin \mathsf{fv}(N) \cup \{x\} \end{array} \tag{3}$$

However, this is only a partial map on Λ^∞. As for Λ, the problem resides in the side-condition $y \notin \mathsf{fv}(N) \cup \{x\}$, which is necessary to avoid the capture of free variables in N by the abstraction in $\lambda y.(P[x := N])$. In the finitary λ-calculus, one can define substitution as a total map on the set Λ_α of α-equivalence classes of λ-terms by choosing a suitable representative $\lambda y.P$ such that $y \notin \mathsf{fv}(N) \cup \{x\}$. This approach is not immediately viable for infinitary λ-calculus, because we may have terms (see also (4) below) that exhaust all the available variables, so that we cannot find a fresh y.

Nominal sets [11] provide an elegant way for establishing an α-structural recursion principle [23] that allows one to define substitution on α-equivalence classes of finite λ-terms starting from a partial function with a side-condition [14,23]. It is one of the aims of this paper to establish an α-*co*recursion principle, which allows us to view (3) as a corecursive definition of substitution on α-equivalence classes of terms in Λ^∞.

Before starting on the details, we need to point out a subtlety that arises when treating infinite λ-terms. An infinite term in Λ^∞ may have an infinite set of variables. Suppose that the set $\mathcal{V}$ of variables $\{x_0, x_1, x_2, \ldots\}$ is countable. Then, we may not be able to have 'enough fresh variables' to work on α-equivalence classes. For example, consider the term $\mathbf{allfv} = x_0(x_1(x_2(\ldots)) \in \Lambda^\infty$ which contains all variables from $\mathcal{V}$. In the following β-step

$$(\lambda x_0 x_1.x_0 x_1)\mathbf{allfv} \to_\beta (\lambda x_1.x_0 x_1)[x_0 := \mathbf{allfv}] \tag{4}$$

we have that $x_1 \in \mathsf{fv}(\mathbf{allfv})$ and, therefore, x_1 should be replaced by some fresh variable, which is impossible because **allfv** contains all of them [24]. However, in this paper, we will restrict our attention to the set

$$\Lambda^\infty_{\mathsf{ffv}} = \{M \in \Lambda^\infty \mid \mathsf{fv}(M) \text{ is finite } \} \tag{5}$$

of λ-terms with *finitely many free variables*, avoiding terms such as **allfv**. On the one hand, finitely many free variables are sufficient in order to capture the infinite

normal forms of terms representing programs, since the limit of an infinite β-reduction sequence starting from a finite term has always a finite number of free variables. As a caveat, notice that such a limit may have infinitely many *bound* variables, since additional fresh variables may be needed at each reduction step to avoid capture. On the other hand, restricting to finitely many free variables has the advantage of allowing us to use the machinery of nominal sets [14].

To come back to α-(co)recursion, Pitts [23] derives structural recursion and induction principles for syntax with binding by using the result from [11] that the set $\Lambda/{=_\alpha}$ of α-equivalence classes of finite λ-terms is the initial algebra for the functor L_α on the category Nom of nominal sets defined by

$$\mathsf{L}_\alpha\ U = \mathcal{V} + \{\bot\} + [\mathcal{V}]U + U \times U \tag{6}$$

where $[\mathcal{V}]U$ is the quotient of $\mathcal{V} \times U$ by α-equivalence (and we added a constant $\{\bot\}$ for reasons that will be explained after Definition 1). We will prove that the final coalgebra of the same functor L_α is the (nominal) set $\Lambda^\infty_{\mathsf{ffv}}/{=_\alpha}$ of α-equivalence classes of finite and infinite λ-terms with finitely many free variables. Using this result we will formulate an α-corecursion principle and give corecursive definitions of substitution and various notions of infinite normal form (Böhm, Lévy-Longo and Berarducci trees) on α-equivalence classes of terms.

The structure of the paper is as follows. Sections 2 and 3 review preliminaries on infinitary lambda calculus and nominal sets, respectively. Section 4 contains our main result on the final L_α-coalgebra and Section 5 its application to substitution and to the notions of Böhm, Lévy-Longo and Berarducci trees. Section 6 compares α-corecursion with α-recursion and comments on the relationship between Set- and Nom-based reasoning.

2 Preliminaries on Infinitary Lambda Calculus

We assume familiarity with basic notions and notations of the finite λ-calculus [5]. First we explain how the set Λ^∞ of finite and infinite λ-terms can be constructed as the metric completion of the set Λ of finite λ-terms. Then we will briefly recall some notions and facts of infinitary λ-calculus [17,15]. We leave the treatment of α-conversion for Section 4.

The idea of putting a metric on a set of terms goes at least back to Arnold and Nivat [4]. To do so we define truncations.

Definition 1 (Truncation). *The truncation of a term $M \in \Lambda$ at depth $n \in \mathbb{N}$ is defined by induction on n:*

$$M^0 = \bot$$

$$M^{n+1} = \begin{cases} x & \text{if } M = x \in \mathcal{V} \\ \bot & \text{if } M = \bot \\ \lambda x.N^n & \text{if } M = \lambda x.N \\ N^n P^n & \text{if } M = NP \end{cases} \tag{7}$$

In this definition, we use the constant $\bot$ to represent "unknown information". Later, in the constructions of normal forms of λ-terms we will, as it is customary in infinitary λ-calculus, use the same constant to replace computationally meaningless terms.

Definition 2 (Metric). *We define a metric* $d : \Lambda \times \Lambda \to [0,1]$ *by*

$$d(M,N) = 2^{-m}, \tag{8}$$

where $m = \sup\{n \in \mathbb{N} \mid M^n = N^n\}$ *and we use the convention* $2^{-\infty} = 0$.

In fact, (Λ, d) is an ultrametric space, since for all $M, N, P \in \Lambda$ we have $d(M,N) \leq \max\{d(M,P), d(P,N)\}$, as one can easily check.

The set Λ^∞ of finite and infinite λ-terms (with the additional constant $\bot$) is now defined as the metric completion of the set Λ of finite terms with respect to the metric d. Alternatively, Λ^∞ can be defined by interpreting (2) as a coinductive definition. The fact that both definitions coincide is a consequence of Barr's theorem on final coalgebras for bicontinuous Set endofunctors [6, Theorem 3.2].

Indeed, interpreting (2) coinductively amounts to taking as λ-terms the elements of the final coalgebra for the Set-endofunctor

$$L\ U = \mathcal{V} + \{\bot\} + \mathcal{V} \times U + U \times U \tag{9}$$

Notice that the set Λ of finite λ-terms constitutes the initial algebra for L. A closer look at the proof of Barr [6, Theorem 3.2 and Proposition 3.1] shows now that the metric d on Λ of Definition 2 coincides with the metric induced by the final coalgebra. Hence, by [6, Proposition 3.1], the completion of the initial L-algebra Λ in the metric d is the final L-coalgebra.

To summarise, the final L-coalgebra $(\Lambda^\infty, \mathbf{unfold} : \Lambda^\infty \to L(\Lambda^\infty))$ is the Cauchy completion of Λ and we have a dense inclusion map $\iota : \Lambda \to \Lambda^\infty$. It is well-known that the structure map of the final coalgebra $\mathbf{unfold} : \Lambda^\infty \to L(\Lambda^\infty)$ is an isomorphism, hence the set $L(\Lambda^\infty)$ can be equipped with a complete metric. The map $\mathbf{unfold} : \Lambda^\infty \to L(\Lambda^\infty)$ is the unique uniformly continuous map from Λ^∞ to $L(\Lambda^\infty)$ making diagram (10) commutative:

$$\begin{array}{ccc} \Lambda & \xrightarrow{\simeq} & L(\Lambda) \\ {\scriptstyle\iota}\downarrow & & \downarrow{\scriptstyle L(\iota)} \\ \Lambda^\infty & \xrightarrow[\mathbf{unfold}]{} & L(\Lambda^\infty) \end{array} \tag{10}$$

Having defined the set Λ^∞ of finite and infinite λ-terms we now extend the usual syntactic conventions for finite λ-calculus to infinitary λ-calculus. Terms and variables will respectively be written with (super- and subscripted) letters M, N and x, y, z. Terms of the form $(M_1 M_2)$ and $(\lambda x M)$ will respectively be called applications and abstractions.

The truncation of an infinite term $M \in \Lambda^\infty$ at depth n is defined just as in Definition 1 by induction on n. Observe that $(M^n)_{n\in\mathbb{N}}$ is a Cauchy sequence in (Λ^∞, d) that converges to M.

The set of free and bound variables of a finite term M is defined as usual and denoted by $\mathsf{fv}(M)$ and $\mathsf{bv}(M)$ respectively. We extend $\mathsf{fv}(M), \mathsf{bv}(M)$ to infinite terms $M \in \Lambda^\infty$ using truncations as below. Also, $\mathsf{var}(M) = \mathsf{fv}(M) \cup \mathsf{bv}(M)$.

$$\mathsf{fv}(M) = \bigcup_{n\in\mathbb{N}} \mathsf{fv}(M^n) \qquad \mathsf{bv}(M) = \bigcup_{n\in\mathbb{N}} \mathsf{bv}(M^n)$$

We define β-reduction on Λ^∞ and denote it as $\to_\beta$ in the usual way: the smallest relation that contains $(\lambda x.P)Q \to_\beta P[x := Q]$ and is closed under contexts. The reflexive and transitive closure of $\to_\beta$ is denoted by $\twoheadrightarrow_\beta$. For the definition of $\twoheadrightarrow_\beta$ that assumes a sequence of reduction steps of any ordinal length, see for instance [16]. Terms of the form $(\lambda x.P)Q$ are called redexes. Normal forms are terms without redexes and hence cannot be changed by further computation.

