Friedrich-Alexander-Universität
Erlangen-Nürnberg

T.cs

Chair for Computer Science 8
Theoretical Computer Science

The rational fixed point in nominal sets and its application to infinitary lambda-calculus

Master Project in Computer Science

Thorsten Wißmann

Advisors:

PD Dr. Stefan Milius

Prof. Dr. Lutz Schröder

October 9, 2014

Contents

1 Introduction

One of the most important concepts in computer science is the λ-calculus. This notion of computation – or call it programming language – is very powerful since it is known to be Turing-complete. On the other hand the λ-calculus is simple in the sense that its syntax consists of only three constructs: variables, abstraction – or function definition – and function application. And the semantics consists of only two concepts: α-conversion – renaming of bound variables – and β-conversion – execution of function applications.[1] Though being simple, programming complex notions known from high level programming languages is not much effort and usually just needs a few characters to write – think of notions as booleans, if-else, natural numbers, arithmetic operations up to exponentiation, lists including mapping and folding, recursion and so on.

In all that shine, there is one detail that dims the shiny light of the beauty of λ-calculus: when performing α- or β-conversion one has to replace bound variables by other variables but one has to take care that free variables are not accidentally bound. So the question is how to define functions inductively on λ-terms that can not mess up the variable bindings. Recent developments brought a new characterization for binding signatures solving this problem: the theory of *nominal sets*.

While [GP99] refer to FM-Sets, which stands for Fraenkel and Mostowski, [GP99] introduced a notion of binding and algebraic data types involving binding on FM-Sets, which is considered the birth of *nominal sets*. Informally speaking, a nominal is a set together with a notion of renaming variables. This allows to define α-equivalence in a general way and furthermore to treat terms modulo α-equivalence. Previously, λ-terms could be defined as the initial algebra of the Set-functor

$$LX = \mathcal{V} + \mathcal{V} \times X + X \times X.$$

where $\mathcal{V}$ is a set of variable names and the three cases of the coproduct describe variables, abstraction and application. With nominal sets, one is able to express the binding by $[\mathcal{V}]X$ denoting the set of equivalence classes on the abstractions $\mathcal{V} \times X$. With that, the λ-terms up to α-equivalence could be characterized as the initial algebra of the functor on the category of nominal sets

$$L_\alpha X = \mathcal{V} + [\mathcal{V}]X + X \times X.$$

More recently, it was shown in [KPSdV13] that the final coalgebra for the same functor L_α describes the potentially *infinite* λ-terms with finitely many free variables up to α-equivalence.

In general for signature functors, beside the initial algebra – the least fixed point – and the final coalgebra – the greatest fixed point – there is another kind of fixed point: *the rational fixed point*. While the initial algebra describes finite, and the final coalgebra describes possibly infinite, the rational fixed point talks about possibly infinite structures which can be described in a finite way. For signature functors on Set and other categories, it is known that the rational fixed point contains exactly the rational trees of the respective signature, i.e. possibly infinite trees that have only finitely many subtrees, see [AMV06, AMV11]. More intuitively, consider the Set-functor

$$FX = 10 \times X + 1$$

[1] Depending on the application, a third conversion η might be of interest. But η-conversion is neither needed for Turing-completeness nor for this work.

with $10 = \{0, 1, \ldots, 9\}$ and 1 denoting the singleton set. The least fixed point of F contains the finite lists of decimal digits, or one could interpret it as decimal representations of natural numbers. The greatest fixed point of F is the set of possibly infinite streams of decimal digits, or say decimal representations of real numbers in the interval $[0, 1]$, e.g. 0.14, $0.3141\ldots$, $0.5\overline{13}$, $0.\overline{9}$. The rational fixed point of F are the possibly infinite streams with finitely many substreams, i.e. eventually periodic streams. So the rational fixed point of F are decimal representations of rational numbers in the interval $[0, 1]$.

This work characterizes the rational fixed point in the category of nominal sets for functors dealing with binding. Therefore, we assume that the reader is familiar with many commonly known notations which are listed in section A.1.

After recalling the necessary preliminaries on finitary and infinitary λ-calculus in section 2, nominal sets are recalled in section 3. While doing that, crucial properties for the later characterization are shown. Then follows the actual description of the rational fixed point for endofunctors on the category of nominal sets in section 4. Like in other categories, here the rational fixed point is the collection of images of orbit-finite coalgebras in the greatest fixed point – the final coalgebra. For the concrete functor L_α it is shown that its rational fixed point is the set of rational λ-terms modulo α-equivalence. As an application, it is shown, how to adapt the definition for the substitution function from infinite λ-terms to rational λ-terms.

2 Introduction to λ-calculus

There are many flavours of the λ-calculus: it spans from the untyped version to very many different type systems, covering different extensions to the language for example abstraction over types, function definitions, or pattern matching.

In this work, we are interested in the most simple version of λ-calculus and its infinitary variant. The most simple version – the untyped finitary λ-calculus – is the well-known system for computable functions.

2.1 Finitary λ-calculus

Let us recall the basics, see [Sel08] for more details. The calculus consists of only three constructs: variables, λ-abstraction, and application. So we can build λ-terms T, S using the grammar

$$T, S ::= \lambda x.T \mid TS \mid x \quad \text{with } x \in \mathcal{V}.$$

The set of *finitary λ-terms* is denoted by the set Λ. Here, $\mathcal{V}$ is any countable set of variable names, e.g.

$$\{v_0, v_1, v_2, \ldots\} \text{ or } \{\text{a,b,c}, \ldots, \text{len}, \ldots, \text{counter}, \ldots\} \text{ or } \ldots$$

The intuition is that $\lambda x.T$ denotes a function which expects some parameter P and maps P to T, where the unbound occurrences of x in the term T stand for P. The parameter P can be an arbitrary λ-term. One says that in $\lambda x.T$, the variable x is bound. In the term $\lambda x.(xy)$ the variable x is bound while the variable y is called free. In general, the set of free variables $\mathsf{FV}(t)$, $\mathsf{FV} : \Lambda \to \mathcal{P}(\mathcal{V})$, of a term is defined as

$$\mathsf{FV}(\lambda x.T) = \mathsf{FV}(T) \setminus \{x\}, \quad \mathsf{FV}(TS) = \mathsf{FV}(T) \cup \mathsf{FV}(S), \quad \mathsf{FV}(x) = \{x\}, \text{ with } x \in \mathcal{V}.$$

So one can build for example $(\lambda x.(\lambda y.x))$, $((\lambda x.(xx))\,(\lambda x.(xx)))$, or $(\lambda x.(\lambda y.(\lambda z.((x\,z)\,(y\,z)))))$. As usual, lower case letters denote variables and upper case letters denote λ-Terms. Furthermore, we assume parentheses are left-associative and the scope of λ-abstraction is as large as possible, so the previous examples can also be rewritten as

$$\lambda x.\lambda y.x, \quad (\lambda x.xx)\,\lambda x.xx, \quad \lambda x.\lambda y.\lambda z.x\,z\,(y\,z).$$

It is crucial that renaming bound variables essentially gives the same term, i.e. all the terms $\lambda x.\lambda y.x$, $\lambda z.\lambda y.z$, $\lambda z.\lambda x.z$, $\lambda y.\lambda x.y$ behave identically. This is formalized by the notion of variable substitution and α-reduction:

$$\begin{aligned}
x[x \mapsto R] &= R, \\
(TS)[x \mapsto R] &= T'S', \text{ if } T[x \mapsto R] = T' \text{ and } S[x \mapsto R] = S' \\
(\lambda v.T)[x \mapsto R] &= \begin{cases} T[x \mapsto R] & \text{if } x \neq v \notin \mathsf{FV}(R) \\ T & \text{if } x = v \end{cases} \\
\lambda x.T &\to_\alpha \lambda y.T', \text{ if } T' = T[x \mapsto y] \\
TS &\to_\alpha T'S, \text{ if } T \to_\alpha T' \\
TS &\to_\alpha TS', \text{ if } S \to_\alpha S'
\end{aligned}$$

It is worth noticing that variable substitution is a partial function, e.g. $(\lambda x.y)[y \mapsto x]$ is not defined because we are not allowed to "make y bound". There are two solutions for this problem. The first is to informally assume all the bound variables to be always "fresh enough". The other solution – the formal flavor of the former – is to consider λ-Terms only up to α-equivalence, that is the least equivalence relation containing $\to_\alpha$. E.g. one writes $\lambda x.x =_\alpha \lambda z.z$. When doing substitution in an α-equivalence class of abstractions, we can pick any representative $(\lambda v.T)$ s.t. the variable for the parameter v is not free in R, $v \notin \mathsf{FV}(R)$. This would make substitution total, but it requires λ-terms up to α-equivalence. An coalgebraic specification of λ-terms up to α-equivalence together with the total substitution function is described in [KPSdV13] and will be recalled in the later section 4.2.

Using the substitution, the main notion of computation is the β-reduction defined as

$$(\lambda x.T)P \to_\beta T', \text{ if } T' = T[x \mapsto P]; \quad TS \to_\beta T'S, \text{ if } T \to_\beta T'; \quad TS \to_\beta TS', \text{ if } S \to_\beta S'.$$

For example we have $(\lambda x.\lambda y.x)\, a\, b \to_\beta (\lambda y.a)\, b \to_\beta a$. For a being a free variable, it is clear that a cannot be reduced any further, i.e. a is a normal form. It is not hard to see, that there are terms without a normal form, e.g.

$$(\lambda x.xx)\,(\lambda x.xx) \to_\beta (\lambda x.xx)\,(\lambda x.xx)$$

This means there is some infinite reduction sequence starting at $(\lambda x.xx)(\lambda x.xx)$.

