2013

Nominal Coalgebraic Data Types with Applications to Lambda Calculus

Alexander Kurz
Chapman University, akurz@chapman.edu

Daniela Petrişan
University of Leicester

Paula Severi
University of Leicester

Fer-Jan de Vries
University of Leicester

Recommended Citation

A. Kurz, D. Petrisan, P. Severi, and F.-J. de Vries, "Nominal Coalgebraic Data Types with Applications to Lambda Calculus," Logical Methods in Computer Science, vol. 9, no. 4, Dec. 2013. DOI: 10.2168/LMCS-9(4:20)2013

Nominal Coalgebraic Data Types with Applications to Lambda Calculus

Comments

This article was originally published in *Logical Methods in Computer Science*, volume 9, issue 4, in 2013. DOI: 10.2168/LMCS-9(4:20)2013

Logical Methods in Computer Science
Vol. 9(4:20)2013, pp. 1–51
www.lmcs-online.org

Submitted Feb. 26, 2013
Published Dec. 11, 2013

NOMINAL COALGEBRAIC DATA TYPES WITH APPLICATIONS TO LAMBDA CALCULUS

ALEXANDER KURZ, DANIELA PETRIŞAN, PAULA SEVERI, AND FER-JAN DE VRIES

Department of Computer Science, University of Leicester, Leicester, UK
e-mail address: {ak155, ps65, fdv1}@mcs.le.ac.uk, daniela.petrisan@gmail.com

Abstract. We investigate final coalgebras in nominal sets. This allows us to define types of infinite data with binding for which all constructions automatically respect alpha equivalence. We give applications to the infinitary lambda calculus.

Contents

2012 ACM CCS: [**Theory of computation**]: Semantics and reasoning—Program semantics—Algebraic semantics.

Key words and phrases: coinduction, corecursion, nominal sets, infinitary lambda calculus.

LOGICAL METHODS
IN COMPUTER SCIENCE

DOI:10.2168/LMCS-9(4:20)2013

1. Introduction

We investigate types of infinite data with binding. A leading example is the infinitary λ-calculus. To construct explicitly a domain of infinitary λ-terms, one usually starts from finite λ-terms and then applies two constructions: Metric completion to obtain infinite terms and quotienting up to α-equivalence. Although each of these constructions appear to be routine, we show that their combination is more subtle than one might think at first. For example, one either needs to assume uncountably many variables or find a solution to the problem that, in the case of countably many variables, metric completion does not commute with quotienting by α-equivalence. On the other hand, general principles suggest to construct infinitary λ-terms more abstractly as final coalgebras in the category of nominal sets. It allows us to treat infinitary data types with binding in general and provides a principle of definition and proof by coinduction. We also show that the syntactic approach of completing and quotienting agrees with the semantic approach of final coalgebras in nominal sets.

To summarise, the paper contributes to coalgebra, to nominal sets, and to the infinitary λ-calculus. To coalgebra, by showing that as for coalgebras over sets also over nominal sets we obtain from finality a definition and proof principle of coinduction. To nominal sets, by investigating limits and introducing notions of safe maps and bound variables. To the infinitary λ-calculus, by clarifying the fundamental constructions in the case of countably many variables and by showing that the informal reasoning with α-equivalence classes of infinitary λ-terms is indeed mathematically precise.

In the remainder of the introduction we outline the contents of the paper.

Nominal sets were introduced in [GP99], but see also [Hof99, FPT99] for related proposals. Roughly speaking, a nominal set is a set X equipped with an action of a set of permutations on some countably infinite set

$$\mathcal{V}$$

of 'names' or 'variables'. One can then define the support of some $x \in X$ as the smallest set of variables on which x depends (we review the precise definitions in Section 4), thus giving an abstract account of the 'free variables' in x. It is characteristic of nominal sets that all elements have finite support. In other words, modelling syntax in nominal sets requires us to only consider terms with finitely many free variables. But, and this is one of the themes of this paper, it is possible to have terms with infinitely many bound variables. (The question what may constitute an abstract account of bound variables in nominal sets will be discussed in Section 5.6.)

Variable binding in nominal sets can be described by a type constructor $X \mapsto [\mathcal{V}]X$ which can be understood as a quotient

$$\mathcal{V} \times X \to [\mathcal{V}]X$$

identifying elements (v, x) up to α-equivalence, that is, up to renaming of the 'bound' variable v. For example, whereas λ-terms are given by the initial algebra of the functor

$$L\ X = \mathcal{V} + \mathcal{V} \times X + X \times X \tag{1.1}$$

it was shown in [GP99] that λ-terms up to α-equivalence are given by the initial algebra of the functor

$$\mathsf{L}_\alpha\ X = \mathcal{V} + [\mathcal{V}]X + X \times X \tag{1.2}$$

Alpha-structural recursion [Pit05, Pit06, Pit11] is the induction principle that ensues from syntax as an initial algebra in the category Nom of nominal sets. For example, the classic definition of substitution in the λ-calculus [Bar84]

$$\begin{aligned} y[x:=N] &= \begin{cases} N & \text{if } y = x, \\ y & \text{otherwise,} \end{cases} \\ (PQ)[x:=N] &= (P[x:=N]Q[x:=N]), \\ (\lambda y.P)[x:=N] &= \lambda y.(P[x:=N]) \qquad\qquad \text{if } y \notin \mathsf{fv}(N) \cup \{x\}. \end{aligned} \tag{1.3}$$

is not an inductive definition in the usual sense. Because of the side condition, substitution is only a partial function on raw terms. But, as explained in detail in [Pit05], (1.3) is an inductive definition according to (1.2). Moreover, [Pit05] establishes a general induction principle for inductively defined data types with variable binding, explaining when partially defined functions in Set give rise to totally defined functions in Nom.

Alpha-structural corecursion introduced in this paper is the analogue of α-structural recursion for coinductive datatypes. For example, in the study of the infinitary λ-calculus [KKSdV95, KKSdV97, KdV03], which we review in Section 3, one is interested in the final coalgebra of L_α. We describe the corecursion principle ensuing from final coalgebras in nominal sets and show that (1.3) is indeed a coinductive definition of substitution for infinitary λ-terms.

Infinitely many free variables in a term (Section 5.4), which do appear if we take the final coalgebra of L in sets, pose a problem. To see this, note that (1.3) becomes an inductive definition by choosing a suitable representative $\lambda y.P$ such that $y \notin \mathsf{fv}(N) \cup \{x\}$. This approach is not immediately viable for the infinitary λ-calculus, because we may have terms that exhaust all the available variables, so that we cannot find a fresh y. For example, consider the infinite λ-term $\mathsf{allfv} = x_0(x_1(x_2(\ldots)))$ which contains all variables from $\mathcal{V}$. In the following β-step

$$(\lambda x_0 x_1.x_0 x_1)\mathsf{allfv} \to_\beta (\lambda x_1.x_0 x_1)[x_0 := \mathsf{allfv}] \tag{1.4}$$

we have that $x_1 \in \mathsf{fv}(\mathsf{allfv})$ and, therefore, the x_1 in $\lambda x_1.x_0x_1$ should be replaced by some fresh variable, which is impossible because allfv contains all of them [Sal01].

Restricting to finitely many free variables, as opposed to allowing $\mathcal{V}$ to be uncountable, is the solution adopted in this paper (but we will come back to infinitely many free variable in Section 6.5). That is, in our example, we will consider the set

$$\Lambda^\infty_{\mathsf{ffv}} = \{M \in \Lambda^\infty \mid \mathsf{fv}(M) \text{ is finite }\} \tag{1.5}$$

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. On the other hand, restricting to finitely many free variables has the advantage of allowing us to work with nominal sets.

Infinitely many bound variables must be allowed, since additional fresh variables may be needed at each β-reduction step to avoid capture. For example, consider the finite term

$\mathsf{Pinfbv} \equiv \mathsf{fix}(\lambda fxy.xy(f(xy)))$ which has the following reduction sequence:

$$\begin{aligned}
\mathsf{Pinfbv} \quad & \twoheadrightarrow_\beta \lambda xy.xy(\mathsf{Pinfbv}(xy)) \\
& =_\alpha \lambda x_0x_1.x_0x_1(\mathsf{Pinfbv}(x_0x_1)) \\
& \twoheadrightarrow_\beta \lambda x_0x_1.x_0x_1(\lambda y.x_0x_1y(\mathsf{Pinfbv}(x_0x_1y))) \\
& =_\alpha \lambda x_0x_1.x_0x_1(\lambda x_2.x_0x_1x_2(\mathsf{Pinfbv}(x_0x_1x_2))) \\
& \twoheadrightarrow_\beta \lambda x_0x_1.x_0x_1(\lambda x_2.x_0x_1x_2(\lambda y.x_0x_1x_2y(\mathsf{Pinfbv}(x_0x_1x_2y)))) \\
& =_\alpha \lambda x_0x_1.x_0x_1(\lambda x_2.x_0x_1x_2(\lambda x_3.x_0x_1x_2x_3(\mathsf{Pinfbv}(x_0x_1x_2x_3)))) \\
& \vdots
\end{aligned}$$

The limit of the above sequence is the infinite term:

$$\mathsf{infbv} \equiv \lambda x_0.\lambda x_1.x_0x_1(\lambda x_2.x_0x_1x_2(\lambda x_3.x_0x_1x_2x_3(\ldots))).$$

The term infbv has an infinite number of bound variables. All the terms in its α-equivalence class have an infinite number of bound variables.

The different classes of λ-terms arising from the discussion above are summarised in the following picture, which is one of the contributions of our work. Previous work on infinitary λ-calculus either assumed uncountably many variables or did not make the careful distinctions discussed below.

$$\begin{tikzcd}
\Lambda \arrow[r, tail] \arrow[dd, two heads] & \Lambda^\infty \arrow[d, two heads] & \Lambda^\infty_{\mathsf{ffv}} \arrow[l, hook'] \arrow[d, two heads] \\
 & \Lambda^\infty/{=_\alpha} \arrow[d, tail] \arrow[r, phantom, "(*)"] & \Lambda^\infty_{\mathsf{ffv}}/{=_\alpha} \arrow[d, "\cong"] \\
\Lambda/{=_\alpha} \arrow[r, tail] & (\Lambda/{=_\alpha})^\infty & (\Lambda/{=_\alpha})^\infty_{\mathsf{fs}} \arrow[l, hook']
\end{tikzcd} \tag{1.6}$$

In the diagram, Λ denotes the set of finite λ-terms. Vertical arrows $\twoheadrightarrow$ denote quotienting by α-equivalence. Infinitary λ-terms are constructed by metric completions $\rightarrowtail$. The rightmost column arises from maps $\hookleftarrow$ that restrict to terms with finitely many free variables. Going first right and then down in the diagram means to first complete to infinitary terms and then to quotient by α-equivalence, whereas going first down and then right, means to first quotient and then to complete. If both ways of constructing infinitary terms up-to α-equivalence coincide, then we say that *metric completion commutes with quotienting by α-equivalence.* The two main results here are the following.

- The vertical map in the middle column $\Lambda^\infty \to (\Lambda/{=_\alpha})^\infty$ is not onto (Example 5.20), hence metric completion and quotienting by α-equivalence do not commute for terms with countably many free variables (as opposed to the case of uncountably many variables, see Theorem 5.19).
- The vertical map $\Lambda^\infty_{\mathsf{ffv}} \to (\Lambda/{=_\alpha})^\infty_{\mathsf{fs}}$ in the right-hand column is onto [KPSdV12, Theorem 22], in other words, restricted to terms with finitely many free variables, the two operations of metric completion and quotienting by α-equivalence do commute.

Nominal coalgebraic datatypes for a binding signature (Section 5.5) generalise (1.6) to the diagram below (where we omitted the middle row obtained from epi-mono factorisations).

$$\begin{tikzcd}
T_\Sigma \arrow[r, tail] \arrow[d, two heads] & T_\Sigma^\infty \arrow[d] & (T_\Sigma^\infty)_{\mathrm{ffv}} \arrow[l, hook'] \arrow[d, two heads] \arrow[dl, phantom, "(*)"] \\
T_\Sigma/{=_\alpha} \arrow[r, tail] & (T_\Sigma/{=_\alpha})^\infty & (T_\Sigma/{=_\alpha})^\infty_{\mathsf{fs}} \arrow[l, hook']
\end{tikzcd} \tag{1.7}$$

Σ is a so-called binding signature [FPT99] and T_Σ and $T_\Sigma/{=_\alpha}$ are initial algebras. The middle column of metric completions arises via unique arrows $\rightarrowtail$ from an ω-colimit into an ω^{op}-limit as in [Bar99, Proposition 3.1]. In the right-hand column, $(T_\Sigma/{=_\alpha})^\infty_{\mathsf{fs}}$ is the final coalgebra in Nom and $(T_\Sigma^\infty)_{\mathrm{ffv}}$ can be defined as making the right-hand square into a pullback. The theorem that metric completion commutes with quotienting by α-equivalence then follows from one of our main technical contributions, namely that pulling back the (not necessarily surjective) middle vertical arrow $T_\Sigma^\infty \to (T_\Sigma/{=_\alpha})^\infty$ along $(T_\Sigma/{=_\alpha})^\infty \hookleftarrow (T_\Sigma/{=_\alpha})^\infty_{\mathsf{fs}}$ yields a surjection $(T_\Sigma^\infty)_{\mathrm{ffv}} \to (T_\Sigma/{=_\alpha})^\infty_{\mathsf{fs}}$.

Representing limits in nominal sets (Section 5.6) provides the setting which enables us to give a semantic proof of the result discussed in the previous paragraph. To see the connection, denote by $U : \mathsf{Nom} \to \mathsf{Set}$ the forgetful functor and use the (well-known) result that the middle column of (1.7) arises as limits of ω^{op}-chains as depicted in

$$\begin{tikzcd}
UX_0 \arrow[d, two heads] & UX_1 \arrow[l] \arrow[d, two heads] & UX_2 \arrow[l] \arrow[d, two heads] & \cdots \arrow[l] & \lim UX_n \arrow[l] \arrow[d] & P \arrow[l, hook'] \arrow[d, two heads] \arrow[dl, phantom, "(*)"] \\
UY_0 & UY_1 \arrow[l] & UY_1 \arrow[l] & \cdots \arrow[l] & \lim UY_n \arrow[l] & U \lim Y_n \arrow[l, hook']
\end{tikzcd} \tag{1.8}$$

Similarly, the bottom right-hand corner of (1.7) is given by the limit in Nom of the lower sequence. The question whether metric completion commutes with α-equivalence now becomes an instance of a more general question. Given an ω^{op}-sequence of surjections $X_n \to Y_n$ in Nom, can the limit $\lim Y_n$ in Nom be represented by a surjection $P \to U \lim Y_n$, where P is defined to be the pullback $(*)$? A careful analysis of this situation is carried out in Section 5.6. In particular, the notions of safe squares, safe maps and of the bound variables relative to a map are introduced and it is shown that $P \to U \lim Y_n$ is onto if all vertical maps and squares in the chain are safe. We also explore the relationship between safe maps and maps with orbit-finite fibres.

As applications, we give a general definition of substitution on the final coalgebra coming from a binding signature. We also give corecursive definitions of various notions of infinite normal form (Böhm, Lévy-Longo and Berarducci trees) on α-equivalence classes of terms. We also show a solution, suggested to us by Pitts, of how to treat infinitely many free variables in nominal sets.

Related Work. This paper generalises [KPSdV12, Theorem 22] from the particular functor describing λ-calculus to arbitrary binding signatures. Along with this generalisation we replaced the syntactic proof (depending on a concrete presentation of the functor) of [KPSdV12, Theorem 22] with a semantic argument for the generalised Theorem 5.34 of this paper. In particular, the new material in Section 5.6 allows us to show that all elements of the final coalgebra are presented by infinite terms with finitely many free variables, using only semantic (that is, category theoretic) properties of the Nom-endofunctors of (5.4).

2. Preliminaries on Algebra and Coalgebra

Finite data types, or *algebraic data types*, can be studied as initial algebras for functors on different categories. Consider an endofunctor F on a category $\mathcal{C}$ and an object X of $\mathcal{C}$. An F-algebra (X, α) with carrier X is a $\mathcal{C}$-morphism $\alpha : FX \to X$. Given two F-algebras (X, α) and (Y, β) an F-algebra morphism is an arrow $f : X \to Y$ such that $f \circ \alpha = \beta \circ Ff$. The F-algebras thus form a category. One can prove the existence of an initial object in this category under certain assumptions on the cocompleteness of $\mathcal{C}$ and on the “size” of the functor F. Assume that $\mathcal{C}$ has colimits. Then we can consider the following sequence, starting from the initial $\mathcal{C}$-object 0.

$$0 \xrightarrow{\;!\;} F0 \xrightarrow{F!} F^2 0 \xrightarrow{F^2!} \ldots \longrightarrow F^\omega 0 \longrightarrow \cdots \tag{2.1}$$

Above $! : 0 \to F0$ is the unique map from the initial object 0 to $F0$. When we reach a limit ordinal α we define $F^\alpha 0$ as $\mathrm{colim}_{\beta<\alpha} F^\beta 0$. The colimit of this sequence, when it exists, is the carrier of the initial F-coalgebra. Notice that for each ordinal ι we have a canonical map from $F^\iota 0$ into the initial algebra. For example, if F preserves colimits of ω-chains, then the initial F-algebra is $F^\omega 0$.

Colimits of ω-chains are an example of a well studied class of colimits, namely *filtered colimits* [AR94]. Recall that a filtered colimit is a colimit of a diagram $J : \mathsf{D} \to \mathsf{C}$ where D is a category such that any finite diagram in D has a cocone. Functors that preserve filtered colimits are called *finitary.* Finitary functors have an initial algebra and the computation of the colimit of the initial chain stops after ω steps.

Infinitary data types, or *coalgebraic data types*, are understood as final coalgebras for suitable functors. An F-coalgebra (X, ξ) is defined as an arrow $\xi : X \to FX$. A coalgebra morphism between (X, ξ) and (Y, ζ) is a $\mathcal{C}$-morphism $f : X \to Y$ such that $Ff \circ \xi = \zeta \circ f$. Similarly to the initial algebra situation, we can consider the final sequence

$$1 \xleftarrow{\;!\;} F1 \xleftarrow{F!} F^2 1 \longleftarrow \ldots \longleftarrow F^\omega 1 \longleftarrow \cdots \tag{2.2}$$

where 1 is the final $\mathcal{C}$-object and ! is the unique arrow from $F1$ to 1. Assume that $\mathcal{C}$ has limits. For limit ordinals we compute the limit of the diagram constructed previously. The limit of the final sequence, if it exists, is the carrier of the final coalgebra. Therefore, functors that preserve limits of ω^{op}-chains, sometimes called *continuous*, have a final coalgebra whose carrier is $F^\omega 1$. We also have canonical maps from the final coalgebra to each $F^\iota 1$ for all ordinals ι.

The metric completion of $F^\omega 0$ is given by $F^\omega 1$, see Barr [Bar99, Proposition 3.1]. In more detail, if F is an endofunctor on Set and $F0$ is nonempty, then one can equip the set $F^\omega 1$ with a metric and prove that it is the metric completion of $F^\omega 0$. The metric on $F^\omega 1$ is obtained using the projections $p_n : F^\omega 1 \to F^n 1$. Explicitly, for $t, t' \in F^\omega 1$ we put $d(t, t') = 2^{-\max\{n \mid p_n t = p_n t'\}}$.

3. Preliminaries on Infinitary Lambda Calculus

3.1. **Infinitary Terms as a Final Coalgebra.** We assume familiarity with basic notions and notations of the finite λ-calculus [Bar84]. The set Λ of finite λ-terms is defined by induction from the grammar:

$$M ::= x \mid (\lambda x.M) \mid (MM) \tag{3.1}$$

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

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 [KKSdV97, KdV03]. The notion of α-equivalence will be given in Definitions 5.11 and 5.17.

The idea of putting a metric on a set of terms goes at least back to Arnold and Nivat [AN80]. To do so we define truncations.

Definition 3.1 (Truncation)**.** The truncation of a term $M \in \Lambda$ at depth $n \in \mathbb{N}$ is defined by induction on n:

$$\begin{aligned} M^0 &= *, \\ M^{n+1} &= \begin{cases} x & \textit{if } M = x \in \mathcal{V}, \\ \lambda x.N^n & \textit{if } M = \lambda x.N, \\ N^n P^n & \textit{if } M = NP. \end{cases} \end{aligned} \tag{3.2}$$

where $*$ is any constant not appearing in the syntax of the λ-calculus, for example $\emptyset$.

Definition 3.2 (Metric)**.** We define a metric $d : \Lambda \times \Lambda \to [0,1]$ by

$$d(M,N) = 2^{-m}, \tag{3.3}$$

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 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 (3.1) as a coinductive definition. The fact that both definitions coincide is a consequence of Barr's theorem on final coalgebras for bicontinuous Set endofunctors.

Indeed, interpreting (3.1) coinductively amounts to taking as λ-terms the elements of the final coalgebra for the Set-endofunctor

$$L\ X = \mathcal{V} + \mathcal{V} \times X + X \times X. \tag{3.4}$$

Notice that the set Λ of finite λ-terms constitutes the initial algebra for L. A closer look at the proof of Barr [Bar99, Theorem 3.2 and Proposition 3.1] shows now that the metric d on Λ of Definition 3.2 coincides with the metric induced by the final coalgebra. Hence, by [Bar99, 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 $(\Lambda^\infty, \mathsf{unfold} : \Lambda^\infty \to L(\Lambda^\infty))$ 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 $\mathsf{unfold} : \Lambda^\infty \to L(\Lambda^\infty)$ is an isomorphism, hence the set $L(\Lambda^\infty)$ can be equipped with a complete metric. The map $\mathsf{unfold} : \Lambda^\infty \to L(\Lambda^\infty)$ is the unique uniformly continuous map from Λ^∞ to $L(\Lambda^\infty)$ making diagram (3.5) commutative:

$$\begin{tikzcd}
\Lambda \arrow[r, "\simeq"] \arrow[d, "\iota"'] & L(\Lambda) \arrow[d, "L(\iota)"] \\
\Lambda^\infty \arrow[r, "\mathsf{unfold}"'] & L(\Lambda^\infty)
\end{tikzcd} \tag{3.5}$$

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 3.1 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 infinitary terms $M \in \Lambda^\infty$ using truncations by

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

Also, $\mathsf{var}(M) = \mathsf{fv}(M) \cup \mathsf{bv}(M)$.

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 $\twoheadrightarrow\!\!\!\!\rightarrow_\beta$ that assumes a sequence of reduction steps of any ordinal length, see for instance [KKSdV95]. 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 enriching the syntax (3.1) with a fresh constant $\perp$ and then adding $\perp$-reduction, denoted by $\to_\perp$, defined as the smallest relation closed under contexts and containing $M \to_\perp \perp$ 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 [KKSdV97, KdV03, SdV11].

3.2. **Computing the Infinite Normal Forms using Corecursion.** 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 [SdV11].

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 and they are recalled in (3.6), (3.7) and (3.8).

In his book [Bar84], 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 $\lambda x_1 \ldots \lambda x_n.xN_1 \ldots N_m$, or it is a term of the form $\lambda x_1 \ldots \lambda x_n.((\lambda y.P)Q)N_1 \ldots N_m$ 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 $BT(M)$ of a term M, that we now recognise as a corecursive definition.

$$BT(M) = \begin{cases} \lambda x_1 \dots \lambda x_n . y BT(M_1) \dots BT(M_m), & \\ & \text{\textit{if} } M \twoheadrightarrow_\beta \lambda x_1 \dots \lambda x_n . y M_1 \dots M_m, \\ \bot & \text{\textit{otherwise.}} \end{cases} \tag{3.6}$$

The image of BT is denoted as $\mathcal{BT}$ and can be explicitly defined as follows.

Definition 3.3 (Set of Böhm trees)**.** The set $\mathcal{BT}$ of *Böhm trees* is defined as the maximal set such that for all $M \in \mathcal{BT}$, either $M = \bot$ or $M = \lambda x_1 \dots \lambda x_n . y M_1 \dots M_m$ where $M_1, \dots, M_m \in \mathcal{BT}$ for some $m, n \geq 0$.

Clearly, any Böhm tree in $\mathcal{BT}$ is an infinitary lambda term over the syntax (3.1) enriched with $\bot$.

Taking for $\mathcal{U}$ the set of terms without hnf, one can show using the confluence property that the normal forms of the corresponding infinitary λ-calculus satisfy the equations in (3.6). 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 $\bot$ [Bar84, KKSdV97].

Alternatively, as Abramsky has forcefully argued in [Abr90], 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 (3.7) that define the Lévy-Longo tree $LLT(M)$ of a term M corecursively [Lon83, Lév76, AO93, KKSdV97].

$$LLT(M) = \begin{cases} y LLT(M_1) \dots LLT(M_m) & \text{\textit{if} } M \twoheadrightarrow_\beta y M_1 \dots M_m, \\ \lambda x . LLT(N) & \text{\textit{if} } M \twoheadrightarrow_\beta \lambda x . N, \\ \bot & \text{\textit{otherwise.}} \end{cases} \tag{3.7}$$

The image of LLT is denoted as $\mathcal{LLT}$ and can be explicitly defined as follows.