The definition of infinitary λ-calculus is completed by adding $\bot$-reduction, denoted by $\to_\bot$: the smallest relation closed under context that contains $M \to_\bot \bot$ for M belonging to some fixed set $\mathcal{U}$ of *meaningless terms*. If and only if the set $\mathcal{U}$ satisfies certain properties, the resulting infinitary calculus is confluent and normalising, in which case each term has a unique normal form [17,15].

The normal form of a λ-term can be thought to represent its meaning, the maximal amount of information embodied in the term, stable in the sense that it cannot be changed by further computation. Note that this concept of meaning depends on the chosen set $\mathcal{U}$ of meaningless terms for which there is ample, uncountable choice [25].

Böhm, Lévy-Longo and Berarducci trees

For concrete sets of meaningless terms an alternative, "informal" corecursive definition of the normal form of a term in the corresponding infinitary λ-calculus can sometimes be given. Three of them are well known.

In his book [5], Barendregt argued that the terms without head normal forms should be considered as meaningless terms. Any finite λ-term is either a head normal form (hnf), that is, a term of the form $\lambda x_1 \dots \lambda x_n.xN_1 \dots N_m$, or it is a term of the form $\lambda x_1 \dots \lambda x_n.((\lambda y.P)Q)N_1 \dots N_m$ where the redex $(\lambda y.P)Q$ is called the head redex. Starting with a term M that is not in hnf one can repeatedly contract the head redex. Either this will go on forever or terminate with a hnf, which represents part of the information embodied in a term. In the latter case one can repeat this process on the subterms N_i to try to compute more information. This idea led Barendregt to his elegant "informal" definition of the Böhm tree $\mathbf{BT}(M)$ of a term M, that we now recognise as a corecursive definition.

$$\mathbf{BT}(M) = \begin{cases} \lambda x_1 \dots \lambda x_n.y\mathbf{BT}(M_1) \dots \mathbf{BT}(M_m) & \\ \qquad\qquad\qquad\qquad\qquad\qquad \textit{if } M \twoheadrightarrow_\beta \lambda x_1 \dots \lambda x_n.yM_1 \dots M_m \\ \bot \quad \textit{otherwise} \end{cases} \tag{11}$$

Taking for $\mathcal{U}$ the set of terms without hnf, one can show using the confluence property that the normal forms of the corresponding infinite λ-calculus satisfy

the equations in (11). That is, the Böhm tree of a term M is the normal form of M in the infinitary λ-calculus that equates all terms without head normal form with $\bot$ [5,17].

Alternatively, as Abramsky has forcefully argued in [1], one can take the set of terms without weak head normal form (whnf) as set of meaningless terms. Any finite λ-term is either a weak head normal form, that is, a term of either of the two forms $xM_1 \dots M_m$, or $\lambda x.N$, or it is a term of the form $((\lambda y.P)Q)M_1 \dots M_m$ where the redex $(\lambda y.P)Q$ is called the weak head redex. In perfect analogy with before, starting with a term M that is not in whnf one can repeatedly contract the weak head redex. Either this will go on forever or terminate with a whnf. In the latter case one can repeat this process on the subterms M_i of the tail of the whnf or on the subterm N of its body to try to compute more information. This describes a lazy computation strategy, that postpones reduction under abstractions as much as possible.

The normal forms of the corresponding infinitary λ-calculus that equates all terms without weak head normal form with $\bot$ satisfy the equations (12) that define the Lévy-Longo tree $\mathbf{LLT}(M)$ of a term M corecursively [19,18,2,17].

$$\mathbf{LLT}(M) = \begin{cases} y\mathbf{LLT}(M_1)\dots\mathbf{LLT}(M_m) & \text{if } M \twoheadrightarrow_\beta yM_1\dots M_m \\ \lambda x.\mathbf{LLT}(N) & \text{if } M \twoheadrightarrow_\beta \lambda x.N \\ \bot & \text{otherwise} \end{cases} \tag{12}$$

The least set of meaningless terms that gives rise to a confluent and normalising infinitary λ-calculus is the set of terms without a top normal form. Here a term M is a top normal form (tnf) if it is either a variable, an abstraction or an application of the form M_1M_2 in which M_1 is a *zero term*, i.e. a term that cannot reduce to an abstraction. The well-known term $\mathbf{\Omega} = (\lambda x.xx)(\lambda x.xx)$ has no tnf. The normal forms of this calculus can alternatively be characterised by the corecursive definition of the Berarducci tree [7,17] $\mathbf{BerT}(M)$ of a term M:

$$\mathbf{BerT}(M) = \begin{cases} x & \text{if } M \twoheadrightarrow_\beta x \\ \lambda x.\mathbf{BerT}(N) & \text{if } M \twoheadrightarrow_\beta \lambda x.N \\ \mathbf{BerT}(P)\mathbf{BerT}(Q) & \text{if } M \twoheadrightarrow_\beta NP \text{ and } N \text{ is a zero term} \\ \bot & \text{otherwise} \end{cases} \tag{13}$$

It is possible to formalise (11)-(13) using corecursion via the final L-coalgebra, provided we give concrete reduction strategies to compute the various forms used in the definitions. However, in order to take into account α-conversion, we will prove an α-corecursion principle based on nominal sets.

3 Preliminaries on Nominal Sets

We recall basic facts on nominal sets [14]. Consider a countable infinite set $\mathcal{V}$ of 'variables' (or 'atoms' or 'names') and the group $\mathfrak{S}(\mathcal{V})$ of permutations on $\mathcal{V}$ generated by transpositions, which are permutations of the form $(x\ y)$ that swap x and y. Consider a set U equipped with an action of the group $\mathfrak{S}(\mathcal{V})$, denoted by $\cdot : \mathfrak{S}(\mathcal{V}) \times U \to U$. We say that $u \in U$ is supported by a set $S \subseteq \mathcal{V}$ when for

all $\pi \in \mathfrak{S}(\mathcal{V})$ such that $\pi(x) = x$ for all $x \in S$ we have $\pi \cdot u = u$. We say that $u \in U$ is finitely supported if there exists a finite $S \subseteq \mathcal{V}$ which supports u.

Definition 3 (Nominal set). *A nominal set $(U, \cdot)$ is set U equipped with a $\mathfrak{S}(\mathcal{V})$-action such that all elements of U are finitely supported. Given nominal sets $(U, \cdot)$ and $(V, \cdot)$, a map $f : U \to V$ is called equivariant when $f(\pi \cdot u) = \pi \cdot f(u)$ for all $\pi \in \mathfrak{S}(\mathcal{V})$ and $u \in U$. The category of nominal sets and equivariant maps is denoted by* Nom.

A crucial property of nominal sets is that each element of a nominal set has a least finite support, see [14]. Indeed, if two finite sets S_1 and S_2 support u, then their intersection also supports u. The smallest finite support of u is denoted by $\mathsf{supp}(u)$. If $x \in \mathcal{V} \setminus \mathsf{supp}(u)$ we say that x is *fresh* for u, and write $x\#u$. More generally, given two nominal sets $(U, \cdot)$ and $(V, \cdot)$, $u \in U$ and $v \in V$, we write $u\#v$ for $\mathsf{supp}(u) \cap \mathsf{supp}(v) = \emptyset$. An important property of supp is that for every equivariant $f : U \to V$ and $u \in U$, we have $\mathsf{supp}(f(u)) \subseteq \mathsf{supp}(u)$.

Example 4. The set of names $\mathcal{V}$ equipped with the evaluation action given by $\pi \cdot x = \pi(x)$ is a nominal set.

Example 5. The set Λ of finite λ-terms with the action $\cdot : \mathfrak{S}(\mathcal{V}) \times \Lambda \to \Lambda$ inductively defined by

$$\begin{aligned} \pi \cdot x &= \pi(x) \\ \pi \cdot \bot &= \bot \\ \pi \cdot (\lambda x.M) &= \lambda \pi(x).(\pi \cdot M) \\ \pi \cdot (MN) &= ((\pi \cdot M)(\pi \cdot N)) \end{aligned} \tag{14}$$

is a nominal set. In this example we do not take into account α-conversion, so the support of a λ-term M is the set of all variables occurring either bound or free in M.

Given a $\mathfrak{S}(\mathcal{V})$-action $\cdot$ on a set U, let U_{fs} denote the set

$$U_{\mathsf{fs}} = \{u \in U \mid u \textit{ is finitely supported}\}. \tag{15}$$

Then $\cdot$ restricts to a $\mathfrak{S}(\mathcal{V})$-action on U_{fs} and $(U_{\mathsf{fs}}, \cdot)$ is a nominal set.

Example 6. The set Λ^∞ of finite and infinite λ-terms can be equipped with the action $\cdot : \mathfrak{S}(\mathcal{V}) \times \Lambda^\infty \to \Lambda^\infty$ defined *coinductively* by (14). Alternatively, $\pi \cdot (-)$ can be defined using the universal property of the metric completion, as the unique map that extends $\Lambda \xrightarrow{\pi \cdot (-)} \Lambda \xrightarrow{\iota} \Lambda^\infty$. Observe that $(\pi \cdot M)^n = \pi \cdot M^n$ for all $M \in \Lambda^\infty$ and $n \in \mathbb{N}$. Notice that $(\Lambda^\infty, \cdot)$ is not a nominal set since the set of variables in a term, and hence its support, can be infinite. But $((\Lambda^\infty)_{\mathsf{fs}}, \cdot)$ is a nominal set and $\mathsf{supp}(M) = \mathsf{var}(M)$ for all $M \in (\Lambda^\infty)_{\mathsf{fs}}$.

