Electronic Notes in Theoretical Computer Science 122 (2005) 147–170 

[www.elsevier.com/locate/entcs](http://www.elsevier.com/locate/entcs)

Iterative Algebras for a Base

# Jiˇr´ı Ada´mek[1](#_bookmark1) ,[2](#_bookmark1) Stefan Milius[2](#_bookmark1)

*Institute of Theoretical Computer Science, Technical University, Braunschweig, Germany*

Jiˇr´ı Velebil[1](#_bookmark1) ,[3](#_bookmark1)

*Faculty of Electrical Engineering, Technical University, Prague, Czech Republic*

Abstract

For algebras *A* whose type is given by an endofunctor, iterativity means that every flat equation morphism in *A* has a unique solution. In our previous work we proved that every object generates a free iterative algebra, and we provided a coalgebraic construction of those free algebras.

Iterativity w.r.t. an endofunctor was generalized by Tarmo Uustalu to iterativity w.r.t. a “base”, i.e., a functor of two variables yielding finitary monads in one variable. In the current paper we introduce iterative algebras in this general setting, and provide again a coalgebraic construction of free iterative algebras.

*Keywords:* free iterative theory, rational monad, coalgebra

# Introduction

In our previous work we introduced, for every finitary [4](#_bookmark2) endofunctor *H*, the concept of an iterative *H*-algebra. The aim was to generalize and simplify the description of free iterative theories of Calvin Elgot and his coauthors [[10](#_bookmark47)], [[11](#_bookmark53)]. In that we followed the footsteps of Evelyn Nelson [[16](#_bookmark55)] who introduced iterative Σ-algebras (in Set) and simplified the description of the free iterative

1 The first and the third author acknowledge the support of the Grant Agency of the Czech Republic under the Grant No. 201/02/0148.

2 Email:[{adamek,milius}@iti.cs.tu-bs.de](mailto:adamek@iti.cs.tu-bs.de)

3 Email:[velebil@math.feld.cvut.cz](mailto:velebil@math.feld.cvut.cz)

4 “finitary” means: preserving filtered colimits

1571-0661 © 2005 Elsevier B.V. Open access under [CC BY-NC-ND license.](http://creativecommons.org/licenses/by-nc-nd/3.0/)

doi:10.1016/j.entcs.2004.06.056

theory *R*Σ on Σ: that theory assigns to every set *Y* the algebra *R*Σ*Y* of all *rational* Σ*-trees* on *Y* , i.e., those Σ-trees that have finitely many subtrees up to isomorphism (a characterization provided by Susanna Ginali [[12](#_bookmark54)]). Whereas the original proof, using constructions on algebraic theories, occupied most of the technical papers [[10](#_bookmark47)], [[8](#_bookmark48)], [[11](#_bookmark53)], the proof due to Evelyn Nelson was concise and intuitive. In [[5](#_bookmark49)] we introduced iterativity for *H*-algebras for every finitary endofunctor *H* of a locally finitely presentable category ffi — this includes categories such as many-sorted sets ffi = Set*S* and the category Fin[Set*,* Set] of all finitary set functors. We proved that every object *Y* of ffi generates a free iterative *H*-algebra, *RY* , and we provided a coalgebraic construction of *RY* : for the case of *Y* = 0 (initial in ffi), *R*0 is a colimit of all finite *H*-coalgebras, and in the general case we work with *H*(−)+ *Y* instead of *H*. And, again, we proved that the monad *R*(−) of free iterative *H*-algebras is a free iterative monad on *H*.

In the present paper we work with a *base* instead of an endofunctor, and study iterativity of base algebras. By a base we understand a finitary endo- functor from ffi to FM(ffi), the category of finitary monads on ffi. This was introduced by Tarmo Uustalu [[18](#_bookmark56)] under the name parametrized monad. The motivating idea is to study iterativity of Σ-algebras, for a signature Σ, where each operation symbol comes with the information in what places iteration is allowed to occur. Let us illustrate this on the simple example of a signature consiting of a single binary operation symbol ∗.

Case 1: full iterativity. This is the concept of iterative algebra of Evelyn Nelson [[16](#_bookmark55)]: An algebra (*A,* ∗) is iterative if every system

*x*1 ≈ *t*1

.

*xm* ≈ *tm*

(1.1)

of equations in variables *X* = { *x*1*,... , xm* } and with right-hand sides *ti* = *xj* ∗ *xk* for *xj, xk* in *X*, or *ti* ∈ *A*, has a unique solution in *A*. Equivalently every system ([1.1](#_bookmark3)) where each *ti* is a term on *X* + *A*, *ti* /∈ *X*, has a unique solution. A free iterative algebra, *RY* , ona set *Y* is the algebra of all rational binary trees on *Y* .

Case 2: restricted iterativity. Here we require that the free variables are only allowed to occur on the left-hand position of ∗. Thus, an iterative algebra is one in which every system ([1.1](#_bookmark3)) with right-hand sides *ti* = *x* ∗ *a* for *x* ∈ *X* and *a* ∈ *A*, or *ti* ∈ *A* has a unique solution. A free iterative algebra, for iterativity w. r. t. these systems of equations, is the algebra of

all right-wellfounded rational binary trees over *Y* , i. e., those which have the right-most path from every node finite. Observe that for an iterative algebra all right-hand sides

*ti* = *xi*1 ∗ (*xi*2 ∗ (··· ∗ (*xin* ∗ *a*)) ··· ) *, xij* ∈ *X, j* = 1*,... ,n* (1.2)

can be allowed: every system ([1.1](#_bookmark3)) with such right-hand sides has a unique solution, too.

Case 3: no iterativity. Here no variable is allowed to occur on right- hand sides of systems ([1.1](#_bookmark3)), i. e., we are left with the ”trivial” systems in which all right-hand sides lie in *A*. Every algebra is then iterative.

In order to formalize such parametrized iterativity, we move from finitary functors *H* : ffi −→ ffi, used for “classical” algebra, to finitary functors *H* : ffi × ffi −→ ffi, called parametrized endofunctors. In the case of one binary operation the “classical” polynomial endofunctor *H* : Set −→ Set, *HX* = *X* × *X*, is now substituted by three parametrized endofunctors: *H*(*X, Y* )= *X* × *X* for Case 1, *H*(*X, Y* )= *X* × *Y* for Case 2, and *H*(*X, Y* )= *Y* × *Y* for Case 3. Let us denote by

*X* *Y* (read “*X* box *Y* ”)

a free *H*(*X,* −)-algebra on *Y* (for all pairs of objects *X, Y* ∈ ffi). More pre- cisely, for every object *X* we denote by *X* − the free monad on the endo- functor *H*(*X,* −) (which, as proved by Micheal Barr [[7](#_bookmark50)] is just the monad of the free algebras of *H*(*X,* −)). This yields a base, i. e., a functor ffi −→ FM(ffi) in the obvious way.