Definition 3.4 (Set of Lévy-Longo trees)**.** The set $\mathcal{LLT}$ of Lévy-Longo trees can be defined as the maximal set that satisfies that whenever $M \in \mathcal{LLT}$ then M has one of the following shapes:

(1) either $M = \bot$, or
(2) $M = \lambda x.N$ for some $N \in \mathcal{LLT}$, or
(3) $M = yM_1 \dots M_n$ for some $M_1, \dots, M_n \in \mathcal{LLT}$.

Term	**Böhm tree**	**Lévy-Longo tree**	**Berarducci tree**
M	$BT(M)$	$LLT(M)$	$BeT(M)$
$\mathsf{fix}\ x$	$x(x(x(\ldots)))$	$x(x(x(\ldots)))$	$x(x(x(\ldots)))$
$\mathsf{fix}(\lambda yx.xy)$	$\lambda x_1.x_1(\lambda x_2.x_2(\ldots))$	$\lambda x_1.x_1(\lambda x_2.x_2(\ldots))$	$\lambda x_1.x_1(\lambda x_2.x_2(\ldots))$
$\lambda x.\Omega$	$\perp$	$\lambda x.\perp$	$\lambda x.\perp$
$\mathsf{fix}\ (\lambda xy.x)$	$\perp$	$\lambda x_1 \lambda x_2 \lambda x_3 \ldots$	$\lambda x_1 \lambda x_2 \lambda x_3 \ldots$
$\Omega\Omega$	$\perp$	$\perp$	$\perp\perp$
$\mathsf{fix}(\lambda y.yx)$	$\perp$	$\perp$	$(((\ldots)x)x)x)$

Figure 1: Examples of Böhm, Lévy-Longo and Berarducci trees

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 $\Omega = (\lambda x.xx)(\lambda x.xx)$ has no tnf. The normal forms of this calculus can alternatively be characterised by the corecursive definition of the Berarducci tree [Ber96, KKSdV97] $BeT(M)$ of a term M:

$$BeT(M) = \begin{cases} x & \text{if } M \twoheadrightarrow_\beta x, \\ \lambda x.BeT(N) & \text{if } M \twoheadrightarrow_\beta \lambda x.N, \\ BeT(N)BeT(P) & \text{if } M \twoheadrightarrow_\beta NP \text{ and } N \text{ is a zero term}, \\ \perp & \text{otherwise}, \end{cases} \tag{3.8}$$

The image of BeT is denoted as $\mathcal{B}e\mathcal{T}$ and can be explicitly defined as follows.

Definition 3.5 (Set of Berarducci trees)**.** The set $\mathcal{B}e\mathcal{T}$ of *Berarducci trees* can be defined as the maximal set that satisfies that whenever $M \in \mathcal{B}e\mathcal{T}$ then M has one of the following shapes:

(1) $M = \lambda x.N$ for some $N \in \mathcal{B}e\mathcal{T}$, or
(2) $M = yM_1 \ldots M_n$ for some $M_1, \ldots, M_n \in \mathcal{B}e\mathcal{T}$, or
(3) $M = \perp M_1 \ldots M_n$ for some $M_1, \ldots, M_n \in \mathcal{B}e\mathcal{T}$, or
(4) $M = ((\ldots)M_2)M_1$ for some $(M_i)_{i \geq 1}$ such that $M_i \in \mathcal{B}e\mathcal{T}$ for all $i \geq 1$.

Some examples of trees are shown in Figure 1.

It is possible to formalise (3.6)-(3.8) 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-equivalence, we will prove an α-corecursion principle based on nominal sets.

4. Preliminaries on Nominal Sets

We recall basic facts on nominal sets [GP01, Pit13].

Consider a countably 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 X equipped with an action of the group $\mathfrak{S}(\mathcal{V})$, denoted by $\cdot : \mathfrak{S}(\mathcal{V}) \times X \to X$. We say that $u \in X$ 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 $u \in X$ is finitely supported if there exists a finite $S \subseteq \mathcal{V}$ which supports u.

Definition 4.1 (Nominal set)**.** A nominal set $(X, \cdot)$ is set X equipped with a $\mathfrak{S}(\mathcal{V})$-action such that all elements of X are finitely supported. Given nominal sets $(X, \cdot)$ and $(Y, \cdot)$, a map $f : X \to Y$ is called *equivariant* when $f(\pi \cdot u) = \pi \cdot f(u)$ for all $\pi \in \mathfrak{S}(\mathcal{V})$ and $u \in X$. 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 [GP01]. Indeed, if two finite sets S_1 and S_2 support u, then their intersection also supports u.

Notation 4.2 (Support and freshness)**.** The smallest finite support of u is denoted by $\mathsf{supp}(u)$. If $x \in \mathcal{V} \backslash \mathsf{supp}(u)$ we say that x is *fresh* for u, and write $x \# u$. More generally, given two nominal sets $(X, \cdot)$ and $(Y, \cdot)$, $u \in X$ and $v \in Y$, we write $u \# v$ for $\mathsf{supp}(u) \cap \mathsf{supp}(v) = \emptyset$. Given $S, T \subseteq \mathcal{V}$, we write $S \# u$ for $\mathsf{supp}(u) \cap S = \emptyset$. We also write $S \# T$ for $S \cap T = \emptyset$.

Remark 4.3. An important property of supp is that for every equivariant $f : X \to Y$ and $u \in X$, we have $\mathsf{supp}(f(u)) \subseteq \mathsf{supp}(u)$.

Example 4.4. The set of names $\mathcal{V}$ equipped with the evaluation action given by $\pi \cdot x = \pi(x)$ is a nominal set.

Example 4.5. The finite subsets of atoms $\mathcal{P}_{\mathit{fin}}(\mathcal{V})$ form a nominal set with the pointwise action $\pi \cdot X = \{\pi(u) \mid u \in X\}$ for all $X \in \mathcal{P}_{\mathit{fin}}(\mathcal{V})$.

Remark 4.6. Notice that taking the support of elements of a nominal set X gives an equivariant map $\mathsf{supp} : X \to \mathcal{P}_{\mathit{fin}}(\mathcal{V})$. Indeed, one can show that for any $u \in X$ and $\pi \in \mathfrak{S}(\mathcal{V})$ we have $\mathsf{supp}(\pi \cdot u) = \pi \cdot \mathsf{supp}(u)$. As a consequence, $\mathsf{supp}(\pi \cdot u)$ and $\mathsf{supp}(u)$ have the same cardinality for any permutation π.

Example 4.7. The set Λ of finite λ-terms with the action $\cdot : \mathfrak{S}(\mathcal{V}) \times \Lambda \to \Lambda$ inductively defined by

$$\begin{aligned} \pi \cdot x &= \pi(x) \\ \pi \cdot (\lambda x.M) &= \lambda \pi(x).(\pi \cdot M) \\ \pi \cdot (MN) &= ((\pi \cdot M)(\pi \cdot N)) \end{aligned} \tag{4.1}$$

is a nominal set. In this example we do not take into account α-equivalence, 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 X, let X_{fs} denote the set

$$X_{\mathsf{fs}} = \{u \in X \mid u \textit{ is finitely supported}\}. \tag{4.2}$$

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

Example 4.8. 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 (4.1). 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 4.9 (Abstraction)**.** Let $(X, \cdot)$ be a nominal set. One defines $\sim_\alpha$ on $\mathcal{V} \times X$ by

$$(x_1, u_1) \sim_\alpha (x_2, u_2) \Leftrightarrow (\exists\ z\#\{x_1, u_1, x_2, u_2\})(x_1\ z) \cdot u_1 = (x_2\ z) \cdot u_2 \tag{4.3}$$

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

$$\pi \cdot \langle x\rangle u = \langle \pi \cdot x\rangle \pi \cdot u. \tag{4.4}$$

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

$$\langle x\rangle u \mapsto \langle x\rangle f(u). \tag{4.5}$$

Definition 4.10. [Concretion] Let $(X, \cdot)$ be a nominal set. Concretion is the partial function $@ : [\mathcal{V}]X \times \mathcal{V} \to X$ 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, observe that $[\mathcal{V}]X \times \mathcal{V}$ is a nominal set with the coordinatewise action of $\mathfrak{S}(\mathcal{V})$. One can show that concretion is equivariant. Indeed, if $z\#\langle y\rangle u$ then $\pi \cdot z\#\langle \pi \cdot y\rangle \pi \cdot u$ and $\pi \cdot (\langle y\rangle u@z) = (\langle \pi \cdot y\rangle \pi \cdot u)@\pi \cdot z$.

Definition 4.11. [Internal hom] Given two nominal sets $(X, \cdot)$ and $(Y, \cdot)$, we define the internal hom $[X, Y]$ as the nominal set of all functions $f : X \to Y$ that are finitely supported with respect to the action

$$(\pi \cdot f)(u) = \pi \cdot f(\pi^{-1} \cdot u).$$

Remark 4.12. A function $f : X \to Y$ is finitely supported if and only if there exists a finite set S of names, such that for all permutations $\pi \in \mathfrak{S}(\mathcal{V})$ that fix the names in S and for all $x \in X$ we have $\pi \cdot f(x) = f(\pi \cdot x)$.

Limits and colimits in Nom. Further, we recall some general results form [Pit03] that will be necessary in the rest of the paper. 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 $(X, \cdot)$ and $(Y, \cdot)$ is $(X \times Y, \cdot)$ 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 $(X_i, \cdot_i)_{i\in I}$, we can equip the set of all tuples $\{(u_i)_{i\in I} \mid u_i \in X_i\}$ with the pointwise action given by

$$\pi \cdot (u_i)_{i\in I} = (\pi \cdot_i u_i)_{i\in I}. \tag{4.6}$$

This is a $\mathfrak{S}(\mathcal{V})$-action, but some tuples may not be finitely supported. The product of $(X_i, \cdot_i)_{i\in I}$ in Nom is the nominal set $(\prod\limits_{i\in I} (X_i, \cdot_i))_{\mathsf{fs}}$ of tuples of the form $(u_i)_{i\in I}$ that are finitely supported with respect to the action of (4.6).

The limit in Nom of an ω^{op}-chain

$$\begin{tikzcd} X_1 & X_2 \arrow[l, "f_1"'] & \dots \arrow[l, "f_2"'] \end{tikzcd}$$

is the nominal set of *finitely supported* tuples $(u_1, u_2, \ldots)$ such that for all $i \in \omega$ we have $u_i \in X_i$ and $f_{i+1}(u_{i+1}) = u_i$.

The initial object of Nom is the empty nominal set with the trivial action. In Nom, all monomorphisms are strong and are precisely the injective equivariant maps.

Categorical properties of Nom. The category Nom is *locally finitely presentable*, see [AR94]. An object X in a category C is called finitely presentable when the hom functor $\mathsf{C}(X, -)$ preserves filtered colimits. For example the finitely presentable sets are the finite ones. A locally small category C is called locally finitely presentable when it is cocomplete and it has a small set A of finitely presentable objects, such that any object in C is a filtered colimit of objects in A.

We describe next the finitely presentable objects in Nom. First we need to define the notion of *orbit*.

Definition 4.13 (Orbit)**.** Consider a nominal set $(X, \cdot)$ and $x, y \in X$. We say that x and y are orbit equivalent when there exists $\pi \in \mathfrak{S}(\mathcal{V})$ such that $\pi \cdot x = y$. An orbit $\mathcal{O} \subseteq X$ is an equivalence class with respect to this equivalence relation.

Remark 4.14. Let $\mathcal{O}$ denote an orbit in a nominal set and consider $x \in \mathcal{O}$. Then $\mathcal{O} = \{\pi \cdot x \mid \pi \in \mathfrak{S}(\mathcal{V})\}$.

A proof of the following proposition is in [Pet12, Proposition 2.3.7].

Proposition 4.15. *A nominal set $(X, \cdot)$ is finitely presentable in Nom if and only if it has finitely many orbits.*

Observe that a nominal set is the disjoint union of its orbits, and hence the directed union of all its nominal subsets with finitely many orbits.

Properties of the abstraction functor. The abstraction functor $[\mathcal{V}](-) : \mathsf{Nom} \to \mathsf{Nom}$ preserves all limits and colimits, see [Pit13]. The remainder of this section is only needed in the proof of Lemma 5.56.

By [Pit13, Theorem 4.13] we know that $[\mathcal{V}](-)$ has a right adjoint $R : \mathsf{Nom} \to \mathsf{Nom}$ defined by

$$RX = \{g \in [\mathcal{V}, X] \mid (\forall x \in \mathcal{V}) x \# g(x)\}.$$

Above, $[\mathcal{V}, X]$ is the nominal set of finitely supported maps from $\mathcal{V}$ to X, as in Definition 4.11. The argument below [Pit13, (4.27)] also implies that R does not preserve colimits, the counterexample being the coproduct of two nominal sets. However we can show that R preserves filtered colimits. But first let us give a simpler description of the nominal set RX.

Lemma 4.16. *Consider a nominal set X and a Set-function $g : \mathcal{V} \to X$. We have that $g \in RX$ if and only if the following conditions are satisfied:*

1. *For all $x \in \mathcal{V}$ we have $x \# g(x)$.*
2. *There exists a finite set of names S such that for all $x \in \mathcal{V}$ we have $\mathsf{supp}(g(x)) \subseteq S$ and g is constant on $\mathcal{V} \setminus S$.*

Proof. For the direct implication, notice that (1) is clearly satisfied. For (2), let S denote the support of g. Observe that, by Remark 4.12 we have that $\mathsf{supp}(g(x)) \subseteq S \cup \{x\}$. Since $x \# g(x)$ we have that $\mathsf{supp}(g(x)) \subseteq S$ for all $x \in \mathcal{V}$. Now consider $x, x' \notin S$. Then, by Remark 4.12, we have $(x\ x') \cdot g(x) = g(x')$. On the other hand, $(x\ x') \cdot g(x) = g(x)$ holds since $x, x' \# g(x)$. Hence $g(x) = g(x')$, thus g is constant outside S.

Conversely, we only have to show that g is finitely supported. Let $\pi \in \mathfrak{S}(\mathcal{V})$ be a permutation that fixes the set S. By Remark 4.12, it is enough to show that $\pi \cdot g(x) = g(\pi \cdot x)$ for all $x \in \mathcal{V}$. Since $\mathsf{supp}(g(x)) \subseteq S$ we have that $\pi \cdot g(x) = g(x)$. We can also prove that $g(\pi \cdot x) = g(x)$. This is clear for $x \in S$. For the case when $x \notin S$, observe that $\pi \cdot x = \pi(x) \notin S$, thus we can use the fact that g is constant outside S. □

Proposition 4.17. *The functor* $[\mathcal{V}](-) : \mathsf{Nom} \to \mathsf{Nom}$ *has a finitary right adjoint.*

Proof. Consider a filtered diagram D and a functor $J : \mathsf{D} \to \mathsf{Nom}$. In order to prove that $R\mathrm{colim}_{d \in \mathsf{D}} Jd \simeq \mathrm{colim}_{d \in \mathsf{D}} RJ(d)$ it is enough to show that each finitely supported map $g : \mathcal{V} \to \mathrm{colim}_{d \in \mathsf{D}} Jd$ such that for all $x \in \mathcal{V}$ we have $x \# g(x)$ factors through a Jd for some $d \in \mathsf{D}$

$$\begin{tikzcd}
\mathcal{V} \arrow[d, dashed, "g_d"'] \arrow[dr, "g"] & \\
Jd \arrow[r, "\iota"] & \mathrm{colim}_{d \in \mathsf{D}} Jd
\end{tikzcd} \tag{4.7}$$

and the map g_d is finitely supported and for all $x \in \mathcal{V}$ we have $x \# g_d(x)$. Let S denote the finite support of g and let $x_0 \notin S$. There exists $d_0 \in \mathsf{D}$ such that $g(x_0) = [y_0]$ for some $y_0 \in Jd_0$. We used square brackets here to denote the equivalence classes needed in the computation of filtered colimits. By Lemma 4.16 we have that $\mathsf{supp}([y_0]) \subseteq S$ and for all $x \notin S$ we have that $g(x) = [y_0]$. Notice that $S = \{x_1, \ldots, x_n\}$ is a finite set and for each x_i there exists $d_i \in \mathsf{D}$ such that $g(x_i) = [y_i]$ for some $y_i \in Jd_i$. We can assume without loss of generality that for all $0 \leq i \leq n$ we have $\mathsf{supp}(y_i) = \mathsf{supp}([y_i])$, see [Pet12, Proposition 2.3.7].

By Lemma 4.16 for all $0 \leq i \leq n$ we have $x_i \# g(x_i) = [y_i]$ and $\mathsf{supp}(g(x_i)) = \mathsf{supp}([y_i]) \subseteq S$. Hence, for all $0 \leq i \leq n$ we have $x_i \# y_i$ and $\mathsf{supp}(y_i) \subseteq S$. Since D is filtered, there exists $d \in \mathsf{D}$ and arrows $u_i : d_i \to d$ in D for all $0 \leq i \leq n$. We define $g_d : \mathcal{V} \to Jd$ by

$$g_d(x) = \begin{cases} Ju_0(y_0) & \text{if } x \notin S, \\ Ju_i(y_i) & \text{if } x = x_i \text{ for some } 1 \leq i \leq n. \end{cases} \tag{4.8}$$

Notice that g_d satisfies the properties (1) and (2) of Lemma 4.16. Indeed, we have that $x \# g_d(x)$ and $\mathsf{supp}(g_d(x)) \subseteq S$ for all $x \in \mathcal{V}$ (because the maps Ju_i are equivariant and thus can only shrink the support of the elements, recall Remark 4.3). Thus g_d is in $R(Jd)$ by Lemma 4.16 and it makes diagram (4.7) commutative. Therefore R preserves filtered colimits. □

Lemma 4.18. *The functor* $[\mathcal{V}](-) : \mathsf{Nom} \to \mathsf{Nom}$ *preserves finitely presentable objects.*

Proof. This is immediate by the previous lemma. Let X be a finitely presentable nominal set and let colim Y_i be a filtered colimit. Then

$$\begin{aligned} \mathsf{Nom}([\mathcal{V}]X, \mathrm{colim}\ Y_i) &\simeq \mathsf{Nom}(X, R(\mathrm{colim}\ Y_i)) \\ &\simeq \mathsf{Nom}(X, \mathrm{colim}\ R(Y_i)) \\ &\simeq \mathrm{colim}\ \mathsf{Nom}(X, R(Y_i)) \\ &\simeq \mathrm{colim}\ \mathsf{Nom}([\mathcal{V}]X, Y_i). \end{aligned} \tag{4.9}$$

This shows that $[\mathcal{V}]X$ is finitely presentable. □

5. Alpha Corecursion Principle for Nominal Coalgebraic Data Types

In this section we introduce nominal coalgebraic data types as a means of studying infinitary data up to α-equivalence. In Section 5.1 we describe final coalgebras of certain Nom-endofunctors. We apply this result for functors arising from signatures with binding. We prove that the set of equivalence classes of infinitary terms with finitely many free variables is the final coalgebra of a Nom-functor. Our running example is the infinitary λ-calculus, and indeed the results of [KPSdV12] are particular instances of the main theorems in this section.

5.1. Final Coalgebras of Nom Functors.

In this section we describe the final coalgebras for certain endofunctors on Nom. It is well known that an endofunctor that preserves limits of ω^{op}-chains has a final coalgebra which is computed as the limit of the final chain. Similarly, endofunctors that preserve colimits of ω-chains have an initial algebra obtained as the colimit of the initial sequence. We will generalise Barr's theorem [Bar99] relating final coalgebras and initial algebras to functors on nominal sets. To this end we need to introduce nominal (complete) metric spaces.

Definition 5.1 (Nominal metric space)**.** A *nominal metric space* is a tuple $(X, \cdot, d)$ such that $(X, \cdot)$ is a nominal set and $d : X \times X \to [0, 1]$ is an equivariant metric when the interval $[0, 1]$ is equipped with the trivial action. That is, $d(x, y) = d(\pi \cdot x, \pi \cdot y)$ for all $x, y \in X$ and $\pi \in \mathfrak{S}(\mathcal{V})$.

Definition 5.2 (Finitely supported Cauchy sequence)**.** A *finitely supported Cauchy sequence* in a nominal metric space is a Cauchy sequence $(x_n)_{n\geq 1}$ such that there exists a finite set of variables $S \subseteq \mathcal{V}$ that supports all elements x_n. A nominal metric space is *complete* when every finitely supported Cauchy sequence converges.

Remark 5.3 (Nominal Completion)**.** Given a nominal metric space $(X, \cdot, d)$ one can construct its nominal completion $(\overline{X}, \cdot, \overline{d})$ by adding the limits of all the finitely supported Cauchy sequences. This construction has the following universal property. For any complete metric space $(Y, \cdot, e)$ and any equivariant uniformly continuous function $f : X \to Y$ there exists a unique equivariant uniformly continuous map $\overline{f} : \overline{X} \to Y$ extending f. The proofs are straightforward.

In what follows, Theorem 5.4 and Theorem 5.5 show that the final coalgebra of certain Nom-functors can be regarded as a nominal metric space and is the nominal completion of the initial algebra. Theorem 5.4 is an instance of Adámek's generalisation of Barr's theorem from [Adá03]. The fact that T is the *nominal* Cauchy completion of I is equivalent to $\hom(B, \mathsf{T})$ being the Cauchy completion of $\hom(B, \mathsf{I})$ for any finitely presentable objects $B \in \mathsf{Nom}$. Nevertheless, we opted to sketch a direct proof below, not only for the sake of completeness, but also because we can work directly with metrics on the initial algebra and final coalgebra, paying attention to some extra conditions regarding equivariance and finite support. Moreover, a careful inspection of the proof allows us to prove a small variation of this result, see Theorem 5.5. One crucial hypothesis in [Adá03] is that the functor at issue, say F, has the property that $\mathsf{F}0$ has an element, where 0 is the initial element in the category. By this, it is meant that a morphism $s : 1 \to \mathsf{F}0$ exists. However, by insights that go back to work by Fraenkel and Mostowski [Jec73, Pit13], the axiom of choice doesn't hold in the topos of nominal sets. So the fact that $\mathsf{F}0 \neq 0$ is not equivalent to the existence of a

Nom-morphism $s : 1 \to \mathsf{F}0$. As an example consider the functor L_α of 5.15, where we can apply Theorem 5.5, but not Theorem 5.4.

We first formulate Theorem 5.4 following [Adá03] in assuming that a morphism $s : 1 \to \mathsf{F}0$ exists and F preserves limits of ω^{op}-chains and monomorphisms. Note that we do not assume that F preserves colimits of ω-chains. The existence of the final coalgebra gives a size constraint on the functor F that ensures also the existence of the initial algebra.

However, in Theorem 5.5 we will see that a variation of Theorem 5.4 in which we only assume that $\mathsf{F}0 \neq 0$ holds provided that F can be extended to finitely-supported maps.

Theorem 5.4. *Let* $\mathsf{F} : \mathsf{Nom} \to \mathsf{Nom}$ *be a functor that preserves limits of* ω^{op}*-chains and monomorphisms and such that a morphism* $s : 1 \to \mathsf{F}0$ *exists. Then* F *has a final coalgebra* T *and an initial algebra* I*, both of which be equipped with equivariant metrics. Moreover the final coalgebra is the nominal completion of the initial algebra.*

Proof. We split the proof in four parts.

(1) *Existence of final* F*-coalgebra and initial* F*-algebra.* Since it preserves limits of ω^{op}-chains, the functor F has a final coalgebra which can be computed as the limit of the ω^{op}-chain

$$1 \xleftarrow{\;!\;} \mathsf{F}1 \xleftarrow{\mathsf{F}!} \mathsf{F}^2 1 \xleftarrow{\mathsf{F}^2!} \cdots \longleftarrow \mathsf{T} \tag{5.1}$$

Since Nom is locally finitely presentable and F preserves monomorphisms we have that the initial F-algebra exists and is a subobject of T. For full details and the general proof see [Adá03, Proposition 3.4]. The idea is to prove by transfinite induction that for every ordinal ι we have a monomorphism $u_\iota : \mathsf{F}^\iota 0 \to \mathsf{F}^\iota 1$. For example for finite ordinals we put $u_i = \mathsf{F}^i(u)$ where $u : 0 \to 1$ is the unique morphism into the final nominal set. For all $\iota \geq \omega$ we have that $\mathsf{F}^\iota 0$ is a subobject of $\mathsf{F}^\iota 1 \cong \mathsf{F}^\omega 1$. Using the fact that Nom is well powered, we know that $\mathsf{F}^\omega 1$ only has a set of subobjects. Hence the initial sequence converges to the initial F-algebra, denoted by I. Moreover since for all $\iota \geq \omega$ we have $\mathsf{F}^\iota 1 \cong \mathsf{F}^\omega 1 \cong \mathsf{T}$ there exists a monomorphism $v : \mathsf{I} \to \mathsf{T}$.

$$\begin{tikzcd}
0 \arrow[r] \arrow[d, "u"] & \mathsf{F}0 \arrow[r] \arrow[d, hook, "u_1"] & \mathsf{F}^2 0 \arrow[r] \arrow[d, hook, "u_2"] & \cdots \arrow[r] & \mathsf{F}^\omega 0 \arrow[r] \arrow[d, hook, "u_\omega"] & \cdots \arrow[r] & \mathsf{F}^\iota 0 \arrow[d, hook, "u_\iota"] & \cong & \mathsf{I} \arrow[d, hook, dashed, "v"] \\
1 & \mathsf{F}1 \arrow[l] & \mathsf{F}^2 1 \arrow[l] & \cdots \arrow[l] & \mathsf{F}^\omega 1 \arrow[l] & \cdots \arrow[l, "\cong"'] & \mathsf{F}^\iota 1 \arrow[l] & \cong & \mathsf{T}
\end{tikzcd} \tag{5.2}$$

(2) *Equivariant metrics on* T *and* I. First we define a metric on T. Let $p_n : \mathsf{T} \to \mathsf{F}^n 1$ denote the projections of the limit in (5.1). Put

$$d(x, y) = 2^{-max\{n \mid p_n x = p_n y\}} \tag{5.3}$$

with the convention that $2^{-\omega} = 0$. Considering the interval $[0, 1]$ as a nominal set with the trivial action the map $d : \mathsf{T} \times \mathsf{T} \to [0, 1]$ becomes equivariant. This follows easily since each p_n is equivariant.