Definition 7 (Abstraction). *Let $(U, \cdot)$ be a nominal set. One defines $\sim_\alpha$ on $\mathcal{V} \times U$ by*

$$(x_1, u_1) \sim_\alpha (x_2, u_2) \Leftrightarrow (\exists\ z\#(x_1, u_1, x_2, u_2))(x_1\ z) \cdot u_1 = (x_2\ z) \cdot u_2 \tag{16}$$

The $\sim_\alpha$-equivalence class of (x,u) is denoted by $\langle x\rangle u$. The abstraction $[\mathcal{V}]U$ of the nominal set U is the quotient $(\mathcal{V}\times U)/\sim_\alpha$. The $\mathfrak{S}(\mathcal{V})$-action on $[\mathcal{V}]U$ is defined by

$$\pi\cdot\langle x\rangle u = \langle \pi\cdot x\rangle \pi\cdot u. \tag{17}$$

Given equivariant $f:(U,\cdot)\to(V,\cdot)$, we define $[\mathcal{V}]f:[\mathcal{V}]U\to[\mathcal{V}]V$ by

$$\langle x\rangle u \mapsto \langle x\rangle f(u). \tag{18}$$

Definition 8 (Concretion). *Let $(U,\cdot)$ be a nominal set. Concretion is the partial function $@:[\mathcal{V}]U\times\mathcal{V}\to U$ with $\langle y\rangle u@z$, the 'concretion of $\langle y\rangle u$ at z', defined as $\langle y\rangle u@z = (z\ y)\cdot u$ if $z\in\mathcal{V}\setminus \mathsf{supp}(\langle y\rangle u)$.*

Notice that $y\#\langle y\rangle u$ and $(\langle y\rangle u)@y = u$. Moreover, concretion is equivariant. If $z\#\langle y\rangle u$ then $\pi\cdot z\#\langle \pi\cdot y\rangle\pi\cdot u$ and $\pi\cdot(\langle y\rangle u@z) = (\langle\pi\cdot y\rangle\pi\cdot u)@\pi\cdot z$.

Further, we recall some general results form [22] that will be necessary to prove our main result on the final L_α-coalgebra. The category Nom is complete and cocomplete. The forgetful functor to Set creates finite products and all colimits. For example, the product of two nominal sets $(U,\cdot)$ and $(V,\cdot)$ is $(U\times V,\cdot)$ where

$$\pi\cdot(u,v) = (\pi\cdot u, \pi\cdot v).$$

Arbitrary products in Nom are computed differently than in Set. Given a family of nominal sets $(U_i,\cdot_i)_{i\in I}$, we can equip the set of all tuples $\{(u_i)_{i\in I} \mid u_i\in U_i\}$ with the pointwise action given by

$$\pi\cdot(u_i)_{i\in I} = (\pi\cdot u_i)_{i\in I}. \tag{19}$$

This is a $\mathfrak{S}(\mathcal{V})$-action, but some tuples may not be finitely supported. The product of $(U_i,\cdot_i)_{i\in I}$ in Nom is the nominal set $(\prod_{i\in I}(U_i,\cdot_i))_{\mathsf{fs}}$ of tuples of the form $(u_i)_{i\in I}$ that are finitely supported with respect to the action of (19).

The abstraction functor $[\mathcal{V}](-):\mathsf{Nom}\to\mathsf{Nom}$ preserves all limits.

4 Infinitary Lambda Calculus in Nominal Sets

This section contains our main technical result, Theorem 22, which is then used to show Theorem 23, stating that the set $\Lambda^\infty_{\mathsf{ffv}}/{=_\alpha}$ of finite and infinite λ-terms up to α-equivalence is the final coalgebra for the functor L_α on Nom, see (6).

On finite terms, α-conversion can be defined inductively using the permutation action $\cdot:\mathfrak{S}(\mathcal{V})\times\Lambda\to\Lambda$ of Example 5, see [14].

Definition 9 (α-conversion on finite λ-terms). *Let $M,N,M',N'\in\Lambda$.*

$$\frac{}{x=_\alpha x}\,(var) \qquad \frac{}{\bot=_\alpha\bot}\,(bot)$$

$$\frac{M=_\alpha N \quad M'=_\alpha N'}{MM'=_\alpha NN'}\,(app) \qquad \frac{(x\ z)\cdot M =_\alpha (y\ z)\cdot N \quad z\#(x,y,M,N)}{\lambda x.M =_\alpha \lambda y.N}\,(abs)$$

The relation $=_\alpha$ is equivariant, that is, $M =_\alpha N$ implies $\pi \cdot M =_\alpha \pi \cdot N$ for all $\pi \in \mathfrak{S}(\mathcal{V})$. Thus we obtain a nominal set $(\Lambda/=_\alpha, \cdot)$ where $\mathsf{supp}(M) = \mathsf{fv}(M)$.

Definition 10 (L_α). *We define* $\mathsf{L}_\alpha : \mathsf{Nom} \to \mathsf{Nom}$ *as follows.*

$$\begin{aligned} \mathsf{L}_\alpha\ U &= \mathcal{V} + \{\bot\} + [\mathcal{V}]U + U \times U \\ \mathsf{L}_\alpha(f) &= [\mathsf{Id}^{\mathcal{V}}, \mathsf{Id}^{\bot}, [\mathcal{V}]f, f \times f] \end{aligned}$$

The nominal set $(\Lambda/=_\alpha, \cdot)$ of α-equivalence classes of finite λ-terms is the initial algebra for the functor L_α [14].

We define α-conversion on the set Λ^∞ using truncations. This definition is slightly different from, though equivalent, to those used in [16,17,15].

Definition 11 (α**-conversion on finite and infinite** λ**-terms).** *We extend the notion of* α*-conversion to the set* Λ^∞ *via*

$$M =_\alpha N \text{ iff } M^n =_\alpha N^n \text{ for all } n \in \mathbb{N}.$$

If $M =_\alpha N$ then $M^n =_\alpha N^n$. The notion of truncation can be extended to $\Lambda^\infty/=_\alpha$ via $[M]^n_\alpha = [M^n]_\alpha$.

Notation 12 (α**-equivalence classes)** *In this section, metavariables of elements in* $\Lambda^\infty/=_\alpha$ *are denoted by* X, Y*. The* α*-equivalence class of* M *is denoted by* $[M]_\alpha$*.*

The reason we need to clearly distinguish between the class X and the term $M \in X$ is that in Theorem 22 we need to construct a Cauchy sequence of representatives $M_1, M_2, \ldots$ from a Cauchy sequence $X_1, X_2, \ldots$ of α-equivalence classes.

A second approach to define the set of α-equivalence classes of infinitary terms is to consider the metric completion of the quotient $\Lambda/=_\alpha$.

Definition 13 (Metric on α**-equivalence classes).** *We define* $d_\alpha : \Lambda \times \Lambda \to [0,1]$ *via*

$$d_\alpha(M, N) = 2^{-m}, \tag{20}$$

where $m = sup\{n \in \mathbb{N} \mid M^n =_\alpha N^n\}$ *and we use the convention* $2^{-\infty} = 0$*.*

We have that d_α is a pseudometric on Λ and $d_\alpha(M, N) = 0$ if and only if $M =_\alpha N$. Thus d_α gives rise to a metric on $\Lambda/=_\alpha$ denoted by abuse of notation also by d_α. Observe that d_α extends to a pseudometric $d^\infty_\alpha : \Lambda^\infty \times \Lambda^\infty \to [0,1]$ also given by

$$d^\infty_\alpha(M, N) = 2^{-m}, \tag{21}$$

where $m = sup\{n \in \mathbb{N} \mid M^n =_\alpha N^n\}$ with the convention $2^{-\infty} = 0$. Then $M =_\alpha N$ in the sense of Definition 11 if and only if $d^\infty_\alpha(M, N) = 0$. Hence we obtain a metric, denoted also by d^∞_α on $\Lambda^\infty/=_\alpha$.

We consider the metric completion of $\Lambda/=_\alpha$ with respect to d_α and denote it by $(\Lambda/=_\alpha)^\infty$.

Theorem 14. *Let $\mathcal{V}$ be uncountable. Then, we have that $(\Lambda^\infty/{=_\alpha}, d_\alpha^\infty)$ is isomorphic to $(\Lambda/{=_\alpha})^\infty$.*

We do not include the proof of this theorem, since in this paper we are only interested in the case where $\mathcal{V}$ is countable. The idea of the proof is to show that $(\Lambda^\infty/{=_\alpha}, d_\alpha^\infty)$ is a complete metric space and then to use the universality property of the metric completion. This argument fails when the set of variables is at most countable. Indeed, we can show that for countable $\mathcal{V}$ the space $(\Lambda^\infty/{=_\alpha}, d_\alpha^\infty)$ is not complete.

Example 15. Assume that $\mathcal{V}$ is countable, say $\mathcal{V} = \{x_0, x_1, \ldots\}$ and consider the sequence $([\lambda x_n.x_n(x_0(\ldots x_{n-1}))]_\alpha)_{n\geq 1}$ in $\Lambda^\infty/{=_\alpha}$. This is a Cauchy sequence with respect to d_α^∞, but has no limit in $\Lambda^\infty/{=_\alpha}$. Indeed, assume for a contradiction that the limit L exists. On the one hand we can prove that $\mathsf{fv}(L) = \mathcal{V}$, on the other hand L should be of the form $\lambda u.ux_0x_1....$ for some variable u. But this contradicts the fact that u is free in L.

These problems of α-equivalence in the presence of countably many variables disappear if we consider the set $\Lambda_{\mathsf{ffv}}^\infty$ of terms with finitely many free variables (5) discussed in the introduction.