Example: for one binary operation with full iterativity, *H*(*X,* −)= *X* ×*X*

is the constant endofunctor whose free algebra on *Y* is

*X* *Y* = *X* × *X* + *Y .*

The case of restricted iterativity, *H*(*X,* −) = *X* × − corresponds to unary operation symbols indexed by *X* — the free algebras are

*X* *Y* = *X*∗ × *Y*

where *X*∗ is a free monoid on *X*.

Finally, the “trivial” case of no iterativity yields the base

*X* *Y* = free algebra on *Y*

independently of *X*.

In general, we use the uncurried form of a base, i. e., a functor

: ffi × ffi −→ ffi

finitary in both variables and equipped with monad units *A* −→ *X* *A* and monad multiplications *X* (*X* *A*) −→ *X* *A* for all pairs of objects *X, A* satisfying some obvious compatibility conditions (see Section [2](#_bookmark5)). A *base alge- bra* is a monad algebra of the monad *A* − on the object *A*. That is, a base algebra is given by a morphism

*α* : *A* *A* −→ *A*

satisfying the Eilenberg-Moore conditions in the second variable.

The bases *X* *Y* = *X* × *X* + *Y* and *X* *Y* = *X*∗ × *Y* on Set yield the usual algebras on one binary operation as base algebras. However, iterativity is different, as we demonstrate below.

For a given base algebra *A* let us call morphisms

*e* : *X* −→ *X* *A, X* finitely presentable,

*equation morphisms*. For a fully iterative binary operation this is *e* : *X* −→ *X* × *X* + *A* expressing precisely a system ([1.1](#_bookmark3)), for the restricted iterativity we get *e* : *X* −→ *X*∗ × *A* as in ([1.2](#_bookmark4)) above.

Definition 1.1 A base algebra *α* : *A* *A* −→ *A* is *iterative* provided that for every equation morphism *e* : *X* −→ *X* *A* there exists a unique *solution*, i. e., a unique morphism *e*† : *X* −→ *A* for which the square

*e*† z,

*X A*

¸

*e α*

, z,

commutes.

*X* *A e*† *A A* *A*

The main result of our paper is that

1. free iterative base algebras always exist, and
2. the monad they present in ffi is a free iterative monad on the given base.

This means that the full strength of the results of [[5](#_bookmark49)], concerning (fully) iterative *H*-algebras of a given endofunctor *H*, generalize to the case of base algebras. In that special case one works with the base

*X* *Y* = *HX* + *Y*

where iterativity of base algebras is the full iterativity above, and the monad of free iterative algebras was proved to be a free iterative monad on *H*.

The notion of a base (under the name parametrized monad) has been in- troduced by Tarmo Uustalu [[18](#_bookmark56)] who generalized some results of our paper [[1](#_bookmark46)]. In the present paper we continue in the same vein by generalizing results of [[5](#_bookmark49)] from *H*-algebras to base algebras. Although the structure of the present pa- per follows that of [[5](#_bookmark49)] closely, it turns out that all the proofs are substantially more difficult. Thus our original hope that we will just indicate how to modify the previous proof ideas to the present generality failed, and we feel obliged to present detailed proofs again. The concrete examples of bases below (see [2.5](#_bookmark8)) and their free iterative algebras (see [3.3](#_bookmark17)–[3.6](#_bookmark20)) were already considered in [[18](#_bookmark56)].

# Bases and Base Algebras

Assumption 2.1 Throughout this section we assume that a locally finitely presentable category ffi is given. We denote by FM(ffi) the category of all finitary monads on ffi (i.e., monads whose underlying functor preserves filtered colimits) and monad morphisms.

Definition 2.2 By a *base* on ffi is understood a finitary functor from ffi to

FM(ffi).

Notation 2.3

* 1. Given a base, we have a finitary underlying functor from ffi to the category Fin[ffi*,* ffi] of finitary endofunctors of ffi. This is a curried form of a functor of two variables, finitary in each variable, which we denote by

: ffi × ffi −→ ffi*.*

* 1. The unit of the monad *X* − is denoted by *uX* : *Id* −→ *X* −; its components are

*uX* : *A* −→ *X* *A.*

*A*

* 1. The multiplication of the monad *X* − is denoted by *mX* ; its components are

*mX* : *X* (*X* *A*) −→ *X* *A.*

*A*

Remark 2.4 Explicitly, to specify a base means to specify a finitary functor of two variables

: ffi × ffi −→ ffi

together with morphisms

*uX* : *A* −→ *X* *A* and *mX* : *X* (*X* *A*) −→ *X* *A*

*A A*

for arbitrary objects *X*, *A* of ffi such that the following diagrams commute:

*X* *uX uX*

*X* *A*¸ *A*z*X*, (*X* *A*) ,*X*r *A X* *A*

¸¸¸¸

¸¸¸

,,

¸¸¸

,,,,

*mX* ,,,

(2.1)

¸¸¸¸

,,

*A* ,,

, ,

*X* *A*

and

*X* *mX*

*X* (*X* (*X* *A*)) *A* z*X*, (*X* *A*)

*X*

*m*

*X* *A*

,

*X* (2.2)

,

*m*

*A*

*X* (*X* *A*) *X*

*m*

*A*

z*X*, *A*

expressing the monad axioms for each *X* −, together with

*uX*

1. *A* z*X*, *A*

*f h* *f*

, ,

(2.3)

1. *Y* z*Y* , *B*

*u*

*B*

and

*mX*

*X* (*X* *A*) *A* z*X*, *A*

*h* (*h* *f* )

,

*h* *f*

,

(2.4)

*Y* (*Y* *B*) *Y*

*m*

*B*

z*Y* , *B*

which express the naturality of *uX* and *mX* and the fact that for every mor- phism *h* : *X* −→ *Y* we have a monad morphism *h* (−): *X* (−) −→ *Y* (−).

Examples 2.5

1. We have “trivial” bases given by all constant functors from ffi to FM(ffi). That is, for every finitary monad S on ffi we form the trivial base

*X* S *A* = *SA*

whose unit and multiplication is that of S.

1. Coproduct is a base

*X* *A* = *X* + *A*

with the obvious unit and multiplication

*uX* = inr : *A* −→ *X*+*A* and *mX* = [inl *,* inl *,* inr ]: *X*+*X*+*A* −→ *X*+*A.*

*A A*

1. Let *X*∗ be a free monoid on the object *X* of ffi with unit *ηX* :1 −→ *X*∗

and multiplication *µX* : *X*∗ × *X*∗ −→ *X*∗. Then we have the base

*X* *A* = *X*∗ × *A*

with base unit *uX* = *ηX* × *A* and base multiplication *mX* = *µX* × *A*.