(3) T *is a nominal complete metric space.* Next we show that (T, d) is a nominal complete metric space. Consider a finitely supported Cauchy sequence $(x_n)_n \subseteq \mathsf{T}$. This implies that there exists a finite set $S \subseteq \mathcal{V}$ that supports all x_n. Without loss of generality we may assume that for every n we have $d(x_n, x_{n+1}) \leq 2^{-n}$. Therefore, we have that $p_n x_n = p_n x_{n+1}$ for all n, or equivalently, $p_n x_n = \mathsf{F}^n!(p_{n+1} x_{n+1})$. Since for all n we have $\mathsf{supp}(x_n) \subseteq S$ and p_n is equivariant, it follows that $\mathsf{supp}(p_n x_n) \subseteq S$ for all n. The

existence of such a common support is essential, recall how limits of ω^{op}-chains are computed in Nom. It follows that there exists an element $x \in \mathsf{T}$ corresponding to the tuple $(p_n x_n)_n$. We thus have that $p_n x = p_n x_n$ for all n and this proves that x is a limit of $(x_n)_n$ with respect to the metric d.

(4) T *is the nominal completion of* I. Since I contains $\mathsf{F}^\omega 0$, see (5.2), it is enough to prove that any element $x \in \mathsf{T}$ can be written as the limit of a Cauchy sequence of elements in $\mathsf{F}^\omega 0$.

Next we use the existence of a morphism $s : 1 \to \mathsf{F}0$. Notice that there exists a unique (finitely supported) map into 1, therefore we have that $!u_1 s = id_1$. We define $x_n \in \mathsf{F}^\omega 0$ as the image under the inclusion $\mathsf{F}^{n+1}0 \to \mathsf{F}^\omega 0$ of $\mathsf{F}^n s(p_n x)$. Notice that the support of x_n is included in the support of x for all natural numbers n. It is easy to check that $p_n u_\omega x_n = p_n x$, thus $d(u_\omega x_n, x) \leq 2^{-n}$. Therefore the sequence $(x_n)_n$ is a finitely supported Cauchy sequence whose limit is x. □

In order to relax the assumption that the map $s : 1 \to \mathsf{F}0$ is equivariant we need to require the functor F to be Nom-enriched. This means that for any two nominal sets X and Y we have an equivariant map $\mathsf{F}_{X,Y} : [X, Y] \to [\mathsf{F}X, \mathsf{F}Y]$ which behaves well with respect to composition [Kel82], where $[X, Y]$ denotes the internal hom in the cartesian closed category Nom, and consists of the finitely supported maps from X to Y, see Definition 4.11.

Theorem 5.5. *Let* $\mathsf{F} : \mathsf{Nom} \to \mathsf{Nom}$ *be a functor that preserves limits of* ω^{op}*-chains and monomorphisms and such that* $\mathsf{F}0 \neq 0$*. Assume further that the functor* F *is* Nom*-enriched. Then* F *has a final coalgebra* T *and an initial algebra* I*, both of which can be equipped with equivariant metrics. Moreover the final coalgebra is the nominal completion of the initial algebra.*

Proof. Notice that in Nom the fact that $\mathsf{F}0 \neq 0$ is equivalent to the existence of a *finitely supported* map $s : 1 \to \mathsf{F}0$. Thus, if F can be extended to finitely supported maps, the proof follows the same lines as Theorem 5.4 with a small difference. The maps $\mathsf{F}^n s$ are finitely-supported rather than equivariant. Since F is Nom-enriched the support of $\mathsf{F}^n s$ is included in the support of s for any finitely supported map s. This follows by Remark 4.3 since the maps $\mathsf{F}_{X,Y} : [X, Y] \to [\mathsf{F}X, \mathsf{F}Y]$ are equivariant. Therefore the support of the Cauchy sequence $(x_n)_n$ is included in the support of x union the support of the map s. □

As a side observation, notice that F being an enriched Nom-functor with respect to the cartesian symmetric monoidal structure of Nom is equivalent to the existence of a strength, a notion that we will also use in Section 6.1.

One can easily show that functors obtained from the grammar (5.4) below have the properties required in Theorems 5.4 and 5.5.

Proposition 5.6. *Endofunctors on* Nom *obtained from the grammar*

$$F ::= \mathcal{V} \mid K \mid \mathsf{Id} \mid \coprod F \mid F \times F \mid [\mathcal{V}]F \tag{5.4}$$

preserve monomorphisms, epimorphims and limits of ω^{op}*-chains and are* Nom*-enriched. Above* K *denotes a constant functor,* $\coprod F$ *denotes at most countable coproducts, while* $[\mathcal{V}]F$ *denotes precomposition with the abstraction functor introduced in Definition 4.9.*

Proof. It is immediate that the constant functors $\mathcal{V}$ and K, the identity functor Id preserve all limits and colimits. Countable coproducts commute in Nom with limits of ω^{op}-chains and preserve monomorphisms and epimorphisms. Binary products commute in Nom with all limits and preserve epimorphisms. Finally, the abstraction functor $[\mathcal{V}](-) : \mathsf{Nom} \to \mathsf{Nom}$ is both a right adjoint, see [Pit13, Theorem 4.12], and a left adjoint, see [Pit13, Theorem 4.13]. Therefore $[\mathcal{V}](-)$ preserves all limits and colimits, in particular limits of ω^{op}-chains, monos and epis. The fact that $[\mathcal{V}](-)$ is Nom-enriched follows from [Pit11, Lemma 4.10]. The functors obtained from products and coproducts can be easily proved to be Nom-enriched, see [Kel82]. □

As a corollary, we can apply Theorem 5.4 to any endofunctor of the form (5.4) with the additional property that a Nom-morphism $s : 1 \to \mathsf{F}0$ exists. Similarly we can apply Theorem 5.5 to any functor of the form (5.4) with the additional property that $\mathsf{F}0 \neq 0$.

As an aside, note that the class of functors with the property that $\mathsf{F}0 \neq 0$ is not closed under countable products:

Example 5.7. Consider the functors $\mathsf{F}_n : \mathsf{Nom} \to \mathsf{Nom}$ defined by $\mathsf{F}_n(X) = P_n(\mathcal{V}) + X$ where $P_n(\mathcal{V})$ is the nominal set of subsets of variables of cardinality n with the pointwise action. Observe that for every n we have $\mathsf{F}_n 0 \neq 0$, but $\prod\limits_{n<\omega} \mathsf{F}_n 0 = 0$.

5.2. **Nominal Algebraic Data Types for Binding Signatures.** In this section we will introduce binding signatures [FPT99] and we will see how the set of finite raw terms, respectively the set of α-equivalence classes of terms for a binding signature can be obtained as initial algebras for Nom-functors. The results in this section are based on [GP01, Pit06, Pit11].

Definition 5.8 (Terms coming from a binding signature)**.** A *binding signature* is a pair (Σ, ar) where Σ is a set of operations and $\mathsf{ar} : \Sigma \to \mathbb{N}^*$ specifies the binding arity of the operations. The set of finite *raw terms* T_Σ for a binding signature (Σ, ar) is defined by the inductive rules

$$\frac{}{x \in T_\Sigma}\,(x \in \mathcal{V}) \qquad\qquad \frac{t_1 \in T_\Sigma, \ldots, t_k \in T_\Sigma}{\mathsf{op}(\langle \overline{x_1} \rangle.t_1, \ldots, \langle \overline{x_k} \rangle.t_k) \in T_\Sigma}\,(\mathsf{ar}(\mathsf{op}) = n_1, \ldots, n_k) \tag{5.5}$$

where in the second rule of (5.5) $\overline{x_i}$ denotes a list $x_i^1, \ldots, x_i^{n_i}$ of of names of length n_i where $n_i \geq 0$ and $1 \leq i \leq k$. If $k = 0$, we write $\mathsf{op}()$ for representing constants. Note that (5.5) defines raw terms, not quotiented by α-equivalence, but the intention here is to say that op binds the names in the list $\overline{x_i}$ in the scope given by the term t_i.

Example 5.9 (λ-terms)**.** The binding signature for λ-calculus consists of two operations Abs (abstraction) and App (application) with respective arities

$$\begin{aligned} \mathsf{ar}(\mathsf{Abs}) &= 1, \\ \mathsf{ar}(\mathsf{App}) &= 0, 0. \end{aligned} \tag{5.6}$$

The raw terms in this signature are the finite λ-terms *not* quotiented by α-equivalence. For example, $\lambda x.x$ is written as $\mathsf{Abs}(\langle \mathsf{x} \rangle \mathsf{x})$ and $(x\ y)$ as $\mathsf{App}(\langle\rangle \mathsf{x}, \langle\rangle \mathsf{y})$. Recall from Example 4.7 that the set of λ-terms can be equipped with an action of the group $\mathfrak{S}(\mathcal{V})$ and thus turned into a nominal set.

We can define inductively a $\mathfrak{S}(\mathcal{V})$-action on raw terms for an arbitrary binding signature

$$\begin{aligned} &\pi \cdot x = \pi(x) && \text{if } x \in \mathcal{V}, \\ &\pi \cdot \mathsf{op}(\langle \overline{x_1} \rangle.t_1, \ldots, \langle \overline{x_k} \rangle.t_k) = \mathsf{op}(\langle \pi \cdot \overline{x_1} \rangle.t_1, \ldots, \langle \pi \cdot \overline{x_k} \rangle.t_k), && \end{aligned} \tag{5.7}$$

where π acts pointwise on the lists x_i, that is, if $\overline{x_i} = x_i^1, \ldots, x_i^{n_i}$ then $\pi \cdot \overline{x_i}$ is the list $\pi \cdot x_i^1, \ldots, \pi \cdot x_i^{n_i}$. Then we have that $(T_\Sigma, \cdot)$ is a nominal set. Notice that the support of a term $t \in T_\Sigma$ is then the set of all variables occurring in t.

Definition 5.10 (Free and bound variables)**.** We can inductively define the set of free variables in a term $t \in T_\Sigma$:

$$\begin{aligned} &\mathsf{fv}(x) = x && \text{if } x \in \mathcal{V}, \\ &\mathsf{fv}(\mathsf{op}(\langle \overline{x_1} \rangle.t_1, \ldots, \langle \overline{x_k} \rangle.t_k)) = \textstyle\bigcup_{i=1}^k \mathsf{fv}(t_i) \setminus \bigcup_{i=1}^k \overline{x_i}. && \end{aligned} \tag{5.8}$$

The set of bound variables is defined similarly:

$$\begin{aligned} &\mathsf{bv}(x) = \emptyset && \text{if } x \in \mathcal{V}, \\ &\mathsf{bv}(\mathsf{op}(\langle \overline{x_1} \rangle.t_1, \ldots, \langle \overline{x_k} \rangle.t_k)) = \textstyle\bigcup_{i=1}^k \mathsf{bv}(t_i) \cup \bigcup_{i=1}^k \overline{x_i}. && \end{aligned} \tag{5.9}$$

For the binding signature in Example 5.9 we obtain the usual definition of free and bound variable in a λ-term.

We will now proceed to define α-equivalence on finite raw terms for a binding signature. We will use the fact that T_Σ is a nominal set. First let us recall the case of λ-calculus. On finite λ-terms, α-equivalence can be defined inductively using the permutation action $\cdot : \mathfrak{S}(\mathcal{V}) \times \Lambda \to \Lambda$ of Example 4.7, see [GP01].

Definition 5.11 (Alpha equivalence on λ-terms)**.** Let $M, N, M', N' \in \Lambda$. The relation $=_\alpha$ is the least equivalence relation closed under the rules:

$$\frac{}{x =_\alpha x}\,(var) \qquad \frac{M =_\alpha N \quad M' =_\alpha N'}{MM' =_\alpha NN'}\,(app)$$

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

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 5.12 (Alpha equivalence on terms coming from a binding signature)**.** For an arbitrary binding signature (Σ, ar), we can define α-equivalence on finite terms inductively:

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

$$\frac{(\overline{x_i}\ \overline{z_i}) \cdot t_i =_\alpha (\overline{y_i}\ \overline{z_i}) \cdot s_i \quad 1 \leq i \leq k \quad \text{for fresh } \overline{z_i}}{\mathsf{op}(\langle \overline{x_1} \rangle.t_1, \ldots, \langle \overline{x_k} \rangle.t_k) =_\alpha \mathsf{op}(\langle \overline{y_1} \rangle.s_1, \ldots, \langle \overline{y_k} \rangle.s_k)}\,(\mathsf{op})$$

where each $\overline{z_i}$ is a list of distinct elements of length n_i which are fresh for all the terms involved. Also, by $(\overline{y_i}\ \overline{z_i})$ we mean the composition of transposition $(x_i^1\ z_i^1) \ldots (x_i^{n_i}\ z_i^{n_i})$.

The relation $=_\alpha$ is equivariant, that is, $t =_\alpha s$ implies $\pi \cdot t =_\alpha \pi \cdot s$ for all $\pi \in \mathfrak{S}(\mathcal{V})$. Thus we obtain a nominal set $(T_\Sigma/{=_\alpha}, \cdot)$. The equivalence class of a term $t \in T_\Sigma$ is denoted by $[t]_\alpha$ and we have that $\mathsf{supp}([t]_\alpha) = \mathsf{fv}(t)$.

We can express the α-equivalence classes of terms in a binding signature as an initial algebra for a Nom-functor. The next result allowed Gabbay and Pitts to formulate α-structural induction and recursion principles, see [Pit11, GP01, Pit06].

Theorem 5.13 (Nominal algebraic data types)**.** *The nominal set $(T_\Sigma/{=_\alpha}, \cdot)$ of finite α-equivalence classes of terms is the initial algebra for the functor $\mathsf{F}_\alpha : \mathsf{Nom} \to \mathsf{Nom}$ given by:*

$$\mathsf{F}_\alpha\ X = \mathcal{V} + \coprod_{\substack{\mathsf{op}\in\Sigma \\ \mathsf{ar}(\mathsf{op})=n_1,\ldots,n_k}} [\mathcal{V}]^{n_1} X \times \ldots \times [\mathcal{V}]^{n_k} X. \tag{5.10}$$

On the other hand, the nominal set $(T_\Sigma, \cdot)$ of finite raw terms is the initial algebra for the functor $\mathsf{F} : \mathsf{Nom} \to \mathsf{Nom}$ given by:

$$\mathsf{F}\ X = \mathcal{V} + \coprod_{\substack{\mathsf{op}\in\Sigma \\ \mathsf{ar}(\mathsf{op})=n_1,\ldots,n_k}} (\mathcal{V}^{n_1} \times X) \times \ldots \times (\mathcal{V}^{n_k} \times X). \tag{5.11}$$

We can also obtain T_Σ as the initial algebra for a Set-functor. Indeed, we define a functor

$$F : \mathsf{Set} \to \mathsf{Set} \tag{5.12}$$

by the same formula as (5.11). The initial F-algebra is the set T_Σ. Notice that F is a lifting of F to nominal sets in the sense that the next diagram commutes.

$$\begin{tikzcd}
\mathsf{Nom} \arrow[r, "\mathsf{F}"] \arrow[d, "U"'] & \mathsf{Nom} \arrow[d, "U"] \\
\mathsf{Set} \arrow[r, "F"] & \mathsf{Set}
\end{tikzcd} \tag{5.13}$$

Remark 5.14. Notice that both functors F_α and F are obtained from grammar (5.4).

Example 5.15. Consider the binding signature for λ-calculus of Example 5.9. Then the nominal set $(\Lambda/{=_\alpha}, \cdot)$ is the initial algebra of the functor $\mathsf{L}_\alpha : \mathsf{Nom} \to \mathsf{Nom}$ already mentioned in the introduction (1.2):

$$\mathsf{L}_\alpha\ X = \mathcal{V} + [\mathcal{V}]X + X \times X$$

while the raw λ-terms for a nominal set which is the initial L-algebra for $\mathsf{L} : \mathsf{Nom} \to \mathsf{Nom}$ given by

$$\mathsf{L}\ X = \mathcal{V} + \mathcal{V} \times X + X \times X.$$

5.3. **An abstract account of α-equivalence.** In the above example α-equivalence is the kernel of the map $\Lambda \twoheadrightarrow \Lambda/{=_\alpha}$. Using Definition 4.9 we see that this map is induced at the level of Nom-endofunctors by the natural transformation

$$\theta_X : \mathcal{V} \times X \twoheadrightarrow [\mathcal{V}]X \tag{5.14}$$

defined by $(x, u) \mapsto \langle x \rangle u$.

Consider an arbitrary $\mathsf{F}_\alpha : \mathsf{Nom} \to \mathsf{Nom}$ obtained from grammar (5.4) or, equivalently, from a binding signature Σ. To F_α we can associate a functor $\mathsf{F} : \mathsf{Nom} \to \mathsf{Nom}$ in which all occurrences of the abstraction functor $[\mathcal{V}](-)$ are replaced by $\mathcal{V} \times -$. Therefore F is a polynomial functor and the natural transformation (5.14) induces a natural transformation $q : \mathsf{F} \to \mathsf{F}_\alpha$ given by the following inductive rules:

$$\frac{\mathsf{F}_\alpha = \mathsf{Id}}{\mathsf{F} = \mathsf{Id} \quad q = id} \qquad \frac{\mathsf{F}_\alpha = \mathcal{V}}{\mathsf{F} = \mathcal{V} \quad q = id} \qquad \frac{\mathsf{F}_\alpha = \mathsf{K}}{\mathsf{F} = \mathsf{K} \quad q = id}$$

$$\begin{gathered}
\frac{\mathsf{F}_\alpha = [\mathcal{V}]\mathsf{F}'_\alpha \qquad q' : \mathsf{F}' \to \mathsf{F}'_\alpha}{\mathsf{F} = \mathcal{V} \times \mathsf{F}' \qquad q = \theta \circ (\mathcal{V} \times q')} \\[2ex]
\frac{\mathsf{F}_\alpha = \mathsf{F}'_\alpha \times \mathsf{F}''_\alpha \qquad q' : \mathsf{F}' \to \mathsf{F}'_\alpha \qquad q'' : \mathsf{F}'' \to \mathsf{F}''_\alpha}{\mathsf{F} = \mathsf{F}' \times \mathsf{F}'' \qquad q = q' \times q''} \\[2ex]
\frac{\mathsf{F}_\alpha = \coprod (\mathsf{F_i})_\alpha \qquad q_i : \mathsf{F_i} \to (\mathsf{F_i})_\alpha}{\mathsf{F} = \coprod \mathsf{F_i} \qquad q = \coprod q_i}
\end{gathered} \tag{5.15}$$

Using the initial chain, this in turn gives a surjective map from the initial F-algebra $\mathsf{I} = (T_\Sigma, \cdot)$ to the initial F_α-algebra $\mathsf{I}_\alpha = (T_\Sigma/{=_\alpha}, \cdot)$. Indeed, we have the following diagram

$$\begin{tikzcd}
0 \arrow[r] \arrow[d, "{[-]_\alpha^{(0)}}"] & \mathsf{F}0 \arrow[r] \arrow[d, two heads, "{[-]_\alpha^{(1)}}"] & \mathsf{F}^2 0 \arrow[r] \arrow[d, two heads, "{[-]_\alpha^{(2)}}"] & \cdots \arrow[r] & \mathsf{I} \arrow[d, dashed, two heads, "{[-]_\alpha}"] \\
0 \arrow[r] & \mathsf{F}_\alpha 0 \arrow[r] & \mathsf{F}_\alpha^2 0 \arrow[r] & \cdots \arrow[r] & \mathsf{I}_\alpha
\end{tikzcd} \tag{5.16}$$

where $[-]_\alpha^{(0)} = id_0$ and $[-]_\alpha^{(n+1)} = q_{\mathsf{F}_\alpha^n 0} \circ \mathsf{F}[-]_\alpha^{(n)}$. The horizontal arrows are defined as in (2.1).

Since the functors F obtained from grammar (5.4) preserve surjections, we have that all the maps $[-]_\alpha^{(n)}$ are surjective. The compositions $\mathsf{F}^n 0 \twoheadrightarrow \mathsf{F}_\alpha^n 0 \to \mathsf{I}_\alpha$ form a cocone, thus using the universal property of the colimit I we obtain a unique map $[-]_\alpha : \mathsf{I} \to \mathsf{I}_\alpha$ such that diagram (5.16) commutes. To show that $[-]_\alpha$ is surjective assume $v, w : \mathsf{I}_\alpha \to B$ are two equivariant maps such that $v[-]_\alpha = w[-]_\alpha$. It follows that v and w equalise $\mathsf{F}^n 0 \twoheadrightarrow \mathsf{F}_\alpha^n 0 \to \mathsf{I}_\alpha$ for all n. Using the universal property of I_α, it follows that $v = w$. Hence $[-]_\alpha$ is surjective.

Remark 5.16. We had to explain why the maps $[-]_\alpha^{(n)}$ are surjections. In Set this follows easily, since all Set-functors preserve surjections (we just have to consider a right inverse). In Nom we have equivariant surjective maps that do not have right inverses. E.g. the unique map from $\mathcal{V}$ to the final nominal set $\{*\}$ has no equivariant right inverse, because the support of $*$ is empty whereas all elements of $\mathcal{V}$ have non-empty support.

Given a binding signature (Σ, ar) we described α-equivalence on finite terms in two equivalent ways:

- *syntactically*, as in Definition 5.12;
- *semantically*, via initial algebras, as in (5.16).

While the latter presentation of α-equivalence may seem rather pedantic, the abstract perspective sheds some light on the problems one has with defining α-equivalence and finding representatives for α-equivalence classes in the infinitary case.

5.4. **Problems with Alpha Equivalence in the Infinitary Case.** As an illustration, let us first look at possible definitions of α-equivalence for the infinitary λ-calculus. We see that there are two possible ways of defining α-equivalence classes and, contrary to all expectations, they are not equivalent in case of a countable set of variables $\mathcal{V}$.

Recall that the set of raw infinitary λ-terms Λ^∞ is the final coalgebra of the Set-functor in (3.4). We define α-equivalence on the set Λ^∞ using truncations. This definition is slightly different from, though equivalent, to those used in [KKSdV95, KKSdV97, KdV03].

Definition 5.17 (Alpha equivalence on infinitary λ-terms)**.** We extend the notion of α-conversion to the set Λ^∞ via

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

We thus obtain the quotient

$$\Lambda^\infty/{=_\alpha}.$$

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

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

$$d_\alpha(M, N) = \inf\{2^{-n} \mid M^n =_\alpha N^n,\ n \in \mathbb{N}\}. \tag{5.17}$$

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.

We consider the metric completion of $\Lambda/{=_\alpha}$ with respect to d_α, denoted by

$$(\Lambda/{=_\alpha})^\infty.$$

Observe that d_α extends to a pseudometric $d^\infty_\alpha : \Lambda^\infty \times \Lambda^\infty \to [0, 1]$ given by the same formula as in (5.17). Then $M =_\alpha N$ in the sense of Definition 5.17 if and only if $d^\infty_\alpha(M, N) = 0$. Hence we obtain a metric on $\Lambda^\infty/{=_\alpha}$, also denoted by d^∞_α.

Theorem 5.19. *Let $\mathcal{V}$ be uncountable. Then we have that $(\Lambda^\infty/{=_\alpha}, d^\infty_\alpha)$ 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 $(\Lambda^\infty/{=_\alpha}, d^\infty_\alpha)$ 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 $(\Lambda^\infty/{=_\alpha}, d^\infty_\alpha)$ is not complete.

Example 5.20. Assume that $\mathcal{V}$ is countable, say $\mathcal{V} = \{x_0, x_1, \dots\}$ and consider the sequence $([\lambda x_n.x_n(x_0(x_1(\dots x_{n-1})))]_\alpha)_{n\geq 1}$ in $\Lambda^\infty/{=_\alpha}$. This is a Cauchy sequence with respect to d^∞_α, but has no limit in $\Lambda^\infty/{=_\alpha}$. Indeed, assume towards a contradiction that the limit ℓ exists. On one hand we can prove that $\mathsf{fv}(\ell) = \mathcal{V}$, on the other hand ℓ should be of the form $[\lambda u.u(x_0(x_1 \dots))]_\alpha$ for some variable u. But this contradicts the fact that u is free in ℓ.

The example shows that with a countable $\mathcal{V}$ the two possible definitions of infinitary λ-terms up to α-equivalence do not coincide. In other words, metric completion and quotienting by α do not commute. We find the following formulation of this phenomenon useful as well.

Remark 5.21. The canonical map

$$[-]_\alpha : \Lambda^\infty \to (\Lambda/{=_\alpha})^\infty$$

taking the α-equivalence class of an infinitary λ-term is not surjective.

These problems of α-equivalence in the presence of countably many variables disappear if we consider the set $\Lambda^\infty_{\mathrm{ffv}}$ of infinitary terms with finitely many free variables.