Remark 16. Note that $\Lambda_{\mathsf{ffv}}^\infty$ is different from the set $(\Lambda^\infty)_{\mathsf{fs}} = \Lambda_{\mathsf{fs}}^\infty$, defined in (15), of λ-terms with finitely many variables, bound or free. To see this, consider the term

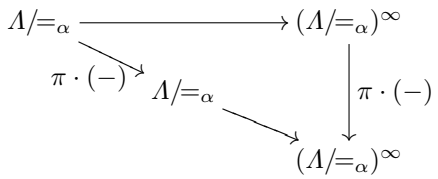
$$\mathbf{allbv} = \lambda x_1.x_1(\lambda x_2.x_1(x_2(\lambda x_3.x_1(x_2(x_3(\ldots)))))).$$

We have that $\mathbf{allbv} \in \Lambda_{\mathsf{ffv}}^\infty$ but $\mathbf{allbv} \notin \Lambda_{\mathsf{fs}}^\infty$. Actually, $[\mathbf{allbv}]_\alpha \cap \Lambda_{\mathsf{fs}}^\infty = \emptyset$ since every term in $[\mathbf{allbv}]_\alpha$ has infinitely many bound variables.

Moreover, the equivalence relation $=_\alpha$ (see Definition 11) restricts to $\Lambda_{\mathsf{ffv}}^\infty$, but not to $\Lambda_{\mathsf{fs}}^\infty$. For example, we have $\lambda x_1.\lambda x_2.\lambda x_3 \ldots =_\alpha \lambda x_1.\lambda x_1.\lambda x_1 \ldots$, but only the latter term is in $\Lambda_{\mathsf{fs}}^\infty$.

For all $M, N \in \Lambda_{\mathsf{ffv}}^\infty$ we have that $M =_\alpha N$ implies $\pi \cdot M =_\alpha \pi \cdot N$. Hence we can equip $\Lambda_{\mathsf{ffv}}^\infty/{=_\alpha}$ with a $\mathfrak{S}(\mathcal{V})$ action given by $\pi \cdot [M]_\alpha = [\pi \cdot M]_\alpha$ and we can easily check that $(\Lambda_{\mathsf{ffv}}^\infty/{=_\alpha},\ \cdot)$ is a nominal set. Indeed, $[M]_\alpha \in \Lambda_{\mathsf{ffv}}^\infty/{=_\alpha}$ is supported by the finite set $\mathsf{fv}(M)$.

The permutation action on $\Lambda/{=_\alpha}$ can be extended to $(\Lambda/{=_\alpha})^\infty$ as follows. For each $\pi \in \mathfrak{S}(\mathcal{V})$ we have that $\pi\cdot(-) : \Lambda/{=_\alpha} \to \Lambda/{=_\alpha}$ is a uniformly continuous function with respect to d_α, thus can be extended to a uniformly continuous map on $(\Lambda/{=_\alpha})^\infty$ using the universal property of the metric completion:



Thus we have a nominal set $((\Lambda/{=_\alpha})_{\mathsf{fs}}^\infty, \cdot)$. A Cauchy sequence $(X_n)_{n\in\mathbb{N}}$ in $\Lambda/{=_\alpha}$ is finitely supported when there exists a finite set $S \subseteq \mathcal{V}$ such that S supports X_n for all $n \in \mathbb{N}$.

Lemma 17. *$X \in (\Lambda/{=_\alpha})^\infty_{\mathsf{fs}}$ if and only if there exists a finitely supported Cauchy sequence of elements of $\Lambda/{=_\alpha}$ converging to X.*

Proof. The right-to-left implication is trivial. For the left-to right implication, consider a Cauchy sequence $(X_n)_{n\in\mathbb{N}}$ in $\Lambda/{=_\alpha}$ converging to X. Then for all $k \in \mathbb{N}$ the sequence of truncations $(X^k_n)_{n\in\mathbb{N}}$ is constant from a point onwards and let Y_k denote its limit in $\Lambda/{=_\alpha}$. We can show that each π that fixes $\mathsf{supp}(X)$ also fixes Y_k. The sequence $(Y_k)_{k\in\mathbb{N}}$ is a finitely supported Cauchy sequence that converges to X.

Remark 18. From the proof of Lemma 17 it follows that each $X \in (\Lambda/{=_\alpha})^\infty$ can be expressed as the limit of a 'canonical' Cauchy sequence $(X_n)_{n\in\mathbb{N}}$ such that the truncation of X_{n+1} at depth n is equal to X_n.

Definition 19 (α-safe term). *Let $M \in \Lambda$. We define the set of α-safe terms $\Gamma \subseteq \Lambda$ by the following inductive rules*

$$\frac{}{x \in \Gamma}\,(var) \qquad \frac{}{\bot \in \Gamma}\,(bot)$$

$$\frac{M \in \Gamma \quad N \in \Gamma \quad \mathsf{bv}(M)\#N \quad \mathsf{bv}(N)\#M}{MN \in \Gamma}\,(app) \qquad \frac{M \in \Gamma \quad x \notin \mathsf{bv}(M)}{\lambda x.M \in \Gamma}\,(abs)$$

Intuitively, a λ-term M is α-safe when $\mathsf{bv}(M) \cap \mathsf{fv}(M) = \emptyset$ and M does not have two different λ's with the same binding variable, i.e. if λx and λy occur in two different positions of M then $x \neq y$. This is slightly stronger than Barendregt's Variable Convention, see [5, Convention 2.1.13].

In Definition 19 we interpret the freshness relation $\#$ in the nominal set $(\Lambda, \cdot)$ of Example 5. So, $\mathsf{bv}(M)\#N$ means that $\mathsf{bv}(M) \cap (\mathsf{bv}(N) \cup \mathsf{fv}(N)) = \emptyset$.

In the next lemma, we will prove that we can always find an α-safe term in an α-equivalence class. Note that the set of α-safe terms is not closed under β. For example, $(\lambda x.xx)(\lambda y.y)$ is α-safe but $(\lambda y.y)(\lambda y.y)$ is not.

Lemma 20 (Choosing representatives). *Let $M \in \Lambda$ and a finite set $S \subseteq \mathcal{V}$. There exists an α-safe term N such that $\mathsf{bv}(N) \cap S = \emptyset$ and $M =_\alpha N$.*

Proof. The proof is by induction on the structure of M. It is immediate for the base cases $M = \bot$ and $M = x \in \mathcal{V}$.

Assume $M = PQ$. By the induction hypothesis there exists a α-safe term P' such that $\mathsf{bv}(P') \cap (\mathsf{fv}(M) \cup S) = \emptyset$ and $P =_\alpha P'$. Applying the induction hypothesis again, there exists a α-safe term Q' such that $\mathsf{bv}(Q') \cap (\mathsf{bv}(P') \cup \mathsf{fv}(M) \cup S) = \emptyset$ and $Q =_\alpha Q'$. Then $M =_\alpha P'Q'$ and we can check that $P'Q'$ is α-safe and $\mathsf{bv}(P'Q') \cap S = \emptyset$.

The case $M = \lambda x.P$ is solved similarly. By the induction hypothesis there exists a α-safe term P' such that $P =_\alpha P'$ and $\mathsf{bv}(P') \cap (S \cup \mathsf{fv}(M) \cup \{x\}) = \emptyset$. Then $\lambda x.P'$ is α-safe by Definition 19, $M =_\alpha \lambda x.P'$ and $S\#\lambda x.P'$.

Lemma 21. *Let $M, P \in \Lambda$ be two α-safe terms such that $P^n =_\alpha M$, $\mathsf{bv}(M)\#P$, and $\mathsf{bv}(P)\#M$. Then there exists a α-safe term N such that*

1. $N =_\alpha P$,
2. *the truncation of* N *at depth* n *is exactly* M *in* Λ,
3. $\mathsf{bv}(N) \subseteq \mathsf{bv}(M) \cup \mathsf{bv}(P)$.

Proof. This is proved by induction on the structure of M. We prove the case that $M = \lambda x.M_0$. Then $P = \lambda y.P_0$. Since $y \# M$, we have that $((y\ x)\cdot P_0)^{n-1} =_\alpha M_0$. By induction hypothesis, there exists N_0 such that $N_0 =_\alpha (y\ x)\cdot P_0$, $N_0^{n-1} = M_0$ and $\mathsf{bv}(N_0) \subseteq \mathsf{bv}(M_0) \cup \mathsf{bv}((y\ x)P_0)$. We set $N = \lambda x.N_0$. We can check that N is α-safe and satisfies the required properties.

In the following theorem, we show that Theorem 14 holds for countable $\mathcal{V}$ if interpreted internally in the category of nominal sets.

Theorem 22 (Completion and quotient by $=_\alpha$ commute in Nom). *Let* $\mathcal{V}$ *be countable. The nominal sets* $(\Lambda^\infty_{\mathsf{ffv}}/{=_\alpha},\ \cdot)$ *and* $((\Lambda/{=_\alpha})^\infty_{\mathsf{fs}}, \cdot)$ *are isomorphic.*

Proof. Define $f : \Lambda^\infty_{\mathsf{ffv}}/{=_\alpha} \to (\Lambda/{=_\alpha})^\infty_{\mathsf{fs}}$ by

$$\Lambda^\infty_{\mathsf{ffv}}/{=_\alpha} \ni X \mapsto \lim_{n\to\infty} X^n \in (\Lambda/{=_\alpha})^\infty_{\mathsf{fs}}$$

where X^n is the truncation of X at depth n. This is well defined by Lemma 17 and the fact that the sequence $(X^n)_{n\in\mathbb{N}}$ is supported by the finite set $\mathsf{supp}(X)$. It is easy to check that f is equivariant.