*A A*

1. Let $ be any locally finitely presentable category. Then so is the category

ffi = Fin[$*,* $] of all finitary endofunctors of $. There we have a base

*X* *A* = J(*X*) · *A*

where a free monad on *X* is denoted by (J(*X*)*, ηX, µX*) — it exists since *X*

is finitary, see [[7](#_bookmark50)]. The base unit is *uX* = *ηXA* and the base multiplication

*A*

*mX* = *µXA*.

*A*

Example 2.6 Let : ffi × ffi −→ ffi be a base. We obtain other bases ˙ by precomposing with finitary endofunctors *H* : ffi −→ ffi:

*X* ˙ *A* = *HX* *A*

with unit

*u*˙*X* = *uHX* : *A* −→ *HX* *A*

*A A*

and multiplication

*m*˙ *X* = *mHX* : *HX* (*HX* *A*) −→ *HX* *A.*

*A A*

Of particular importance is the base obtained from the base +, i. e.,

*X* ˙

*A* = *HX* + *A.*

We will see below that our previous results of [[5](#_bookmark49)] on *H*-algebras are special cases of the results concerning these bases.

Definition 2.7 Given a base , by a *base algebra* is understood an object *A*

of ffi together with a monadic algebra on *A* of the monad *A* −.

That is, a base algebra is given by an object *A* and a morphism *α* : *A*

*A* −→ *A* such that the following two diagrams

*uA*

*A* ¸¸¸ *A* z*A*, *A*

¸**¸**

**¸¸**¸

*A* (*A* *A*) *A**α* z*A*, *A*

¸**¸¸**¸ *A*

*m*

*α*

¸**¸**

*A*

, ,

*α* (2.5)

,

*A A* *A α* z*A*,

commute.

Notation 2.8 We denote by

Alg(ffi*,* )

the category of all base algebras and all *homomorphisms* from (*A, α*) to (*B, β*), i.e., morphisms *h* : *A* −→ *A*' of ffi such that the square

*A* *A*  *α* z*A*,

*h* *h h*

, ,

(2.6)

*B* *B β* z*B*,

commutes.

Examples 2.9

1. Algebras of the base *X* *A* = *X* + *A* are given by an object *A* and an endomorphism of *A* (i.e., these are just unary algebras in ffi). Homomor- phisms are also the usual homomorphisms of unary algebras.
2. Algebras of the base *X* *A* = *HX* + *A* are the usual *H*-algebras, i.e., pairs consisting of an object *A* and a morphism *α* : *HA* −→ *A*. Also, homomorphisms are the usual *H*-algebra homomorphisms. Thus,

Alg(ffi*,* )= Alg *H*

is the category of *H*-algebras and homomorphisms.

1. Algebras of the base on Set given by

*X* *A* = *X*∗ × *A* (see [2.5](#_bookmark8)([iii](#_bookmark10)))

are precisely the usual algebras on one binary operation. In fact, given the latter, say,

define *α* : *A*∗ × *A* −→ *A* by

⊕ : *A* × *A* −→ *A*

*α*(*a*1*a*2 *... an, a*)= *a*1 ⊕ (*a*2 ⊕ *...* (*an* ⊕ *a*) *...* )*.*

This satisfies ([2.5](#_bookmark13)). Conversely, given *α* : *A*∗ × *A* −→ *A* satisfying ([2.5](#_bookmark13)), it is given by the above formula where ⊕ denotes the restriction of *α* to all pairs in *A* × *A*. Consequently, the bases

*X* *A* = (*X* × *X*)+ *A* and *X* *A* = *X*∗ × *A*

on Set define the same categories of algebras.

1. Let $ be any locally finitely presentable category and put ffi = Fin[$*,* $] with the base

*X* *A* = J(*X*) · *A* (see [2.5](#_bookmark8)([iv](#_bookmark11)))

An algebra is a pair (*A, α*) consisting of a finitary endofunctor *A* : $ −→

$ and a natural transformation *α* : *A* · *A* −→ *A*.

More precisely: each such pair defines a unique natural transformation *α* from *A* to ⟨*A, A*⟩ (the right Kan extension of *A* along *A*). Since ⟨*A, A*⟩ is always a monad and J(*A*) is a free monad on *A*, *α* yields a unique monad morphism

*α*˜ : J(*A*) −→ ⟨*A, A*⟩*.*

The unique natural transformation

*α*^ : J(*A*) · *A* −→ *A*

corresponding to *α*˜ defines an algebra of our base — in fact, the condi- tion ([2.5](#_bookmark13)) above is equivalent to *α*˜ being a monad morphism.

Proposition 2.10 *The category* Alg(ffi*,* ) *is locally ﬁnitely presentable, and its forgetful functor into* ffi *has a left adjoint, i.e., free base algebras exist on every object of* ffi*.*

Proof. The endofunctor *SA* = *A* *A* of ffi is finitary, and thus the category Alg *S* is locally finitely presentable and its forgetful functor has a left adjoint, see [[6](#_bookmark51)].

The category Alg(ffi*,* ) is a full subcategory of Alg *S*, and it is easy to verify that it is closed under limits and filtered colimits in Alg *S*. It follows that it is a reflective subcategory, see Theorem 2.48 in [[6](#_bookmark51)]. Since the forgetful functor of Alg(ffi*,* ) is a restriction of that of Alg *S*, the proposition follows.

# Iterative Base Algebras

Assumption 3.1 Throughout this section denotes a base on a locally finitely presentable category ffi.

Definition 3.2

* 1. By a (finitary, flat) *equation morphism* in an object *A* is meant a mor- phism

*e* : *X* −→ *X* *A, X* finitely presentable.

* 1. Suppose that *A* is the underlying object of a base algebra *α* : *A* *A* −→ *A*. Then by a *solution* of *e* is meant a morphism *e*† : *X* −→ *A* such that the

square

*e*† z,

*X A*

¸

*e α*

, z,

(3.1)

commutes.

*X* *A e*† *A A* *A*

* 1. A base algebra is called *iterative* provided that every equation morphism has a unique solution.

Example 3.3 Consider the base *X* *A* = *X* + *A* in Set. Then a base algebra,

i. e., a unary algebra *α* : *A* −→ *A*, is iterative if and only if *αk* has a unique fixed point for all *k*, see [[5](#_bookmark49)].

Example 3.4 Algebras of the base

*X* *A* = (*X* × *X*)+ *A*

on ffi = Set are the usual algebras on one binary operation: see Example [2.9](#_bookmark14)([ii](#_bookmark15)) and put *HX* = *X* ×*X*. There is no easy criterion for an algebra to be iterative. But there are nice examples of iterative algebras, see [[5](#_bookmark49)], e.g.,

*A* = {1*,* 2*,* 3*,...* }∪ {∞} with addition

*A* = (0*,* ∞] with addition

*A* = (1*,* ∞] with multiplication

A free iterative algebra on a set *Y* (of generators) can be described as the algebra *RY* of all rational binary trees on *Y* , see Section [1](#_bookmark0).