Notation 5.22 (Restriction to finitely many free variables)**.** Let $\Lambda^\infty_{\mathrm{ffv}}$ denote the set $\{M \in \Lambda^\infty \mid \mathsf{fv}(M) \text{ is finite }\}$.

Remark 5.23. Note that $\Lambda^\infty_{\mathrm{ffv}}$ is different from the set $(\Lambda^\infty)_{\mathsf{fs}} = \Lambda^\infty_{\mathsf{fs}}$, defined in (4.2), of λ-terms with finitely many variables, bound or free. We do have that $\Lambda^\infty_{\mathsf{fs}} \subseteq \Lambda^\infty_{\mathrm{ffv}}$, but the inclusion is strict. Indeed, the following two terms belong to $\Lambda^\infty_{\mathrm{ffv}}$ but not to $\Lambda^\infty_{\mathsf{fs}}$:

- $\mathsf{ogre} \equiv \lambda x_1.\lambda x_2.\lambda x_3 \ldots$
- $\mathsf{infbv} \equiv \lambda x_0.\lambda x_1.x_0 x_1(\lambda x_2.x_0 x_1 x_2(\lambda x_3.x_0 x_1 x_2 x_3(\ldots)))$.

For ogre we can find $N \in \Lambda^\infty_{\mathsf{fs}}$ such that $N =_\alpha \mathsf{ogre}$, e.g. $N \equiv \lambda x_1.\lambda x_1.\lambda x_1 \ldots$. For infbv this is not possible.

Remark 5.24. The equivalence relation $=_\alpha$ of Definition 5.17 restricts to $\Lambda^\infty_{\mathrm{ffv}}$, but not to $\Lambda^\infty_{\mathsf{fs}}$, as shown by $\mathsf{ogre} =_\alpha \lambda x_1.\lambda x_1.\lambda x_1 \ldots$ where only the latter term is in $\Lambda^\infty_{\mathsf{fs}}$.

For all $M, N \in \Lambda^\infty_{\mathrm{ffv}}$ we have that $M =_\alpha N$ implies $\pi \cdot M =_\alpha \pi \cdot N$. Hence we can equip $\Lambda^\infty_{\mathrm{ffv}}/{=_\alpha}$ with a $\mathfrak{S}(\mathcal{V})$-action given by $\pi \cdot [M]_\alpha = [\pi \cdot M]_\alpha$ and we can easily check that $(\Lambda^\infty_{\mathrm{ffv}}/{=_\alpha},\ \cdot)$ is a nominal set. Indeed, $[M]_\alpha \in \Lambda^\infty_{\mathrm{ffv}}/{=_\alpha}$ 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. Using the universal property of the metric completion, this function can be extended to a uniformly continuous map on $(\Lambda/{=_\alpha})^\infty$:

$$\begin{tikzcd}
\Lambda/{=_\alpha} \arrow[rr] \arrow[dr, swap, "\pi \cdot (-)"] & & (\Lambda/{=_\alpha})^\infty \arrow[dd, "\pi \cdot (-)"] \\
 & \Lambda/{=_\alpha} \arrow[dr] & \\
 & & (\Lambda/{=_\alpha})^\infty
\end{tikzcd}$$

Thus we have a nominal set $((\Lambda/{=_\alpha})^\infty_{\mathsf{fs}}, \cdot)$. In [KPSdV12] we showed that this nominal set is isomorphic to $(\Lambda^\infty_{\mathrm{ffv}}/{=_\alpha},\ \cdot)$ and to the carrier of the final coalgebra of the Nom-functor L_α. Hence, for each α-equivalence class of infinitary terms with finitely many variables we can find a representative. This means that we have a surjective map

$$[-]_\alpha : \Lambda^\infty_{\mathrm{ffv}} \to (\Lambda/{=_\alpha})^\infty_{\mathsf{fs}} \tag{5.18}$$

whose kernel is the α-equivalence relation on $\Lambda^\infty_{\mathrm{ffv}}$.

Remark 5.25. On the other hand, the restriction $\Lambda^\infty_{\mathsf{fs}} \to (\Lambda/{=_\alpha})^\infty_{\mathsf{fs}}$ to $\Lambda^\infty_{\mathsf{fs}}$ of (5.18) is not surjective. For example the equivalence class of $\mathsf{infbv} \in \Lambda^\infty_{\mathrm{ffv}}$ is obtained as the limit of a finitely supported (actually emptily supported) sequence:

$$
\begin{aligned}
&[\lambda x_0.*]_\alpha\\
&[\lambda x_0 x_1.*]_\alpha\\
&[\lambda x_0 x_1.x_0*]_\alpha\\
&\dots
\end{aligned}
$$

and thus belongs to $(\Lambda/{=_\alpha})^\infty_{\mathsf{fs}}$. However, for the term infbv there is no $N \in \Lambda^\infty_{\mathsf{fs}}$ such that $N =_\alpha \mathsf{infbv}$.

To summarise, we have seen the following classes of λ-terms

$$
\begin{tikzcd}
\Lambda \arrow[r, tail] \arrow[dd, two heads] & \Lambda^\infty \arrow[d, two heads] & \Lambda^\infty_{\mathrm{ffv}} \arrow[l, hook'] \arrow[d, two heads] & \Lambda^\infty_{\mathsf{fs}} \arrow[l, hook'] \arrow[ddl] \\
 & \Lambda^\infty/{=_\alpha} \arrow[d, tail] & \Lambda^\infty_{\mathrm{ffv}}/{=_\alpha} \arrow[d, tail, two heads, "\cong"'] & \\
\Lambda/{=_\alpha} \arrow[r, tail] & (\Lambda/{=_\alpha})^\infty & (\Lambda/{=_\alpha})^\infty_{\mathsf{fs}} \arrow[l, hook'] &
\end{tikzcd}
\tag{5.19}
$$

with inclusions, injections and surjections as indicated by the arrows. Λ and $\Lambda/{=_\alpha}$ are initial algebras for L and L_α, respectively. Similarly $\Lambda^\infty_{\mathsf{fs}}$ and $(\Lambda/{=_\alpha})^\infty_{\mathsf{fs}}$ are final coalgebras for the same functors. But the coinductive situation is complicated by the fact that the canonical map $\Lambda^\infty_{\mathsf{fs}} \to (\Lambda/{=_\alpha})^\infty_{\mathsf{fs}}$ is not onto, that is, infinitary λ-terms upto α-equivalence do not arise by quotienting, in Nom, the raw infinitary λ-terms (which do allow only finitely many bound variables). Instead of $\Lambda^\infty_{\mathsf{fs}}$ we need to work with $\Lambda^\infty_{\mathrm{ffv}}$, which is not a final coalgebra. That $\Lambda^\infty_{\mathrm{ffv}}$ can be given a semantic characterisation is one of the topics of the next subsection.

5.5. **Nominal Coalgebraic Data Types for Binding Signatures.** The aim of this section is to introduce nominal coalgebraic data types in their generality. We will generalise the previous subsection to arbitrary binding signatures and give semantic characterisations of all the vertices of (5.19).

In particular, at the end of the section, we will have explained the following diagram, which generalises (5.19) (eliding the middle row of (5.19) obtained by epi-mono factorisations).

$$
\begin{tikzcd}
T_\Sigma \arrow[r, tail] \arrow[d, two heads] & T^\infty_\Sigma \arrow[d] & (T^\infty_\Sigma)_{\mathrm{ffv}} \arrow[l, hook'] \arrow[d, two heads] & (T^\infty_\Sigma)_{\mathsf{fs}} \arrow[l, hook'] \arrow[dl] \\
T_\Sigma/{=_\alpha} \arrow[r, tail] & (T_\Sigma/{=_\alpha})^\infty & (T_\Sigma/{=_\alpha})^\infty_{\mathsf{fs}} \arrow[l, hook'] &
\end{tikzcd}
\tag{5.20}
$$

Recall from Section 5.2 the definition of T_Σ and $T_\Sigma/{=_\alpha}$. Also recall that T_Σ and $T_\Sigma/{=_\alpha}$ are initial algebras of the Nom-endofunctors F and F_α, (see respectively (5.11) and (5.10)). The transformation $T_\Sigma \twoheadrightarrow T_\Sigma/{=_\alpha}$ induced by a natural transformation $\mathsf{F} \to \mathsf{F}_\alpha$ is quotienting by α-equivalence. Next, we define T^∞_Σ.

Definition 5.26 (Infinitary terms coming from a binding signature)**.** Consider a binding signature (Σ, ar).

(1) The set of infinitary raw terms T^∞_Σ is defined coinductively by

$$\frac{}{x \in T^\infty_\Sigma}\ (x \in \mathcal{V}) \qquad\qquad \frac{t_1 \in T^\infty_\Sigma, \dots, t_k \in T^\infty_\Sigma}{\mathsf{op}(\langle\overline{x_1}\rangle.t_1, \dots, \langle\overline{x_k}\rangle.t_k) \in T^\infty_\Sigma}\ (\mathsf{ar}(\mathsf{op}) = n_1, \dots, n_k) \tag{5.21}$$

(2) Truncation of raw terms at depth n is defined by induction on n:

$$\begin{aligned} t^0 &= * \\ t^{n+1} &= \begin{cases} x & \textit{if } t = x \in \mathcal{V} \\ \mathsf{op}(\langle\overline{x_1}\rangle.t_1^n, \dots, \langle\overline{x_k}\rangle.t_k^n) & \textit{if } t = \mathsf{op}(\langle\overline{x_1}\rangle.t_1, \dots, \langle\overline{x_k}\rangle.t_k) \end{cases} \end{aligned} \tag{5.22}$$

where $\{*\}$ is a terminal object in Nom.

(3) To define α-equivalence, let t and s be two infinitary raw terms in T^∞_Σ. We say that $t =_\alpha s$ when the truncations at all depths are α-equivalent in the sense of Definition 5.12, that is, for all n we have $t^n =_\alpha s^n$.

(4) The sets $\mathsf{fv}(t)$ and $\mathsf{bv}(t)$ of free and bound variables of an infinitary raw term $t \in T^\infty_\Sigma$ are defined as follows.

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

Remark 5.27. T^∞_Σ is the final coalgebra for the Set-functor defined in (5.12).

Definition 5.28. We denote by $(T_\Sigma/{=_\alpha})^\infty$ the metric completion of $T_\Sigma/{=_\alpha}$ with respect to the metric d_α given by

$$d_\alpha([t]_\alpha, [s]_\alpha) = \inf\{2^{-n} \mid t^n =_\alpha s^n,\ n \in \mathbb{N}\}.$$

Notice that $(T_\Sigma/{=_\alpha})^\infty$ is equipped with a canonical permutation action, but it is not a nominal set, since not all elements are finitely supported (namely those terms with infinitely many free variables).

Remark 5.29. Going back to Section 2, and in the notation of (5.16), we have that $U\mathsf{I} \cong U\mathsf{F}^\omega 0 \cong T_\Sigma$ and $U\mathsf{I}_\alpha \cong U\mathsf{F}^\omega_\alpha 0 \cong T_\Sigma/{=_\alpha}$. The completions T^∞_Σ and $(T_\Sigma/{=_\alpha})^\infty$ then can be obtained as limits of ω^{op}-sequences:

$$\begin{tikzcd} 1 \arrow[d, tail, two heads] & U\mathsf{F}1 \arrow[l] \arrow[d, two heads, "{U[-]^{(1)}_\alpha}"] & U\mathsf{F}^2 1 \arrow[l] \arrow[d, two heads, "{U[-]^{(2)}_\alpha}"] & \cdots \arrow[l] & \lim U\mathsf{F}^n 1 \cong T^\infty_\Sigma \arrow[l] \arrow[d, "{[-]_\alpha}"] \\ 1 & U\mathsf{F}_\alpha 1 \arrow[l] & U\mathsf{F}^2_\alpha 1 \arrow[l] & \cdots \arrow[l] & \lim U\mathsf{F}^n_\alpha 1 \cong (T_\Sigma/{=_\alpha})^\infty \arrow[l] \end{tikzcd} \tag{5.23}$$

Moreover, the horizontal arrows in the diagram are precisely the truncations (with $1 = \{*\}$) and the kernels of the vertical arrows capture the α-equivalence of Definition 5.26. Since all maps $[-]^{(n)}_\alpha : \mathsf{F}^n 1 \to \mathsf{F}^n_\alpha 1$ are surjective, each element of $\mathsf{F}^n_\alpha 1$ can be expressed as the α-equivalence class of a raw term $t \in \mathsf{F}^n 1$. If $[t]_\alpha \in \mathsf{F}^n_\alpha 1$ we have that $\mathsf{supp}([t]_\alpha) = \mathsf{fv}(t)$.

The set $\lim U\mathsf{F}^n_\alpha 1 \cong (T_\Sigma/{=_\alpha})^\infty$ appears to be a natural domain for infinitary terms up to α-equivalence and it will reappear in Section 6.5. But it fails to have desirable properties. Indeed, it is not a final coalgebra of a Set-functor in any obvious way, nor is it a nominal set. Moreover the map $[-]_\alpha : T^\infty_\Sigma \to (T_\Sigma/{=_\alpha})^\infty$ is not surjective in general, as shown by Example 5.20. In the case of the infinitary λ-calculus we solved this issue by restricting our attention to terms with finitely many free variables [KPSdV12]. We do the same in the case of a general binding signature.

Recalling from (4.2) the notation $(-)_{\mathsf{fs}}$, we now obtain from the rightmost edge of (5.23)

$$(T^\infty_\Sigma)_{\mathsf{fs}} \quad \text{and} \quad (T_\Sigma/{=_\alpha})^\infty_{\mathsf{fs}}. \tag{5.24}$$

In the first case we restrict to finitely many variables and in the second case to finitely many free variables. More precisely, $(T_\Sigma/{=_\alpha})^\infty_{\mathsf{fs}}$ consists of limits of Cauchy sequences $([t_n]_\alpha)_n$ of α-equivalence classes of finite terms which altogether have only finitely many free variables, that is, $\bigcup\limits_n \mathsf{fv}(t_n)$ is finite. Similarly $(T^\infty_\Sigma)_{\mathsf{fs}}$ consists of limits of Cauchy sequences $(t_n)_n$ of finite terms which altogether have only finitely many variables.

Remark 5.30. According to Theorem 5.5, we have that $(T^\infty_\Sigma)_{\mathsf{fs}}$, respectively $(T_\Sigma/{=_\alpha})^\infty_{\mathsf{fs}}$, can be taken to be the final F-coalgebra T, respectively the final F_α-coalgebra T_α.

Just as in the case of the initial chains (see (5.16)), the natural transformation $q : \mathsf{F} \to \mathsf{F}_\alpha$ defined by (5.15) induces a unique map $\mathsf{T} \to \mathsf{T}_\alpha$ from the final F-coalgebra to the final F_α-coalgebra. However, unlike in the initial algebra situation (5.16), the induced map $[-]_\alpha$ is not surjective in general, see Remark 5.25. The aim of the remainder of this section is to prove that, nevertheless, T_α is the quotient by α-equivalence of the infinitary terms with finitely many free variables for which we introduce the following notation.

Definition 5.31. Denote by

$$(T^\infty_\Sigma)_{\mathrm{ffv}} \tag{5.25}$$

the set of elements of T^∞_Σ having only finitely many free variables.

Notice that $(T^\infty_\Sigma)_{\mathrm{ffv}}$ consists of limits of Cauchy sequences $(t_n)_n$ of finite terms which altogether have only finitely many free variables. The definition above relies on the syntactic notion of free variable. Proposition 5.33 shows that a semantic definition is possible. To this end, we first give a semantic definition of the inclusion $(T_\Sigma/{=_\alpha})^\infty_{\mathsf{fs}} \hookrightarrow (T_\Sigma/{=_\alpha})^\infty$ as the map ι_α arising in (5.26).

Remark 5.32. As a final F-coalgebra, T induces a cone over the sequence $(\mathsf{F}^n 1)_{n<\omega}$, dualising (5.16).

This induces a unique map $\iota : U\mathsf{T} \to \lim U\mathsf{F}^n 1$. In the same way, by finality of T_α, one obtains $\iota_\alpha : U\mathsf{T}_\alpha \to \lim U\mathsf{F}^n_\alpha 1$.

$$\begin{tikzcd}
1 \arrow[d] & U\mathsf{F}1 \arrow[l] \arrow[d, two heads] & U\mathsf{F}^2 1 \arrow[l] \arrow[d, two heads] & \cdots \arrow[l] & \lim U\mathsf{F}^n 1 \arrow[l] \arrow[d, "{[-]_\alpha}"'] \arrow[r, phantom, "(*)" description, shift right=6] & U\mathsf{T} \arrow[l, "\iota"'] \arrow[d, "{U[-]_\alpha}"] \\
1 & U\mathsf{F}_\alpha 1 \arrow[l] & U\mathsf{F}^2_\alpha 1 \arrow[l] & \cdots \arrow[l] & \lim U\mathsf{F}^n_\alpha 1 \arrow[l] & U\mathsf{T}_\alpha \arrow[l, "\iota_\alpha"]
\end{tikzcd} \tag{5.26}$$

Proposition 5.33. *The set $(T^\infty_\Sigma)_{\text{ffv}}$ of infinitary raw terms with finitely many free variables is the pullback of the maps $\iota_\alpha : U\mathsf{T}_\alpha \to \lim U\mathsf{F}^n_\alpha 1$ and $[-]_\alpha : \lim U\mathsf{F}^n 1 \to \lim U\mathsf{F}^n_\alpha 1$.*

$$\begin{tikzcd}
\lim U\mathsf{F}^n 1 \arrow[d, "{[-]_\alpha}"'] & (T^\infty_\Sigma)_{\text{ffv}} \arrow[l, hook'] \arrow[d, "{[-]_\alpha}"] \arrow[dl, phantom, "\llcorner", very near start] \\
\lim U\mathsf{F}^n_\alpha 1 & U\mathsf{T}_\alpha \arrow[l, hook', "\iota_\alpha"']
\end{tikzcd} \tag{5.27}$$

Proof. First we have to define the map $[-]_\alpha : (T^\infty_\Sigma)_{\text{ffv}} \to U\mathsf{T}_\alpha$. Given $t \in (T^\infty_\Sigma)_{\text{ffv}}$, notice that we can construct a finitely supported sequence in the sets $\mathsf{F}^n_\alpha 1$ by applying the map $[-]^{(n)}_\alpha$ to the truncations at depth n of t. Indeed, each $[t^n]^{(n)}_\alpha$ is supported by the finite set of free variables in t. We define $[t]_\alpha \in \mathsf{T}_\alpha$ to be the unique element whose projection in $\mathsf{F}^n_\alpha 1$ is exactly $[t^n]^{(n)}_\alpha$ for all n. We can easily check that the square (5.27) commutes.

Consider a pair (t, s) with $t \in \lim U\mathsf{F}^n 1 \simeq T^\infty_\Sigma$ and $s \in U\mathsf{T}_\alpha$ such that $\iota_\alpha(s) = [t]_\alpha$. This implies that $[t^n]^{(n)}_\alpha$ is equal to the projection of s into $\mathsf{F}^n_\alpha 1$, and thus the free variables of each truncation t^n are contained in the finite set that supports s. Therefore t has finitely many free variables, that is, $t \in (T^\infty_\Sigma)_{\text{ffv}}$ and $s = [t]_\alpha$. So $(T^\infty_\Sigma)_{\text{ffv}}$ is indeed a pullback. □

To summarise, we are now ready to give a semantic version of diagram (5.20) – which we set about to prove at the beginning of the section:

$$\begin{tikzcd}
\mathsf{I} \arrow[r, tail] \arrow[d, two heads] & \lim U\mathsf{F}^n 1 \arrow[d] & P \arrow[l, hook'] \arrow[d, two heads] \arrow[dl, phantom, "\llcorner", very near start] & \mathsf{T} \arrow[l, hook'] \arrow[dl] \\
\mathsf{I}_\alpha \arrow[r, tail] & \lim U\mathsf{F}^n_\alpha 1 & \mathsf{T}_\alpha \arrow[l, hook'] &
\end{tikzcd} \tag{5.28}$$

with I and I_α the initial algebras as well as T and T_α the final (or terminal) coalgebras of F and F_α, and P being a pullback in Set. To improve readability we omitted writing the forgetful functor $U : \mathsf{Nom} \to \mathsf{Set}$ in (5.28).

The map $\mathsf{T} \hookrightarrow P$ above is obtained using the universal property of P and the commutativity of the square $(*)$ in (5.26) and corresponds to the inclusion $(T^\infty_\Sigma)_{\mathsf{fs}} \hookrightarrow (T^\infty_\Sigma)_{\text{ffv}}$, (see Proposition 5.33).

Theorem 5.34. *Completing T_Σ by Cauchy sequences $(t_n)_n$ such that $\bigcup_n \mathsf{fv}(t_n)$ is finite and quotienting by α-equivalence commute. This means that*

- *the two equivalent diagrams below commute*

$$\begin{tikzcd}
U\mathsf{I} \arrow[r] \arrow[d, two heads, "{[-]_\alpha}"'] & P \arrow[d, two heads, "{[-]_\alpha}"] \\
U\mathsf{I}_\alpha \arrow[r] & U\mathsf{T}_\alpha
\end{tikzcd}
\qquad
\begin{tikzcd}
T_\Sigma \arrow[r] \arrow[d, two heads, "{[-]_\alpha}"'] & (T^\infty_\Sigma)_{\text{ffv}} \arrow[d, two heads, "{[-]_\alpha}"] \\
T_\Sigma/{=_\alpha} \arrow[r] & (T_\Sigma/{=_\alpha})^\infty_{\mathsf{fs}}
\end{tikzcd} \tag{5.29}$$

semantic version *syntactic version;*

- *the map $[-]_\alpha : P \to U\mathsf{T}_\alpha$ is surjective, or equivalently each element in $(T_\Sigma/{=_\alpha})^\infty_{\mathsf{fs}}$ can be represented as an equivalence class of an infinitary raw term with finitely many free variables.*

Proof. The first bullet is easier to prove. We show that the semantic version of (5.29) commutes, using the commutativity of

$$\begin{tikzcd}
\mathsf{I} \arrow[r] \arrow[d, two heads, "{[-]_\alpha}"'] & \mathsf{T} \arrow[d, "{[-]_\alpha}"] \\
\mathsf{I}_\alpha \arrow[r] & \mathsf{T}_\alpha
\end{tikzcd} \tag{5.30}$$

The argument uses the finality of T_α and the fact that all the four arrows in (5.30) are F_α-coalgebra morphisms.

By pasting the right-hand triangle of (5.28), we obtain the commutativity of the desired diagram:

$$\begin{tikzcd}
U\mathsf{I} \arrow[r] \arrow[d, two heads, "{[-]_\alpha}"'] & U\mathsf{T} \arrow[r, hook] \arrow[d, "{[-]_\alpha}"'] & P \arrow[dl] \\
U\mathsf{I}_\alpha \arrow[r] & U\mathsf{T}_\alpha &
\end{tikzcd} \tag{5.31}$$

The second part of the theorem, stating that the map $P \to U\mathsf{T}_\alpha$ in

$$\begin{tikzcd}
1 & U\mathsf{F}1 \arrow[l] & U\mathsf{F}^2 1 \arrow[l] & \cdots \arrow[l] & \lim U\mathsf{F}^n 1 \arrow[l] & P \arrow[l] \\
1 \arrow[from=u] & U\mathsf{F}_\alpha 1 \arrow[l] \arrow[from=u, two heads] & U\mathsf{F}_\alpha^2 1 \arrow[l] \arrow[from=u, two heads] & \cdots \arrow[l] & \lim U\mathsf{F}_\alpha^n 1 \arrow[l] \arrow[from=u] & U\mathsf{T}_\alpha \arrow[l, "\iota_\alpha"] \arrow[from=u]
\end{tikzcd} \tag{5.32}$$

is surjective can be proved by going back to the syntax as in [KPSdV12], just that this time, due to generalising from λ-calculus to binding signatures, the notation becomes even heavier and quite unpleasant. Therefore, we will give a semantic proof in the next section, so that surjectivity becomes a consequence of Theorem 5.72. This, in turn, is a consequence of a more general result Theorem 5.46 about limits of sequences in nominal sets and will be proved in the next section. □

We state explicitly the most important consequence of the theorem as a corollary.

Corollary 5.35 (Nominal coalgebraic data types)**.** *The nominal set* $((T_\Sigma^\infty)_{\mathrm{ffv}}/{=_\alpha}, \cdot)$ *of α-equivalence classes of infinitary terms with finitely many free variables is the final coalgebra for the functor* $\mathsf{F}_\alpha : \mathsf{Nom} \to \mathsf{Nom}$ *corresponding to a binding signature* (Σ, ar)*:*

$$\mathsf{F}_\alpha\ X = \mathcal{V} + \coprod_{\substack{\mathsf{op} \in \Sigma \\ \mathsf{ar}(\mathsf{op}) = n_1, \ldots, n_k}} [\mathcal{V}]^{n_1} X \times \ldots \times [\mathcal{V}]^{n_k} X. \tag{5.33}$$

Remark 5.36. Let us point out that Diagram (5.28) does not actually depend on the functors $\mathsf{F}, \mathsf{F}_\alpha$ arising from a binding signature and makes sense for any pair of Nom-endofunctors $\mathsf{F}, \mathsf{F}_\alpha$ and any component-wise surjective natural transformation $\mathsf{F} \to \mathsf{F}_\alpha$ subject to some natural conditions (which are satisfied by functors that do arise from binding signatures), namely that F preserves surjections and that both $\mathsf{F}, \mathsf{F}_\alpha$ have initial algebras and final coalgebras. Finally, we want the surjectivity of $P \to \mathsf{T}_\alpha$ and we give a semantic analysis of it in the next subsection.