Next we give $g : (\Lambda/{=_\alpha})^\infty_{\mathsf{fs}} \to \Lambda^\infty_{\mathsf{ffv}}/{=_\alpha}$. Consider $X \in (\Lambda/{=_\alpha})^\infty_{\mathsf{fs}}$. By Lemma 17 we have a finitely supported Cauchy sequence $(X_n)_{n\in\mathbb{N}} \subseteq \Lambda/{=_\alpha}$ converging to X with $\mathsf{supp}(X_n) \subseteq \mathsf{supp}(X)$ for all $n \in \mathbb{N}$. Moreover we can assume that the truncation of X_{n+1} at depth n is X_n.

Claim. There exist representatives $M_n \in X_n$ such that the sequence $(M_n)_{n\in\mathbb{N}}$ is Cauchy in the metric space (Λ, d).

Moreover we have that $\mathsf{fv}(M_n) \subseteq \mathsf{supp}(X)$ for all $n \in \mathbb{N}$, thus the limit $M \in \Lambda^\infty$ of the sequence $(M_n)_{n\in\mathbb{N}}$ has finitely many free variables. We put $g(X) = [M]_\alpha$. We can show that g is well defined, equivariant and is the inverse of f.

We now sketch the proof of the claim. The idea is to start with an arbitrary sequence of representatives and to inductively rename the bound variables, so that we preserve α-equivalence. In a first step, we construct a sequence of representatives $P_n \in X_n$ by induction on $n \in \mathbb{N}$ such that $\mathsf{bv}(P_n) \cap S_n = \emptyset$ where $S_n = \mathsf{supp}(X) \cup \bigcup_{i=1}^{n-1} \mathsf{var}(P_i)$ and P_n is α-safe using Lemma 20. Notice that S is finite because $P_1, \ldots, P_{n-1}$ are all finite. Notice that this sequence satisfies that the truncation of P_{n+1} at depth n is α-convertible to P_n.

In a second step we define a sequence $(M_n)_{n\in\mathbb{N}}$ of α-safe terms by induction that satisfy the following:

1. $M_n =_\alpha P_n$,
2. the truncation of M_{n+1} at depth n is exactly M_n,
3. $\mathsf{bv}(M_n) \subseteq \bigcup_{i=1}^{n} \mathsf{bv}(P_i)$.

For $n = 0$, set $M_0 = P_0$. For the induction step, we have that:

$$\begin{aligned}(P_{n+1})^n &=_\alpha P_n \quad \text{by assumption}\\ &=_\alpha M_n \text{ by induction hypothesis}\end{aligned}$$

Since $\mathsf{bv}(M_n) \subseteq \bigcup_{i=1}^{n} \mathsf{bv}(P_i)$, we have that $\mathsf{bv}(M_n) \cap \mathsf{bv}(P_{n+1}) = \emptyset$. Since $\mathsf{fv}(P_{n+1})$ and $\mathsf{fv}(M_n)$ are subsets of $\mathsf{supp}(X)$ and every $\mathsf{bv}(P_i)$ is disjoint from $\mathsf{supp}(X)$, we have that $\mathsf{bv}(M_n) \cap \mathsf{fv}(P_{n+1}) = \emptyset$ and $\mathsf{bv}(P_{n+1}) \cap \mathsf{fv}(M_n) = \emptyset$. Therefore we can apply Lemma 21 to M_n and P_{n+1}, hence there exists $M_{n+1} =_\alpha P_{n+1}$ such that the truncation of M_{n+1} at depth n is exactly M_n. We also have that

$$\begin{aligned}\mathsf{bv}(M_{n+1}) &\subseteq \mathsf{bv}(M_n) \cup \mathsf{bv}(P_{n+1}) \text{ by Lemma 21}\\ &\subseteq \textstyle\bigcup_{i=1}^{n+1} \mathsf{bv}(P_i) \qquad \text{by induction hypothesis}\end{aligned}$$

Theorem 23 (Final L_α-coalgebra). *We have that* $((\Lambda^\infty_{\mathsf{ffv}}/{=_\alpha}, \cdot), \mathbf{unfold}_\alpha)$ *is the final coalgebra for the functor* $\mathsf{L}_\alpha : \mathsf{Nom} \to \mathsf{Nom}$ *given by*

$$\mathsf{L}_\alpha\ U = \mathcal{V} + \{\bot\} + [\mathcal{V}]U + U \times U.$$

Proof. The functor L_α is continuous, that is, it preserves limits of ω^{op}-chains. Therefore a final coalgebra for L_α is obtained as the limit of the diagram

$$1 \xleftarrow{!} \mathsf{L}_\alpha(1) \xleftarrow{\mathsf{L}_\alpha(!)} \mathsf{L}^2_\alpha(1) \xleftarrow{\mathsf{L}^2_\alpha(!)} \quad \dots \quad \longleftarrow \lim_{n<\omega} \mathsf{L}^n_\alpha(1) \tag{22}$$

If we think of the singleton 1 as $\{\bot\}$, we can identify elements of $\mathsf{L}^n_\alpha(1)$ with truncations of elements in $\Lambda/{=_\alpha}$ at depth n. Recall how limits are computed in Nom. An element of $\lim_{n<\omega} \mathsf{L}^n_\alpha(1)$ is a *finitely supported* tuple $(X_n)_{n<\omega} \in \prod \mathsf{L}^n_\alpha(1)$ such that for all $n < \omega$ we have

$$\mathsf{L}^{n+1}_\alpha(!)(X_{n+1}) = X_n, \tag{23}$$

or equivalently, that the truncation of X_{n+1} at depth n is X_n.

Then $(X_n)_{n<\omega}$ is a finitely supported Cauchy sequence. By Lemma 17, its limit is in $(\Lambda/{=_\alpha})^\infty_{\mathsf{fs}}$. Conversely, by Remark 18, each $X \in (\Lambda/{=_\alpha})^\infty_{\mathsf{fs}}$ is the limit of a finitely supported Cauchy sequence $(X_n)_{n<\omega} \in \prod \mathsf{L}^n_\alpha(1)$ satisfying (23).

Thus we have an isomorphism $\lim_{n<\omega} \mathsf{L}^n_\alpha(1) \simeq (\Lambda/{=_\alpha})^\infty_{\mathsf{fs}}$, whose equivariance can be easily checked. By Theorem 22 we have

$$\lim_{n<\omega} \mathsf{L}^n_\alpha(1) \simeq \Lambda^\infty_{\mathsf{ffv}}/{=_\alpha}\ .$$

Remark 24 (Alternative proof to Theorem 23). To prove Theorem 23, we could have applied a generalisation of Barr's theorem on the existence of final coalgebras to locally finitely presentable categories given by Adámek [3]. We know that the initial algebra of L_α is the nominal set $\Lambda/{=_\alpha}$. To prove that $(\Lambda/{=_\alpha})^\infty_{\mathsf{fs}}$ is the carrier of the final coalgebra amounts to showing that for all finitely presentable nominal sets B we have that $\mathsf{Nom}(B, (\Lambda/{=_\alpha})^\infty_{\mathsf{fs}})$ is the metric completion of $\mathsf{Nom}(B, \Lambda/{=_\alpha})$.

5 An α-corecursion principle and applications

Using corecursion, we define an α-invariant substitution into infinitary λ-terms and we define α-invariant notions of the infinite normal forms (Böhm, Lévy-Longo and Berarducci) discussed in Section 2. The defined notions will arise from a unique arrow into the final L_α-coalgebra given by the nominal set $\Lambda^\infty_{\mathsf{ffv}}/{=_\alpha}$ of Theorem 23.

Notation 25 (Injections) *The injections for the coproduct $U+V$ are denoted as $\mathbf{inl}^{U,V} : U \to U+V$ and $\mathbf{inr}^{U,V} : V \to U+V$. But for the case of $L(U)$ and $\mathsf{L}_\alpha(U)$, we denote them as*

$$\begin{array}{ll} \mathbf{inbot}^U : \{\bot\} \to L(U) & \mathbf{inbot}^U_\alpha : \{\bot\} \to \mathsf{L}_\alpha(U) \\ \mathbf{invar}^U : \mathcal{V} \to L(U) & \mathbf{invar}^U_\alpha : \mathcal{V} \to \mathsf{L}_\alpha(U) \\ \mathbf{inabs}^U : \mathcal{V} \times U \to L(U) & \mathbf{inabs}^U_\alpha : [\mathcal{V}]U \to \mathsf{L}_\alpha(U) \\ \mathbf{inapp}^U : U \times U \to L(U) & \mathbf{inapp}^U_\alpha : U \times U \to \mathsf{L}_\alpha(U) \end{array}$$

We drop the superscripts when they are clear from the context.

Notation 26 (α-equivalence classes) *We write Λ_α for the (nominal) set $\Lambda/{=_\alpha}$ of finite λ-terms up to α-equivalence and Λ^∞_α for the (nominal) set $\Lambda^\infty_{\mathsf{ffv}}/{=_\alpha}$ of infinitary λ-terms with finitely many free variables up to α-equivalence (see Theorem 23). We continue to denote terms in Λ or $\Lambda^\infty_{\mathsf{ffv}}$ by M, N, but will denote terms in Λ_α or Λ^∞_α by M, N.*

Notation 27 (Elements of Λ^∞_α) *Since $\mathbf{unfold}_\alpha$ is an isomorphism that partitions its domain Λ^∞_α into four disjoint components, see (6), we write typical elements of Λ^∞_α as x, $\bot$, $\mathsf{M_1M_2}$, $\lambda\mathsf{y}.\mathsf{M}$ where*

$$\begin{array}{ll} \mathsf{x} & = \mathbf{unfold}^{-1}_\alpha(\mathbf{invar}_\alpha \mathsf{x}) \\ \bot & = \mathbf{unfold}^{-1}_\alpha(\mathbf{inbot}_\alpha \bot) \\ \lambda\mathsf{y}.\mathsf{M} & = \mathbf{unfold}^{-1}_\alpha(\mathbf{inabs}_\alpha \langle \mathsf{y} \rangle \mathsf{M}) \\ \mathsf{M_1M_2} & = \mathbf{unfold}^{-1}_\alpha(\mathbf{inapp}_\alpha\ (\mathsf{M_1}, \mathsf{M_2})) \end{array} \tag{24}$$

We use x to denote both an element in $\mathcal{V}$ and also its copy in Λ^∞_α.