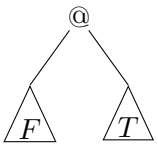
2.2 Infinitary λ-calculus

When examining infinite reduction sequences of finitary λ-terms, one can notice that there are sequences in which the terms will not stop growing. One prominent is Yf for f being any λ-term – e.g. some free variable – and Y being the fixed point combinator Y. So with $Y := (\lambda g.(\lambda x.g(x\,x))\,(\lambda x.g(x\,x)))$ we have:

$$\begin{aligned}(\lambda g.(\lambda x.g(x\,x))\,(\lambda x.g(x\,x)))f &\to_\beta (\lambda x.f(x\,x))\,(\lambda x.f(x\,x)) \to_\beta f((\lambda x.f(x\,x))\,(\lambda x.f(x\,x)))\\ &\to_\beta f(f((\lambda x.f(x\,x))\,(\lambda x.f(x\,x)))) \to_\beta \dots\end{aligned}$$

Informally speaking, this converges to an infinite term $f(f(f(\cdots)))$. This demands a notion of infinitary λ-calculus. To make it more intuitive, let us visualize λ-terms by their syntax trees:

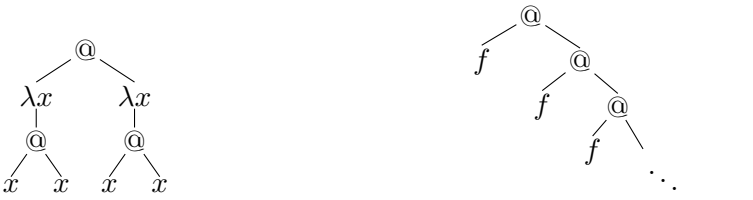
$v \mapsto v$ for $v \in \mathcal{V}$ $\qquad$ $\lambda x.T \mapsto$ 

$\qquad$ $FT \mapsto$ 

Variables are represented by leaves. E.g. $(\lambda x.xx)\ (\lambda x.xx)$ is represented by the tree in Figure 2.1(a). The infinite term $f(f(f(...))) \notin \Lambda$ is visualized by Figure 2.1(b). To emphasize that thinking of the trees is more intuitive, we will name λ-terms also λ-trees. Of course, infinite trees may have more than one infinite branch, and so may λ-trees. One way of defining infinite trees – namely via metric completion – was introduced by [AN80] and can be applied to λ-calculus as shown in [KPSdV13]:

Definition 2.1. Assuming an additional constant $\star$, not in the syntax of λ-calculus. The *truncation* of a λ-term $t \in \Lambda$ at depth $n \in \mathbb{N}$ is denoted by t^n and defined as:

$$t^0 = \star, \quad t^{n+1} = \begin{cases} x & \text{if } t = x \in \mathcal{V}, \\ \lambda x.s^n & \text{if } t = \lambda x.s \\ f^n s^n & \text{if } t = fs \end{cases}$$



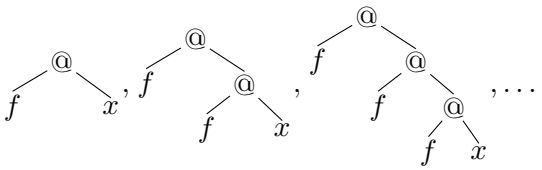
(a) Syntax tree of $(\lambda x.xx)\ (\lambda x.xx)$ (b) Syntax tree of $f(f(f(...)))$

Figure 2.1: Different syntax trees of λ-terms

Definition 2.2. Using the truncation we can define a metric $d : \Lambda \times \Lambda \to [0, 1]$

$$d(s,t) = \begin{cases} 0 & \text{if } s = t \\ 2^{-m} & \text{otherwise, where } m = \sup\{n \in \mathbb{N} \mid t^n = s^n\}. \end{cases}$$

Then we can define the set of possibly infinite λ-trees as the metric completion of Λ of finite trees with respect to the metric d. That means that we can think of infinitary λ-trees as Cauchy sequences of finitary λ-trees. For example, the tree in Figure 2.1(b) is exactly the limit of the Cauchy sequence



The notion of α-equivalence can be defined component-wise: two infinitary λ-trees are α-equivalent if the ith components of the corresponding Cauchy-sequence are α-equivalent for all i.

2.3 Rational λ-trees

Another observation is, that the above example has some self-similarity: when cropping the first f of $f(f(f(\cdots)))$ the identical term $f(f(f(\cdots)))$ is obtained. Furthermore, the term $f(f(f(\cdots)))$ has only one – in particular only finitely many – "subterm", namely itself. This leads to the notion of *rational λ-trees*.

Definition 2.3. A λ-tree having only finitely many subtrees (up to isomorphism) is called *rational*. A λ-tree modulo α-conversion, i.e. an α-equivalence class of λ-trees, is called rational if it contains at least one rational λ-tree.

Every finite λ-tree obviously is rational. Other examples for infinite rational λ-trees can be seen in Figure 2.2. The uplink from some node s to some other node r indicate that the entire tree starting at r occurs as a subtree of s. In other words, such a λ-tree with uplinks represents its tree unravelling, i.e. Figure 2.2(a) and Figure 2.2(b) both represent the same rational infinite λ-tree shown in Figure 2.1(b).

Things get more complicated, if abstractions come into play, as in Figure 2.2(c). Here the x clearly refers to the λx in the root, but some of the "copies" of x are bound by the λx in the left branch and other copies are bound to the abstraction in the root. Something similar can be observed in Figure 2.2(d), which has two free variables x, y, but all "copies" of x and y are bound by the previous copy of λx and λy respectively. Figure 2.2(e) shows some tree

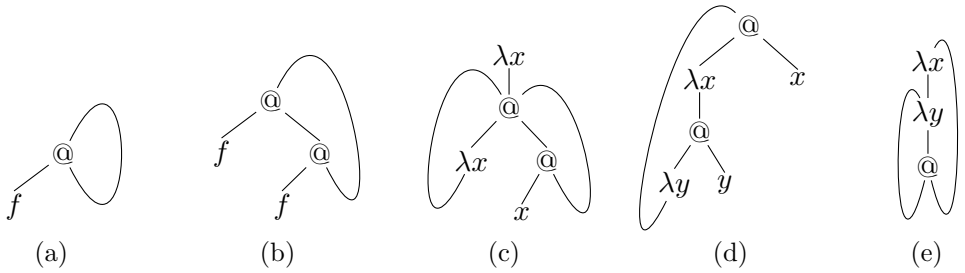


Figure 2.2: Examples for rational λ-trees

that consists of applications and abstractions only. One can think of some infinite λ-term $\lambda xy.(\lambda y \lambda y \ldots)(\lambda xy.(\lambda y \lambda y \ldots)(\lambda xy.(\lambda y \lambda y \ldots)\ldots))$ which is represented by that tree.

Note that in general the property that an infinite term has only finitely many subterms is not always fulfilled. Not only in theory but also when looking at the limit of computation sequences of finite λ-terms. Think of an λ-term P computing the infinite sequence of digits of the real number π, i.e. P converges to

$$(\text{cons } 3\,(\text{cons } 1\,(\text{cons } 4\,(\text{cons } 1\,(\text{cons } 5\,(\text{cons } 9\,(\cdots)))))))$$

with cons denoting the λ-term for prepending a single element to a list. Due to the irrationality of π, this λ-term cannot have only finitely many subterms but has infinitely many subterms.

It is worth noticing, that for the infinitary case, it is crucial whether one allows the λ-tree to have infinitely many free variables or not [KPSdV13]. In the following, we always assume that "infinite λ-tree" refers to some infinite λ-tree with only finitely many free variables. For the rational case, it is a lot easier: a rational λ-tree has only finitely many subtrees, so it can mention only finitely many variables. This means, that we do not need a distinction for the rational case.

Roughly, one can say that a λ-tree t is rational, iff it can be drawn as such a finite λ-tree with possible uplinks whose unravelling is t. This statement will be formalized in the following chapters. Such a finite tree with uplinks – i.e. some finite graph – will be formalized as a coalgebra. The tree unravelling is the unique homomorphism into the final coalgebra. To talk about λ-trees modulo α-equivalence, this will happen in the category of nominal sets, which is introduced in the following chapter before going further to the coalgebraic setting.

3 Basics on Nominal Sets

As mentioned in the introduction, nominal sets were introduced by Gabbay and Pitts [GP99]. In this section we will recall nominal sets and show some of their properties. As before, let $\mathcal{V}$ be any countable set of variable names. Using these variable names, the target is to build terms equipped with a notion of variable renaming. Therefore, let $\mathfrak{S}(\mathcal{V})$ be the group of *finite permutations* of $\mathcal{V}$. A permutation π is called finite iff $\{v \in \mathcal{V} \mid \pi(v) \neq v\}$ is finite. It is known that every $\pi \in \mathfrak{S}(\mathcal{V})$ is equal to a finite sequence of transpositions:

$$(a_1 b_1) \circ \cdots \circ (a_n b_n) \quad \text{with } a_i \neq b_i \text{ and } \pi a_i \neq a_i \text{ and } \pi b_i \neq b_i.$$

Definition 3.1. For a set X and a group $(G, \circ)$ with identity e, a map $\cdot : G \times X \to X$ is called a *group action* if

$$h \cdot (g \cdot x) = (h \circ g) \cdot x \quad \text{and} \quad e \cdot x = x \quad \text{for all } h, g \in G, x \in X.$$

Equivalently, $\cdot$ is called a group action if it is a group homomorphism from G to the group of all permutations of X.

Now consider a set X together with the finite permutations of $\mathcal{V}$ acting on it via $\cdot : \mathfrak{S}(\mathcal{V}) \times X \to X$. This means we can translate each permutation of variable names to a map on X.

Intuitively speaking, X is a set of terms and for a finite permutation of variable names π and for some term x, $\pi \cdot x$ denotes the new term obtained after renaming the variables in x according to π.

One might be interested in the variables "occurring" in $x \in X$. The given group action already provides enough information for this: we can check which variable renamings fix the term x. This is captured by the formal notion of support:

Definition 3.2. For $x \in X$ with $\cdot : \mathfrak{S}(\mathcal{V}) \times X \to X$, a set $S \subseteq \mathcal{V}$ *supports* x if

$$\text{for all } \pi \in \mathfrak{S}(\mathcal{V}) \text{ with } \pi(v) = v \text{ for all } v \in S \text{ we have } \pi \cdot x = x.$$

Some $x \in X$ is *finitely supported* if there is a finite $S \subseteq \mathcal{V}$ supporting x.