Example 3.5 Consider the base *X* *A* = *X*∗ × *A* on ffi = Set, see Exam- ple [2.5](#_bookmark8)([iii](#_bookmark10)). Although its algebras are, again, the usual binary algebras, the concept of iterative algebras differs from the above example. Recall that an algebra (*A,* ⊕) leads to *α* : *A*∗ × *A* −→ *A* with

*α*(*a*1*a*2 *... an, a*)= *a*1 ⊕ (*a*2 ⊕ *...* (*an* ⊕ *a*) *...* )*.*

It is iterative if and only if for every equation morphism *e* : *X* −→ *X* ∗ × *A* (*X* finite) there exists a unique *e*† : *X* −→ *A* such that for every variable *x* we have that

1. *e*(*x*)= (*ε, a*) implies *e*†(*x*)= *a*, and
2. *e*(*x*)= (*x*1 *... xn, a*) implies *e*†(*x*)= *e*†(*x*1) ⊕(*e*†(*x*2) ⊕ *...* (*e*†(*xn*) ⊕ *a*) *...* ).

Thus, for example, the empty algebra is iterative in the present sense (but it is not iterative for (*X* × *X*)+ *A*).

A free iterative algebra, *R*^*Y* , on a set *Y* can be described as a subalgebra of the above algebra *RY* of all rational binary trees on *Y* . Let us call a binary tree *right-wellfounded* if from every node the right-most path is always finite (i.e., it leads to a leaf). It is obvious that the set *R*^*Y* of all rational right- wellfounded trees is a subalgebra of *RY* . This subalgebra is iterative (w.r.t. the present base *X*∗ × *A*), in fact, *R*^*Y* is a free iterative algebra on *Y* .

Example 3.6 We know that the classical Σ-algebras for a signature Σ = (Σ*n*)*n*∈N are just algebras of the corresponding “polynomial” endofunctor *H*Σ of Set. The above Example [3.4](#_bookmark19) immediately generalizes to the base *X* ˙ *Y* =

*H*Σ*X* + *Y* of Example [2.6](#_bookmark12): a free iterative algebra on a set *Y* is the algebra

*R*Σ*Y*

of all rational Σ-trees on *Y* , see Section [1](#_bookmark0).

Open Problem 3.7 Describe, for ffi = Fin[Set*,* Set], free iterative algebras of the bases

and

*X* *A* = (*X* × *X*)+ *A*

*X* *A* = J(*X*) · *A.*

Example 3.8 For the trivial bases S, see Example [2.5](#_bookmark8)([i](#_bookmark9)), all algebras are iterative. In fact, given an Eilenberg-Moore algebra *α* : *SA* −→ *A* and an equation morphism *e* : *X* −→ *SA*, the unique solution is *e*† = *α* · *e* : *X* −→ *A*.

Notation 3.9 Let *e* : *X* −→ *X* *A* be an equation morphism with parameters in *A*. Every morphism *h* : *A* −→ *B* in ffi yields an equation morphism

*h* • *e* ≡ *X*  *e* z*X*, *A**X**h* z*X*, *A.* (3.2)

Lemma 3.10 *Let* (*A, α*) *and* (*B, β*) *be iterative algebras. Then a morphism*

*h* : *A* −→ *B in* ffi *is a homomorphism if and only if it preserves solutions,*

*i. e., for every equation morphism e* : *X* −→ *X* *A the solution of h* • *e in B*

*is h* · *e*†*.*

Lemma [3.10](#_bookmark22) explains that the choice of “plain” homomorphisms between iterative algebras is adequate.

Proposition 3.11 *Every object of* ffi *generates a free iterative algebra.*

Proof. It is easy to prove that iterative algebras are closed under limits and filtered colimits in Alg(ffi*,* ). By Theorem 2.48 in [[6](#_bookmark51)] they form a reflective subcategory of Alg(ffi*,* ). Now apply Proposition [2.10](#_bookmark16).

# A Coalgebraic Construction

We know from the preceding section that, for every base, free iterative algebras exist. The aim of the present section is to show that a free iterative algebra *RY* on an object *Y* can be constructed as a filtered colimit of all equation morphisms *e* : *X* −→ *X* *Y* (where *X* ranges through a set ffi*fp* representing all finitely presentable objects of ffi up to isomorphism).

More precisely, consider the coalgebras of the endofunctor − *Y* together with the usual coalgebra homomorphisms. We denote by

EQ*Y*

the category of all equation morphisms as the full subcategory of Coalg (− *Y* ) on all objects from ffi*fp*. Since Coalg (− *Y* ) is cocomplete, with colimits formed on the level of ffi, it is obvious that EQ*Y* is closed in it under finite colimits. In particular, EQ*Y* is a filtered category. We also denote by

Eq*Y* : EQ −→ ffi*,* (*e* : *X* −→ *X* *Y* ) '→ *X*

the forgetful functor. This defines a filtered diagram in ffi.

Notation 4.1 *RY* denotes a (filtered) colimit of the diagram Eq*Y* with colimit cocone

*e* : *X* −→ *RY* (for all *e* : *X* −→ *X* *Y* in EQ*Y* ).

Remark 4.2 The aim of the present section is to prove that *RY* carries a structure of an iterative algebra making it a free iterative algebra on *Y* . We proceed in two steps: we first assume that *Y* is a finitely presentable ob- ject. This enables us, for example, to define the universal arrow immediately:

observe that *uY* : *Y* −→ *Y* *Y* is an object of EQ*Y* and denote by

*Y*

*ηY* = (*uY* )

*Y*

: *Y* −→ *RY*

the corresponding colimit morphism.

An extension of all the results to arbitrary objects *Y* is then easy. For example, *ηY* is defined as follows: express *Y* as a colimit of a filtered diagram of finitely presentable objects *Yt*, (*t* ∈ *T* ), then it is easy to verify that *RY* is a filtered colimit of *RYt*, (*t* ∈ *T* ), and we put *ηY* = colim *ηYt* .

*t*∈*T*

Notation 4.3 For every object *Y* we denote by

*iY* : *RY* −→ *RY* *Y*

the unique morphism for which the squares

*X*  *e* z*X*, *Y*

*e e* *Y*

, ,

z,

(4.1)

*RY iY RY* *Y*

commute for every *e* in EQ*Y* . This is well defined since the morphisms (*e* *Y* )·*e* are easily seen to form a cocone of the diagram Eq*Y* .

Lemma 4.4 *For every equation morphism e in* EQ*Y there exists a unique morphism e* : *X* −→ *RY such that the square* ([4.1](#_bookmark23)) *commutes.*

Notation 4.5 We introduce notation here for “adding variables” to equations: given an equation morphism

*e* : *X* −→ *X* *Y* in EQ*Y*

and an object *Q*, when are we able to form “canonically” an equation mor- phism

*X* + *Q* −→ (*X* + *Q*) *Y* in EQ*Y* ?