The following lemma [21, Lemma 2.1] allows parameters in coinductive definitions. It dualises the way in which primitive recursion strengthens induction. In order to express substitution, the set U will be used for the term N in $\mathsf{M}[\mathsf{x} := \mathsf{N}]$ which is not subject to recursion and the set V will be used for the recursion.

Lemma 28. *Let $\delta : D \to F(D)$ be a final coalgebra and $g : U \to F(U)$ an arbitrary F-coalgebra. Then, there is a unique map $f : V \to D$ such that for any $h : V \to F(U+V)$, the following diagram commutes:*

$$\begin{array}{ccc} V & \xrightarrow{\quad f \quad} & D \\ {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle \delta} \\ F(U+V) & \xrightarrow{\quad F([g^*, f]) \quad} & F(D) \end{array}$$

where $g^* : U \to D$ is the unique homomorphism between (U, g) and (D, δ).

We need to fix some notation:

Notation 29 (α-corecursion) *We consider the final L_α-coalgebra $\mathbf{unfold}_\alpha : \Lambda_\alpha^\infty \to \mathsf{L}_\alpha(\Lambda_\alpha^\infty)$.*

1. *Given $g : U \to \mathsf{L}_\alpha(U)$, the unique map to the final L_α-coalgebra $\mathbf{unfold}_\alpha : \Lambda^\infty \to \mathsf{L}_\alpha(\Lambda^\infty)$ is denoted by $\mathsf{corec}_\alpha(g)$.*
2. *Given $g : U \to \mathsf{L}_\alpha(U)$ and $h : V \to \mathsf{L}_\alpha(U + V)$, the unique map from h to $\mathbf{unfold}_\alpha : \Lambda^\infty \to \mathsf{L}_\alpha(\Lambda^\infty)$ given by Lemma 28 is denoted by $\mathbf{corec}_\alpha(g, h)$.*

We now define substitution using the finality of $\mathbf{unfold}_\alpha : \Lambda_\alpha^\infty \to \mathsf{L}_\alpha(\Lambda_\alpha^\infty)$.

Definition 30 (Substitution on α-equivalence classes). *We define substitution $\mathbf{subs}_\alpha : \Lambda_\alpha^\infty \times \mathcal{V} \times \Lambda_\alpha^\infty \to \Lambda_\alpha^\infty$ as*

$$\mathbf{subs}_\alpha = \mathbf{corec}_\alpha(\mathbf{unfold}_\alpha, \mathbf{h}_{\mathbf{subs}_\alpha}),$$

where $\mathbf{subs}_\alpha$ arises from the instance of Lemma 28, see the diagram below,

$$\begin{array}{ccc} \Lambda_\alpha^\infty \times \mathcal{V} \times \Lambda_\alpha^\infty & \xrightarrow{\quad\mathbf{subs}_\alpha\quad} & \Lambda_\alpha^\infty \\ \Big\downarrow{\scriptstyle \mathbf{h}_{\mathbf{subs}_\alpha}} & & \Big\downarrow{\scriptstyle \mathbf{unfold}_\alpha} \\ \mathsf{L}_\alpha(\Lambda_\alpha^\infty + \Lambda_\alpha^\infty \times \mathcal{V} \times \Lambda_\alpha^\infty) & \xrightarrow{\mathsf{L}_\alpha([\mathbf{unfold}_\alpha^*, \mathbf{subs}_\alpha])} & \mathsf{L}_\alpha(\Lambda_\alpha^\infty) \end{array} \tag{25}$$

where $\mathbf{h}_{\mathbf{subs}_\alpha} : \Lambda_\alpha^\infty \times \mathcal{V} \times \Lambda_\alpha^\infty \to \mathsf{L}_\alpha(\Lambda_\alpha^\infty + \Lambda_\alpha^\infty \times \mathcal{V} \times \Lambda_\alpha^\infty)$ is given by

$$\begin{array}{lll} \mathbf{h}_{\mathbf{subs}_\alpha}(\mathsf{x}, \mathsf{x}, \mathsf{N}) & = \mathsf{L}_\alpha\mathbf{inl}(\mathbf{unfold}_\alpha(\mathsf{N})) & \\ \mathbf{h}_{\mathbf{subs}_\alpha}(\mathsf{x}, \mathsf{y}, \mathsf{N}) & = \mathbf{invar}_\alpha\ \mathsf{x} & \text{if } \mathsf{y} \neq \mathsf{x} \\ \mathbf{h}_{\mathbf{subs}_\alpha}(\bot, \mathsf{y}, \mathsf{N}) & = \mathbf{inbot}_\alpha\ \bot & \\ \mathbf{h}_{\mathbf{subs}_\alpha}(\mathsf{M_1M_2}, \mathsf{x}, \mathsf{N}) & = \mathbf{inapp}_\alpha\ (\mathbf{inr}\ (\mathsf{M}_1, \mathsf{x}, \mathsf{N}), \mathbf{inr}\ (\mathsf{M}_2, \mathsf{x}, \mathsf{N})) & \\ \mathbf{h}_{\mathbf{subs}_\alpha}(\lambda\mathsf{y}.\mathsf{M}, \mathsf{x}, \mathsf{N}) & = \mathbf{inabs}_\alpha\ \langle\mathsf{z}\rangle(\mathbf{inr}\ ((\langle\mathsf{y}\rangle\mathsf{M})@\mathsf{z}, \mathsf{x}, \mathsf{N})) & \text{if } \mathsf{z}\#(\lambda\mathsf{y}.\mathsf{M}, \mathsf{x}, \mathsf{N}). \end{array}$$

The map $\mathbf{h}_{\mathbf{subs}_\alpha}$ is well-defined since the last clause is independent of the choice of z (which always exists due to all terms being finitely supported) and $\mathbf{h}_{\mathbf{subs}_\alpha}$ is equivariant since all operations involved, in particular abstraction and concretion, are equivariant.

Remark 31. To see how Definition 30 captures (3), note that, by (25), we have that $\mathbf{subs}_\alpha = \mathbf{unfold}_\alpha^{-1} \circ \mathsf{L}_\alpha([\mathbf{unfold}_\alpha^*, \mathbf{subs}_\alpha]) \circ \mathbf{h}_{\mathbf{subs}_\alpha}$, that is, using (24) and the fact that one can choose $\mathsf{z} = \mathsf{y}$ in the last clause of $\mathbf{h}_{\mathbf{subs}_\alpha}$,

$$\begin{array}{lll} \mathbf{subs}_\alpha(\mathsf{x}, \mathsf{x}, \mathsf{N}) & = \mathsf{N} & \\ \mathbf{subs}_\alpha(\mathsf{x}, \mathsf{y}, \mathsf{N}) & = \mathsf{x} & \text{if } \mathsf{y} \neq \mathsf{x} \\ \mathbf{subs}_\alpha(\bot, \mathsf{y}, \mathsf{N}) & = \bot & \\ \mathbf{subs}_\alpha(\mathsf{M_1M_2}, \mathsf{x}, \mathsf{N}) & = \mathbf{subs}_\alpha(\mathsf{M}_1, \mathsf{x}, \mathsf{N})\mathbf{subs}_\alpha(\mathsf{M}_2, \mathsf{x}, \mathsf{N}) & \\ \mathbf{subs}_\alpha(\lambda\mathsf{y}.\mathsf{M}, \mathsf{x}, \mathsf{N}) & = \lambda\mathsf{y}.\mathbf{subs}_\alpha(\mathsf{M}, \mathsf{x}, \mathsf{N}) & \text{if } \mathsf{y}\#(\mathsf{x}, \mathsf{N}) \end{array}$$

which looks indeed just like a notational variant of the Set-based (3), but is now fully justified as a coinductive definition on α-equivalence classes of λ-terms.

Definition 32 (β-reduction on α-equivalence classes). *We define β_α-reduction as the smallest relation on $\Lambda_\alpha^\infty \times \Lambda_\alpha^\infty$ that satisfies*

$$\frac{}{(\lambda \mathsf{x}.\mathsf{P})\mathsf{Q} \to_{\beta_\alpha} \mathbf{subs}_\alpha(\mathsf{P},\mathsf{x},\mathsf{N})}\,(\beta_\alpha) \qquad \frac{\mathsf{P} \to_{\beta_\alpha} \mathsf{P}'}{\lambda \mathsf{x}.\mathsf{P} \to_{\beta_\alpha} \lambda \mathsf{x}.\mathsf{P}'}\,(abs)$$

$$\frac{\mathsf{P} \to_{\beta_\alpha} \mathsf{P}'}{\mathsf{PQ} \to_{\beta_\alpha} \mathsf{P}'\mathsf{Q}}\,(app_L) \qquad \frac{\mathsf{Q} \to_{\beta_\alpha} \mathsf{Q}'}{\mathsf{PQ} \to_{\beta_\alpha} \mathsf{PQ}'}\,(app_R)$$

The informal definitions of Böhm, Lévy-Longo and Berarducci trees given in (11), (12) and (13) use the notion of β-reduction relation which is not a function. To formally define the notions of trees, we need to define evaluation strategies, i.e. some restricted notions of β-reduction which are actually partial functions.

We first define the notion of β-head reduction which contracts only the redex at the head position and corresponds to the normalising leftmost strategy. This reduction is used to define Böhm trees.