Definition 3.3. A *nominal set* is a set X together with a $\mathfrak{S}(\mathcal{V})$-action such that all elements of X are finitely supported.

Example 3.4. The set of variable names $\mathcal{V}$ itself has nominal structure by $\pi \cdot v = \pi(v)$. For each $v_i \in \mathcal{V}$, both the singleton $\{v_i\}$ and the set $\{v_0, \ldots, v_i\}$ support v_i.

Example 3.5. Each set X can be extended to a nominal by equipping it with the trivial action $\pi \cdot x = x \ \forall x \in X, \pi \in \mathfrak{S}(\mathcal{V})$. So each $x \in X$ can be thought of a term not containing any variable, i.e. the empty set supports x.

Notice that generally if $S \subseteq V$ supports $x \in X$, then $S \subseteq S'$ also supports $x \in X$. So S supporting x only means that not touching the members S will not modify the term x. But it is more interesting to talk about the variables actually occurring in x.

Definition 3.6. Let $\mathsf{supp}(x)$ be the smallest set supporting x. If $v \in \mathcal{V} \setminus \mathsf{supp}(x)$, we say that v is *fresh for* x, denoted by $v\#x$. In some cases we will refer to $\mathsf{supp}(x)$ as the free variables x.

Example 3.7. The set $\mathcal{P}_f(\mathcal{V})$, the finite subsets of $\mathcal{V}$, together with the point-wise action is a nominal. The support of each $u \in \mathcal{P}_f(\mathcal{V})$ is u itself.

$$\pi \cdot u = \{\pi \cdot x \mid x \in u\}, \quad \mathsf{supp}(u) = u.$$

Note that $\mathcal{P}(\mathcal{V})$ with the point-wise action is not a nominal because the infinite $\{v_0, v_2, v_4, \ldots\}$ does not have any finite support.

3.1 Equivariants

The maps preserving the structure of nominal sets are called equivariants and are defined as follows.

Definition 3.8. An *equivariant* $f : (X, \cdot) \to (Y, \star)$ is a map $f : X \to Y$ with

$$f(\pi \cdot x) = \pi \star f(x) \quad \text{for all } \pi \in \mathfrak{S}(\mathcal{V}), x \in X.$$

Let us look at some of their properties.

Remark 3.9. For any equivariant $f : (X, \cdot) \to (Y, \star)$, we have $\mathsf{supp}(f(x)) \subseteq \mathsf{supp}(x)$ for any $x \in X$. To see this, assume $\pi \in \mathfrak{S}(\mathcal{V})$ with $\pi(v) = v$ for all $v \in \mathsf{supp}(x)$. So we have $\pi \cdot f(x) = f(\pi \cdot x) = f(x)$, so $\mathsf{supp}(x)$ also supports $f(x)$ and thus $\mathsf{supp}(f(x)) \subseteq \mathsf{supp}(x)$.

The nominal sets – together with the equivariants as morphisms – form a category, denoted by Nom. Let us look at some example equivariant first.

Lemma 3.10. *The function $\mathsf{supp} : X \to \mathcal{P}_f(\mathcal{V})$ mapping each element to its (finite) support is an equivariant.*

Proof. To be able to show $\mathsf{supp}(\pi \cdot x) = \pi \cdot \mathsf{supp}(x)$, for arbitrary $x \in X, \pi \in \mathfrak{S}(\mathcal{V})$, consider for $u \in \mathcal{P}_f(\mathcal{V})$:

$$\begin{aligned}
u \text{ supports } x &\Leftrightarrow \forall \sigma \in \mathfrak{S}(\mathcal{V}) \colon \big(\forall v' \in u \colon \sigma \cdot v' = v'\big) \to \sigma \cdot x = x \\
&\Leftrightarrow \forall \sigma \in \mathfrak{S}(\mathcal{V}) \colon \big(\forall v \in \pi \cdot u \colon \sigma \cdot \pi^{-1} \cdot v = \pi^{-1} \cdot v\big) \to \sigma \cdot x = x \\
&\Leftrightarrow \forall \sigma \in \mathfrak{S}(\mathcal{V}) \colon \big(\forall v \in \pi \cdot u \colon \pi \cdot \sigma \cdot \pi^{-1} \cdot v = v\big) \to \sigma \cdot x = x \quad (\rho := \pi \cdot \sigma \cdot \pi^{-1}) \\
&\Leftrightarrow \forall (\pi^{-1} \cdot \rho \cdot \pi) \in \mathfrak{S}(\mathcal{V}) \colon \big(\forall v \in \pi \cdot u \colon \rho \cdot v = v\big) \to \pi^{-1} \cdot \rho \cdot \pi \cdot x = x \\
&\Leftrightarrow \forall (\pi^{-1} \cdot \rho \cdot \pi) \in \mathfrak{S}(\mathcal{V}) \colon \big(\forall v \in \pi \cdot u \colon \rho \cdot v = v\big) \to \rho \cdot \pi \cdot x = \pi \cdot x \\
&\Leftrightarrow \forall \rho \in \mathfrak{S}(\mathcal{V}) \colon \big(\forall v \in \pi \cdot u \colon \rho \cdot v = v\big) \to \rho \cdot \pi \cdot x = \pi \cdot x \\
&\Leftrightarrow \pi \cdot u \text{ supports } \pi \cdot x
\end{aligned}$$

Note that depending on the context, the operator $\cdot$ denotes the action on X, on $\mathcal{V}$, on $\mathcal{P}_f(\mathcal{V})$, or the composition of permutations.

The equality can be adapted to the smallest set supporting the respective element, hence $\pi \cdot \mathsf{supp}(x)$ is the smallest support for $\pi \cdot x$. So the action commutes with supp, i.e. supp is an equivariant. □

The result $\forall X \in \mathsf{Nom} : X \xrightarrow{\mathsf{supp}} \mathcal{P}_f(\mathcal{V}) \in \mathbf{mor}\,\mathsf{Nom}$ gives that $\mathcal{P}_f(\mathcal{V})$ is weakly terminal, i.e. for every nominal set X there exists an equivariant map $X \to \mathcal{P}_f(\mathcal{V})$. Though, $\mathcal{P}_f(\mathcal{V})$ is not the terminal object in Nom, but the set $1 = \{\emptyset\}$ is. Let us characterize more categorical notions of Nom:

Proposition 3.11. *The monomorphisms in* Nom *are precisely the injective equivariants.*

Proof. For the direction left to right consider a monomorphism $m : A \to B$ of Nom and its kernel, defined as $\ker m := \{(x, y) \in A^2 \mid mx = my\}$. This defines a nominal set with component-wise action $\pi \cdot (x, y) = (\pi x, \pi y)$. This is still in $\ker m$, because

$$m(\pi \cdot x) = \pi \cdot mx = \pi \cdot my = m(\pi \cdot y), \text{ thus } (\pi x, \pi y) \in \ker m.$$

The projections $\mathsf{outl}, \mathsf{outr} : \ker m \to A, (x, y) \mapsto x, (x, y) \mapsto y$ fulfill $m \circ \mathsf{outl} = m \circ \mathsf{outr}$. As m is a monomorphism, $\mathsf{outl} = \mathsf{outr}$. So for any $x, y \in A$, $mx = my$ implies $x = y$, i.e. m is injective.

For the converse direction, assume $m : A \to B$ is an injective equivariant. For any $f, g : X \to A$ with $mf = mg$, i.e. $\forall x \in X : m(fx) = m(gx)$, $fx = gx$ by the injectivity of m. Hence m is a monomorphism. □

Lemma 3.12. *For an equivariant $f : A \to B$, the image of f, $f[A] \subseteq B$, and $B \setminus f[A]$ are nominal sets with the restricted action of B.*

Proof. Consider $b \in f[A], \pi \in \mathfrak{S}(\mathcal{V}), b = f(a)$. Then $\pi \cdot b$ still is in $f[A]$, because

$$\pi \cdot b = \pi \cdot f(a) = f(\pi \cdot a).$$

So $B \setminus f[A]$ must be a nominal, too. Otherwise, for $b' \in B \setminus f[A]$ with $\pi \cdot b' \in f[A]$ we would have $b' = \pi^{-1} \cdot (\pi \cdot b') \in f[A]$. □

Proposition 3.13. *The epimorphisms in* Nom *are precisely the surjective equivariants.*

Proof. To see the direction left to right, let $e : A \to B$ be an epimorphism. Define two equivariants $f, g : B \to 2$, with 2 denoting the two element set $\{0, 1\}$:

$$g(b) := 0, \quad f(b) := \begin{cases} 0 & \text{if } \exists a \in A : e(a) = b \\ 1 & \text{otherwise.} \end{cases}$$

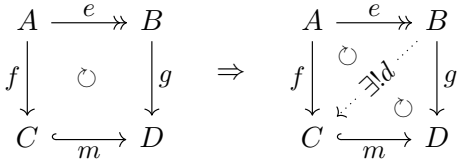
Obviously, g is an equivariant, since $g(\pi \cdot b) = 0 = \pi \cdot 0 = \pi \cdot g(b)$ for all $b \in B$. Similarly, f is an equivariant; if b is the image of $a \in A$, then:

$$f(\pi \cdot b) = f(\pi \cdot e(a)) = f(e(\pi \cdot a)) = 0 = \pi \cdot 0 = \pi \cdot f(b)$$

If b is not in the image of e, then neither is $\pi \cdot b$ by Lemma 3.12. So $f(\pi \cdot b) = 1 = \pi \cdot 1 = \pi \cdot f(b)$, and f is an equivariant. Obviously, $g \circ e = 0! = f \circ e$, so by the epimorphism property of e, $0! = g = f$. So $\forall b \exists a : e(a) = b$, i.e. e is surjective.