5.6. **Presenting Limits in Nominal Sets.** The motivation of this section is to give a semantic proof of the fact that the final T_α-coalgebra is the quotient by α-equivalence of $(T^\infty_\Sigma)_{\mathrm{ffv}}$, the infinitary terms with finitely many free variables. We move this proof into a separate subsection because the semantic analysis depends on certain facts on limits in nominal sets and leads to a novel notion of 'bound variable relative to a map' which may be of independent interest.

5.6.1. *Bound variables, safe maps, and safe squares.*

In nominal sets, the syntax dependent notion of free variable is replaced by the semantic concept of minimal finite support. What about bound variables? Consider $[-]_\alpha : \Lambda \to \Lambda/=_\alpha$ and $x(\lambda y.y) \in \Lambda$. Then the bound variables of $x\lambda y.y$ can be computed as $\mathsf{supp}(x\lambda y.y) \setminus \mathsf{supp}([x\lambda y.y]_\alpha) = \{x, y\} \setminus \{x\} = \{y\}$. Of course, this calculation depends on being able to assume that the bound variables and free variables of $x\lambda y.y$ do not overlap, or, in the terminology of [KPSdV12], that $x\lambda y.y$ is α-safe. The next definition gives a semantic formulation of an element being safe with respect to a map, which now does not need to be a quotient by α-equivalence.

Definition 5.37 (Safe element)**.** Let $f : X \to Y$ be an equivariant function. We call $u \in X$ f-safe when

$$|\mathsf{supp}(u)| = \max\{\ |\mathsf{supp}(v)| \mid v \in f^{-1}(f(u))\}. \tag{5.34}$$

The maximum in the right-hand side of 5.34 does not always exist, see Example 5.44.

Example 5.38 ($[-]_\alpha$-Safe Terms)**.** We consider the equivariant map $[-]_\alpha : \Lambda \to \Lambda/=_\alpha$. Then M is an $[-]_\alpha$-safe term if it has a maximal number of variables among all the representatives of its α-equivalence class. The terms $x(\lambda y.y)$ and $\lambda x.x(\lambda y.y)$ are $[-]_\alpha$-safe but $x(\lambda x.x)$ and $\lambda x.x(\lambda x.x)$ are not.

Remark 5.39 (α-safe Term)**.** A term $M \in \Lambda$ is α-safe in the sense of [KPSdV12, Definition 19] if and only if M is $[-]_\alpha$-safe in the sense of Definition 5.37. 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 $x \neq y$.

If a λ-term M is $[-]_\alpha$-safe then the set $\mathsf{bv}(M)$ of bound variables of M is equal to $\mathsf{var}(M) \setminus \mathsf{fv}(M) = \mathsf{supp}(M) \setminus \mathsf{supp}([M]_\alpha)$. This motivates the following notation.

Notation 5.40. If $f : X \to Y$ is an equivariant map then we define

$$\mathsf{bv}_f(u) = \mathsf{supp}(u) \setminus \mathsf{supp}(f(u)).$$

When no confusion may arise, we omit the subscript and write $\mathsf{bv}(u)$ instead of $\mathsf{bv}_f(u)$.

Remark 5.41. Let $f : X \to Y$ be equivariant and $u \in X$. Then $\mathsf{supp}(f(u)) \subseteq \mathsf{supp}(u)$ and $|\mathsf{bv}_f(u)| = |\mathsf{supp}(u)| - |\mathsf{supp}(f(u))|$.

Lemma 5.42. *Let $f : X \to Y$ be an equivariant map. Then u is f-safe if and only if*

$$|\mathsf{bv}_f(u)| = \max\{\ |\mathsf{bv}_f(v)| \mid v \in f^{-1}(f(u))\}. \tag{5.35}$$

Proof. Assume u is f-safe and consider $v \in Y$ such that $f(v) = f(u)$. Then

$$|\mathsf{bv}_f(u)| = |\mathsf{supp}(u)| - |\mathsf{supp}(f(u))| \geq |\mathsf{supp}(v)| - |\mathsf{supp}(f(v))| = |\mathsf{bv}_f(v)|.$$

The converse is similar. □

Definition 5.43 (Safe map)**.** Let $f : X \to Y$ be an equivariant map in Nom. We call f *safe* when for all $v \in Y$ there exists an f-safe $u \in X$ such that $f(u) = v$.

Example 5.44. The map $[-]_\alpha : \Lambda \to \Lambda/{=_\alpha}$ is safe [KPSdV12, Lemma 20]. But the map $! : \mathcal{P}_{\mathit{fin}}(\mathcal{V}) \to \{*\}$ is not.

A diagram such as (5.36) is a weak pullback if for all identified u and v, there is a z witnessing this fact, that is, if $f(u) = q(v)$ then there exists $z \in Z$ such that $u = p(z)$ and $v = g(z)$. In the following we will need a similar but weaker condition, which, intuitively, requires the existence of a witness only up to the renaming of bound variables.

Definition 5.45 (Safe square)**.** A square

$$\begin{tikzcd}
X \arrow[d, "f"'] & Z \arrow[l, "p"'] \arrow[d, "g"] \\
W & Y \arrow[l, "q"]
\end{tikzcd} \tag{5.36}$$

is a *safe square* when for all f-safe $u \in X$ and for all $v \in Y$ such that $f(u) = q(v)$ and $\mathsf{bv}_f(u)\#v$ there exists a g-safe $z \in Z$ such that $p(z) = u$ and $g(z) = v$.

5.6.2. *Representing limits in nominal sets.*

Consider two ω^{op}-chains in Nom and let $\lim X_n$, respectively $\lim Y_n$ denote their limits in Nom.

$$\begin{tikzcd}
X_0 \arrow[d, two heads, "f_0"] & X_1 \arrow[l, "p_1"'] \arrow[d, two heads, "f_1"] & X_2 \arrow[l, "p_2"'] \arrow[d, two heads, "f_2"] & \cdots \arrow[l] & \lim X_n \arrow[l] \arrow[d, dashed, "f"] \\
Y_0 & Y_1 \arrow[l, "q_1"] & Y_1 \arrow[l, "q_2"] & \cdots \arrow[l] & \lim Y_n \arrow[l]
\end{tikzcd} \tag{5.37}$$

By the universal property of the limits we obtain a map $f : \lim X_n \to \lim Y_n$. In the category of sets, we have the theorem that if all the squares are weak pullbacks and all the f_n are surjective, then f is surjective. But in our main example where the f_n quotient by α-equivalence, the squares are not weak pullbacks.

Recalling that $U : \mathsf{Nom} \to \mathsf{Set}$ denotes the forgetful functor, consider the limits of the two chains in Set

$$\begin{tikzcd}
UX_0 \arrow[d, two heads, "f_0"] & UX_1 \arrow[l] \arrow[d, two heads, "f_1"] & UX_2 \arrow[l] \arrow[d, two heads, "f_2"] & \cdots \arrow[l] & \lim UX_n \arrow[l] \arrow[d, dashed, "g"] \arrow[dr, phantom, "(*)"] & U \lim X_n \arrow[l, dashed, "a"'] \arrow[d, "Uf"] \\
UY_0 & UY_1 \arrow[l] & UY_1 \arrow[l] & \cdots \arrow[l] & \lim UY_n \arrow[l] & U \lim Y_n \arrow[l, dashed, "b"]
\end{tikzcd} \tag{5.38}$$

By the universal property of limits, there exist unique maps $a : U \lim X_n \to \lim UX_n$ and $b : U \lim Y_n \to \lim UY_n$ making the square $(*)$ commutative. Again, the map $g : \lim UX_n \to \lim UY_n$ induced in the limit may not be surjective in general, see Example 5.20.

However we can prove the following general result:

Theorem 5.46. *Assume that diagram* (5.37) *is such that for all n the square*

$$\begin{tikzcd}
X_n \arrow[d, two heads, "f_n"] & X_{n+1} \arrow[l, "p_n"'] \arrow[d, two heads, "f_{n+1}"] \\
Y_n & Y_{n+1} \arrow[l, "q_n"]
\end{tikzcd} \tag{5.39}$$

is safe and f_n is a safe map. Let P denote the pullback

$$\begin{tikzcd}
\lim UX_n \arrow[d, "g"'] & P \arrow[l, "r"'] \arrow[d, "h"] \arrow[dl, phantom, very near start, "\llcorner"] \\
\lim UY_n & U\lim Y_n \arrow[l, "b"]
\end{tikzcd} \tag{5.40}$$

Then $h : P \to U \lim Y_n$ is a surjection.

Proof. We start with $v \in \lim Y_n$. Consider the projections $v_n \in Y_n$ of v obtained via the projections of the limiting cone. Since $q_n(v_{n+1}) = v_n$ and q_n are equivariant maps we have the following inclusions of finite sets

$$\mathsf{supp}(v_0) \subseteq \mathsf{supp}(v_1) \subseteq \ldots \subseteq \mathsf{supp}(v).$$

Thus the sequence stabilises eventually, that is, there exists n such that $\mathsf{supp}(v_n) = \mathsf{supp}(v_{n+k})$ for all natural numbers k. Moreover since $v = (v_0, v_1, \ldots)$ we have that for all $k \geq 0$

$$\mathsf{supp}(v_{n+k}) = \mathsf{supp}(v).$$

We will now construct $u = (u_1, u_2, \ldots)$ in $\lim UX_n$ such that $g(u) = b(v)$. First let $u_n \in X_n$ be an f_n-safe element such that $f_n(u_n) = v_n$. For each $k \geq 1$, we define an f_{n+k}-safe $u_{n+k} \in X_{n+k}$ such that $p_{n+k}(u_{n+k}) = u_{n+k-1}$ and $f_{n+k}(u_{n+k}) = v_{n+k}$.

The proof is by induction on k. Since $\mathsf{supp}(v_{n+1}) = \mathsf{supp}(v_n)$, we have that $(\mathsf{supp}(u_n) \setminus \mathsf{supp}(v_n)) \cap \mathsf{supp}(v_{n+1}) = \emptyset$. Since all the squares (5.39) are safe squares, there exists an f_{n+1}-safe element $u_{n+1} \in X_{n+1}$ in the preimage of v_{n+1} such that $p_{n+1}(u_{n+1}) = u_n$. This shows that the claim is true for $k = 1$.

For the inductive step $k \to k+1$, notice that $(\mathsf{supp}(u_{n+k}) \setminus \mathsf{supp}(v_{n+k})) \cap \mathsf{supp}(v_{n+k+1}) = \emptyset$ and that u_{n+k} is f_{n+k}-safe. Using the safe square property, there exists f_{n+k+1}-safe $u_{n+k+1} \in X_{n+k+1}$ with the desired properties.

For $m < n$ define $u_m = p_m p_{m+1} \ldots p_n(u_n)$. Now observe that $(u_0, u_1, \ldots)$ is an element of $\lim UX_n$ whose image in $\lim UY_n$ is $(v_0, v_1, \ldots)$. This means that

$$g((u_0, u_1, \ldots)) = b(v).$$

Since P is the pullback of g and b, there exists $w \in P$ such that $r(w) = (u_0, u_1, \ldots)$ and $h(w) = v$. Thus $h : P \to U \lim Y_n$ is surjective. □

Going back to Theorem 5.34 and looking at (5.32), we find that so far we established the following corollary of Theorem 5.34.

Corollary 5.47. *Let $\mathsf{F}, \mathsf{F}_\alpha$ be endofunctors on* Nom *having final coalgebras, and let $\mathsf{F} \to \mathsf{F}_\alpha$ be a component-wise surjective natural transformation, and P be a pullback as in* (5.20). *Moreover assume that* F *preserves surjections. If*

- *the induced maps $[-]_\alpha^{(n)} : \mathsf{F}^n 1 \to \mathsf{F}_\alpha^n 1$ are safe*

- *and the squares*

$$\begin{tikzcd}
\mathsf{F}^n 1 \arrow[d, two heads, "{[-]_\alpha^{(n)}}"'] & \mathsf{F}^{n+1} 1 \arrow[l] \arrow[d, two heads, "{[-]_\alpha^{(n+1)}}"] \\
\mathsf{F}_\alpha^n 1 & \mathsf{F}_\alpha^{n+1} 1 \arrow[l]
\end{tikzcd} \tag{5.41}$$

are safe squares,

then $P \to U\mathsf{T}_\alpha$ is onto.

In the following we will prove the second part of Theorem 5.34 using Corollary 5.47, that is, by establishing the two bullet points above. Let us briefly outline the structure of that argument. Recall that the maps $[-]_\alpha^{(n)}$ are defined inductively as follows: $[-]_\alpha^{(0)} = id_1$ and $[-]_\alpha^{(n+1)} = q_{\mathsf{F}_\alpha^n 1} \circ \mathsf{F}([-]_\alpha^{(n)})$. Also recall that the natural transformation $q : \mathsf{F} \to \mathsf{F}_\alpha$ is defined inductively depending on the structure of F and F_α using the rules (5.15). Therefore, the argument will proceed by induction on n and on the structure of F. For this, we need some structural closure properties for safe maps. To this end, we also study a special case of safe maps, namely maps with orbit-finite fibres. This will help us to prove the first bullet point of Corollary 5.47. For the second bullet we need a detailed study of safe squares. This is eventually done in Section 5.6.5.

5.6.3. *Some properties of safe elements and safe maps.*

Lemma 5.48. *If $u \in X$ is f-safe and π is an arbitrary finite permutation then $\pi \cdot u$ is f-safe.*

Proof. Suppose $u \in X$ is f-safe. Consider $v \in X$ such that $v \in f^{-1}(f(\pi \cdot u))$. Then $\pi^{-1} \cdot v \in f^{-1}(f(u))$ because f is equivariant. Using Remark 4.6 we conclude

$$|\mathsf{supp}(\pi \cdot u)| = |\mathsf{supp}(u)| \geq |\mathsf{supp}(\pi^{-1} \cdot v)| = |\mathsf{supp}(v)|.$$

□

Lemma 5.49. *Assume $f : X \to Y$ is safe. Consider $v \in Y$ and let S be an arbitrary finite set of names. Then there exists an f-safe $u \in f^{-1}(v)$ such that $\mathsf{bv}(u) \cap S = \emptyset$.*

Proof. Since f is safe, there exists an f-safe $u \in f^{-1}(v)$. Let $T = S \cap \mathsf{bv}(u)$. If T is empty we are done. If not, let T' be a finite set of names fresh for u, v, S which has the same number of elements as T. Let π denote a finite permutation that swaps the elements of T and T' and fixes the remaining names in $\mathcal{V}$. Then $\pi \cdot x = x$ for all $x \in \mathsf{supp}(v)$, thus $\pi \cdot u \in f^{-1}(v)$. By Lemma 5.48, $\pi \cdot u$ is f-safe. Moreover, by Remark 4.6 we have $\mathsf{supp}(\pi \cdot u) = \pi \cdot \mathsf{supp}(u)$ and thus $\mathsf{bv}(\pi \cdot u) \cap S = \emptyset$. □

Lemma 5.50. *Assume $f_1 : X_1 \to Y_1$ and $f_2 : X_2 \to Y_2$ are safe maps. Then (u_1, u_2) is $(f_1 \times f_2)$-safe if and only if the following hold*

$$\begin{aligned} &u_i \textit{ is } f_i\textit{-safe for } i = 1, 2 \\ &\mathsf{bv}(u_1) \# u_2 \\ &\mathsf{bv}(u_2) \# u_1. \end{aligned} \tag{5.42}$$

Proof. Assume (u_1, u_2) satisfy (5.42) and let v_i denote $f_i(u_i)$ for $i = 1, 2$. We will show that $(u_1, u_2) \in (f_1 \times f_2)^{-1}(v_1, v_2)$ is $(f_1 \times f_2)$-safe. First observe that

$$\begin{aligned}\mathsf{bv}(u_1, u_2) &= (\mathsf{supp}(u_1) \cup \mathsf{supp}(u_2)) \setminus (\mathsf{supp}(v_1) \cup \mathsf{supp}(v_2)) \\ &= (\mathsf{supp}(u_1) \setminus \mathsf{supp}(v_1)) \setminus \mathsf{supp}(v_2) \cup \\ &\cup \ (\mathsf{supp}(u_2) \setminus \mathsf{supp}(v_2)) \setminus \mathsf{supp}(v_1) \\ &= \mathsf{bv}(u_1) \uplus \mathsf{bv}(u_2).\end{aligned} \tag{5.43}$$

The last equality holds by (5.42). Therefore

$$|\mathsf{bv}(u_1, u_2)| = |\mathsf{bv}(u_1)| + |\mathsf{bv}(u_2)|. \tag{5.44}$$

In order to show that (u_1, u_2) is indeed $(f_1 \times f_2)$-safe consider $(u_1', u_2') \in (f_1 \times f_2)^{-1}(v_1, v_2)$. We have the inequalities

$$\begin{aligned}|\mathsf{supp}(u_1', u_2') \setminus \mathsf{supp}(v_1, v_2)| &= |(\mathsf{supp}(u_1') \cup \mathsf{supp}(u_2')) \setminus (\mathsf{supp}(v_1) \cup \mathsf{supp}(v_2))| \\ &\leq |(\mathsf{supp}(u_1') \setminus (\mathsf{supp}(v_1) \cup \mathsf{supp}(v_2)))| + \\ &+ \ |(\mathsf{supp}(u_2') \setminus (\mathsf{supp}(v_1) \cup \mathsf{supp}(v_2)))| \\ &\leq |(\mathsf{supp}(u_1') \setminus \mathsf{supp}(v_1)| + |(\mathsf{supp}(u_2') \setminus \mathsf{supp}(v_2)| \\ &\leq |\mathsf{bv}(u_1)| + |\mathsf{bv}(u_2)|.\end{aligned} \tag{5.45}$$

The last inequality holds because u_1 and u_2 are f_1-safe, respectively, f_2-safe. Thus, using (5.44), we can conclude that

$$|\mathsf{supp}(u_1', u_2') \setminus \mathsf{supp}(v_1, v_2)| \leq |\mathsf{bv}(u_1, u_2)|.$$

Conversely, assume that $(u_1', u_2') \in (f_1 \times f_2)^{-1}(v_1, v_2)$ is $(f_1 \times f_2)$-safe. By Lemma 5.49, there exists $u_1 \in f_1^{-1}(v_1)$ an f_1-safe element such that

$$\mathsf{bv}(u_1) \cap \mathsf{supp}(u_2) = \emptyset. \tag{5.46}$$

Similarly, there exists $u_2 \in f_2^{-1}(v_2)$ an f_2-safe element such that

$$\mathsf{bv}(u_2) \cap \mathsf{supp}(u_1) = \emptyset. \tag{5.47}$$

From (5.46) and (5.47) we can derive (5.42). By the first part of the proof we know that (u_1, u_2) is $(f_1 \times f_2)$-safe and that the inequalities of (5.45) hold. Since (u_1', u_2') is also $(f_1 \times f_2)$-safe we know that all the inequalities of (5.45) are equalities. For the last inequality of (5.45), this implies that each u_i' is f_i-safe. For the second inequality of (5.45), this implies that $\mathsf{bv}(u_1')\#f_2(u_2)$ and $\mathsf{bv}(u_2)\#f_1(u_1)$. Hence, by the fact that also the first inequality of (5.45) is actually an equality we have that $\mathsf{bv}(u_1')\#\mathsf{bv}(u_2')$. Since $\mathsf{supp}(u_1') = \mathsf{bv}(u_1') \cup \mathsf{supp}(f_1(u_1'))$ we conclude that $\mathsf{bv}(u_2')\#u_1'$. Similarly $\mathsf{bv}(u_1')\#u_2'$. □

Lemma 5.51. *Let $f : X \to Y$ and $g : Y \to W$ be equivariant maps. If u is gf-safe, then u is f-safe.*

Proof. Consider $v \in f^{-1}(f(u))$. Then $gf(v) = gf(u)$. Since u is gf-safe, we have that $|\mathsf{supp}(v)| \leq |\mathsf{supp}(u)|$. □

Unfortunately, safe maps are not closed under composition as the next example shows.

Example 5.52. Consider the set $\mathbb{N}$ of natural numbers as a nominal set equipped with the trivial action and let $\mathcal{P}_{\mathit{fin}}(\mathcal{V})$ denote the nominal set of all finite subsets of $\mathcal{V}$. Let $f : \mathcal{P}_{\mathit{fin}}(\mathcal{V}) \to \mathbb{N}$ denote the map which sends any finite set of names to its cardinal and let $g : \mathbb{N} \to \{*\}$ denote the unique map from $\mathbb{N}$ into the final nominal set $\{*\}$. Both f and g are equivariant and safe, but their composition $g \circ f$ is not safe.

Therefore we need a stronger notion of maps that still accommodates our examples but with better closure properties. This is the purpose of the next section.

5.6.4. *Maps with orbit-finite fibers.*

In this section, we introduce the notion of maps with orbit-finite fibers and use it to prove that the maps $[-]_\alpha^{(n)} : \mathsf{F}^n 1 \to \mathsf{F}_\alpha^n 1$ are safe.

Definition 5.53 (Orbite-finite fibers)**.** We say that a Nom-morphism $f : X \to Y$ has orbit-finite fibers when for all $v \in Y$ we have that $f^{-1}(v)$ is included in the union of finitely many orbits of X.

Lemma 5.54. *The following are equivalent:*

(1) *f has orbit-finite fibers.*
(2) *For all $v \in Y$ there exists a finitely presentable nominal subset $X_v \subseteq X$ such that $f^{-1}(v) \subseteq X_v$.*
(3) *For all finitely presentable nominal subset $Y' \subseteq Y$ the nominal subset $f^{-1}(Y')$ of X is finitely presentable.*

Proof. (1) $\Longleftrightarrow$ (2) is immediate.

(2) $\Longrightarrow$ (3): Consider $Y' \subseteq Y$ a finitely presentable nominal subset. Then Y' is a finite union of orbits $O_{v_1} \cup \ldots \cup O_{v_n}$. We have picked generators $v_1, \ldots, v_n$ for these orbits. For each i there exists a finitely presentable nominal subset $X_i \subseteq X$ with $f^{-1}(v_i) \subseteq X_i$. Then for all π we have that $f^{-1}(\pi \cdot v_i) \subseteq \pi \cdot X_i = X_i$. Therefore $f^{-1}(O_{v_i}) \subseteq X_i$. Therefore, $f^{-1}(Y') \subseteq X_1 \cup \ldots \cup X_n$. Since a finite union of finitely presentable nominal sets is a finitely presentable nominal set, we conclude that $f^{-1}(Y')$ is a nominal subset of a finitely presentable nominal set, thus it is itself a finitely presentable nominal set. (3) $\Longleftarrow$ (2): Consider $v \in Y$. Put Y' to be the one-orbit nominal set generated by v. By (3) we have that $f^{-1}(Y')$ is a finitely presentable nominal subset of X. We obviously have that $f^{-1}(v) \subseteq f^{-1}(Y')$. □

Lemma 5.55. *If $f : X \to Y$ is surjective and has orbit-finite fibers, then f is safe.*

Proof. If two elements of X are in the same orbit of $f^{-1}(v)$ their supports have the same number of variables. Indeed, if $u, u' \in f^{-1}(v)$ and $u' = \pi \cdot u$ for some permutation π, we have that $\mathsf{supp}(u') = \pi \cdot \mathsf{supp}(u)$, thus $\mathsf{supp}(u)$ and $\mathsf{supp}(u')$ have the same number of elements. By hypothesis $f^{-1}(v)$ is included in the union $O_1 \cup \ldots \cup O_n$ of finitely many orbits of X. By the above observation, the set

$$\{|\mathsf{supp}(u)| \mid u \in O_i \cap f^{-1}(v)\}$$

is a singleton for each $1 \leq i \leq n$. Since we have only finitely many orbits, it follows that

$$\{|\mathsf{supp}(u)| \mid u \in f^{-1}(v)\}$$

has at most n elements, and therefore has a maximum. Thus f is safe. □

Lemma 5.56. *Functions with orbit-finite fibers are closed under (1) finite products, (2) coproducts, (3) abstraction, and (4) composition.*

Proof.