One possibility is to assume that a morphism

*q* : *Q* −→ *X* *X*

is given. Then we can define an equation morphism

*eq* = (inl *Y* ) · [*X* *Y, mX* · (*X* *e*)] · (*e* + *q*) i.e., *eq* is given by the following diagram

*Y*

*X*  *e* z*X*, *Y*

inl

, *eq*

inl *Y*

,

*X* + *Q*

¸

inr

z(*X*,+ *Q*) *Y*

¸

inl *Y*

*Q q* z*X*, *X X* *e* z*X*, (*X* *Y* ) *X*

*m*

*Y*

z*X*, *Y*

Theorem 4.6 *For every ﬁnitely presentable object Y there exists a unique structure of a base algebra ρY* : *RY* *RY* −→ *RY such that the squares*

*Q*  inr z*X*,+ *Q*

*q*

,

*X* *X*

*e* *e*

*eq*

(4.2)

, ,

*RY* *RY ρY* z*R*,*Y*

*(where e and q are as in Notation* [*4.5*](#_bookmark25)*) commute.*

Remark 4.7 In the proof below we will denote by Eq*Y* Eq*Y* : EQ*Y* −→ ffi the diagram given by objects *X* *X*, where *e* : *X* −→ *X* *Y* ranges over EQ*Y* , and by morphisms *h* *h*, for *h* in EQ*Y* . Since is finitary in both variables and the diagonal EQ*Y* −→ EQ*Y* × EQ*Y* is cofinal, the colimit of Eq*Y* Eq*Y* is *RY* *RY* with colimit injections *e* *e*. Analogously, *X* *RY* = colim *X* Eq*Y* , etc.

Proof. We establish that there is a unique morphism *ρY* for which ([4.2](#_bookmark26)) com- mutes, and postpone the verification that this yields a base algebra to the full version of our paper.

1. Consider an arbitrary morphism *p* : *Q* −→ *RY* *RY* where *Q* ∈ ffi*fp*. In every locally finitely presentable category each object is a canonical filtered colimit of morphisms from finitely presentable objects, thus, *RY* *RY* is a colimit of the corresponding diagram *D* with the colimit cocone formed by all *p*’s. We show that each *p* factors as

*p* = (*e* *e*) · *q*

for some *e* and *q*, and that the morphisms *eq* · inr : *Q* −→ *RY* (which, as we show, are independent of the choice of such a factorization) form a cocone of *D*.

The existence of the above factorization follows from the above filtered colimit *RY* *RY* = colim(*EqY* Eq*Y* ). The morphism *p* : *Q* −→ *RY* *RY* , having a finitely presentable domain, factors through one of the colimit injections.

We claim that the upper passage *eq* · inr of the square ([4.2](#_bookmark26)) is independent of the choice of the factorization. Thus, let *f* : *Z* −→ *Z* *Y* be an equation

morphism and let

*Q* ¸¸¸¸

*r*

¸¸ ¸

˛r

¸¸¸*p*

¸

*Z* *Z f* *f* z*R*,*Y* *RY*

be another factorization of *p* through a colimit morphism of the diagram Eq*Y* Eq*Y* . Since that diagram is filtered, we can assume, without loss of generality, that a morphism *h* from *e* to *f* in EQ*Y* exists, and that the equation *r* = (*h* *h*) · *q* holds. It follows that *h* + *Q* is a morphism from *eq* to *fr*:

r~ *e*+*q*

*eq*

[*X* *Y,mX* ·(*X* *e*)]

~v,

*X*+*Q*

z(*X*, *Y* )+(*X* *X*) *Y* z*X* ,*Y*  inl *Y* z(*X*,+*Q*) *Y*

*h*+*Q*

(*h* *Y* )+(*h* *h*)

*h* *Y*

(*h*+*Q*) *Y*

,,¸ z, ,

z,, z, ,

*Z*+*Q*

*f* +*r*

(*Z* *Y* )+(*Z* *Z*)

[*Z* *Y,mY* ·(*Z* *f* )]

*fr*

*Z*

*Z* *Y*

inl *Y*

(*Z*+*Q*) *Y*

¸

,

Consequently,

*eq* = *fr* · (*h* + *Q*)*.*

This proves the desired independence:

*eq* · inr = *fr* · (*h* + *Q*) · inr = *fr* · inr *.*

1. The above morphisms *eq* · inr : *Q* −→ *RY* form a cocone of the diagram

*D*. That is, given an arrow *p*' : *Q*' −→ *RY* *RY* , *Q*' ∈ ffi*fp*, and given a morphism

*Q* ¸¸

¸¸

¸¸¸

¸¸

*p* ¸¸

*t* z*Q*,'

.....

. *p*'

˛r

*RY* *RY*

of the diagram *D*, we prove that for any factorization

*Q*' ¸¸

*r*

¸¸

˛r

¸¸ *p*'

¸¸¸

¸

*Z* *Z f* *f* z*R*,*Y* *RY*

we have

*eq* · inr = *fr* · inr · *t.* (4.3)

In fact, the following factorization of *p*:

*Q* ¸¸¸

*t*

*Q*' ˛r

¸¸¸¸

¸¸¸¸*p*

¸¸¸¸

'

*r*

˛r

*p* ¸¸¸

s¸ r¸

¸

*Z* *Z f* *f* z*R*,*Y* *RY*

yields, by the independence proved in part (1):

*eq* · inr = *frt* · inr *.*

Moreover, *Z* + *t* is an equation morphism from *frt* into *fr*:

*Z* + *Q*

*f* +*rt*

[*Z* *Y,mZ* (*Z* *f* )]

z(*Z*, *Y* )+ (*Z* *Z*) z*Z*, *Y* z(*Z*,+ *Q*) *Y*

*Y*  inl *Y*

*Z*+*t*

,

*Z* ' z(*Z*, *Y* )+ (*Z* *Z*)

z*Z*, *Y*  z(*Z*,+

(*Z*+*t*) *Y*

,'

+ *Q f* +*r*

[*Z* *Y,mZ* (*Z* *f* )]

*Y*

inl *Y*

*Q* ) *Y*

Consequently, *frt* = *fr* · (*Z* + *t*), which implies ([4.3](#_bookmark28)):

*eq* · inr = *fr* · (*Z* + *t*) · inr = *fr* · inr · *t.*

The cocone *eq* · inr has precisely one factorization through the colimit cocone

— this proves that ([4.2](#_bookmark26)) defines a unique *ρY* .