For the inverse direction, let $e : A \to B$ be a surjective equivariant and consider two arbitrary parallel equivariants $f, g : B \to C$ with $fe = ge$, i.e. $\forall a \in A : f(e(a)) = g(e(a))$. As e is surjective, this implies that $\forall b \in B : f(b) = g(b)$, so e is an epimorphism. □

Proposition 3.14. *In* Nom*, every epimorphism $e : A \to B$ is strong. I.e. for any monomorphism $m : C \to D$ and any commutative square $f : A \to C, g : B \to D, mf = ge$, there is a unique diagonal $d : B \to C$ with $de = f$, $md = g$.*



Proof. As e is surjective, it is sufficient to define d for all $e(a), a \in A$, namely as $m^{-1}[g(e(a))]$. Here, the preimage $m^{-1}[\ldots]$ is well-defined, because m is injective and $mf[A] = ge[A]$. For $m(c) = y \in D$ we have $m(\pi \cdot c) = \pi \cdot y$ and $m^{-1}[\pi \cdot y] = \pi \cdot c = \pi \cdot m^{-1}[y]$. This implies, that $d : B \to C$ is an equivariant:

$$d(\pi \cdot b) = m^{-1}[g(\pi \cdot b)] = m^{-1}[\pi \cdot g(b)] = \pi \cdot m^{-1}[g(b)] = \pi \cdot d(b).$$

The commutativity of the triangles follows from the definition of d:

$$m(d(b)) = m(m^{-1}[g(\pi \cdot b)]) = g(b), \quad d(e(a)) = m^{-1}[g(e(a))] = m^{-1}[m(f(a))] = f(a).$$

For the uniqueness, consider d' with $f = d'e$ and $g = md'$, i.e. $de = f = d'e$. As e is surjective, this gives $d = d'$. □

Remark 3.15. Nom is both complete and cocomplete. The coproducts and the finite products are in Set [Pit13, Section 2.2].

Remark 3.16. In contrast to Set, in Nom the non-empty monomorphisms are not always split-monos. Consider a monomorphism $m : X \hookrightarrow Y$. If there is a $y \in Y$ with $\mathsf{supp}(y)$ smaller than any $\mathsf{supp}(x)$, $x \in X$, then there is no equivariant $f : Y \to X$ at all, as it had to fulfill $\mathsf{supp}(f(y)) \subseteq \mathsf{supp}(y)$. For a concrete example where the split-mono property fails, consider $\mathrm{inl}: \mathcal{V} \hookrightarrow \mathcal{V} + 1$. There is no $f : \mathcal{V} + 1 \to \mathcal{V}$, because for $v_i = f(1)$ we have

$$v_i = f(1) = f((v_i\ v_j) \cdot 1) = (v_i\ v_j) \cdot f(1) = v_j \quad \forall v_j \in V.$$

3.2 Notion of finiteness

Definition 3.17 (orbit-finite)**.** Two elements $x, x' \in X$ are in the same orbit if $x' = \pi \cdot x$ for some finite permutation π. A nominal X is said to be *orbit-finite* if it consists of the union of finitely many orbits of X.

For example $\mathcal{V}$ has one orbit and $\mathcal{V} + \mathcal{V}$ has two orbits. In the following, many results about orbit-finite nominals are presented, which will help to understand the constructions in the next chapter.

Remark 3.18. If $x, x' \in X$ are in the same orbit then their images via an equivariant $f : X \to Y$ are still in the same orbit. This follows directly by:

$$x = \pi \cdot x' \quad \Rightarrow \quad f(x) = f(\pi \cdot x') = \pi \cdot f(x')$$

This gives that orbit-finite nominal sets are closed under quotients, i.e. for $e : X \twoheadrightarrow Y$ with X finitely presentable, Y is finitely presentable as well, because Y has at most as many orbits as X.

Recall that in general, an object X of a category $\mathcal{C}$ is called finitely presentable, if $\mathcal{C}(X, -)$ preserves filtered colimits. It has been shown by Petrişan [Pet11, Proposition 2.3.7], that the finitely presentable objects in Nom are precisely the orbit-finite nominals.

Even though an orbit of a nominal $(X, \cdot)$ may be infinitely large, we will see that we can reach only finitely many elements using arbitrary equivariants $f : X \to X$.

Lemma 3.19. *For any $x_1, x_2 \in X$ in the same orbit, the equation $|\mathsf{supp}(x_1)| = |\mathsf{supp}(x_2)|$ holds.*

Proof. Assuming $\pi \cdot x_1 = x_2$, the equivariant supp provides the equation

$$\mathsf{supp}(x_1) = \mathsf{supp}(\pi \cdot x_2) = \pi \cdot \mathsf{supp}(x_2) = \{\pi(x) \mid x \in \mathsf{supp}(x_2)\}.$$

So π can be considered as a map $\mathsf{supp}(x_2) \to \mathsf{supp}(x_1)$, which is injective because $\pi \in \mathfrak{S}(\mathcal{V})$ and is surjective because of the above equation and thus is a bijection between these two finite sets. □

Lemma 3.20. *For an element x of a nominal set X, there are at most $|\mathsf{supp}(x)|!$ many elements with support $\mathsf{supp}(x)$ in the orbit of x.*

Proof. Consider the set

$$\begin{aligned} N &= \{\pi \cdot x \mid \pi \in \mathfrak{S}(\mathcal{V}) \text{ and } \mathsf{supp}(\pi \cdot x) = \mathsf{supp}(x)\} \\ &= \{\pi \cdot x \mid \pi \in \mathfrak{S}(\mathcal{V}) \text{ and } \pi \cdot \mathsf{supp}(x) = \mathsf{supp}(x)\}. \end{aligned}$$

All π with $\pi \cdot x \in N$ have to fulfill the property $\mathsf{supp}(x) = \pi \cdot \mathsf{supp}(x)$. So π maps elements of $\mathsf{supp}(x)$ to elements of $\mathsf{supp}(x)$, i.e.

$$\pi(v) \in \mathsf{supp}(x) \Leftrightarrow v \in \mathsf{supp}(x).$$

So each such π can be factorized into two permutations $f_\pi, g_\pi \in \mathfrak{S}(\mathcal{V})$ with

$$f_\pi(v) = \begin{cases} \pi(v) & \text{if } v \in \mathsf{supp}(x) \\ v & \text{if } v \notin \mathsf{supp}(x) \end{cases}, \quad g_\pi(v) = \begin{cases} v & \text{if } v \in \mathsf{supp}(x) \\ \pi(v) & \text{if } v \notin \mathsf{supp}(x) \end{cases}$$

and with $f_\pi \circ g_\pi = g_\pi \circ f_\pi = \pi$. As $\mathsf{supp}(x)$ supports x, $g_\pi \cdot x = x$. So for all π with $\pi \cdot x \in N$,

$$\begin{aligned} N &= \{\pi \cdot x \quad\quad\; \mid \pi \in \mathfrak{S}(\mathcal{V}) \text{ and } \pi \cdot \mathsf{supp}(x) = \mathsf{supp}(x)\} \\ &= \{f_\pi \cdot g_\pi \cdot x \mid \pi \in \mathfrak{S}(\mathcal{V}) \text{ and } \pi \cdot \mathsf{supp}(x) = \mathsf{supp}(x)\} \\ &= \{f_\pi \cdot x \quad\quad\; \mid \pi \in \mathfrak{S}(\mathcal{V}) \text{ and } \pi \cdot \mathsf{supp}(x) = \mathsf{supp}(x)\} \end{aligned}$$

The set of permutations $\{f_\pi \mid \pi \cdot x \in N\}$ is bound by $|\mathsf{supp}(x)|!$, hence N also is. □

Lemma 3.21. *For a set $W \subseteq \mathcal{V}$ and an orbit $\mathcal{O}$ of the nominal set X there are only finitely many elements in $\mathcal{O}$ whose support is W.*

Proof. If there is no element in $\mathcal{O}$ with support W, there are in particular only finitely many. For some x in $\mathcal{O}$ with $\mathsf{supp}(x) = W$, there are only finitely many elements in $\mathcal{O}$ with the same support by Lemma 3.20. □

Lemma 3.22. *For a finite set $W \subseteq \mathcal{V}$ and an orbit $\mathcal{O}$ of the nominal set X there are only finitely many elements in $\mathcal{O}$ whose support is contained in W.*

Proof. W has only finitely many subsets, and for each such subset $W' \subseteq W$, everything follows from the previous lemma. □

Proposition 3.23. *If $f : X \to X$ is an equivariant and if x and $f(x)$ are in the same orbit, then $\mathsf{supp}(x) = \mathsf{supp}(f(x))$.*

Proof. Combine Lemma 3.19 with Remark 3.9 and use that the support is finite:

$$\left.\begin{aligned} |\mathsf{supp}(f(x))| &= |\mathsf{supp}(x)| \\ \mathsf{supp}(f(x)) &\subseteq \mathsf{supp}(x) \end{aligned}\right\} \Longrightarrow \mathsf{supp}(x) = \mathsf{supp}(f(x))$$ □

Note that even though $f : X \to X$ is an equivariant and $x, f(x)$ are in the same orbit, it might be that $x \neq f(x)$. For example consider the equivariant swap : $\mathcal{V} \times \mathcal{V} \to \mathcal{V} \times \mathcal{V}, (a, b) \mapsto (b, a)$. Then for $a \neq b$:

$$\mathsf{supp}(a, b) = \{a, b\} = \mathsf{supp}(b, a) \text{ but } (a, b) \neq (b, a).$$

The generalization of that example directly leads to the following.

Proposition 3.24. *For a fixed $x \in X$ there are finitely many possibilities for $f(x)$ if $f : X \to X$ is an equivariant and $f(x), x$ are in the same orbit. In fact, there are at most $|\mathsf{supp}(x)|!$ many possibilities, corresponding to the permutations on the support of x.*

Proof. For $x \in X$, define the subset of the orbit of x which is "reachable via equivariants" as

$$N = \{\pi \cdot x \mid \text{ there is some equivariant } f : X \to X \text{ and some } \pi \text{ with } f(x) = \pi \cdot x\}.$$

By Proposition 3.23, $\mathsf{supp}(f(x)) = \mathsf{supp}(x)$. By Lemma 3.20, there are at most $|\mathsf{supp}(x)|!$ elements with the same support as x in the same orbit. Hence, N is bound by $|\mathsf{supp}(x)|!$. □

All these properties of Nom will come in handy when constructing the rational fixed point in section 4.1 and when characterizing it in section 4.2.