(1) Consider $f_i : X_i \to Y_i$ for $i = 1, 2$ with orbit-finite fibers. Consider $(v_1, v_2) \in Y_1 \times Y_2$. There exists finitely presentable nominal subsets $X'_i \subseteq X_i$ such that $f_i^{-1}(v_i) \subseteq X'_i$. Then $(f_1 \times f_2)^{-1}(v_1, v_2) = f_1^{-1}(v_1) \times f_2^{-1}(v_2) \subseteq X'_1 \times X'_2$. But, since orbit-finite nominal sets are closed under finite products, see for example [BBKL12, Lemma 2], we have that $X'_1 \times X'_2$ is orbit-finite, so we are done.
(2) That's easy, component-wise.
(3) Consider $f : X \to Y$ with orbit-finite fibers. We want to prove that $[\mathcal{V}]f$ also has orbit-finite fibers. Notice that
$$([\mathcal{V}]f)^{-1}(\langle x\rangle v) = \{\langle x\rangle u \mid u \in f^{-1}(v)\}.$$
There exist a finitely presentable nominal subset $X' \subseteq X$ such that $f^{-1}(v) \subseteq X'$. Therefore $([\mathcal{V}]f)^{-1}(\langle x\rangle v) \subseteq [\mathcal{V}]X'$. By Lemma 4.18 we know that $[\mathcal{V}]X'$ is finitely presentable.
(4) Assume $f : X \to Y$ and $g : Y \to W$ have orbit-finite fibers. We show that $g \circ f$ also has orbit-finite fibers. Let $w \in W$. Then $g^{-1}(w) \subseteq Y'$ for some finitely presentable nominal subset Y' of Y. By Lemma 5.54 we have that $f^{-1}(Y')$ is finitely presentable. Since $(g \circ f)^{-1}(w) = f^{-1}(g^{-1}(w)) \subseteq f^{-1}(Y')$ we are done. □

Example 5.57. The function $\theta_X : \mathcal{V} \times X \to [\mathcal{V}]X$ has orbit-finite fibers. Indeed, notice that $\theta^{-1}(\langle x\rangle v) \subseteq \mathcal{V} \times O_v$ where O_v is the orbit spanned by v. But $\mathcal{V} \times O_v$ is finitely presentable, so we are done.

Lemma 5.58. *The maps $q_X : \mathsf{F}X \to \mathsf{F}_\alpha X$ have orbit-finite fibers.*

Proof. This is proved by induction on the structure of F, since the map q_X is obtained from the identity map and θ_X, via products, coproducts and composition. We can therefore apply Lemma 5.56. □

Proposition 5.59. *The maps $[-]_\alpha^{(n)} : \mathsf{F}^n 1 \to \mathsf{F}_\alpha^n 1$ have orbit-finite fibers and are safe.*

Proof. We use induction on n. The base case is clear since $[-]_\alpha^{(0)} = id_1$. For the inductive step, notice that $[-]_\alpha^{(n+1)} = q_{\mathsf{F}_\alpha^n 1} \circ \mathsf{F}([-]_\alpha^{(n)})$. By items (1) and (2) of Lemma 5.56 we have that $\mathsf{F}([-]_\alpha^{(n)})$ has orbit-finite fibers. Then we can apply item (4) of Lemma 5.56 and Lemma 5.58 to derive that $[-]_\alpha^{(n+1)}$ also has orbit-finite fibers. Since the $[-]_\alpha^{(n)}$ are surjective, it follows from Lemma 5.55 that they are safe. □

The maps with orbit-finite fibers have nice closure properties, but safe elements do not behave well with respect to composition, as shown in the next example.

Example 5.60. Let $\mathcal{P}_2(\mathcal{V})$ be the nominal set of two-element sets of names and let $f : \mathcal{V} + \mathcal{P}_2(\mathcal{V}) \to \mathcal{V} + 1$ denote the map $id_{\mathcal{V}}+!$ where ! denotes the unique map into the final nominal set. Let g denote the unique map from $\mathcal{V} + 1$ to 1. Notice that f, g (and therefore their composition) have orbit-finite fibers and are safe. Nevertheless

- $u \in \mathcal{V}$ is f-safe and $f(u) = u$ is g safe, but u is not $(g \circ f)$-safe.
- $\{u, v\}$ is $(g \circ f)$-safe, but $f(\{u, v\})$ is not g-safe.

Therefore, in the next section we need to study the properties of safe squares.

5.6.5. *Properties of safe squares.*

In the first part of this section we will show that safe squares are closed under products and coproducts. Then we will show that also the 'vertical' composition of safe squares is safe provided that some additional properties are satisfied by the maps at issue. This allows us to prove the second bullet point of Corollary 5.47 and conclude our main result on the surjectivity of $P \to U\mathsf{T}_\alpha$ in Theorem 5.72.

Lemma 5.61. *Consider a safe square*

$$\begin{tikzcd}
X \arrow[d, "f"'] & Z \arrow[l, "p"'] \arrow[d, "g"] \\
W & Y \arrow[l, "q"]
\end{tikzcd} \tag{5.48}$$

If $u \in X$ is f-safe, $v \in Y$ and S is a finite subset of $\mathcal{V}$ such that $f(u) = q(v)$, $\mathsf{bv}_f(u)\#v$ and $\mathsf{bv}_f(u)\#S$ there exists a g-safe $z \in Z$ such that $p(z) = u$, $g(z) = v$ and $\mathsf{bv}_g(z)\#S$.

Proof. By the definition of safe squares there exists a g-safe z such that $g(z) = v$ and $p(z) = u$. Let T denote the intersection $S \cap \mathsf{bv}_g(z)$. If T is empty we are done. Otherwise consider a set T' of names fresh for u, v, z, S having the same cardinality as T and let π denote a finite permutation that swaps the elements of T with the elements of T' and fixes all the other elements of $\mathcal{V}$. We will show that $\pi \cdot z$ has all the required properties. By Lemma 5.48, $\pi \cdot z$ is g-safe. In order to prove that $g(\pi \cdot z) = v$ it is enough to check that $\pi \cdot x = x$ for all $x \in \mathsf{supp}(v)$. This is true because $T \subseteq \mathsf{bv}_g(z)$ and $\mathsf{bv}_g(z) \cap \mathsf{supp}(v) = \emptyset$, hence $T \cap \mathsf{supp}(v) = \emptyset$. In order to check that $p(\pi \cdot z) = u$, it is enough to show that $\pi \cdot x = x$ for all $x \in \mathsf{supp}(u)$. But $\mathsf{supp}(u) = \mathsf{bv}_f(u) \cup \mathsf{supp}(f(u)) \subseteq \mathsf{bv}_f(u) \cup \mathsf{supp}(v)$. We have established that $T \cap \mathsf{supp}(v) = \emptyset$. On the other hand, $T \subseteq S$ and $S\#\mathsf{bv}_f(u)$ imply that $T \cap \mathsf{bv}_f(u) = \emptyset$. Hence $T\#u$. Since $T'\#u$ we obtain that π fixes all the names in $\mathsf{supp}(u)$. □

Lemma 5.62. *Safe squares are closed under finite products and coproducts.*

Proof. The case of coproducts is easy. We show the closure of safe squares under finite products. Assume

$$\begin{tikzcd}
X_i \arrow[d, "f_i"'] & Z_i \arrow[l, "p_i"'] \arrow[d, "q_i"] \\
W_i & Y_i \arrow[l, "g_i"]
\end{tikzcd} \tag{5.49}$$

for $i \in \{1, 2\}$ are safe squares. We will show that

$$\begin{tikzcd}
X_1 \times X_2 \arrow[d, "f_1 \times f_2"'] & Z_1 \times Z_2 \arrow[l, "p_1 \times p_2"'] \arrow[d, "q_1 \times q_2"] \\
W_1 \times W_2 & Y_1 \times Y_2 \arrow[l, "g_1 \times g_2"]
\end{tikzcd} \tag{5.50}$$

is a safe square. Consider $f_1 \times f_2$-safe (u_1, u_2) and $(v_1, v_2) \in Y_1 \times Y_2$ such that $\mathsf{bv}(u_1, u_2)\#(v_1, v_2)$ and $f_i(u_i) = g_i(v_i)$. By Lemma 5.50 we know that u_i is f_i-safe and that $\mathsf{bv}(u_1)\#\mathsf{bv}(u_2)$.

Moreover we can compute that

$$\mathsf{bv}(u_1, u_2) = \mathsf{bv}(u_1) \uplus \mathsf{bv}(u_2).$$

Since $\mathsf{bv}(u_1)\#v_2$ and $\mathsf{bv}(u_1)\#\mathsf{bv}(u_2)$, by Lemma 5.61 (applied for $S = \mathsf{bv}(u_2)$) we can find $z_1 \in Z_1$ such that

- $\mathsf{bv}(z_1)\#v_2$ and $\mathsf{bv}(z_1)\#\mathsf{bv}(u_2)$
- z_1 is q_1-safe.
- $q_1(z_1) = v_1$ and $p_1(z_1) = u_1$.

Since $\mathsf{bv}(u_2)\#v_1$ and $\mathsf{bv}(u_2)\#\mathsf{bv}(z_1)$, by Lemma 5.61 (applied for $S = \mathsf{bv}(z_1)$) we can find $z_2 \in Z_2$ such that

- $\mathsf{bv}(z_2)\#v_1$ and $\mathsf{bv}(z_2)\#\mathsf{bv}(z_1)$.
- z_2 is q_2-safe.
- $q_2(z_2) = v_2$ and $p_2(z_2) = u_2$.

By construction we have that $\mathsf{bv}(z_1)\#v_2$, $\mathsf{bv}(z_2)\#v_1$ and $\mathsf{bv}(z_2)\#\mathsf{bv}(z_1)$. It follows that $\mathsf{bv}(z_1)\#z_2$ and $\mathsf{bv}(z_2)\#z_1$. By Lemma 5.50 we know that (z_1, z_2) is $(q_1 \times q_2)$-safe. □

Next we will show that under some mild conditions safe squares are closed under vertical composition. Safe squares resemble weak pullbacks. It is straightforward to show that vertical composition of weak pullbacks gives a weak pullback. However, in the case of safe squares, some additional constraints are imposed on the elements, such as f-safety. Therefore, safe elements should behave well with respect to the vertical composition of the maps. We will find the following definitions handy.

Definition 5.63 (Forward-safe)**.** Let $f : X \to Y$ and $g : Y \to W$ be safe maps. We say that the pair (f, g) is *forward-safe* if, for all $u \in X$ such that u is $(g \circ f)$-safe, we have that $f(u)$ is g-safe.

Definition 5.64 (Backward-safe)**.** Let $f : X \to Y$ and $g : Y \to W$ be safe maps. We say that the pair (f, g) is *backward-safe* if, for all $u \in X$ such that u is f-safe and $f(u)$ is g-safe, we have that u is $(g \circ f)$-safe.

Lemma 5.65. *Consider the following diagram*

$$\begin{tikzcd}
X_1 \arrow[d, "f"'] \arrow[dr, phantom, "(1)"] & X_2 \arrow[l, "p"'] \arrow[d, "h"] \\
Y_1 \arrow[d, "g"'] \arrow[dr, phantom, "(2)"] & Y_2 \arrow[l, "q"] \arrow[d, "k"] \\
W_1 & W_2. \arrow[l, "s"]
\end{tikzcd} \tag{5.51}$$

such that (1) *and* (2) *are safe squares, the pair* (f, g) *is forward-safe and the pair* (h, k) *is backward-safe. Then the outer square in* (5.51) *is a safe square.*

Proof. Consider $u_1 \in X_1$ and $w_2 \in W_2$ such that u_1 is $(g \circ f)$-safe, $\mathsf{bv}_{gf}(u_1)\#w_2$ and $gf(u_1) = s(w_2)$. Since (f, g) is forward-safe we have that $f(u_1)$ is g-safe. We also have that $\mathsf{bv}_g(f(u_1))\#w_2$, because $\mathsf{bv}_g(f(u_1)) \subseteq \mathsf{bv}_{gf}(u_1)$. Since (2) is a safe square there exists a k-safe $v_2 \in Y_2$ such that $k(v_2) = w_2$ and $q(v_2) = f(u_1)$.

Moreover by Lemma 5.61 we can assume that

$$\mathsf{bv}_k(v_2)\#\mathsf{bv}_f(u_1). \tag{5.52}$$

We can apply Lemma 5.61 in this case because $\mathsf{bv}_f(u_1)\#\mathsf{bv}_g(f(u_1))$.

By Lemma 5.51 u_1 is f-safe. We also have that $\mathsf{bv}_f(u_1)\#v_2$. This holds because $\mathsf{supp}(v_2) = \mathsf{bv}_k(v_2) \uplus \mathsf{supp}(w_2)$ and both $\mathsf{bv}_k(v_2)$ and $\mathsf{supp}(w_2)$ are fresh for $\mathsf{bv}_f(u_1)$. The former is by (5.52) while the latter holds because $\mathsf{bv}_f(u_1) \subseteq \mathsf{bv}_{gf}(u_1)$ and $\mathsf{bv}_{gf}(u_1)\#w_2$. Since $q(v_2) = f(u_1)$, we use that (1) is a safe square to derive the existence of an h-safe $u_2 \in X_2$ that satisfies $p(u_2) = u_1$ and $h(u_2) = v_2$. Since (h, k) is a backward-safe pair of maps, u_2 is h-safe and $h(u_2) = v_2$ is k-safe, we conclude that u_2 is also $(k \circ h)$-safe and thus satisfies all the requirements. □

Lemma 5.66. *If u is f-safe and $x\#\mathsf{bv}_f(u)$ then (x, u) is $(\theta \circ (\mathcal{V} \times f))$-safe.*

Proof. By Lemma 5.50, (x, u) is $(\mathcal{V} \times f)$-safe. Consider (y, v) such that $\langle x\rangle f(u) = \langle y\rangle f(v)$. We will show that $|\mathsf{supp}(y, v)| \leq |\mathsf{supp}(x, u)|$. We have that $f(u) = (y\ x) \cdot f(v)$. Then $f(u) = f((y\ x)\cdot v)$ because f is equivariant. Since (x, u) is $(\mathcal{V} \times f)$-safe, $|\mathsf{supp}(x, (y\ x)\cdot v)| \leq |\mathsf{supp}(x, u)|$. We also have that $|\mathsf{supp}(y, v)| = |(y\ x) \cdot \mathsf{supp}(y, v)| = |\mathsf{supp}(x, (y\ x) \cdot v)| \leq |\mathsf{supp}(x, u)|$ □

Lemma 5.67. *Back- and forward-safe pairs have the following closure properties:*

1. *If (f_1, g_1) and (f_2, g_2) are forward-safe (backward-safe) pairs of maps then $(f_1 \times f_2, g_1 \times g_2)$ is forward-safe (backward-safe).*
2. *If (f_i, g_i) are forward-safe (backward-safe) then $(\coprod f_i, \coprod g_i)$ is forward-safe (backward-safe).*
3. *If (f, g) is a forward-safe (backward-safe) pair of maps then $(\mathcal{V} \times f, \theta \circ (\mathcal{V} \times g))$ is forward-safe (backward-safe).*

Proof.

1. Let us show first that $(f_1 \times f_2, g_1 \times g_2)$ is forward-safe. Assume (u_1, u_2) is $(g_1 \times g_2) \circ (f_1 \times f_2)$-safe. By Lemma 5.50 we know that each u_i is $g_i \circ f_i$-safe, $\mathsf{bv}_{g_1 f_1}(u_1)\#u_2$ and $\mathsf{bv}_{g_2 f_2}(u_2)\#u_1$. Since each (f_i, g_i) is forward-safe we have that $f_i(u_i)$ is g_i-safe. Moreover since $\mathsf{bv}_{g_i}(f_i(u_i)) \subseteq \mathsf{bv}_{g_i f_i}(u_i)$ and $\mathsf{supp}(f_i(u_i)) \subseteq \mathsf{supp}(u_i)$ we conclude that $\mathsf{bv}_{g_1}(f_1(u_1))\#f_2(u_2)$ and $\mathsf{bv}_{g_2}(f_2(u_2))\#f_1(u_1)$. Therefore we can apply again Lemma 5.50 to conclude that $(f_1(u_1), f_2(u_2))$ is $g_1 \times g_2$-safe.

   Next we show that $(f_1 \times f_2, g_1 \times g_2)$ is backward-safe when each (f_i, g_i) is. To this end assume (u_1, u_2) is $f_1 \times f_2$-safe and $(f_1(u_1), f_2(u_2))$ is $g_1 \times g_2$-safe. We want to show that (u_1, u_2) is $(g_1 \times g_2) \circ (f_1 \times f_2)$-safe. By Lemma 5.50 we have that
   - u_i is f_i-safe and $f_i(u_i)$ is g_i-safe,
   - $\mathsf{bv}_{f_1}(u_1)\#u_2$ and $\mathsf{bv}_{f_2}(u_2)\#u_1$,
   - $\mathsf{bv}_{g_1}(f_1(u_1))\#f_2(u_2)$ and $\mathsf{bv}_{g_2}(f_2(u_2))\#f_1(u_1)$.

   Since (f_i, g_i) are backward-safe the first item above implies that each u_i is $g_i \circ f_i$-safe. The next two items imply together that $\mathsf{bv}_{g_1 f_1}(u_1)\#u_2$ and $\mathsf{bv}_{g_2 f_2}(u_2)\#u_1$. By Lemma 5.50 we conclude that (u_1, u_2) is $(g_1 \times g_2) \circ (f_1 \times f_2)$-safe.
2. As usual the case of coproducts seems trivial.
3. We first prove the lemma for forward-safe maps. Assume (x, u) is $(\theta_Y \circ (\mathcal{V} \times g) \circ (\mathcal{V} \times f))$-safe. We will show that $(x, f(u))$ is $(\theta_Y \circ (\mathcal{V} \times g))$-safe. By Lemma 5.51 we have that (x, u) is $((\mathcal{V} \times g) \circ (\mathcal{V} \times f))$-safe. By Lemma 5.50 we have that u is $(g \circ f)$-safe and $x\#\mathsf{bv}_{gf}(u)$. Since (f, g) is forward-safe we have that $f(u)$ is g-safe. Moreover since

$\mathsf{bv}_g(f(u)) \subseteq \mathsf{bv}_{gf}(u)$ we have that $x\#\mathsf{bv}_g(f(u))$. Therefore, by Lemma 5.50 we have that $(x, f(u))$ is $(\mathcal{V}\times g)$-safe. By Lemma 5.66 we know that $(x, f(u))$ is $(\theta_Y \circ (\mathcal{V}\times g))$-safe.

Now let us prove this closure property for backward-safe maps. Assume (x, u) is $(\mathcal{V} \times f)$-safe and $(x, f(u))$ is $(\theta \circ (\mathcal{V} \times g))$-safe. We want to show that (x, u) is $(\theta \circ (\mathcal{V} \times g) \circ (\mathcal{V} \times f))$-safe. By Lemma 5.50 we know that u is f-safe and $x\#\mathsf{bv}_f(u)$. By Lemma 5.51 we know that $(x, f(u))$ is $(\mathcal{V} \times g)$-safe, thus by Lemma 5.50 we know that $f(u)$ is g-safe and $x\#\mathsf{bv}_g(f(u))$. Since (f, g) is backward-safe we have that u is $(g \circ f)$-safe. We can also check that $x\#\mathsf{bv}_{gf}(u)$. Thus (x, u) is $(\mathcal{V} \times (g \circ f))$-safe. Applying Lemma 5.66 we get that (x, u) is $(\theta \circ (\mathcal{V} \times (g \circ f)))$-safe. □

Lemma 5.68. *For every safe map $f : X \to Y$ the pair of maps $(\mathsf{F}(f), q_Y)$ is both backward-safe and forward-safe.*

Proof. The proof is by induction on the grammar of F_α and Lemma 5.67. □

Lemma 5.69. *If*

$$\begin{tikzcd}
X \arrow[d, "f"'] & Z \arrow[l, "p"'] \arrow[d, "q"] \\
W & Y \arrow[l, "g"]
\end{tikzcd} \tag{5.53}$$

is a safe square then

$$\begin{tikzcd}
\mathcal{V} \times X \arrow[d, "\theta \circ (\mathcal{V} \times f)"'] & \mathcal{V} \times Z \arrow[l, "\mathcal{V} \times p"'] \arrow[d, "\theta \circ (\mathcal{V} \times q)"] \\
{[\mathcal{V}]W} & {[\mathcal{V}]Y} \arrow[l, "{[\mathcal{V}]g}"]
\end{tikzcd} \tag{5.54}$$

is a safe square.

Proof. We apply Lemma 5.65. Note that the pairs of maps $(\mathcal{V} \times f, \theta)$ and $(\mathcal{V} \times q, \theta)$ are forward and backward-safe by Lemma 5.67 (3). □

Lemma 5.70. *For all equivariant $f : X \to Y$ the square*

$$\begin{tikzcd}
\mathsf{F}Y \arrow[d, two heads, "q_Y"'] & \mathsf{F}X \arrow[l] \arrow[d, two heads, "q_X"] \\
\mathsf{F}_\alpha Y & \mathsf{F}_\alpha X \arrow[l]
\end{tikzcd} \tag{5.55}$$

is a safe square.

Proof. This is proved by induction on F_α using Lemmas 5.62 and 5.69. □

Proposition 5.71. *For all n the squares*

$$\begin{tikzcd}
\mathsf{F}^n 1 \arrow[d, two heads, "{[-]_\alpha^{(n)}}"'] & \mathsf{F}^{n+1} 1 \arrow[l] \arrow[d, two heads, "{[-]_\alpha^{(n+1)}}"] \\
\mathsf{F}_\alpha^n 1 & \mathsf{F}_\alpha^{n+1} 1 \arrow[l]
\end{tikzcd} \tag{5.56}$$

are safe squares.

Proof. We use induction on n. For the inductive step observe that the $(n+1)^{th}$ square is the composition

$$\begin{tikzcd}
\mathsf{F}\mathsf{F}^{n}1 \arrow[d, two heads, "\mathsf{F}[-]_{\alpha}^{(n)}"'] & \mathsf{F}\mathsf{F}^{n+1}1 \arrow[l] \arrow[d, two heads, "\mathsf{F}[-]_{\alpha}^{(n+1)}"] \\
\mathsf{F}\mathsf{F}_{\alpha}^{n}1 \arrow[d, "q_{\mathsf{F}_{\alpha}^{n}1}"'] & \mathsf{F}\mathsf{F}_{\alpha}^{n+1}1 \arrow[l] \arrow[d, "q_{\mathsf{F}_{\alpha}^{n+1}1}"] \\
\mathsf{F}_{\alpha}\mathsf{F}_{\alpha}^{n}1 & \mathsf{F}_{\alpha}\mathsf{F}_{\alpha}^{n+1}1 \arrow[l]
\end{tikzcd} \tag{5.57}$$

By Lemma 5.62 we know that the upper square in (5.57) is a safe square. By Lemma 5.70 the lower square in (5.57) is also safe. Moreover, by Lemma 5.68 we know that $(\mathsf{F}[-]_{\alpha}^{(n)}, q_{\mathsf{F}_{\alpha}^{n}1})$ is forward-safe and $(\mathsf{F}[-]_{\alpha}^{(n+1)}, q_{\mathsf{F}_{\alpha}^{n+1}1})$ is backward-safe. Thus, we can apply Lemma 5.65 to conclude that the outer square in (5.57) is a safe square. □

We established the two bullet points of Corollary 5.47 in Propositions 5.59 and 5.71:

Theorem 5.72. *Let $\mathsf{F}, \mathsf{F}_{\alpha}$ be endofunctors on* Nom *obtained from a binding signature and $q : \mathsf{F} \to \mathsf{F}_{\alpha}$ be the natural transformation defined in 5.12. Let P be a pullback as in* (5.20)*. Then $P \to U\mathsf{T}_{\alpha}$ is onto.*

6. Applications

In this section, we first give a general definition of substitution on the final coalgebra T_{α} of a functor F_{α} coming from a binding signature. We, then, apply the general results given in the previous sections to the infinitary λ-calculus by defining substitution and the notions of Böhm, Lévy-Longo and Berarducci trees on α-equivalence classes of λ-terms.

6.1. **Substitution on an Arbitrary Coalgebraic Data Type.**

The following lemma [Mos01, 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 X will be used for the term N in $\mathsf{M}[\mathsf{x} := \mathsf{N}]$ which is not subject to recursion and the set Y will be used for the recursion.

Lemma 6.1. *Let $\delta : D \to F(D)$ be a final coalgebra and $g : X \to F(X)$ an arbitrary F-coalgebra. Then, there is a unique map $f : Y \to D$ such that for any $h : Y \to F(X+Y)$, the following diagram commutes:*

$$\begin{tikzcd}
Y \arrow[r, "f"] \arrow[d, "h"'] & D \arrow[d, "\delta"] \\
F(X+Y) \arrow[r, "{F([g^*, f])}"] & F(D)
\end{tikzcd}$$

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

We apply Lemma 6.1 to define substitution on the final coalgebra T_{α} of a functor F_{α} coming from a binding signature.

Definition 6.2 (Substitution on α-equivalence classes of infinitary terms coming from a binding signature)**.** Substitution on T_α is defined as the unique map such that the diagram below commutes:

$$\begin{tikzcd}
\mathsf{T}_\alpha \times \mathcal{V} \times \mathsf{T}_\alpha \arrow[r, dashed, "\mathsf{subs}_\alpha"] \arrow[d, "\mathsf{h}_{\mathsf{subs}_\alpha}"'] & \mathsf{T}_\alpha \arrow[d, "\mathsf{unfold}"] \\
\mathsf{F}_\alpha(\mathsf{T}_\alpha + \mathsf{T}_\alpha \times \mathcal{V} \times \mathsf{T}_\alpha) \arrow[r, "{\mathsf{F}_\alpha([id, \mathsf{subs}_\alpha])}"] & \mathsf{F}_\alpha(\mathsf{T}_\alpha)
\end{tikzcd}$$

where $\mathsf{h}_{\mathsf{subs}_\alpha}$ is defined in (6.1), (6.2) and (6.3).

Intuitively $\mathsf{subs}_\alpha([t]_\alpha, x, [s]_\alpha)$ is given by $[t[x := s]]_\alpha$. However, as explained in the introduction, substitution cannot be formally defined as a total function on raw terms: additional freshness side conditions are required. See also the explanation following (6.7).