Lemma 4.8 *For every ﬁnitely presentable object Y the morphism iY (see Diagram* ([4.1](#_bookmark23))*) is an isomorphism with the inverse*

*i*−1 = *j*

≡ *RY* *Y*  *RY* *ηY* z*R*,*Y* *RY*  *ρY* z*R*,*Y* (4.4)

*Y Y*

Theorem 4.9 *For every ﬁnitely presentable object Y , the algebra* (*RY, ρY* ) *is a free iterative algebra on Y w.r.t. the universal arrow ηY* : *Y* −→ *RY .*

Proof. (1) (*RY, ρY* ) is iterative. In fact, every equation morphism

*e* : *X* −→ *X* *RY , X* finitely presentable,

has a unique solution obtained as follows. Since *X* *RY* = colim *X* Eq*Y* , see Remark [4.7](#_bookmark27), *e* factors through the colimit injection *X* *f* for some *f* :

*V* −→ *V* *Y* in EQ*Y* . Thus, we have a commutative triangle

*X* ¸¸ *e* z*X*, *RY*

¸¸¸¸

¸¸¸

*e*0 ¸¸¸¸

¸

*X* *f*

(4.5)

¸t

*X* *V*

We form an equation *e*˜ : *X* + *V* −→ (*X* + *V* ) *Y* as follows:

*X*  *e*0 z*X* ,*V X* *f* z*X* ,(*V* *Y* ) inl (inr *Y* ) z(*X*,+*V* ) ((*X*+*V* ) *Y* )

inl

, *e*e

*mX*+*V Y*

z, ,

*X*+ *V* ¸

inr

(*X*+*V* ) *Y*

¸

inr *Y*

(4.6)

*V f* z*V* ,*Y*

We will prove that the given equation morphism *e* has the solution

*e*† ≡ *X*  inl z*X*,+ *V e*ez*R*,*Y .* (4.7)

From ([4.4](#_bookmark29)) and ([4.1](#_bookmark23)) we have

*jY* · (*f* *Y* ) · *f* = *i*−1 · (*f* *Y* ) · *f* = *f.* (4.8)

*Y*

Furthermore, inr : *V* −→ *X* + *V* is a morphism of equations from *f* to *e*˜ (see the lower square of ([4.6](#_bookmark30))), thus,

*e*˜· inr = *f.* (4.9)

This proves that the diagram

*X* *f*

inl (inr *Y* )

*mX*+*V*

*X* *V*  z*X* ,(*V* *Y* )

z(*X*,+*V* ) ((*X*+*V* ) *Y* ) *Y* z(*X*,+*V* ) *Y*

inl (*V* *Y* )

*RY* ¸*Y*

([4.9](#_bookmark34))

*e*e (*e*e *Y* )

([2.4](#_bookmark7))

, *e*e (*f* *Y* )

, *mRY*

*e*e *Y*

inl *V*

(*X*+*V* ) (*V* *Y* )

z*RY*, (*RY* *Y* ) *Y* z, ˛r

.. ¸¸

¸¸

*e*e *ηY*

([4.8](#_bookmark33))

,

.....

.....

*RY* (*RY* *ηY* )

,

([2.4](#_bookmark7))

¸*RY* *ηY*

¸¸¸r,

(4.10)

(*X*+*V* ) *V*

*RY* *j* ([4.4](#_bookmark29))

*RY* (*RY* *RY* )

,,,,,

,

*RY* z*RY*, *RY*

*RY*

*m*

. ,,,,,,,

*Y*..

*e*e *f*

. ,,,,,,

([2.5](#_bookmark13)) *ρY*

... ,,,,,

.,.,.,.,,,,

,˛ ¸rr.

*RY* *ρY*

,

*RY* *RY ρY* z*RY*,

commutes. Denote by

*q* : *X* *V* −→ (*X* + *V* ) *Y*

the upper horizontal morphism of ([4.10](#_bookmark35)).

The proof that ([4.7](#_bookmark32)) is a solution of *e* follows from the fact that the outward square of the diagram

*X*  inl z*X*+,*V*  *e*e *RY*

z,

¸¸¸¸¸

¸¸¸¸¸*e*0

¸

([4.6](#_bookmark30))

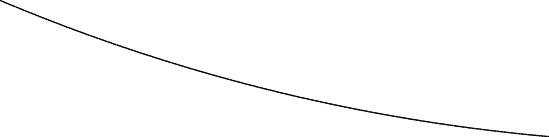
*e*e ([4.1](#_bookmark23))

˛ ,¸

*iY*

¸¸¸¸¸

¸¸r *q*



z(*X*,+*V* ) *Y*¸¸¸

z*RY*, *Y*¸¸ ([4.4](#_bookmark29))

(∗)

¸¸¸¸

¸¸¸¸¸¸

¸¸

¸¸ *RY* *ηY* ¸¸

¸¸¸

*e*† *f*

*e*e *ηY*

¸¸¸¸

¸¸¸

¸¸¸¸¸¸¸¸¸

¸¸

¸¸

¸¸¸¸¸

s

, *e*e *Y*

*jY*

˛ r

*e* ([4.5](#_bookmark31))

*X* *V ρY*

*X* *f*

,˛ r

r

*X* *RY*

*e*† *RY*

**z***RY***,** *RY*

commutes — in fact, all inner parts except (∗) commute and by ([4.10](#_bookmark35)) the triangle (∗) commutes when extended by *ρY* , the right-hand vertical arrow.

It remains to prove that the solution *e*† is unique. Given another solution

*X*  *s* z*R*,*Y*

¸

*e ρY*

, z,

(4.11)

*X* *RY s* *RY RY* *RY*

we prove that the square

*X* + *V*  *e*e z(*X*,+ *V* ) *Y*

[*s,f* ]

[*s,f* ] *Y*

, ,

z,

(4.12)

*RY iY*

*RY* *Y*

commutes. By Lemma [4.4](#_bookmark24), it follows that *e*˜= [*s, f*], thus

*e*† = *e*˜· inl = [*s, f*] · inl = *s.*

The right-hand component of ([4.12](#_bookmark37)) with domain *V* commutes:

r~ *V*  *f* z*V* , *Y* ~v

inr

*f* *X*

([4.6](#_bookmark30))

, *e*e z(*X*,+

inr *Y*

,

*f* *Y*

+ *V*

[*s,f* ]

,¸

,

*V* ) *Y*

[*s,f* ] *Y*

,

,

z*R*,*Y i* z*R*,*Y* *Y* ,r

*Y*

For the left-hand component notice first that the equation

*mRY* · (*RY* *iY* )= *iY* · *ρY* (4.13) holds. In fact, in the diagram

*Y*

*RY* *RY RY* *iY*z*R*,*Y* (*RY* *Y* )*RY* (*RY* *ηY* )*RY* (*RY* *RY* )*RY* *ρYRY* *RY*

*ρ RY*

*m*

*Y*

*Y*

z,

([2.4](#_bookmark7)) *RY*

*m*

*RY*

z,

([2.5](#_bookmark13)) *ρY*

, ,

z, z,

, ,

*RY iY*

*RY* *Y RY* *ηY*

*RY* *RY ρY* z*R*,*Y*

the horizontal morphisms are both identity morphisms (see ([4.4](#_bookmark29))), which proves that the outward square commutes. Consequently, the left-hand square commutes, since *ρY* · (*RY* *ηY* ) is an isomorphism.