4 The rational fixed point in Nom

Let us look at the rational fixed point in Nom in general, before taking a closer look on the concrete example of λ-calculus.

4.1 General observations for Nom-endofunctors

For the following, the reader is assumed being familiar with algebras and coalgebras as they are described in [JR97]. In general, the initial F-algebra for some endofunctor $F : \mathcal{C} \to \mathcal{C}$ talks about finite structures and the final F-coalgebra talks about possibly infinite structures. The compromise between them are possibly infinite structures that can be described in a finite way. I.e. while the greatest fixed point of F is defined as the final coalgebra νF, the rational fixed point ϱF is defined as a coalgebra which is final for all the coalgebras with finitely presentable carrier.

Theorem 4.1. *For an lfp category $\mathcal{C}$ and an finitary $F : \mathcal{C} \to \mathcal{C}$ preserving monos, let $\mathsf{CoAlg}F$ denote the category of F-coalgebras and $\mathsf{CoAlg}_f F$ denote the category of F-coalgebras (B, b) with B finitely presentable. The colimit*

$$\varrho F := \operatorname{colim} \mathsf{CoAlg}_f F \quad \text{in } \mathsf{CoAlg}F.$$

has a unique coalgebra structure $r : \varrho F \to F(\varrho F)$ and is "final" for all coalgebras from $\mathsf{CoAlg}_f F$ in the sense that for each coalgebra with finitely presentable carrier, there is a unique coalgebra-homomorphism into ϱF.

As a first result, we can show how ϱF can be derived from νF in Nom.

Proposition 4.2. *Let F be an finitary endofunctor $\mathsf{Nom} \to \mathsf{Nom}$ preserving monomorphisms, let $\nu F \xrightarrow{f} F\nu F$ be its final coalgebra, and for any coalgebra (A, a) let $a^\dagger : X \to \nu F$ be the unique homomorphism into νF. Then*

$$\operatorname{colim} \mathsf{CoAlg}_f F = \bigcup_{(A,a) \in \mathsf{CoAlg}_f F} a^\dagger[B].$$

I.e. the rational fixed point in Nom can be computed by collecting all the "solutions" of finite F-coalgebras.

Proof. As an improvement of readability, we will omit the F in this proof when denoting categories of coalgebras like $\mathsf{CoAlg}F$. Consider the subcategory $\mathsf{CoAlg}_f{}'$ of CoAlg_f consisting of only those coalgebras (B, b) for which $b^\dagger$ is injective i.e. a monomorphism. The proof consists of three steps:

$$\forall (A, a) \in \mathsf{CoAlg}_f \ \exists (B, b) \in \mathsf{CoAlg}_f{}' \text{ and } e : (A, a) \to (B, b) \text{ s.t. } b^\dagger \circ e = a^\dagger \tag{4.1}$$

$$\operatorname{colim} \mathsf{CoAlg}_f = \operatorname{colim} \mathsf{CoAlg}_f{}' \tag{4.2}$$

$$\operatorname{colim} \mathsf{CoAlg}_f{}' = \bigcup_{(B,b) \in \mathsf{CoAlg}_f{}'} b^\dagger[B]. \tag{4.3}$$

For the first step, consider $a^\dagger : A \to \nu F$ for $A \xrightarrow{a} FA$. $a^\dagger$ factorizes in Nom into a surjective equivariant e followed by an injective equivariant m.

$$\overbrace{A \xrightarrow{e} B \xrightarrow{m} \nu F}^{a^\dagger} \quad \overbrace{FA \xrightarrow{Fe} FB \xrightarrow{Fm} F\nu F}^{Fa^\dagger}$$

As F preserves monomorphisms, Fm is a mono. After glueing these two diagrams together via $a : A \to FA$ and $f : \nu F \to F(\nu F)$, Proposition 3.14 gives us a unique diagonal $b : B \to FB$ such that both small squares commute:

$$\begin{array}{ccc} A & \xrightarrow{a} & FA \\ {\scriptstyle e}\downarrow & & \downarrow{\scriptstyle Fe} \\ B & \xrightarrow{b} & FB \\ {\scriptstyle m}\downarrow & & \downarrow{\scriptstyle Fm} \\ \nu F & \xrightarrow[f]{} & F(\nu F) \end{array} \qquad \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \\ \end{array}\right\} a^\dagger \\ \\ \left.\begin{array}{l} \\ \\ \\ \\ \end{array}\right\} Fa^\dagger \end{array}$$

This gives, that e and m are coalgebra-homomorphisms. For the coalgebra (B, b), $b^\dagger : B \to \nu F$ is the unique coalgebra-homomorphism making the bottom square commute, so it follows $m = b^\dagger$. By $A \xrightarrow{e} B$, B is finitely presentable, so $(B, b) \in \mathsf{CoAlg}_f{}'$, which establishes (4.1).

With (4.1), all conditions for Lemma A.3 are met: The inclusion $\mathsf{CoAlg}_f{}' \hookrightarrow \mathsf{CoAlg}_f$ is a full subdiagram, and both are filtered. So the inclusion $\mathsf{CoAlg}_f{}' \hookrightarrow \mathsf{CoAlg}_f$ is a cofinal subdiagram and their colimits are isomorphic, i.e. (4.2).

As a last step let us describe how the colimit is constructed. So set

$$U := \bigcup_{(B,b) \in \mathsf{CoAlg}_f{}'} b^\dagger[B], \quad (u_B = b^\dagger : B \to U)_{(B,b) \in \mathsf{CoAlg}_f{}'}.$$

To prove that $(u_B : B \to U)_{(B,b)}$ is a cocone, it is sufficient to show that for any $f : (B, b) \to (B', b')$, $b^\dagger = b'^\dagger \circ f$, which holds because of the universal property of νF.

Now assume another cocone $(c_B : B \to C)_{(B,b) \in \mathsf{CoAlg}_f{}'}$ of $\mathsf{CoAlg}_f{}'$. Define the morphism $u : U \to C$ pointwise. Let $x \in b^\dagger[B]$ for some $(b, B) \in \mathsf{CoAlg}_f{}'$ and set $u(x) := c_B(b^{\dagger -1}(x))$. This is well-defined: for another $x \in b'^\dagger[B']$ there is some B'' with $(B, b) \xhookrightarrow{i} (B'', b'') \xhookleftarrow{i'} (B', b')$, because $\mathsf{CoAlg}_f{}'$ is filtered, as well as with $b''^\dagger \circ i = b^\dagger$ and $b''^\dagger \circ i' = b'^\dagger$, because of the universal property of νF. This ensures that $c_B(b^{\dagger -1}(x)) = c_{B'}(b'^{\dagger -1}(x))$. Obviously, it also holds that $c_B = u \circ u_B$ for any $B \in \mathsf{CoAlg}_f{}'$.

To see that u is unique, assume another $\tilde{u} : U \to C$ with $c_B = \tilde{u} \circ u_B$ for any $B \in \mathsf{CoAlg}_f{}'$. It follows directly that

$$\tilde{u} \circ u_B(x) = c_B(x) = u \circ u_B(x)$$

Because all elements $y \in U$ are $y = u_B(x) = b^\dagger(x)$ for some $(B, b) \in \mathsf{CoAlg}_f{}'$ and $x \in B$, it is $\tilde{u} = u$. □

The remaining question is, what this rational fixed point of some endofunctor F exactly looks like in Nom. In detail, what do we get after having collected all the images of finitely presentable coalgebras in νF? This is difficult to tell for arbitrary endofunctors F, so let us look at the concrete example of the functor corresponding to the λ-calculus.

4.2 Characterization of rational λ-trees up to α-equivalence

Consider the Nom-endofunctor

$$LX := \mathcal{V} + \mathcal{V} \times X + X \times X.$$

L-coalgebras are systems $C \xrightarrow{c} \mathcal{V} + \mathcal{V} \times C + C \times C$ reminding us of λ-trees: the elements of C are the subtrees. Each tree is either a variable, an abstraction of some other tree, or an application of a tree to a tree. With this intuition, we can understand the $\mathfrak{S}(\mathcal{V})$-action on C as renaming the variables in such a tree.

As observed by [KPSdV13], there is a problem with this definition. Though the λ-trees may be infinitely large, they may mention only finitely many variables, i.e. νL is the set of possibly infinite λ-trees with finitely many variables. Furthermore, it is impossible to define substitution on these terms as a total function, as terms are not treated up to α-equivalence. The solution is to use another construct for the case of an abstraction, that really "hides" the abstracted variable:

Definition 4.3 (Abstraction, [GP99, Lemma 5.1])**.** For an nominal $(X, \cdot)$, we can define the α-equivalence $\sim_\alpha$ generally as the relation on $\mathcal{V} \times X$ as

$$(v_1, x_1) \sim (v_2, x_2) :\Leftrightarrow \text{there exists } z\#\{v_1, v_1\}, z\#x_1, z\#x_2 \text{ with } (v_1\ z)x_1 = (v_2\ z)x_2$$

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

$$\pi \cdot \langle v \rangle x = \langle \pi(v) \rangle (\pi \cdot x).$$

For an equivariant $f : X \to Y$, $[\mathcal{V}]f : [\mathcal{V}]X \to [\mathcal{V}]Y$ is defined by

$$\langle v \rangle x \mapsto \langle v \rangle (f(x)).$$

Definition 4.4 (Concretion, [GP99])**.** Let $(X, \cdot)$ be a nominal set. Concretion is the partial function $@ : [\mathcal{V}]X \times \mathcal{V} \to X$ with $\langle v \rangle x @ w$, the "concretion of $\langle v \rangle x$ at w", defined as $\langle v \rangle x @ w = (w\ v) \cdot x$ if $w \in \mathcal{V} \setminus \mathsf{supp}(\langle v \rangle x)$.