In order to define the equivariant map $\mathsf{h}_{\mathsf{subs}_\alpha}$ we can use the properties of the functor F_α. As observed in Proposition 5.6 such functors are Nom-enriched, or equivalently (see [Koc72]) *strong*. That is, there exists a natural transformation

$$\tau_{X,Y} : \mathsf{F}_\alpha X \times Y \to \mathsf{F}_\alpha(X \times Y).$$

Each functor obtained from the grammar in (5.4) can be equipped with a strength. Most constructions are standard and if two functors are strong so is their composition, product or coproduct. The only interesting case is that of the abstraction functor. We define a strength using the concretion of Definition 4.10. Explicitly $\tau_{X,Y} : [\mathcal{V}]X \times Y \to [\mathcal{V}](X \times Y)$ is defined by

$$(\langle x\rangle u, v) \mapsto \langle y\rangle(\langle x\rangle u@y, v)$$

where y is some/any fresh variable for x, u, v. By construction, τ is a well-defined and natural in both X and Y.

The map $\mathsf{h}_{\mathsf{subs}_\alpha}$ is defined as follows.

- For tuples of the form (x, x, N) we define $\mathsf{h}_{\mathsf{subs}_\alpha}(x, x, N)$ as the composite

$$\mathsf{T}_\alpha \times \mathcal{V} \times \mathsf{T}_\alpha \xrightarrow{\pi_3} \mathsf{T}_\alpha \xrightarrow{\mathsf{unfold}} \mathsf{F}_\alpha(\mathsf{T}_\alpha) \xrightarrow{\mathsf{F}_\alpha(\mathsf{inl})} \mathsf{F}_\alpha(\mathsf{T}_\alpha + \mathsf{T}_\alpha \times \mathcal{V} \times \mathsf{T}_\alpha) \tag{6.1}$$

- For tuples of the form (y, x, N) with $x \neq y$ or for tuples of the form (k, x, N) where k is a constant, we define $\mathsf{h}_{\mathsf{subs}_\alpha}(y, x, N)$ as the composite

$$\mathsf{T}_\alpha \times \mathcal{V} \times \mathsf{T}_\alpha \xrightarrow{\pi_1} \mathsf{T}_\alpha \xrightarrow{\mathsf{unfold}} \mathsf{F}_\alpha(\mathsf{T}_\alpha) \xrightarrow{\mathsf{F}_\alpha(\mathsf{inl})} \mathsf{F}_\alpha(\mathsf{T}_\alpha + \mathsf{T}_\alpha \times \mathcal{V} \times \mathsf{T}_\alpha) \tag{6.2}$$

- For tuples (M, x, N) such that M is not a variable we define $\mathsf{h}_{\mathsf{subs}_\alpha}(M, x, N)$ as the composite:

$$\begin{tikzcd}
\mathsf{T}_\alpha \times \mathcal{V} \times \mathsf{T}_\alpha \arrow[r, dashed, "\mathsf{h}_{\mathsf{subs}_\alpha}"] \arrow[d, "\mathsf{unfold} \times \mathcal{V} \times \mathsf{T}_\alpha"'] & \mathsf{F}_\alpha(\mathsf{T}_\alpha + \mathsf{T}_\alpha \times \mathcal{V} \times \mathsf{T}_\alpha) \\
\mathsf{F}_\alpha(\mathsf{T}_\alpha) \times \mathcal{V} \times \mathsf{T}_\alpha \arrow[r, "\tau_{\mathsf{T}_\alpha, \mathcal{V} \times \mathsf{T}_\alpha}"] & \mathsf{F}_\alpha(\mathsf{T}_\alpha \times \mathcal{V} \times \mathsf{T}_\alpha) \arrow[u, "\mathsf{F}_\alpha(\mathsf{inr})"']
\end{tikzcd} \tag{6.3}$$

Since $\{(x,x,N) \mid x \in \mathcal{V}\}$, $\{(y,x,N) \mid x \neq y \in \mathcal{V}\}$, $\{(k,x,N) \mid k \text{ constant}\}$ and $\{(M,x,N) \mid M \notin \mathcal{V}\}$ are nominal sets that form a partition of $\mathsf{T}_\alpha \times \mathcal{V} \times \mathsf{T}_\alpha$ the map $\mathsf{h}_{\mathsf{subs}_\alpha}$ is well-defined and equivariant, thus we can apply Lemma 6.1 to prove the existence of a unique substitution map subs_α.

6.2. **Substitution on α-Equivalence Classes of Infinitary λ-Terms.** As an example we spell out the concrete calculations for substitution on the nominal set $\Lambda^\infty_\perp$ of finite and infinite λ-terms with $\perp$. The set $\Lambda^\infty_\perp$ is defined as the final coalgebra of the functor Lb defined by:

$$Lb\ X = \mathcal{V} + \{\perp\} + \mathcal{V} \times X + X \times X. \tag{6.4}$$

Adding the extra constant $\perp$ is needed in order to write corecursive functions that compute the Böhm, Lévy-Longo and Berarducci trees. We also consider the functor

$$\mathsf{Lb}_\alpha\ X = \mathcal{V} + \{\perp\} + [\mathcal{V}]X + X \times X. \tag{6.5}$$

Notation 6.3. We write $\mathsf{\Lambda}^\infty_\alpha$ for the final coalgebra of Lb_α, omitting the $\perp$ in the notation to improve readability. We continue to denote terms in $\Lambda^\infty_{\perp\mathrm{ffv}}$ by M, N, but will denote terms in $\mathsf{\Lambda}^\infty_\alpha$ by M, N. By Corollary 5.35 we have that $\mathsf{\Lambda}^\infty_\alpha$ is isomorphic to $\Lambda^\infty_{\perp\mathrm{ffv}}/{=_\alpha}$.

The injections for the coproduct $X+Y$ are denoted as $\mathsf{inl}^{X,Y} : X \to X+Y$ and $\mathsf{inr}^{X,Y} : Y \to X+Y$. But for the case of $\mathsf{L}_\alpha(X)$, we denote them as

$$\begin{aligned}
&\mathsf{inbot}^X : \{\perp\} \to \mathsf{Lb}_\alpha(X)\\
&\mathsf{invar}^X : \mathcal{V} \to \mathsf{Lb}_\alpha(X)\\
&\mathsf{inabs}^X : [\mathcal{V}]X \to \mathsf{Lb}_\alpha(X)\\
&\mathsf{inapp}^X : X \times X \to \mathsf{Lb}_\alpha(X).
\end{aligned}$$

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

Since unfold is an isomorphism that partitions its domain $\mathsf{\Lambda}^\infty_\alpha$ into four disjoint components, see (1.2), we write typical elements of $\mathsf{\Lambda}^\infty_\alpha$ as x, $\perp$, $\mathsf{M}_1\mathsf{M}_2$, $\lambda\mathsf{y}.\mathsf{M}$ where

$$\begin{aligned}
\mathsf{x} &= \mathsf{unfold}^{-1}(\mathsf{invar}\ \mathsf{x})\\
\perp &= \mathsf{unfold}^{-1}(\mathsf{inbot}\ \perp)\\
\lambda\mathsf{y}.\mathsf{M} &= \mathsf{unfold}^{-1}(\mathsf{inabs}\ \langle\mathsf{y}\rangle\mathsf{M})\\
\mathsf{M}_1\mathsf{M}_2 &= \mathsf{unfold}^{-1}(\mathsf{inapp}\ (\mathsf{M}_1,\mathsf{M}_2)).
\end{aligned} \tag{6.6}$$

We use x to denote both an element in $\mathcal{V}$ and also its copy in $\mathsf{\Lambda}^\infty_\alpha$.

Example 6.4 (Substitution on α-equivalence classes of infinitary λ-terms with $\perp$)**.** By instantiating the definition of the map $\mathsf{h}_{\mathsf{subs}_\alpha}$ given in (6.1), (6.2) and (6.3) for the functor

Lb_α we obtain $\mathsf{h}_{\mathsf{subs}_\alpha} : \Lambda^\infty_\alpha \times \ \mathcal{V} \ \times \ \Lambda^\infty_\alpha \ \to \mathsf{Lb}_\alpha(\Lambda^\infty_\alpha + \ \Lambda^\infty_\alpha \times \ \mathcal{V} \ \times \Lambda^\infty_\alpha)$ given by

$$\begin{aligned}
&\mathsf{h}_{\mathsf{subs}_\alpha}(\mathsf{x},\mathsf{x},\mathsf{N}) &&= \mathsf{unfold}(\mathsf{N}) && \\
&\mathsf{h}_{\mathsf{subs}_\alpha}(\mathsf{y},\mathsf{x},\mathsf{N}) &&= \mathsf{invar}\ \mathsf{y} && \textit{if}\ \ \mathsf{y} \neq \mathsf{x} \\
&\mathsf{h}_{\mathsf{subs}_\alpha}(\bot,\mathsf{x},\mathsf{N}) &&= \mathsf{inbot}\ \bot && \\
&\mathsf{h}_{\mathsf{subs}_\alpha}(\mathsf{M}_1\mathsf{M}_2,\mathsf{x},\mathsf{N}) &&= \mathsf{inapp}\ ((\mathsf{M}_1,\mathsf{x},\mathsf{N}),(\mathsf{M}_2,\mathsf{x},\mathsf{N})) && \\
&\mathsf{h}_{\mathsf{subs}_\alpha}(\lambda\mathsf{y}.\mathsf{M},\mathsf{x},\mathsf{N}) &&= \mathsf{inabs}\ \langle \mathsf{z}\rangle(((\langle \mathsf{y}\rangle \mathsf{M})@\mathsf{z},\mathsf{x},\mathsf{N}) && \textit{if}\ \ \mathsf{z}\#(\lambda\mathsf{y}.\mathsf{M},\mathsf{x},\mathsf{N}).
\end{aligned}$$

To improve readability we omitted $\mathsf{Lb}_\alpha\mathsf{inr}$ or $\mathsf{Lb}_\alpha\mathsf{inl}$ in the definition of $\mathsf{h}_{\mathsf{subs}_\alpha}$. We obtain the substitution function

$$\mathsf{subs}_\alpha : \Lambda^\infty_\alpha \times \ \mathcal{V} \times \Lambda^\infty_\alpha \ \to \ \Lambda^\infty_\alpha$$

given by

$$\begin{aligned}
&\mathsf{subs}_\alpha(\mathsf{x},\mathsf{x},\mathsf{N}) &&= \mathsf{N} && \\
&\mathsf{subs}_\alpha(\mathsf{x},\mathsf{y},\mathsf{N}) &&= \mathsf{x} && \textit{if}\ \mathsf{y} \neq \mathsf{x} \\
&\mathsf{subs}_\alpha(\bot,\mathsf{y},\mathsf{N}) &&= \bot && \\
&\mathsf{subs}_\alpha(\mathsf{M}_1\mathsf{M}_2,\mathsf{x},\mathsf{N}) &&= \mathsf{subs}_\alpha(\mathsf{M}_1,\mathsf{x},\mathsf{N})\ \mathsf{subs}_\alpha(\mathsf{M}_2,\mathsf{x},\mathsf{N}) && \\
&\mathsf{subs}_\alpha(\lambda\mathsf{y}.\mathsf{M},\mathsf{x},\mathsf{N}) &&= \lambda\mathsf{y}.\mathsf{subs}_\alpha(\mathsf{M},\mathsf{x},\mathsf{N}) && \textit{if}\ \ \mathsf{y}\#(\mathsf{x},\mathsf{N}).
\end{aligned} \tag{6.7}$$

It should be pointed out that subs_α defined in (6.7) is the 'semantic' version of the substitution map since it is defined on the (or any) final Lb_α-coalgebra. It is Corollary 5.35 that allows us to identify the element $\lambda\mathsf{y}.\mathsf{M} \in \Lambda^\infty_\alpha$ with the α-equivalence class of an infinitary term $[\lambda y.M]_\alpha$ having finitely many free variables. Thus we obtain a 'syntactic' version of the substitution map which looks indeed just like a notational variant of the Set-based (1.3), but is now fully justified as a coinductive definition on α-equivalence classes of λ-terms.

We can now define β-reduction using subs_α.

Definition 6.5 (β-reduction on α-equivalence classes)**.** We define β_α-reduction as the smallest relation on $\Lambda^\infty_\alpha \times \Lambda^\infty_\alpha$ that satisfies

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

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

We 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 6.6 (Head β-reduction on α-equivalence classes)**.** We define β_1-reduction as the smallest relation on $\Lambda^\infty_\alpha \times \Lambda^\infty_\alpha$ closed under

$$\frac{}{(\lambda \mathsf{x}.\mathsf{P})\mathsf{Q} \to_{\beta_1} \mathsf{subs}_\alpha(\mathsf{P},\mathsf{x},\mathsf{Q})}\,(\beta_1) \qquad \frac{\mathsf{P} \to_{\beta_1} \mathsf{P}'}{\lambda \mathsf{x}.\mathsf{P} \to_{\beta_1} \lambda \mathsf{x}.\mathsf{P}'}\,(abs)$$

$$\frac{\mathsf{P} \to_{\beta_1} \mathsf{P}' \quad \mathsf{P} \text{ is not an abstraction}}{\mathsf{PQ} \to_{\beta_1} \mathsf{P}'\mathsf{Q}}\,(app_L)$$

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

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 6.7 (Weak head β-reduction on α-equivalence classes)**.** We define β_2-reduction as the smallest relation on $\Lambda^\infty_\alpha \times \Lambda^\infty_\alpha$ closed under

$$\frac{}{(\lambda \mathsf{x}.\mathsf{P})\mathsf{Q} \to_{\beta_2} \mathsf{subs}_\alpha(\mathsf{P},\mathsf{x},\mathsf{Q})}\,(\beta_2) \qquad \frac{\mathsf{P} \to_{\beta_1} \mathsf{P}'}{\mathsf{PQ} \to_{\beta_2} \mathsf{P}'\mathsf{Q}}\,(app_L)$$

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

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 6.8 (Top β-reduction on α-equivalence classes)**.** We define β_3-reduction as the smallest relation on $\Lambda^\infty_\alpha \times \Lambda^\infty_\alpha$ closed under

$$\frac{\mathsf{M} \twoheadrightarrow_{\beta_2} (\lambda \mathsf{x}.\mathsf{P})}{\mathsf{MQ} \to_{\beta_3} \mathsf{subs}_\alpha(\mathsf{P},\mathsf{x},\mathsf{Q})}\,(\beta_3)$$

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, transitive closures of $\to_{\beta_1}$, $\to_{\beta_2}$ and $\to_{\beta_3}$ are denoted by $\twoheadrightarrow_{\beta_1}$, $\twoheadrightarrow_{\beta_2}$ and $\twoheadrightarrow_{\beta_3}$, respectively. The corresponding normal forms, head normal form (hnf), weak head normal form (whnf) and top normal form (tnf), should they exist are unique.

6.3. **Computing the Infinite Normal Form of α-Equivalence Classes of λ-Terms.**

We now define the notions of Böhm tree, Lévy-Longo tree, and Berarducci tree using the finality of $\mathsf{unfold} : \Lambda^\infty_\alpha \to \mathsf{Lb}_\alpha(\Lambda^\infty_\alpha)$.

Definition 6.9 (Böhm tree on α-equivalence classes)**.** We define the Böhm tree of M as $\mathsf{BT}_\alpha(\mathsf{M})$ where BT_α is the unique map such that

$$\begin{tikzcd}
\Lambda^\infty_\alpha \arrow[r, "\mathsf{BT}_\alpha"] \arrow[d, "\mathsf{g}_{\mathsf{BT}_\alpha}"'] & \Lambda^\infty_\alpha \arrow[d, "\mathsf{unfold}"] \\
\mathsf{Lb}_\alpha(\Lambda^\infty_\alpha) \arrow[r, "\mathsf{Lb}_\alpha(\mathsf{BT}_\alpha)"'] & \mathsf{Lb}_\alpha(\Lambda^\infty_\alpha)
\end{tikzcd}$$

commutes, with $\mathsf{g}_{\mathsf{BT}_\alpha} : \Lambda^\infty_\alpha \to \mathsf{Lb}_\alpha(\Lambda^\infty_\alpha)$ being defined as

$$\mathsf{g}_{\mathsf{BT}_\alpha}(\mathsf{M}) = \begin{cases} \mathsf{unfold}(\mathsf{N}) & \textit{if } \mathsf{M} \twoheadrightarrow_{\beta_1} \mathsf{N} \textit{ and } \mathsf{N} \textit{ is in hnf,} \\ \mathsf{inbot}\ \bot & \textit{otherwise.} \end{cases}$$

Definition 6.10 (Lévy-Longo tree on α-equivalence classes)**.** We define the Lévy-Longo tree of M as $\mathsf{LLT}_\alpha(\mathsf{M})$ where LLT_α is the unique map such that

$$\begin{tikzcd}
\Lambda^\infty_\alpha \arrow[r, "\mathsf{LLT}_\alpha"] \arrow[d, "\mathsf{g}_{\mathsf{LLT}_\alpha}"'] & \Lambda^\infty_\alpha \arrow[d, "\mathsf{unfold}"] \\
\mathsf{Lb}_\alpha(\Lambda^\infty_\alpha) \arrow[r, "\mathsf{Lb}_\alpha(\mathsf{LLT}_\alpha)"'] & \mathsf{Lb}_\alpha(\Lambda^\infty_\alpha)
\end{tikzcd}$$

commutes, with $\mathsf{g}_{\mathsf{LLT}_\alpha} : \Lambda^\infty_\alpha \to \mathsf{Lb}_\alpha(\Lambda^\infty_\alpha)$ being defined as

$$\mathsf{g}_{\mathsf{LLT}_\alpha}(\mathsf{M}) = \begin{cases} \mathsf{unfold}(\mathsf{N}) & \textit{if } \mathsf{M} \twoheadrightarrow_{\beta_2} \mathsf{N} \textit{ and } \mathsf{N} \textit{ is in whnf,} \\ \mathsf{inbot}\ \bot & \textit{otherwise.} \end{cases}$$

Definition 6.11 (Berarducci tree on α-equivalence classes)**.** We define the Berarducci tree of M as $\mathsf{BeT}_\alpha(\mathsf{M})$ where BeT_α is the unique map such that

$$\begin{tikzcd}
\Lambda^\infty_\alpha \arrow[r, "\mathsf{BeT}_\alpha"] \arrow[d, "\mathsf{g}_{\mathsf{BeT}_\alpha}"'] & \Lambda^\infty_\alpha \arrow[d, "\mathsf{unfold}"] \\
\mathsf{Lb}_\alpha(\Lambda^\infty_\alpha) \arrow[r, "\mathsf{Lb}_\alpha(\mathsf{BeT}_\alpha)"'] & \mathsf{Lb}_\alpha(\Lambda^\infty_\alpha)
\end{tikzcd}$$

commutes, with $\mathsf{g}_{\mathsf{BeT}_\alpha} : \Lambda^\infty_\alpha \to \mathsf{Lb}_\alpha(\Lambda^\infty_\alpha)$ being defined as follows.

$$\mathsf{g}_{\mathsf{BeT}_\alpha}(\mathsf{M}) = \begin{cases} \mathsf{unfold}(\mathsf{N}) & \textit{if } \mathsf{M} \twoheadrightarrow_{\beta_3} \mathsf{N} \textit{ and } \mathsf{N} \textit{ is in tnf,} \\ \mathsf{inbot}\ \bot & \textit{otherwise.} \end{cases}$$

Similar to the remark on the substitution map mentioned at the end of Example 6.4, the maps BT_α, LLT_α and BeT_α are the 'semantic' counterparts of the maps that compute the Böhm, Lévy-Longo and Berarducci trees. Corollary 5.35 allows us to obtain a 'syntactic' version of these maps which look like a notational variant of the Set-based (3.6), (3.7) and (3.8), but are now fully justified as coinductive definitions on α-equivalence classes of λ-terms.

6.4. **Nominal Sets of Infinite Normal Forms and Bisimulations.** Recall the *informal* corecursive definitions for the sets of Böhm, Lévy-Longo and Berarducci Trees given in Definitions 3.3, 3.4 and 3.5. In this section, we formally define the first two ones on α-equivalence classes as an application of Corollary 5.35. In order to define the set of Berarducci trees, one needs to extend the notion of binding signature to include infinite products, which is beyond the scope of our paper.

Example 6.12 (Böhm trees up to α-equivalence)**.** We define the functor S_{BT} by

$$\mathsf{S}_{\mathsf{BT}}(X) \quad = \{\bot\} + \mathcal{V} + \textstyle\coprod_k [\mathcal{V}]^k(\mathcal{V} \times \mathsf{List}(X)).$$

The final coalgebra of S_{BT}, denoted by $\mathcal{BT}_\alpha$, exists and it is isomorphic to the set $(\mathcal{BT})_{\mathrm{ffv}}/{=_\alpha}$ of α-equivalence classes of Böhm trees with finitely many free variables by Corollary 5.35 (see also (3.6)).

The nominal set Λ^∞_α can be equipped with a S_{BT}-coalgebra structure. Explicitly, consider $\xi_{\mathsf{BT}} : \Lambda^\infty_\alpha \to \mathsf{S}_{\mathsf{BT}}(\Lambda^\infty_\alpha)$ defined by

$$\xi_{\mathsf{BT}}(\mathsf{M}) = \begin{cases} \mathsf{x} & \textit{if } \mathsf{M} \twoheadrightarrow_{\beta_1} \mathsf{x}, \\ \langle \mathsf{x}_1\rangle \ldots \langle \mathsf{x}_n\rangle(\mathsf{x}, \mathsf{M}_1, \ldots, \mathsf{M}_n) & \textit{if } \mathsf{M} \twoheadrightarrow_{\beta_1} \lambda\mathsf{x}_1 \ldots \lambda\mathsf{x}_n.\mathsf{x}\mathsf{M}_1 \ldots \mathsf{M}_m, \\ \bot & \textit{otherwise.} \end{cases}$$

Since $\mathcal{BT}_\alpha$ is the final S_{BT}-coalgebra, we have a unique morphism $\mathsf{BT}_\alpha : \Lambda^\infty_\alpha \to \mathcal{BT}_\alpha$ such that the diagram below commutes. Notice that this morphism is obtained by restricting the codomain of the map BT_α from Definition 6.9.

$$\begin{tikzcd}
\Lambda^\infty_\alpha \arrow[r, "\mathsf{BT}_\alpha"] \arrow[d, "\xi_{\mathsf{BT}}"'] & \mathcal{BT}_\alpha \arrow[d, "\simeq"] \\
\mathsf{S}_{\mathsf{BT}}(\Lambda^\infty_\alpha) \arrow[r, "\mathsf{S}_{\mathsf{BT}}(\mathsf{BT}_\alpha)"'] & \mathsf{S}_{\mathsf{BT}}(\mathcal{BT}_\alpha)
\end{tikzcd}$$

We can now define the head bisimulation $\sim_{hnf}$ on Λ^∞_α as the kernel pair of the map BT_α, that is, $\mathsf{M} \sim_{hnf} \mathsf{N}$ if and only if M and N have the same Böhm tree. Explicitly, $\sim_{hnf}$ is defined as the pullback

$$\begin{tikzcd}
\sim_{hnf} \arrow[r, "\pi_1"] \arrow[d, "\pi_2"'] \arrow[dr, phantom, "\lrcorner", very near start] & \Lambda^\infty_\alpha \arrow[d, "\mathsf{BT}_\alpha"] \\
\Lambda^\infty_\alpha \arrow[r, "\mathsf{BT}_\alpha"'] & \mathcal{BT}_\alpha
\end{tikzcd}$$

Since $[\mathcal{V}](-)$ preserves pullbacks and pullbacks commute with coproducts and limits in Nom, we have that S_{BT} preserves pullbacks, thus the outer square of the diagram

$$\begin{tikzcd}
\mathsf{S}_{\mathsf{BT}}(\sim_{hnf}) \arrow[rrr, "\mathsf{S}_{\mathsf{BT}}(\pi_1)"] \arrow[ddd, "\mathsf{S}_{\mathsf{BT}}(\pi_2)"'] & & & \mathsf{S}_{\mathsf{BT}}(\Lambda^\infty_\alpha) \arrow[ddd, "\mathsf{S}_{\mathsf{BT}}(\mathsf{BT}_\alpha)"] \\
& \sim_{hnf} \arrow[ul, dashed] \arrow[r, "\pi_1"] \arrow[d, "\pi_2"'] \arrow[dr, phantom, "\lrcorner", very near start] & \Lambda^\infty_\alpha \arrow[ur, "\xi_{\mathsf{BT}}"'] \arrow[d, "\mathsf{BT}_\alpha"] & \\
& \Lambda^\infty_\alpha \arrow[r, "\mathsf{BT}_\alpha"'] \arrow[dl, "\xi_{\mathsf{BT}}"] & \mathcal{BT}_\alpha \arrow[dr] & \\
\mathsf{S}_{\mathsf{BT}}(\Lambda^\infty_\alpha) \arrow[rrr, "\mathsf{S}_{\mathsf{BT}}(\mathsf{BT}_\alpha)"'] & & & \mathsf{S}_{\mathsf{BT}}(\mathsf{BT}_\alpha)
\end{tikzcd}$$

is also a pullback. Therefore we obtain an S_{BT}-coalgebra structure on $\sim_{hnf}$ such that the diagram

$$\begin{tikzcd}
\Lambda^{\infty}_{\alpha} \arrow[d, "\xi_{\mathsf{BT}}"'] & \sim_{hnf} \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d] & \Lambda^{\infty}_{\alpha} \arrow[d, "\xi_{\mathsf{BT}}"] \\
\mathsf{S}_{\mathsf{BT}}(\Lambda^{\infty}_{\alpha}) & \mathsf{S}_{\mathsf{BT}}(\sim_{hnf}) \arrow[l, "\mathsf{S}_{\mathsf{BT}}(\pi_1)"] \arrow[r, "\mathsf{S}_{\mathsf{BT}}(\pi_2)"'] & \mathsf{S}_{\mathsf{BT}}(\Lambda^{\infty}_{\alpha})
\end{tikzcd} \tag{6.8}$$

commutes. This shows that $\sim_{hnf}$ is a bisimulation in the sense of [AM89]. The commutativity of (6.8) means that $\sim_{hnf}$ is a binary relation on $\Lambda^{\infty}_{\alpha}$ such that for all M and N, if $\mathsf{M} \sim_{hnf} \mathsf{N}$ and $\mathsf{M} \twoheadrightarrow_{\beta_1} \lambda \mathsf{x}_1 \ldots \lambda \mathsf{x}_\mathsf{n}.\mathsf{x}\mathsf{M}_1 \ldots \mathsf{M}_\mathsf{m}$, then there are $\mathsf{N}_1, \ldots, \mathsf{N}_\mathsf{m}$ such that $\mathsf{N} \twoheadrightarrow_{\beta_1} \lambda \mathsf{x}_1 \ldots \lambda \mathsf{x}_\mathsf{n}.\mathsf{x}\mathsf{N}_1 \ldots \mathsf{N}_\mathsf{m}$ and $\mathsf{M}_\mathsf{i} \sim_{hnf} \mathsf{N}_\mathsf{i}$ for all $1 \leq i \leq m$.