Definition 33 (Head β-reduction on α-equivalence classes). *We define $\beta_{\mathsf{h}_\alpha}$-reduction as the smallest relation on $\Lambda_\alpha^\infty \times \Lambda_\alpha^\infty$ closed under*

$$\frac{}{(\lambda \mathsf{x}.\mathsf{P})\mathsf{Q} \to_{\beta_{\mathsf{h}_\alpha}} \mathbf{subs}_\alpha(\mathsf{P},\mathsf{x},\mathsf{Q})}\,(\beta_{\mathsf{h}_\alpha})$$

$$\frac{\mathsf{P} \to_{\beta_{\mathsf{h}_\alpha}} \mathsf{P}' \quad \textit{P is not an abstraction}}{\mathsf{PQ} \to_{\beta_{\mathsf{h}_\alpha}} \mathsf{P}'\mathsf{Q}}\,(app_L) \qquad \frac{\mathsf{P} \to_{\beta_{\mathsf{h}_\alpha}} \mathsf{P}'}{\lambda \mathsf{x}.\mathsf{P} \to_{\beta_{\mathsf{h}_\alpha}} \lambda \mathsf{x}.\mathsf{P}'}\,(abs)$$

A term M is in head normal form (hnf) if it is of the form $\lambda \mathsf{x}_1 \ldots \mathsf{x}_\mathsf{n}.\mathsf{y}\mathsf{N}_1 \ldots \mathsf{N}_\mathsf{m}$.

We restrict the β-head reduction by not contracting β-redexes in the body of an abstraction and obtain the weak head β-reduction which is needed to define the notion of Lévy-Longo tree.

Definition 34 (Weak head β-reduction on α-equivalence classes). *We define $\beta_{\mathsf{wh}_\alpha}$-reduction as the smallest relation on $\Lambda_\alpha^\infty \times \Lambda_\alpha^\infty$ closed under*

$$(\lambda \mathsf{x}.\mathsf{P})\mathsf{Q} \to_{\beta_{\mathsf{wh}_\alpha}} \mathbf{subs}_\alpha(\mathsf{P},\mathsf{x},\mathsf{Q})\ (\beta_{\mathsf{wh}_\alpha}) \qquad \frac{\mathsf{P} \to_{\beta_{\mathsf{h}_\alpha}} \mathsf{P}'}{\mathsf{PQ} \to_{\beta_{\mathsf{wh}_\alpha}} \mathsf{P}'\mathsf{Q}}\,(app_L)$$

A term M is in weak head normal form (whnf) if it is either a head normal form or an abstraction.

The reflexive, transitive closure of $\to_{\beta_\alpha}$, $\to_{\beta_{\mathsf{h}_\alpha}}$ and $\to_{\beta_{\mathsf{wh}_\alpha}}$ are denoted by $\twoheadrightarrow_{\beta_\alpha}$, $\twoheadrightarrow_{\beta_{\mathsf{h}_\alpha}}$ and $\twoheadrightarrow_{\beta_{\mathsf{wh}_\alpha}}$, respectively.

We now define the notion of top β-reduction which only contracts β-weak head redexes at depth 0 and it will be used to define Berarducci trees.

Definition 35 (Top β-reduction on α-equivalence classes). *We define $\beta_{\mathsf{t}_\alpha}$-reduction as the smallest relation on $\Lambda_\alpha^\infty \times \Lambda_\alpha^\infty$ closed under*

$$\frac{\mathsf{M} \twoheadrightarrow_{\beta_{\mathsf{wh}_\alpha}} (\lambda \mathsf{x}.\mathsf{P})}{\mathsf{MQ} \to_{\beta_{\mathsf{t}_\alpha}} \mathbf{subs}_\alpha(\mathsf{P}, \mathsf{x}, \mathsf{Q})} \, (\beta_{\mathsf{t}_\alpha})$$

A term M is a top normal form (tnf) if it is either a weak head normal form or an application of the form NP where N cannot reduce to an abstraction.

The reflexive and transitive closure of $\to_{\beta_\mathsf{t}}$ is denoted by $\twoheadrightarrow_{\beta_\mathsf{t}}$.

We now define the notion of Böhm tree, Lévy-Longo tree, and Berarducci tree using the finality of $\mathbf{unfold}_\alpha : \Lambda_\alpha^\infty \to \mathsf{L}_\alpha(\Lambda_\alpha^\infty)$.

Definition 36 (Böhm tree on α-equivalence classes). *We define the Böhm tree of M as $\mathbf{BT}_\alpha(\mathsf{M})$ where $\mathbf{BT}_\alpha = \mathsf{corec}_\alpha(\mathbf{g}_{\mathbf{BT}_\alpha})$ is the unique map such that*

$$\begin{array}{ccc} \Lambda_\alpha^\infty & \xrightarrow{\quad\mathbf{BT}_\alpha\quad} & \Lambda_\alpha^\infty \\ {\scriptstyle \mathbf{g}_{\mathbf{BT}_\alpha}}\downarrow & & \downarrow{\scriptstyle \mathbf{unfold}_\alpha} \\ \mathsf{L}_\alpha(\Lambda_\alpha^\infty) & \xrightarrow[\mathsf{L}_\alpha(\mathbf{BT}_\alpha)]{} & \mathsf{L}_\alpha(\Lambda_\alpha^\infty) \end{array}$$

commutes, with $\mathbf{g}_{\mathbf{BT}_\alpha} : \Lambda_\alpha^\infty \to L(\Lambda_\alpha^\infty)$ being defined as

$$\mathbf{g}_{\mathbf{BT}_\alpha}(\mathsf{M}) = \begin{cases} \mathbf{unfold}_\alpha \mathsf{N} & \text{if } \mathsf{M} \twoheadrightarrow_{\beta_{\mathsf{h}_\alpha}} \mathsf{N} \text{ and } \mathsf{N} \text{ is in hnf} \\ \mathbf{inbot}_\alpha \, \bot & \text{otherwise} \end{cases}$$

Definition 37 (Lévy-Longo tree on α-equivalence classes). *We define the Lévy-Longo tree of M as $\mathbf{LLT}_\alpha(\mathsf{M})$ where $\mathbf{LLT}_\alpha = \mathsf{corec}_\alpha(\mathbf{g}_{\mathbf{LLT}_\alpha})$ is the unique map such that*

$$\begin{array}{ccc} \Lambda_\alpha^\infty & \xrightarrow{\quad\mathbf{LLT}_\alpha\quad} & \Lambda_\alpha^\infty \\ {\scriptstyle \mathbf{g}_{\mathbf{LLT}_\alpha}}\downarrow & & \downarrow{\scriptstyle \mathbf{unfold}_\alpha} \\ \mathsf{L}_\alpha(\Lambda_\alpha^\infty) & \xrightarrow[\mathsf{L}_\alpha(\mathbf{LLT}_\alpha)]{} & \mathsf{L}_\alpha(\Lambda_\alpha^\infty) \end{array}$$

commutes, with $\mathbf{g}_{\mathbf{LLT}_\alpha} : \Lambda_\alpha^\infty \to \mathsf{L}_\alpha(\Lambda_\alpha^\infty)$ being defined as

$$\mathbf{g}_{\mathbf{LLT}_\alpha}(\mathsf{M}) = \begin{cases} \mathbf{unfold}_\alpha \mathsf{N} & \text{if } \mathsf{M} \twoheadrightarrow_{\beta_{\mathsf{wh}_\alpha}} \mathsf{N} \text{ and } \mathsf{N} \text{ is in whnf} \\ \mathbf{inbot}_\alpha \, \bot & \text{otherwise} \end{cases}$$

Definition 38 (Berarducci tree on α-equivalence classes). *We define the Berarducci tree of M as $\mathbf{BerT}_\alpha(\mathsf{M})$ where $\mathbf{BerT}_\alpha = \mathsf{corec}_\alpha(\mathbf{g}_{\mathbf{BerT}_\alpha})$ is the unique map such that*

$$\begin{array}{ccc} \Lambda_\alpha^\infty & \xrightarrow{\quad\mathbf{BerT}_\alpha\quad} & \Lambda_\alpha^\infty \\ {\scriptstyle \mathbf{g}_{\mathbf{BerT}_\alpha}}\downarrow & & \downarrow{\scriptstyle \mathbf{unfold}_\alpha} \\ \mathsf{L}_\alpha(\Lambda_\alpha^\infty) & \xrightarrow[\mathsf{L}_\alpha(\mathbf{BerT}_\alpha)]{} & \mathsf{L}_\alpha(\Lambda_\alpha^\infty) \end{array}$$

commutes, with $\mathbf{g}_{\mathbf{BerT}_\alpha} : \Lambda_\alpha^\infty \to \mathsf{L}_\alpha(\Lambda_\alpha^\infty)$ *being defined as follows.*

$$\mathbf{g}_{\mathbf{BerT}_\alpha}(\mathsf{M}) = \begin{cases} \mathbf{unfold}_\alpha \mathsf{N} & \text{if } \mathsf{M} \twoheadrightarrow_{\beta_{\mathsf{t}_\alpha}} \mathsf{N} \text{ and } \mathsf{N} \text{ is in tnf} \\ \mathbf{inbot}_\alpha \perp & \text{otherwise} \end{cases}$$

6 Comparison of α-recursion and corecursion

The α-recursion principle of [23] is based on the initial L_α-algebra Λ_α (the principle applies of course to general functors, but we only consider the example L_α here). Thus, in order to make an inductive definition, that is, in order to define an arrow from Λ_α into some set U, one needs, in particular, to give a map $[\mathcal{V}]U \to U$. This is done in [23] by giving a map $f : \mathcal{V} \times U \to U$ which has to satisfy a certain condition, the freshness condition for binders (FCB). FCB makes sure that f factors (uniquely) through $[\mathcal{V}]U$, thus giving the required $[\mathcal{V}]U \to U$.