One can show that $[\mathcal{V}] : \mathsf{Nom} \to \mathsf{Nom}$ is a functor – called the *binding functor*. It can be observed that v is fresh for $\langle v \rangle x$. Using concretion, one can equip the functor $[\mathcal{V}]$ with a strength $\tau_{X,Y} : [\mathcal{V}]X \times Y \to [\mathcal{V}](X \times Y)$ defined by

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

where w is fresh for v, x, y. Now we can define the Nom-endofunctor

$$L_\alpha X := \mathcal{V} + [\mathcal{V}]X + X \times X.$$

As shown by [KPSdV13], νL_α consists precisely of the infinite λ-trees up to α-equivalence with finitely many free variables – but with possibly infinitely many bound variables. With that knowledge, we are able to describe $\varrho L_\alpha \subseteq \nu L_\alpha$ further.

Note that by [KPSdV13, Proof of Proposition 5.6] both L and L_α preserve monos and are finitary. So all assumptions of Theorem 4.1 and Proposition 4.2 are met.

Theorem 4.5. *Let X be orbit-finite and equipped with a coalgebra-structure $X \xrightarrow{a} L_\alpha X$. Then for all* root $\in X$*, $a^\dagger(\text{root}) \in \varrho L_\alpha$ is a rational λ-tree.*

Proof. Let m be the maximal number of free variables in any element of X, i.e.

$$m := \max_{x \in X} \big| \mathsf{supp}(x) \big|$$

The maximum exists, because X consists of finitely many orbits and because all the supports of elements of the same orbit have the same finite cardinality, see Lemma 3.19.

Define $W \subseteq \mathcal{V}$ as some set of $m+1$ variable names containing $\mathsf{supp}(\text{root})$. This gives us

$$\forall x \in X \; \exists w \in W \colon w \# x. \tag{4.4}$$

In the following, a rational λ-tree is built, which will be in the α-equivalence class of $a^\dagger(\text{root})$. First, define an L-coalgebra $C \xrightarrow{c} LC = \mathcal{V} + \mathcal{V} \times C + C \times C$ in Set:

$$C := \{x \in X \mid \mathsf{supp}(x) \subseteq W\}, \quad c(x) = \begin{cases} w & \text{if } a(x) = w \in W \subseteq \mathcal{V} \\ (\ell, r) & \text{if } a(x) = (\ell, r) \in X \times X \\ (w, y) & \text{if } a(x) = \langle v \rangle y' \text{ and } y = (v \; w) y' \\ & \text{for some } w \in W \setminus \mathsf{supp}(x) \end{cases}$$

The function c defined on C really has its image in LC:

- For the case $a(x)$ in $\mathcal{V}$, $a(x)$ is also in W because $x \in C$ and thus
$$\mathsf{supp}(a(x)) \subseteq \mathsf{supp}(x) \subseteq W.$$
- For the case $a(x) = (\ell, r)$ in $X \times X$,
$$\mathsf{supp}(\ell) \cup \mathsf{supp}(r) \subseteq \mathsf{supp}(x) \subseteq W.$$
So $\ell, r \in C$.
- For $a(x) = \langle v \rangle y' \in [\mathcal{V}]X$, we know by (4.4), that there is such a fresh $w \in W$. So $\langle w \rangle y = \langle v \rangle y'$. In particular, $y \in C$, because
$$\mathsf{supp}(y) \subseteq \{w\} \cup \mathsf{supp}(a(x)) \subseteq \{w\} \cup \mathsf{supp}(x) \subseteq \{w\} \cup W \subseteq W.$$

The last item gives that $\langle w \rangle y = \langle v \rangle y'$, so the following commutes in Set.

$$\begin{array}{ccccc} C & \hookrightarrow & X & \xrightarrow{\quad a \quad} & L_\alpha X = \mathcal{V} + [\mathcal{V}]X + X \times X \\ {\scriptstyle c}\downarrow & & & & \uparrow{\scriptstyle \mathrm{id}_{\mathcal{V}} + \mathrm{bind} + (\mathrm{id}_X \times \mathrm{id}_X)} \\ LC & \hookrightarrow & LX & = & \mathcal{V} + \mathcal{V} \times X + X \times X \end{array} \tag{4.5}$$

Here, bind denotes the equivariant $\mathcal{V} \times X \to [\mathcal{V}]X$, $(v, x) \mapsto \langle v \rangle x$. It is used heavily, that coproducts and finite products in Nom are build the same way as in Set. The commutativity of the diagram tells us that the L-coalgebra (C, c) in Set plus binding behaves the same as the L_α-coalgebra (X, a) in Nom.

Observe that C is finite, because X is orbit-finite and within an orbit there are only finitely many elements with a support contained in W by Lemma 3.22. Let $c^\dagger$ denote the unique L-coalgebra morphism into the final L-coalgebra in Set. As C is finite, we also can consider $c^\dagger$ as the unique morphism into the rational fixed point of L making the following diagram commute:

$$\begin{array}{ccc} C & \xrightarrow{\quad c \quad} & LC \\ {\scriptstyle c^\dagger}\downarrow & & \downarrow{\scriptstyle Lc^\dagger} \\ \varrho L & \xrightarrow{\quad f \quad} & L(\varrho L) \end{array}$$

Observe, that root $\in C$ and thus $c^\dagger(\text{root})$ is a rational tree, by [AMV06]. Using coinduction together with (4.5), one can see that it is

$$\forall x \in C \subseteq X : [c^\dagger(x)]_\alpha = a^\dagger(x).$$

For $x := \text{root}$ this gives that $c^\dagger(\text{root})$ is the desired rational λ-tree in $a^\dagger(\text{root})$ and thus the α-equivalence class $a^\dagger(\text{root})$ is a rational λ-tree. □

In the previous proof, for the L-coalgebra (C, c) in Set, we know that for any $x \in C$, the resulting tree $c^\dagger(x)$ has at most $|C|$ subtrees. This does not hold for an L_α-coalgebra (X, a) in Nom: if X has a non-trivial action, then the cardinality of X is at least infinite, i.e. the cardinality does not give a reasonable bound for the number of subtrees. And the number of orbits n is not a bound either. The problem is that multiple elements from the same orbit may represent different subtrees. For example, consider the rational tree

$$t := \begin{array}{ccc} & @ & \\ v_0 & & v_1 \end{array}, \quad X := \mathcal{V} + \{(\ell, r) \in \mathcal{V} \times \mathcal{V} \mid \ell \neq r\},$$

equipped with the coalgebra structure

$$X \xrightarrow{a} L_\alpha X, \quad a(x) = \begin{cases} v & \text{if } x = v \in \mathcal{V} \\ (\ell, r) & \text{if } x = (\ell, r) \in \mathcal{V} \times \mathcal{V}. \end{cases}$$

X is constructed to have two orbits: one consisting of single variables and one consisting of unequal ordered pairs of variables. We have $(v_0, v_1) \in X$ and $a^\dagger\big((v_0, v_1)\big) = [t]_\alpha$. But t has three subtrees, namely t itself, v_0, and v_1. This example is expanded later in Example 4.7.

But when looking closer at the construction of C in the previous proof, we can give a bound on the number of subtrees.

Proposition 4.6. *Let (X, a) as in the previous theorem and m be the maximal number of free variables in any element of X as in the previous proof and n be the number of orbits, then the number of subtrees of the constructed tree (the cardinality of the coalgebra (C, c) in the previous proof) is bound by:*

$$n \cdot (m + 1)!$$

Proof. Recall from the proof of Theorem 4.5

$$C := \{x \in X \mid \mathsf{supp}(x) \subseteq W\},$$

where W is a set of $m + 1$ variables. For an orbit, whose support has cardinality k, there are at most $\binom{m+1}{k}$ possibilities of choosing a concrete k-element subset S of W and there are at most $k!$ elements in the orbit with support S, by Lemma 3.20. For the orbit, there are

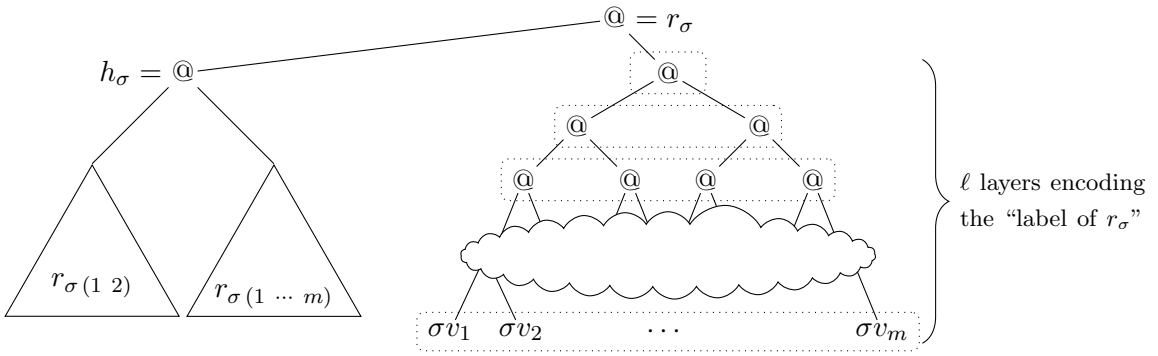
$$\binom{m+1}{k} \cdot k! = \frac{(m+1)!}{k! \cdot (m+1-k)!} \cdot k! = \frac{(m+1)!}{(m+1-k)!} \overset{k \leq m}{\leq} \frac{(m+1)!}{(m+1-m)!} = (m+1)!$$

elements from that orbit in C. In total, the cardinality of C is bound by $n \cdot (m + 1)!$. □

That the number of orbits n occours linearly is not surprising, because if we have some finite carrier set X with the trivial action that "uses" all its elements for the coalgebra structure, we have exactly one subtree per element of X.

At a first glance it is not clear that one has such a blowup in the number of subtrees when increasing the number of variables used. But even if we omit the usage of abstraction, we are able to approach this bound, as the following example shows.

Example 4.7. Define the following rational λ-tree r_σ, parametrized by the number of layers $\ell \in \mathbb{N}$:



Before describing the idea, let us define the new variables: let m be $2^{\ell-1}$, i.e. the number of leaves in a complete binary tree with ℓ layers. Let σ denote some permutation of m variables $\{v_1, \ldots, v_m\}$, i.e. $\sigma \in S_m$.