Example 6.13 (Lévy-Longo trees up to α-equivalence)**.** We define the functor $\mathsf{S}_{\mathsf{LLT}}$ by

$$\mathsf{S}_{\mathsf{LLT}}(X) = \{\perp\} + [\mathcal{V}]X + \mathcal{V} \times \mathsf{List}(X).$$

The final coalgebra of $\mathsf{S}_{\mathsf{LLT}}$, denoted by $\mathcal{LLT}_\alpha$, exists and is isomorphic to the set $(\mathcal{LLT})_{\mathrm{ffv}}/{=_\alpha}$ by Corollary 5.35 (see also (3.7)).

The nominal set $\Lambda^{\infty}_{\alpha}$ can be equipped with an $\mathsf{S}_{\mathsf{LLT}}$-coalgebra structure $\xi_{\mathsf{LLT}} : \Lambda^{\infty}_{\alpha} \to \mathsf{S}_{\mathsf{LLT}}(\Lambda^{\infty}_{\alpha})$ as follows:

$$\xi_{\mathsf{LLT}}(\mathsf{M}) = \begin{cases} \langle \mathsf{x} \rangle \mathsf{N} & \textit{if } \mathsf{M} \twoheadrightarrow_{\beta_2} \lambda \mathsf{x}.\mathsf{N}, \\ (\mathsf{x}, \mathsf{M}_1, \ldots, \mathsf{M}_\mathsf{n}) & \textit{if } \mathsf{M} \twoheadrightarrow_{\beta_2} \mathsf{x}\mathsf{M}_1 \ldots \mathsf{M}_\mathsf{m}, \\ \perp & \textit{otherwise.} \end{cases}$$

The unique map from ξ_{LLT} into the final $\mathsf{S}_{\mathsf{LLT}}$-coalgebra is given by the restriction of the map LLT_α from Definition 6.10 and maps the equivalence class of an infinitary λ-term to its Lévy-Longo tree.

The weak head bisimulation can be defined as the kernel pair of the map

$$\mathsf{LLT}_\alpha : \Lambda^{\infty}_{\alpha} \to \mathcal{LLT}_\alpha.$$

Similarly to the case of head simulation, we can show that $\sim_{whnf}$ is a bisimulation in the sense of [AM89], namely we have an $\mathsf{S}_{\mathsf{LLT}}$-coalgebra structure on $\sim_{whnf}$ such that

$$\begin{tikzcd}
\Lambda^{\infty}_{\alpha} \arrow[d, "\xi_{\mathsf{LLT}}"'] & \sim_{whnf} \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d] & \Lambda^{\infty}_{\alpha} \arrow[d, "\xi_{\mathsf{LLT}}"] \\
\mathsf{S}_{\mathsf{LLT}}(\Lambda^{\infty}_{\alpha}) & \mathsf{S}_{\mathsf{LLT}}(\sim_{whnf}) \arrow[l, "\mathsf{S}_{\mathsf{LLT}}(\pi_1)"] \arrow[r, "\mathsf{S}_{\mathsf{LLT}}(\pi_2)"'] & \mathsf{S}_{\mathsf{LLT}}(\Lambda^{\infty}_{\alpha})
\end{tikzcd} \tag{6.9}$$

The commutativity of the above diagram means that the weak head bisimulation $\sim_{whnf}$ is a binary relation on $\Lambda^{\infty}_{\alpha}$ such that for all M and N, if $\mathsf{M} \sim_{whnf} \mathsf{N}$, the following hold:

(1) $\mathsf{M} \twoheadrightarrow_{\beta_2} \lambda \mathsf{x}.\mathsf{M}$ then $\mathsf{N} \twoheadrightarrow_{\beta_2} \lambda \mathsf{x}.\mathsf{N}$ and $\mathsf{M} \sim_{whnf} \mathsf{N}$.
(2) $\mathsf{M} \twoheadrightarrow_{\beta_2} \mathsf{x}\mathsf{M}_1 \ldots \mathsf{M}_\mathsf{m}$ then $\mathsf{N} \twoheadrightarrow_{\beta_1} \mathsf{x}\mathsf{N}_1 \ldots \mathsf{N}_\mathsf{m}$ and $\mathsf{M}_\mathsf{i} \sim_{whnf} \mathsf{N}_\mathsf{i}$ for all $1 \leq i \leq m$.

Example 6.14 (Berarducci trees up to α-equivalence)**.** By Theorem 5.4, the final coalgebra of $\mathsf{S}_{\mathsf{BerT}}$ exists where

$$\mathsf{S}_{\mathsf{BerT}}(X) = [\mathcal{V}]X + \{\perp\} \times \mathsf{List}(X) + \mathcal{V} \times \mathsf{List}(X) + \mathsf{Stream}(X)$$

However, we cannot apply Corollary 5.35 in this case because our current definition of binding signatures does not include infinite products, e.g. Stream.

6.5. **Infinitely Many Free Variables.** In this section we follow a suggestion from Pitts to treat the case of terms with infinitely many free variables. In Theorem 5.19 we proved that $(\Lambda^\infty/{=_\alpha}, d^\infty_\alpha)$ is isomorphic to $(\Lambda/{=_\alpha})^\infty$ provided that $\mathcal{V}$ is uncountable. However, the isomorphism does not hold if $\mathcal{V}$ is countable (Example 5.20). Moreover, even if $\mathcal{V}$ is countable and the set $(\Lambda/{=_\alpha})^\infty$ can be equipped with a permutation action, $(\Lambda/{=_\alpha})^\infty$ is not a nominal set, since the terms with infinitely many free variables are not finitely supported. One way to deal with terms with infinitely many free variables, without leaving the world of nominal sets, is to extend the calculus with constants and to regard the free variables as constants.

Let $\mathcal{C}$ be a countable set of names. The sets $\Lambda(\mathcal{C})$ and $\Lambda(\mathcal{C})^\infty$ are the sets Λ and Λ^∞ extended with a set of constants $\mathcal{C}$. We can equip the set $\mathcal{C}$ with the trivial permutation action and consider the functor $\mathsf{L}_\alpha + \mathcal{C} : \mathsf{Nom} \to \mathsf{Nom}$. The following proposition follows from [Pit11, Theorem 5.12], or [Pit06, Theorem 5.1, Remark 5.3].

Proposition 6.15. *The initial algebra of $\mathsf{L}_\alpha + \mathcal{C}$ is the nominal set $\Lambda(\mathcal{C})/{=_\alpha}$ of λ-terms extended with $\mathcal{C}$-constants up to α-equivalence.*

The following proposition follows from Corollary 5.35 since $\mathsf{L}_\alpha + \mathcal{C}$ is a functor obtained from a binding signature (see Definition 5.8 and Proposition 5.6).

Proposition 6.16. *The final coalgebra of $\mathsf{L}_\alpha + \mathcal{C}$ is isomorphic to the nominal sets $(\Lambda(\mathcal{C})/{=_\alpha})^\infty_{\mathsf{fs}}$ and $\Lambda(\mathcal{C})^\infty_{\mathsf{ffv}}/{=_\alpha}$.*

Proposition 6.17. *Let $\rho : \mathcal{V} \to \mathcal{C}$ be a bijection. We have an isomorphism Tr between $\Lambda/{=_\alpha}$ and the set of closed terms in $(\Lambda(\mathcal{C})/{=_\alpha})^0$.*

Proof. The map $\mathsf{Tr} : \Lambda \to \Lambda(\mathcal{C})^0$ is defined inductively as follows:

$$\begin{array}{lcl} \mathsf{Tr}(x) & = & \rho(x) \\ \mathsf{Tr}(MN) & = & \mathsf{Tr}(M)\mathsf{Tr}(N) \\ \mathsf{Tr}(\lambda x.M) & = & \lambda x.\mathsf{Tr}(M)[x/\rho(x)] \end{array}$$

where $\mathsf{Tr}(M)[x/\rho(x)]$ is the result of replacing $\rho(x)$ by x in $\mathsf{Tr}(M)$.

It is easy to check that if $M =_\alpha N$ then $\mathsf{Tr}(M) =_\alpha \mathsf{Tr}(N)$. Hence, Tr can be defined on equivalence classes and we have $\mathsf{Tr} : \Lambda/{=_\alpha} \to (\Lambda(\mathcal{C})/{=_\alpha})^0$. Note that the map Tr is not finitely supported. For example, $\mathsf{Tr}((xy) \cdot x) \neq (xy) \cdot \mathsf{Tr}(x)$ if $\rho(x) \neq \rho(y)$. Hence, we cannot apply Pitts' alpha structural induction principle.

The inverse of Tr is defined as follows. Given an α-equivalence class $[M]_\alpha \in (\Lambda(\mathcal{C})/{=_\alpha})^0$, we can find a representative $M \in \Lambda(\mathcal{C})^0$ such that the set of bound variables of M is disjoint from the image under ρ^{-1} of the constants occurring in M. We put $\mathsf{Tr}^{-1}(M)$ to be the equivalence class of the term obtained by replacing each constant b occurring in M by $\rho^{-1}(b)$. □

Proposition 6.18. *The set* $(\Lambda/{=_\alpha})^\infty$ *is isomorphic to* $(\Lambda(\mathcal{C})^\infty_{\mathrm{ffv}}/{=_\alpha})^0$.

Proof. By the universal property of the Cauchy completion, Tr is extended to an isomorphism between $(\Lambda/{=_\alpha})^\infty$ and the completion of $(\Lambda(\mathcal{C})/{=_\alpha})^0$. The proof is complete by observing that the isomorphism between $(\Lambda(\mathcal{C})/{=_\alpha})^\infty_{\mathsf{fs}}$ and $\Lambda(\mathcal{C})^\infty_{\mathrm{ffv}}/{=_\alpha}$ (see Proposition 6.16) cuts down to an isomorphism between the completion of $(\Lambda(\mathcal{C})/{=_\alpha})^0$ and $(\Lambda(\mathcal{C})^\infty_{\mathrm{ffv}}/{=_\alpha})^0$. Explicitly, given a Cauchy sequence in $\Lambda/{=_\alpha}$, the image under Tr is a *finitely supported* Cauchy sequence in $(\Lambda(\mathcal{C})/{=_\alpha})^0$, and thus also in $\Lambda(\mathcal{C})/{=_\alpha}$. Thus it converges to a unique element of $(\Lambda(\mathcal{C})^\infty_{\mathrm{ffv}}/{=_\alpha})^0$. Conversely, given $M \in (\Lambda(\mathcal{C})^\infty_{\mathrm{ffv}}/{=_\alpha})^0$, we consider the truncations M^n. Their translations $\mathsf{Tr}^{-1}(M^n)$ in $\Lambda/{=_\alpha}$ form a Cauchy sequence and we map M to its limit. □

Substitution is defined on $\Lambda(\mathcal{C})^\infty_{\mathrm{ffv}}/{=_\alpha}$ by instantiating Definition 6.2. Now, β-reduction restricts to $(\Lambda(\mathcal{C})^\infty_{\mathrm{ffv}}/=_\alpha)^0$ and can be defined on $(\Lambda/=_\alpha)^\infty$ via the translation Tr. We illustrate how this works with an example.

Example 6.19. Consider the term $(\lambda x_0 x_1.x_0 x_1)\mathsf{allfv}$, where $\mathsf{allfv} = x_0(x_1(x_2(\ldots)))$ given in (1.4). Suppose $\mathcal{C} = \{c_0, c_1, \ldots\}$ and $\rho(x_i) = c_i$ for all i.

Then,

$$\mathsf{Tr}((\lambda x_0 x_1.x_0 x_1)\mathsf{allfv}) = (\lambda x_0 x_1.x_0 x_1)\mathsf{allconst}$$

where $\mathsf{allconst} = c_0(c_1(c_2(\ldots)))$. The translated term does not contain free variables, but bound variables and constants only. It is important to stress the fact that the constants represent the free variables of the original term. We can now safely perform the β-step

$$(\lambda x_0 x_1.x_0 x_1)\mathsf{allconst} \to_\beta (\lambda x_1.x_0 x_1)[x_0 := \mathsf{allconst}] = \lambda x_1.\mathsf{allconst}\ x_1. \tag{6.10}$$

This β-step is possible because substitution is defined on the set $(\Lambda(\mathcal{C})^\infty_{\mathrm{ffv}}/{=_\alpha})$. According to Proposition 6.18, the equivalence class $[\lambda x_1.\mathsf{allconst}\ x_1]_\alpha$ translates back to an element in $(\Lambda/{=_\alpha})^\infty$ which is the limit of the Cauchy sequence $([\lambda x_n.x_0(x_1 \ldots (x_{n-1} *))x_n]_\alpha)_n$. Similar to Example 5.20, it does not have any preimage under $[-]_\alpha : \Lambda^\infty \to (\Lambda/{=_\alpha})^\infty$.

7. Related and Future Work

The problem of having insufficiently many fresh variables does not arise if we use de Bruijn indices [dB72, Dup00]. However, it is unclear whether using de Bruijn indices could lead to a coalgebraic treatment of the corecursion principle.

It would also be interesting to investigate nominal coalgebraic data types with infinitely many free variables based on either Section 6.5 or on variations of nominal sets allowing countable supports, see e.g. [Che06, DG12]. This could have applications to the semantics of processes which are able to generate infinitely many fresh names.

Using the 'same' endofunctor as [GP99], but on a different category, namely the category $\mathsf{Set}^{\mathbb{F}}$ of presheaves on finite sets, [FPT99] also exhibits finite λ-terms as an initial algebra. Roughly speaking, the difference between Nom and $\mathsf{Set}^{\mathbb{F}}$ is that the latter already comes equipped with a notion of substitution, see [Sta07, Section 7.3] for details. [MU04] further develop substitution for algebraic and coalgebraic datatypes over presheaf-categories and describe the set of infinitary λ-terms as a final coalgebra. [AMV11] furthermore study the so-called rational fixed point, again over $\mathsf{Set}^{\mathbb{F}}$, as a semantic universe for solutions of higher-order recursion schemes.

In [HvO03], the authors define a calculus with an operator called *adbmal* to deal with α-conversion. This operator removes the scope of a variable. It will be interesting to extend this calculus for infinite terms. Using this operator, it would give an alternative approach to dealing with the problem of having insufficient fresh variables.

Nominal Isabelle provides infrastructure for declaring nominal data types and defining recursive functions over them [Urb08]. Proposals for codata types in Isabelle are presented in [TPB12]. It will be nice to include nominal codata types in Isabelle in order to formalise the proofs of some theorems on λ-calculus concerning Böhm trees and infinitary λ-calculus.

Nominal extensions of typed λ-calculus have been proposed in [Pit10, Che09] for system T, in [Che12] for LF and in [WSA09] for the Calculus of Inductive Constructions. Further research could include a study of typing that combines nominal syntax with coinductive data types.

The corecursion principle presented in this paper cannot handle infinitary meta-terms as defined for Infinitary Combinatory Reduction Systems (iCRS) [KS11]. It will be interesting to prove an α-coinduction principle that includes meta-variables and meta-terms. However, it is not straightforward to define α-equivalence on them [FG07].

One could also study how to extend the notion of binding signature and Corollary 5.35 to include infinite products for representing the set of Berarducci trees up to α (see Example 6.14). Another possible solution, suggested by a referee, would be to use a version of Bekič lemma [LS81, BBvGvdW95, Fre92] and to replace a nested final coalgebra by a non-nested many-sorted one.

It will also be worthwhile to study α-corecursion principles for sets of infinitary terms obtained from alternative metrics such as the 001- and 101-metrics [KKSdV97] or the metric that captures the infinite normal forms of reactive programs [SdV12]. Also, meta-terms for Infinitary Combinatory Reduction Systems that satisfy the finite property chain can be defined using an alternative metric ([KS11, page 20]).

Acknowledgements

We are grateful to Andy Pitts for suggesting us to treat the infinitely many free variables of a term as constants. We would also like to thank Christian Urban for helpful discussions. Finally, we acknowledge insightful improvements suggested by the referees.

References

[Abr90] S. Abramsky. The lazy lambda calculus. In *Research Topics in Functional Programming*, pages 65–116. Addison-Wesley, 1990.

[Adá03] J. Adámek. On final coalgebras of continuous functors. *Theoretical Computer Science*, 294(1/2):3–29, 2003.

[AM89] P. Aczel and N. P. Mendler. A final coalgebra theorem. In *CTCS*, pages 357–365, 1989.

[AMV11] J. Adámek, S. Milius, and J. Velebil. Semantics of higher-order recursion schemes. *Logical Methods in Computer Science*, 7(1), 2011.

[AN80] A. Arnold and M. Nivat. The metric space of infinite trees. algebraic and topological properties. *Fundamenta Informaticae*, 4:445–476, 1980.

[AO93] S. Abramsky and C.-H. Luke Ong. Full abstraction in the lazy lambda calculus. *Information and Computation*, 105(2):159–267, 1993.

[AR94] J. Adámek and J. Rosicky. *Locally Presentable and Accessible Categories*. Cambridge University Press, 1994.

[Bar84] H. P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*. North-Holland, Amsterdam, Revised edition, 1984.

[Bar99] M. Barr. Terminal coalgebras for endofunctors on sets. *Theoretical Computer Science*, 114(2):299–315, 1999.

[BBKL12] M. Bojanczyk, L. Braud, B. Klin, and S. Lasota. Towards nominal computation. In *POPL*, pages 401–412, 2012.

[BBvGvdW95] R. C. Backhouse, M. Bijsterveld, R. van Geldrop, and J. van der Woude. Categorical fixed point calculus. In *Category Theory and Computer Science*, pages 159–179, 1995.

[Ber96] A. Berarducci. Infinite λ-calculus and non-sensible models. In *Logic and algebra (Pontignano, 1994)*, pages 339–377. Dekker, New York, 1996.

[Che06] J. Cheney. Completeness and Herbrand theorems for nominal logic. *Journal of Symbolic Logic*, 71(1):299–320, 2006.

[Che09] J. Cheney. A simple nominal type theory. *Electronic Notes on Theoretical Computer Science*, 228:37–52, 2009.

[Che12] J. Cheney. A dependent nominal type theory. *Logical Methods in Computer Science*, 8(1), 2012.

[dB72] N. G. de Bruijn. Lambda Calculus Notation with Nameless Dummies, a Tool for Automatic Formula Manipulation, with applications to the Church-Rosser Theorem. *Indagationes Mathematicae*, 34:381–392, 1972.

[DG12] G. Dowek and M. J. Gabbay. Permissive-nominal logic: First-order logic over nominal terms and sets. *ACM Transactions on Computational Logic*, 13(3):A1–A37, 2012. http://www.odysci.com/article/1010113018587267.

[Dup00] Y. D. Duppen. A coalgebraic approach to lambda calculus. Master's thesis, Vrije Universiteit Amsterdam, 2000.

[FG07] M. Fernández and M. J. Gabbay. Nominal rewriting. *Information and Computation*, 205(6):917–965, 2007.

[FPT99] M. Fiore, G. D. Plotkin, and D. Turi. Abstract syntax and variable binders. In *LICS*, pages 193–202, 1999.

[Fre92] P. Freyd. Remarks on algebraically compact categories. In *Applications of Categories in Computer Science*, volume 77 of *London Math. Soc. Lecture Notes Series*, pages 95 – 106. Cambridge University Press, 1992.

[GP99] M. J. Gabbay and A. M. Pitts. A new approach to abstract syntax involving binders. In *LICS*, pages 214–224, 1999.

[GP01] 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.

[Hof99] M. Hofmann. Semantical analysis of higher-order abstract syntax. In *LICS*, pages 204–213, 1999.

[HvO03] D. Hendriks and V. van Oostrom. Adbmal. In *CADE*, pages 136–150, 2003.

[Jec73] T. Jech. *The axiom of choice*. North-Holland, 1973.

[KdV03] J. R. Kennaway and F. J. de Vries. Infinitary rewriting. In Terese, editor, *Term Rewriting Systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*, pages 668–711. Cambridge University Press, 2003.

[Kel82] M. Kelly. *Basic Concepts of Enriched Category Theory*. Number 64 in London Mathematical Society Lecture Notes. Cambridge University Press, 1982.

[KKSdV95] J. R. Kennaway, J. W. Klop, M. R. Sleep, and F. J. de Vries. Infinite lambda calculus and Böhm models. In *RTA*, pages 257–270, 1995.

[KKSdV97] J. R. Kennaway, J. W. Klop, M. R. Sleep, and F. J. de Vries. Infinitary lambda calculus. *Theoretical Computer Science*, 175(1):93–125, 1997.

[Koc72] A. Kock. Strong functors and monoidal monads. *Archiv der Mathematik*, 23(1):113–120, 1972.

[KPSdV12] A. Kurz, D. Petrişan, P. G. Severi, and F. J. de Vries. An alpha-corecursion principle for the infinitary lambda calculus. In *CMCS*, pages 130–149, 2012.

[KS11] J. Ketema and J. Grue Simonsen. Infinitary combinatory reduction systems. *Information and Computation*, 209(6):893–926, 2011.

[Lév76] J.-J. Lévy. An algebraic interpretation of the $\lambda\beta K$-calculus, and an application of a labelled λ-calculus. *Theoretical Computer Science*, 2(1):97–114, 1976.

[Lon83] G. Longo. Set-theoretical models of λ-calculus: theories, expansions, isomorphisms. *Annals of Pure and Applied Logic*, 24(2):153–188, 1983.

[LS81] D. J. Lehmann and M. B. Smyth. Algebraic specification of data types: A synthetic approach. *Mathematical Systems Theory*, 14:97–139, 1981.

[Mos01] L. S. Moss. Parametric corecursion. *Theoretical Computer Science*, 260:139–163, 2001.

[MU04] R. Matthes and T. Uustalu. Substitution in non-wellfounded syntax with variable binding. *Theoretical Computer Science*, 327:155–174, 2004.

[Pet12] D. Petrişan. *Investigations into Algebra and Topology over Nominal Sets*. PhD thesis, University of Leicester, 2012.

[Pit03] A. M. Pitts. Nominal logic, a first order theory of names and binding. *Information and Computation*, 186:165–193, 2003.

[Pit05] A. M. Pitts. Alpha-structural recursion and induction. In *TPHOLs*, pages 17–34, 2005.

[Pit06] A. M. Pitts. Alpha-structural recursion and induction. *Journal of the ACM*, 53:459–506, 2006.

[Pit10] A. M. Pitts. Nominal system T. In *POPL*, pages 159–170, 2010.

[Pit11] A. M. Pitts. Nominal sets. Notes for Midlands Graduate School, 2011.

[Pit13] A. M. Pitts. *Nominal Sets*. Cambridge University Press, 2013.

[Sal01] A. Salibra. Nonmodularity results for lambda calculus. *Fundamenta Informaticae*, 45:379–392, 2001.

[SdV11] P. G. Severi and F. J. de Vries. Weakening the axiom of overlap in the infinitary lambda calculus. In *RTA*, pages 313–328, 2011.

[SdV12] P. G. Severi and F. J. de Vries. Pure type systems with corecursion on streams: from finite to infinitary normalisation. In *ICFP*, pages 141–152, 2012.

[Sta07] S. Staton. Name-passing process calculi: operational models and structural operational semantics. Technical Report UCAM-CL-TR-688, University of Cambridge, Computer Laboratory, June 2007.

[TPB12] D. Traytel, A. Popescu, and J. C. Blanchette. Foundational, compositional (co)datatypes for higher-order logic: Category theory applied to theorem proving. In *LICS*, pages 596–605, 2012.

[Urb08] C. Urban. Nominal techniques in Isabelle/HOL. *Journal of Automated Reasoning*, 40(4):327–356, 2008.

[WSA09] E. M. Westbrook, A. Stump, and E. Austin. The calculus of nominal inductive constructions: an intensional approach to encoding name-bindings. In *LFMTP*, pages 74–83, 2009.