The left-hand component of ([4.12](#_bookmark37)) is, due to ([4.6](#_bookmark30)), the outward square of the commutative diagram

r~ *e*

*X* ¸¸¸

¸¸*e*¸

¸¸¸

*s*

,z *RY*,,

*ρY*,,,

,,

0 ¸¸¸¸

¸

, zt,

([4.11](#_bookmark36))

*s* *RY* z,

,,,,,

*X* *V*

*X* *f*

*X* *f*

*X* *RY*

([4.1](#_bookmark23))

*RY* *RY*

*RY* *iY*

([4.13](#_bookmark38))

, *s* (*f* *Y* ) z, ,

,,,,,

*e*e·inl

*X* (*V* *Y* )

*RY* (*RY* *Y* ) *iY*

inl (inr *Y* )

,,,,, ,

,,,,,,,

,,,

,,

,

,, *Y*

, ,,,,

(*X*+*V* ) ((*X*+*V* ) *Y* )

,¸ *Y*

*mX*+*V*

[*s,f* ] ([*s,f* ] *Y* )

([2.4](#_bookmark7))

,, *mRY*

,,,,

,

,,

z, , z,r,

(*X*+*V* ) *Y*

[*s,f* ] *Y*

*RY* *Y*

(2) *RY* is free, i.e., for every iterative algebra *α* : *A* *A* −→ *A* and for every morphism *h*0 : *Y* −→ *A* there exist a unique homomorphism *h* : *RY* −→ *A* with *h* · *ηY* = *h*0.

(2a) Existence of *h*. For every equation morphism *g* : *X* −→ *X* *Y* in EQ*Y* we form *h*0 • *g* : *X* −→ *X* *A*, see Notation [3.9](#_bookmark21), and obtain the unique solution

(*h*0 • *g*)† : *X* −→ *A*. This is a cocone of the diagram Eq . In fact, given a morphism

*Y*

*g*

*X* z*X*, *Y*

*p*

,

z,'

'

*p* *Y*

,

(4.14)

*X g*'

*X* *Y*

in EQ , then (*h*0 • *g*')† · *p* is a solution of *h*0 • *g*:

*Y*

*p* (*h*0•*g*')†

*X*  z*X*,' z*A*,

¸

*g* ([4.14](#_bookmark39)) *g*'

, *p* *Y* z,' ,

([3.1](#_bookmark18))

*X* *Y*

*X* *h*

*X* *Y*

*X*' *h*0 *α*

' ,

0 ,*X* , *A*¸¸ †

,,,,,,

¸¸¸ ¸¸

*p* *A*,,,,,,

, ,

*X* *A* †

¸¸(¸*h*¸0¸•*g*') *A*

¸

s z*A*, *A*

((*h*0 •*g*) ·*p*) *A*

Thus, (*h*0 • *g*')† = (*h*0 • *g*)† · *p* as desired. Consequently, we can define *h* by the commutativity of the triangles

*RY h*  z *A*,,

¸

(4.15)

*g*

†

(*h*0•*g*)

*X*

for all *g* in EQ*Y* . We will show that this is the unique algebra homomorphism extending *h*0.

We first prove that

*Y*

*h*0 = *h* · *ηY*

= *h* · (*uY* ) *.*

In fact, the commutative diagram

*Y*  *h*0 z*A*, *A*

,

,

*Y* ([2.3](#_bookmark6))

*u*

*Y*

,

¸

,,,([2.5](#_bookmark13))

,,,,

*α*

*Y* *Y* ¸¸

¸¸¸¸

¸¸*h*¸0¸ *h*0

*A*

*A* ,,,

*u*

,

*Y* *h*0

¸¸¸¸¸¸¸¸¸ ,,

, s r˛

*Y* *A h*0 *A* z*A*, *A*

shows that *h*0 is a solution of *h* • *uY*

*Y*

in *A*, thus, by ([4.15](#_bookmark40)) we have

*h*0 = (*h* • *uY* )

*Y*

†

= *h* · (*uY* ) *.*

Next we will prove that *h* is a homomorphism. Due to Lemma [3.10](#_bookmark22), it is sufficient to prove that *h* preserves solutions, explicitly, for every equation morphism *e* : *X* −→ *X* *RY* we form

*Y*

*e* = *h* • *e* ≡ *X*  *e* z*X*, *RY* *X**h* z*X*, *A* (4.16)

and we prove that

*e*† = *h* · *e*†*.* (4.17)

Recall from ([4.6](#_bookmark30)) the equation morphisms *f* : *V* −→ *V* *Y* and *e*˜ : *X* + *V* −→

(*X* + *V* ) *Y* , and put

*f* = *h*0 • *f* ≡ *V* z*V* , *Y* z*V* , *A.* (4.18)

*f*  *V* *h*0

Let us prove that

[*e*†*, f* †]= (*h* • *e*˜)† : *X* + *V* −→ *A.* (4.19)

0

Then ([4.17](#_bookmark41)) follows: ([4.19](#_bookmark42)) implies due to ([4.7](#_bookmark32)) and ([4.15](#_bookmark40)) applied to *g* = *e*˜ the equation

*h* · *e*† = *h* · *e*˜· inl = (*h*0 • *e*˜)† · inl = *e*†*.*

Thus, the proof of (2a) will be complete by proving that the square

† †

*X* + *V*  [*e ,f* ] z*A*,

¸

*e*e

,

(*X* + *V* ) *Y α* (4.20)

(*X*+*V* ) *h*0

,

(*X* + *V* ) *A*

†

†

[*e ,f* ] *A*

z*A*, *A*

commutes.

For the right-hand component with domain *V* this follows from the com-

mutative diagram

†

*f*

r~ *V* ¸¸ z,

¸

*A*

¸

¸

*e*e·inr

*f* ¸¸¸

, ¸¸*f*

¸

([3.1](#_bookmark18))

*V* *Y*¸¸¸¸ ¸¸¸

inr *Y*

,¸

,

*V* *h*0 ¸¸¸ *α*

¸¸ ¸

¸¸ ¸

z(*X*,+ *V* ) *Y*

r ¸

,*V* *A*¸ †

(*X*+*V* ) *h*0

,,,,,,

, ¸r,

,,,,inr *A*

¸¸¸¸¸*f* *A*

¸¸¸¸

¸

r

(*X* + *V* ) *A*

†

[*e ,f* ] *A*

†

z*A*, *A*

For the right-hand component with domain *X*, first observe that the defi- nition of solution yields

*e*† = *α* · (*e*† *A*) · (*X* *h*) · *e* = *α* · (*e*† *h*) · *e,* (4.21)

and

† † †

*f* = *α* · (*f* *A*) · (*X* *h*0) · *f* = *α* · (*f* *h*0) · *f.* (4.22)

Therefore, we get a commutative diagram

r~ *X* ¸¸¸¸¸¸

*e*† z*A* ,

....˛ ,¸

¸

*e*0 ¸*e*¸¸¸

([4.21](#_bookmark44))

*α*...