The idea is, that our coalgebra encodes a tree of permutations. Each permutation is encoded as a full binary tree of ℓ layers each. Each layer can be encoded as a single orbit, because all elements of the same layer – indicated by the dashed rectangle – can be transformed into each other by renaming the variables accordingly. Abstractly, we will refer to this tree on the right-hand-side of r_σ, as the "label σ of r_σ".

The h_σ just is some helper node, to connect the "actual" subtrees of r_σ. We will refer to $r_{\sigma\cdot(1\ 2)}$ as the "actual left subtree" of r_σ, and to $r_{\sigma\cdot(1\ 2\ \cdots\ m)}$ as the "actual right subtree".

To see, that it is $\tau \cdot r_\sigma = r_{\tau\cdot\sigma}$, consider the following bisimulation on infinite binary trees with labels in S_m:

$$R := \{(\tau \cdot r_\sigma, r_{\tau\cdot\sigma}) \mid \tau, \sigma \in S_m\}$$

Clearly, the label of $\tau \cdot r_\sigma$ is $\tau \cdot \sigma$, i.e. equal to the label of $r_{\tau\cdot\sigma}$. The actual left subtree of $\tau \cdot r_\sigma$, namely $\tau \cdot r_{\sigma\cdot(1\ 2)}$, is in relation to $r_{\tau\cdot\sigma\cdot(1\ 2)}$, which is the left subtree of $r_{\tau\cdot\sigma}$. The analgogous argument holds for the right subtree. So in total R is a bisimulation, so it is

$$\tau \cdot r_\sigma = r_{\tau\cdot\sigma}.$$

This gives, that the subtrees of r_σ are in the same orbit as r_σ, because

$$r_{\sigma\cdot(1\ 2)} = \sigma \cdot (1\ 2) \cdot \sigma^{-1} \cdot r_\sigma, \quad r_{\sigma\cdot(1\ 2\ \cdots\ n)} = \sigma \cdot (1\ 2\ \cdots\ n) \cdot \sigma^{-1} \cdot r_\sigma.$$

There is another orbit needed for encoding the junction h_σ of the two subtrees of r_σ. So we can encode this whole tree as a L_α-coalgebra with $\ell + 2$ orbits.

The tree's α-equivalence class – i.e. the solution in ϱL_α – is a singleton, as there are no abstractions at all. The only λ-tree in it has much more subtrees than the coalgebra has orbits: consider the tree starting at r_{id}. The whole group of permutations of m elements is generated by $(1\ 2)$ and $(1\ 2\ \cdots\ m)$. As there are $m!$ permutations, there are at least

$$m! = (2^{\ell-1})!$$

subtrees, though having only $n = \ell + 2$ orbits.

As there are no abstractions, we can reach all subtrees via an equivariant, and there are quite a lot – namely $m!$ – many subtrees to reach, which reminds us of the result in Proposition 3.24.

This shows that the size bounds of the construction in Theorem 4.5 are reasonable and necessary when characterizing ϱL_α. But we can strengthen this statement about the content even more, obtaining the desired characterization of ϱL_α:

Theorem 4.8. *The rational fixed point ϱL_α, contains exactly the rational λ-trees.*

Proof. After Theorem 4.5 it only remains to show that all rational λ-trees are in ϱL_α. Let $t_\alpha \in \nu L_\alpha$ be rational, witnessed by some rational representative $t \in t_\alpha$ with only finitely many subtrees. We can consider this tree as a L-coalgebra $(C, c : C \to LC)$ in Set. $C \subseteq \varrho L$ is the set of subtrees of t and c the restriction of $f : \varrho L \to L\varrho L$.

The nominal X can be defined as

$$X := \bigcup_{s \in C} O([s]_\alpha) \subseteq \nu L_\alpha$$

where $[s]_\alpha \in \nu L_\alpha$ is the α-equivalence class of the subtree s and $O(y) \subseteq \nu L_\alpha$ denotes the orbit of a given element $y \in \nu L_\alpha$. X is closed under "$\cdot$" of νL_α restricted to X, because X is a union of orbits. As t has only finitely many subtrees, each of these can mention only finitely many variables, so indeed each x has finite support and X is a nominal.

The L_α-coalgebra $a : X \to L_\alpha X$ is inherited from νL_α as follows: for a given $x \in X$, there are three cases:

- $f(x) \in \mathcal{V}$ poses no problem and $a(x) := f(x)$.
- For $f(x) = \langle v \rangle y \in [\mathcal{V}](\nu L_\alpha)$, $x = \pi \cdot [\lambda w.s]_\alpha$ for some $\pi \in \mathfrak{S}(\mathcal{V})$ and some subtree $\lambda w.s$ of t. So we have

  $$\langle w \rangle [s]_\alpha = f([\lambda w.s]_\alpha) = f(\pi^{-1} \cdot x) = \pi^{-1} \cdot \langle v \rangle y = \big\langle \pi^{-1}(v) \big\rangle (\pi^{-1} \cdot y)$$

  By the definition of abstraction, we have some $z \in \mathcal{V}$ with

  $$(w\ z) \cdot [s]_\alpha = (\pi^{-1}(v)\ z) \cdot \pi^{-1} \cdot y,$$

  hence, y is in the orbit of $[s]_\alpha$, $f(x)$ is in $[\mathcal{V}]X$, and we can set $a(x) := f(x)$.
- For $f(x) = (\ell, r) \in \varrho L_\alpha \times \varrho L_\alpha$, let $x \in X$ be $\pi \cdot [(s_\ell, s_r)]_\alpha$ for some $\pi \in \mathfrak{S}(\mathcal{V})$. Analogously to the previous case, we have

  $$(\ell, r) = f(x) = f(\pi \cdot [(s_\ell, s_r)]_\alpha) = \pi \cdot ([s_\ell]_\alpha, [s_r]_\alpha) = (\pi \cdot [s_\ell]_\alpha, \pi \cdot [s_r]_\alpha) \in X \times X.$$

  With ℓ, r in X, we can set $a(x) := f(x)$.

By construction, $t_\alpha \in X$ and $a^\dagger(x) = x\ \forall x \in X$. By the finiteness of C, X is orbit-finite and thus $a^\dagger[X] \subseteq \varrho L_\alpha$, especially $t_\alpha \in \varrho L_\alpha$. □

4.3 Corecursive functions on rational λ-trees

When performing operations known from (infinitary) λ-calculus on rational λ-trees, it is not clear whether the resulting λ-tree still is rational in general. One such operation is the substitution function

$$\mathsf{subs} : \nu L_\alpha \times \mathcal{V} \times \nu L_\alpha \longrightarrow \nu L_\alpha,$$

which replaces each occourence of some variable in some λ-tree by another λ-tree. Kurz et al. [KPSdV13] show how to define subs on infinite λ-trees: roughly speaking, they define some coalgebra whoose unique homomorphism into the final coalgebra is subs. It is possible to adapt this to the orbit-finite case as follows.

For arbitrary coalgebras with orbit-finite carriers $A \xrightarrow{a} L_\alpha A$, $B \xrightarrow{b} L_\alpha B$, define $\mathsf{subs}_{A,B} : A \times \mathcal{V} \times B \to \varrho L_\alpha$. $\mathsf{subs}_{A,B}$ describes the substitution of some variable within some tree in the coalgebra (A, a) with some tree from the coalgebra (B, b). Define

$$B + A \times \mathcal{V} \times B \xrightarrow{[g,h]} L_\alpha(B + A \times \mathcal{V} \times B)$$

where $g = L_\alpha(\mathsf{inl}) \circ b$ and $h = [h_{\mathrm{Var}}, h_{\mathrm{Abs}}, h_{\mathrm{App}}] \circ (a \times \mathrm{id} \times \mathrm{id})$, using that $\times$ distributes over $+$.

- $h_{\mathrm{Var}} : \mathcal{V} \times \mathcal{V} \times B \to L_\alpha(B + A \times \mathcal{V} \times B)$, $h_{\mathrm{Var}}(v, w, t) = \begin{cases} \mathsf{in}_1 v & \text{if } v \neq w \\ L_\alpha(\mathsf{inl})(\underbrace{b(t)}_{\in L_\alpha(B)}) & \text{if } v = w \end{cases}$
- $h_{\mathrm{App}} : A \times A \times \mathcal{V} \times B \to L_\alpha(B + A \times \mathcal{V} \times B)$ is the composition of the obvious steps $A \times A \times \mathcal{V} \times B \to (A \times \mathcal{V} \times B) \times (A \times \mathcal{V} \times B) \to L_\alpha(B + A \times \mathcal{V} \times B)$
- $h_{\mathrm{Abs}} : ([\mathcal{V}]A) \times \mathcal{V} \times B \to L_\alpha(B + A \times \mathcal{V} \times B)$, $h_{\mathrm{Abs}} := \mathsf{in}_2 \circ \tau_{A,\mathcal{V}\times B}$, using the strength τ of the functor $[\mathcal{V}]$ with $\tau_{A,\mathcal{V}\times B} : ([\mathcal{V}]A) \times \mathcal{V} \times B \to [\mathcal{V}](A \times \mathcal{V} \times B)$.

$B + A \times \mathcal{V} \times B$ is orbit-finite, so we get a unique L_α-coalgebra homomorphism $[g', h']$ into ϱL_α, making the following commute:

$$\begin{array}{ccc} B + A \times \mathcal{V} \times B & \xrightarrow{[g,h]} & L_\alpha(B + A \times \mathcal{V} \times B) \\ {\scriptstyle [g',h']}\downarrow & & \downarrow{\scriptstyle L_\alpha([g',h'])} \\ \varrho L_\alpha & \xrightarrow{\quad f_\alpha \quad} & L_\alpha(\varrho L_\alpha) \end{array}$$

Here, we define $\mathsf{subs}_{A,B} := h'$. The only thing left is to extend this to the desired domain $\varrho L_\alpha \times \mathcal{V} \times \varrho L_\alpha$. We know that

$$\varrho L_\alpha \quad = \quad \bigcup_{(A,a)\text{ f.p.}} a^\dagger[A] \quad = \quad \bigcup_{\substack{(A,a)\text{ f.p.} \\ a^\dagger \text{ injective}}} a^\dagger[A] \quad = \quad \bigcup_{\substack{(A,a)\text{ f.p.} \\ a^\dagger \text{ injective} \\ a^\dagger \text{ is a inclusion}}} a^\dagger[A] \quad = \quad \left.\bigcup_{\substack{(A,a)\text{ f.p.} \\ a^\dagger \text{ injective} \\ a^\dagger \text{ is an inclusion}}} A \right\} := P(A)$$

With that we can define substitution on rational λ-trees $\mathsf{subs}_{\mathrm{rat}}$ as

$$\mathsf{subs}_{\mathrm{rat}}(t, v, s) = \mathsf{subs}_{A,B}(t, v, s), \quad \mathsf{subs}_{\mathrm{rat}} : \varrho L_\alpha \times \mathcal{V} \times \varrho L_\alpha \to \varrho L_\alpha$$

for some A with $P(A), t \in A$ and some B with $P(B), s \in B$.