In the coinductive case, we need instead give a map $U \to [\mathcal{V}]U$. Note that not every equivariant $U \to [\mathcal{V}]U$ factors through an equivariant $U \to \mathcal{V} \times U$. So we could say that a not necessarily equivariant $U \to \mathcal{V} \times U$ *satisfies the dual of FCB* if the composition $U \to \mathcal{V} \times U \to [\mathcal{V}]U$ is equivariant. In our examples, 'the dual of FCB' is automatic because of the equivariance of the ingredients used, see e.g. the definition of $\mathbf{h}_{\mathbf{subs}_\alpha}$ in Definition 30.

Next we compare α-corecursion with corecursion. Duppen [9] attempts to obtain a corecursive definition of substitution, as in (3), using the unique arrow into the final coalgebra for an endofunctor on the category Set of sets and functions. However, this approach does not take into account α-conversion: (3) defines a *partial* map on λ-terms which induces a total map only on 'λ-terms up to α-equivalence' (as can be seen, for example, by using Barendregt's variable convention [5] and comparing his definition of substitution [5, 2.1.15] with (3)).

For a general introduction to the problems arising in Set-based as opposed to Nom-based reasoning with binders, see [23, Section 1] and [22, Section 2]. Typically, Set-based induction in the presence of binders is complicated by side conditions as in (3) which lead to partial functions with the domain of definition restricted to 'safe terms' having no conflicts between free and bound variables. But 'safe terms' are not closed under the operations of abstraction and binding and so do not carry an algebra structure that can serve as a codomain U to define an arrow $\Lambda \to U$ by initiality.

In the coinductive case, however, we can equip the safe terms with a coalgebra structure. The details are as follows. In analogy to Notation 29, we consider the final L-coalgebra $\mathbf{unfold} : \Lambda^\infty \to L(\Lambda^\infty)$ and, given $g : U \to L(U)$ and $h : V \to L(U + V)$, we denote the unique map from $h : V \to L(U + V)$ to $\mathbf{unfold} : \Lambda^\infty \to L(\Lambda^\infty)$ by $\mathbf{corec}(g, h)$. The set of *safe terms* (with respect to substitution) is defined as $(\Lambda^\infty \times \mathcal{V} \times \Lambda^\infty)_{\mathsf{safe}} = \{(M, x, N) \in \Lambda^\infty \times \mathcal{V} \times \Lambda^\infty \mid \mathsf{bv}(M) \cap \mathsf{fv}(N) = \emptyset\}$. In analogy with Definition 30, we then define substitution

$$\mathbf{subs} : (\Lambda^\infty \times \mathcal{V} \times \Lambda^\infty)_{\mathsf{safe}} \to \Lambda^\infty$$

as $\mathbf{corec}(\mathbf{unfold}, \mathbf{h_{subs}})$, where $\mathbf{h_{subs}} : (\Lambda^\infty \times \mathcal{V} \times \Lambda^\infty)_{\mathsf{safe}} \to L(\Lambda^\infty + (\Lambda^\infty \times \mathcal{V} \times \Lambda^\infty)_{\mathsf{safe}})$ is as in Definition 30 but with

$$\mathbf{h_{subs}}((\lambda x.M), x, N) = \mathbf{inabs}\ (x, \mathbf{inl}\ (M, x, N))$$
$$\mathbf{h_{subs}}((\lambda y.M), x, N) = \mathbf{inabs}\ (y, \mathbf{inr}\ (M, x, N)) \text{ if } y \neq x$$

where we have that $y \notin \mathsf{fv}(N)$ because $\mathsf{bv}(\lambda y.M) \cap \mathsf{fv}(N) = \emptyset$. Although **subs** is well-defined on safe tuples (M, x, N), it is not immediately clear how to extend the definition of substitution to tuples of the form $([M]_\alpha, x, [N]_\alpha)$. As illustrated by the term $(\lambda x_0 x_1.x_0 x_1)$**allfv** in the introduction, finding "safe" representatives for α-equivalence classes of arbitrary infinitary terms is problematic.

Further, the β-rule could be defined on the set $(\Lambda^\infty)_{\mathsf{safe}} = \{M \in \Lambda^\infty_{\mathsf{ffv}} \mid \mathsf{fv}(M) \cap \mathsf{bv}(M) = \emptyset\}$ using **subs**. Since $(\Lambda^\infty)_{\mathsf{safe}}$ is closed under $\twoheadrightarrow_{\beta_{\mathsf{wh}}}$, in most lazy functional programming languages which use β_{wh}-reduction such as Haskell we could work with representatives without having to consider equivalence classes. In spite of this, to define the notion of Lévy-Longo tree, we also need to extend the β_{wh}-reduction to Λ^∞_α because the function that computes the β_{wh}-normal form is not an L-coalgebra since P may not belong to $(\Lambda^\infty)_{\mathsf{safe}}$ when $\lambda x.P \in (\Lambda^\infty)_{\mathsf{safe}}$.

To summarize, it pays off to work in Nom from the beginning, because all constructions are automatically invariant under α-conversion.

7 Conclusion

As far as we know this is the first paper applying nominal techniques to the infinitary λ-calculus. In particular, we have derived a principle for making coinductive definitions up to α-equivalence and applied it to fully formalising some important concepts from infinitary λ-calculus. In the context of presheaf models substitution into non-well founded syntax such as the infinitary lambda calculus has been investigated in [20].

Let us emphasise that Theorem 23 extends to other signatures. Not only is the presence of $\bot$ irrelevant for Theorem 23, it also applies if we replace L_α by other polynomial functors, for example by those that fall into the nominal (universal) algebra of [13]. We are confident that the future will uncover more interesting examples of α-corecursion.

In future work we will study α-corecursion for λ-terms that may contain infinitely many free variables, for example using the generalised theory of names [12] or the ideal-supported subsets of [8].

It will also be interesting to explore whether nominal rewriting [10] has applications in the infinitary λ-calculus.

References

1. S. Abramsky. The lazy lambda calculus. In *Research Topics in Functional Programming*, pages 65–116. Addison-Wesley, 1990.

2. S. Abramsky and C.-H. Luke Ong. Full abstraction in the lazy lambda calculus. *Inform. and Comput.*, 105(2):159–267, 1993.
3. J. Adámek. On final coalgebras of continuous functors. *Theor. Comput. Sci.*, 294(1/2):3–29, 2003.
4. A. Arnold and M. Nivat. The metric space of infinite trees. algebraic and topological properties. *Fundamenta Informaticae*, 4:445–476, 1980.
5. H. P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. North-Holland, Amsterdam, Revised edition, 1984.
6. M. Barr. Terminal coalgebras for endofunctors on sets. *Theor. Comp. Sci.*, 114(2):299–315, 1999.
7. A. Berarducci. Infinite λ-calculus and non-sensible models. In *Logic and algebra (Pontignano, 1994)*, pages 339–377. Dekker, New York, 1996.
8. J. Cheney. Completeness and Herbrand theorems for nominal logic. *J. Symb. Log.*, 71(1):299–320, 2006.
9. Y. D. Duppen. A coalgebraic approach to lambda calculus. Master's thesis, Vrije Universiteit Amsterdam, 2000.
10. M. Fernández and M. Gabbay. Nominal rewriting. *Inf. Comput.*, 205(6):917–965, 2007.
11. M. Gabbay and A. M. Pitts. A new approach to abstract syntax involving binders. In *LICS*, pages 214–224, 1999.
12. M. J. Gabbay. A general mathematics of names. *Inf. Comput.*, 205(7):982–1011, 2007.
13. M. J. Gabbay and A. Mathijssen. Nominal (universal) algebra: Equational logic with names and binding. *J. Log. Comput.*, 19(6):1455–1508, 2009.
14. M. J. Gabbay and A. M. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13(3–5):341–363, 2001.
15. J. R. Kennaway and F. J. de Vries. Infinitary rewriting. In Terese, editor, *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theor. Comp. Sci.*, pages 668–711. Cambridge University Press, 2003.
16. J. R. Kennaway, J. W. Klop, M. R. Sleep, and F. J. de Vries. Infinite lambda calculus and Böhm models. In *RTA*, volume 914 of *LNCS*, pages 257–270. Springer, 1995.
17. J. R. Kennaway, J. W. Klop, M. R. Sleep, and F. J. de Vries. Infinitary lambda calculus. *Theor. Comp. Sci.*, 175(1):93–125, 1997.
18. J.-J. Lévy. An algebraic interpretation of the $\lambda\beta K$-calculus, and an application of a labelled λ-calculus. *Theor. Comp. Sci.*, 2(1):97–114, 1976.
19. G. Longo. Set-theoretical models of λ-calculus: theories, expansions, isomorphisms. *Ann. Pure Appl. Logic*, 24(2):153–188, 1983.
20. R. Matthes and T. Uustalu. Substitution in non-wellfounded syntax with variable binding. *Theor. Comput. Sci*, 327:155–174, 2004.
21. L. S. Moss. Parametric corecursion. *Theor. Comp. Sci.*, 260:139–163, 2001.
22. A. M. Pitts. Nominal logic, a first order theory of names and binding. *Information and Computation*, 186:165–193, 2003.
23. A. M. Pitts. Alpha-structural recursion and induction. In Joe Hurd and Thomas F. Melham, editors, *TPHOLs*, volume 3603 of *LNCS*, pages 17–34. Springer, 2005.
24. A. Salibra. Nonmodularity results for lambda calculus. *Fundamenta Informaticae*, 45:379–392, 2001.
25. P. G. Severi and F. J. de Vries. Weakening the axiom of overlap in the infinitary lambda calculus. In *RTA*, volume 10 of *LIPIcs*, pages 313–328. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2011.