It remains to prove that the result does not depend on the choice of A and B. We have $A, A' \subseteq \varrho L_\alpha$. We also have for all $x \in A$ and $x \in A$ that $f_\alpha \circ a^\dagger = L_\alpha(a^\dagger) \circ a$ so $f_\alpha(x) = a(x)$. The analogous argument holds for (A', a'). The respective function h is defined by pattern matching – i.e. on the alternative indicated by a – so it behaves independently from the choice of A, analogously for the choice of B.

To summarize, we can say that one can define operations not escaping the rational λ-trees if these n-ary operations can be defined parametrized by n finitely presentable coalgebras as it was shown for the binary case of substitution.

5 Conclusion

To conclude, we have seen that the set of rational λ-trees modulo α-equivalence can be described as the rational fixed point of $L_\alpha X = \mathcal{V} + [\mathcal{V}]X + X \times X$ – in addition to the earlier definition as infinite trees with finitely many subtrees. For a concrete L_α-coalgebra, we have seen how the number of subtrees explodes when increasing the number of used variables.

The close relation of ϱL_α to the greatest fixed point νL_α showed us how to adapt definitions, e.g. for the substitution function. And as all trees were considered modulo α-equivalence, we obtained a total substitution function, totally analogous to the infinitary case in [KPSdV13]. One could look further into their further applications like definitions of Böhm, Lévy-Longo, and Berarducci trees and how they are adaptable from the general infinite case to the rational case.

It remains open, how ϱF looks like for other Nom-endofunctors than L_α. One can suggest, that the result is absolutely analogous for functors obtained by the grammar

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

where K are constant functors, Id is the identity functor, and $\coprod F$ denotes countable coproducts. This family of functors is examined in [KPSdV13] for the infinitary case. But all our results should be adaptable from L_α to this generalized family of functors.

But for Nom-endofunctors, which do not match the above pattern, a concrete description of the final coalgebra and of the rational fixed point are open.

A Common definitions and facts

A.1 Common notation

The basic set theoretic notation includes the following conventions:

- The power set of some set X – i.e. the set of all subsets of X – is denoted by $\mathcal{P}(X)$.
- The set of all *finite* subsets of X is denoted by $\mathcal{P}_f(X)$.
- A transposition of two elements a, b is a permutation that swaps these two elements and leaves anything else fix. I.e. for $a, b \in X$, $(a\ b) : X \to X$, with

$$(a\ b)(x) = \begin{cases} b & \text{if } a = x \\ a & \text{if } b = x \\ x & \text{otherwise.} \end{cases}$$

- The image of some map $f : A \to B$ with respect to some domain $D \subseteq A$ is denoted by $f[D]$.
- For some $b \in B$, the constant function $b! : A \to B$ is defined as $a \mapsto b$ for all $a \in A$.

We assume that the reader is familiar with basic category theory, as described in the first chapters of [Awo10], the objects of a category $\mathcal{C}$ are denoted by $\mathbf{obj}\,\mathcal{C}$, the morphisms by $\mathbf{mor}\,\mathcal{C}$. For products, the projections are denoted by outl and outr. For coproducts, the injection into the ith component is denoted by in_i, for binary coproducts with inl, inr.

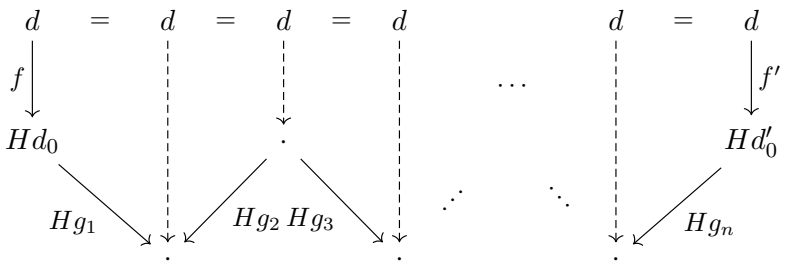
A.2 Known facts

There are different definitions of cofinal, here the following definition is used:

Definition A.1 (see [Lan98, p. 217])**.** A functor $H : \mathcal{D}_0 \to \mathcal{D}$ is said to be *cofinal* provided that for each $d \in \mathcal{D}$ the comma category $d \downarrow H$ is non-empty and connected. This means that for each object d in $\mathcal{D}$

(a) there exists a morphism $f : d \to Hd_0$ for some object d_0 in $\mathcal{D}_0$.

(b) given two such morphisms $f : d \to Hd_0$ and $f' : d \to Hd'_0$, there exists a finite commutative diagram with finitely many morphisms $g_i \in \mathcal{D}_0$ of the following form.



Theorem A.2 (see [Lan98, p. 217]). *If $H : \mathcal{D}_0 \to \mathcal{D}$ is cofinal and $D : \mathcal{D} \to \mathcal{C}$ is a functor such that* $\operatorname{colim} DH$ *exists, then* $\operatorname{colim} D$ *exists and there is a canonical map* $\operatorname{colim} DH \to \operatorname{colim} D$ *which is an isomorphism.*

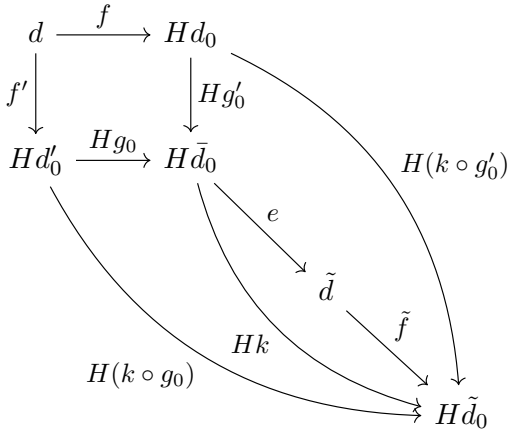
So each colimit for $D \circ H$ is a colimit for D. For the case of filtered subdiagrams, it is commonly known that the criterion can be simplified as described in the following lemma. This lemma could not be found in the literature, so we add the proof here for the sake of completeness.

Lemma A.3. *Assume that $H : \mathcal{D}_0 \hookrightarrow \mathcal{D}$ is a full subdiagram and that $\mathcal{D}, \mathcal{D}_0$ are filtered. Then (a) implies (b) in the previous Definition A.1.*

Proof. Assume (a), i.e. for any d in $\mathcal{D}$ there is an $f : d \to Hd_0$. Furthermore, assume another such morphism $f' : d \to Hd_0'$. By the filteredness of $\mathcal{D}_0$, there is some $\bar{d}_0$ together with morphisms g_0, g_0':

$$\begin{array}{ccc} d & \xrightarrow{f} & Hd_0 \\ {\scriptstyle Hf'}\downarrow & & \downarrow{\scriptstyle Hg_0'} \\ Hd_0' & \xrightarrow{Hg_0} & H\bar{d}_0 \end{array}$$

This square does not necessarily commute. But by the filteredness of $\mathcal{D}$, we get some morphism $e : H\bar{d}_0 \to \tilde{d}$, making the parallel morphisms $Hg_0' \circ f$ and $Hg_0 \circ f'$ equal. Using the property (a) for object $\tilde{d}$ gives another morphism $\tilde{f} : \tilde{d} \to H\tilde{d}_0$ for some $\tilde{d}_0$ in $\mathcal{D}_0$. As $\mathcal{D}_0$ is a full subdiagram of $\mathcal{D}$, there is some morphism k in $\mathcal{D}_0$ with $Hk = \tilde{f} \circ e$.



The outer square commutes, i.e. property (b) holds. □

Bibliography

[AMV06] Jiří Adámek, Stefan Milius, and Jiri Velebil. Iterative algebras at work. *Mathematical Structures in Computer Science*, 16(6):1085–1131, 2006.

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

[AN80] André Arnold and Maurice Nivat. The metric space of infinite trees. algebraic and topological properties. *Fundam. Inform.*, 3(4):445–476, 1980.

[Awo10] Steve Awodey. *Category Theory*. Oxford Logic Guides. OUP Oxford, 2010.

[GP99] Murdoch Gabbay and Andrew M. Pitts. A new approach to abstract syntax involving binders. In Giuseppe Longo, editor, *Proceedings of the Fourteenth Annual IEEE Symp. on Logic in Computer Science, LICS 1999*, pages 214–224. IEEE Computer Society Press, July 1999.

[JR97] Bart Jacobs and Jan Rutten. A tutorial on (co)algebras and (co)induction. *EATCS Bulletin*, 62:62–222, 1997.

[KPSdV13] Alexander Kurz, Daniela Petrisan, Paula Severi, and Fer-Jan de Vries. Nominal coalgebraic data types with applications to lambda calculus. *Logical Methods in Computer Science*, 9(4), 2013.

[Lan98] Saunders Mac Lane. *Categories for the Working Mathematician*. Graduate Texts in Mathematics. Springer New York, 1998.

[Pet11] Daniela Petrişan. *Investigations into Algebra and Topology over Nominal Sets*. dissertation, University of Leicester, 2011.

[Pit13] Andrew M. Pitts. *Nominal Sets: Names and Symmetry in Computer Science*, volume 57 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2013.

[Sel08] Peter Selinger. Lecture notes on the lambda calculus. *ArXiv e-prints*, April 2008.