, ¸¸¸t

....

*X* *V*

*X* *f* z*X* ,*RY e*† *h* z*A* , *A*¸

*X* *f*

,

([4.22](#_bookmark45))&([4.15](#_bookmark40))

†

†

*e* (*f* *h*0 )

*A* *α*

*e*e·inl

([4.6](#_bookmark30))

*X* (*V* *Y* )

inl (inr *Y* )

† † † †

z*A* ,(*A* *A*)

,,,, , ,

, , ,

,

([2.5](#_bookmark13))

*α*

[*e ,f* ] ([*e ,f* ] *h* )

, 0

, ,,,

(*X*+*V* ) ((*X*+*V* ) *Y* )

*X*+*V*

,¸

,,,

,,,

,,*mA*

*m*

*Y*

z, , ¸¸

([2.4](#_bookmark7))

,, *A*

,,,

(*X*+*V* ) *Y*

(*X*+*V* ) *h*

¸¸¸¸¸¸¸¸¸¸ ,,

0 [*e ,f* ] *h*0 ¸¸¸¸¸¸¸¸¸¸ ,,

,

† †

,

, ¸¸¸¸zz' ,¸

(*X*+*V* ) *A*

[*e*†*,f* †] *A*

*A* *A*

which shows that the left-hand component of ([4.20](#_bookmark43)) commutes.

(2b) Uniqueness. Let *k* : *RY* −→ *A* be a homomorphism (i.e., *k* · *ρY* =

*α*·(*k* *k*)) with *h*0 = *k*·*ηY* . Then for every equation morphism *g* : *X* −→ *X* *Y*

of EQ*Y* the diagram

*X g* z*R*,*Y*  *k* z*A*,

r~ *i*

¸,¸¸,¸ *ρ*  ¸

*g* ([4.1](#_bookmark23))

*Y*

¸ r

*RY* ¸*Y*¸¸

([4.4](#_bookmark29))

¸¸¸*Y*¸¸¸

*R*,*Y* ,,*R*!*Y* ,

¸¸

,,,

¸¸¸¸¸¸¸¸¸¸

¸¸¸¸¸¸

,

,,,,

,,,, *RY* *ηY*

,,,,

,,

*α*

*h*0 •*g*

*X* *Y*

*g* *Y*

z*R*,*Y* *Y*¸¸¸¸

¸¸¸¸¸¸¸

¸

,,,,

,,

,*k*, *k*

*X* *h*0

¸¸¸¸¸ ¸ ,,,,

*k* *h*0

¸ ¸¸¸ ,

,¸ ,

*X* *A*

z,

¸¸¸¸ ,,

(*k*·*g* ) *A* z*A*, *A*

t

commutes, proving

*k* · *g* = (*h*0 • *g*)† = *h* · *g ,*

see ([4.15](#_bookmark40)). Thus, *k* = *h*, since the morphisms *g*’s form a colimit cocone.

# Conclusions and Future Work

In our paper we provided a coalgebraic construction of free iterative algebras of a general “environment” called base. This is analogous to (but, unfortunately, technically more involved than) the coalgebraic construction given in [[5](#_bookmark49)] for *H*-algebras, where *H* is an arbitrary finitary endofunctor. The next goal is to introduce iterative bases in the spirit of iterative theories of C. Elgot, and prove that the monad of free iterative algebras yields a free iterative base.

The technical result will serve us to describe algebraic trees of B. Cour- celle [[9](#_bookmark52)] (i. e., trees resulting from a semantics of recursive program schemes) in a manner similar to the description of rational trees via *R*Σ, the rational monad of Σ-algebras. For that, we need to introduce a suitable base on the category FM(Set) of all finitary monads on Set.

# References

1. P. Aczel, J. Ad´amek, S. Milius and J. Velebil, Infinite Trees and Completely Iterative Theories: A Coalgebraic View, *Theoret. Comput. Science* 300 (2003), 1–45.
2. J. Ad´amek, On a Description of Terminal Coalgebras and Free Iterative Theories,

*Electron. Notes Theor. Comput. Sci.* 82.1 (2003).

1. J. Ad´amek, S. Milius and J. Velebil, Free Iterative Theories — A Coalgebraic View,

*Math. Structures Comput. Sci.* 13 (2003), 259–320.

1. J. Ad´amek, S. Milius and J. Velebil, On Rational Monads and Free Iterative Theories,

*Electron. Notes Theoret. Comput. Sci.* 69 (2003).

1. J. Ad´amek, S. Milius and J. Velebil, From Iterative Algebras to Iterative Theories (Extended Abstract), accepted for publication in *Electron. Notes Theor. Comput. Sci.*, full version available via <http://www.iti.cs.tu-bs.de/~milius>.
2. J. Ad´amek and J. Rosicky´, *Locally presentable and accessible categories*, Cambridge University Press, 1994.
3. M. Barr, Coequalizers and free triples, *Math. Z.* 116 (1970), 307–322.
4. S. Bloom and Z. E´sik, *Iterative Theories: The Equational Logic of Iterative Processes*, EATCS Monograph Series on Theoretical Computer Science, Springer-Verlag, 1993.
5. B. Courcelle, Fundamental Properties of Infinite Trees, *Theoret. Comput. Science* 25 (1983), 95–169.
6. C. C. Elgot, Monadic Computation and Iterative Algebraic Theories, in: *Logic Colloquium ’73*

(eds: H. E. Rose and J. C. Shepherdson), North-Holland Publishers, Amsterdam, 1975.

1. C. C. Elgot, S. L. Bloom and R. Tindell, On the Algebraic Structure of Rooted Trees,

*J. Comp. Syst. Sci.* 16, (1978), 361–399.

1. S. Ginali, Regular Trees and the Free Iterative Theory, *J. Comput. System Sci.* 18 (1979), 228–242.
2. P. Gabriel and F. Ulmer, *Lokal pra¨sentierbare Kategorien*, Lecture N. Math. 221, Springer- Verlag, Berlin 1971.
3. S. MacLane, *Categories for the Working Mathematician*, Springer-Verlag, 2nd edition, 1998.
4. S. Milius, On Iteratable Endofunctors, *Electron. Notes Theor. Comput. Sci.* 69 (2003).
5. E. Nelson, Iterative Algebras, *Theoret. Comput. Science* 25 (1983), 67–94.
6. J. Tiuryn, Unique Fixed Points vs. Least Fixed Points, *Theoret. Comput. Science* 12 (1980), 229–254.
7. T. Uustalu, Generalizing substitution, *Theor. Inform. Appl.* 37.4 (2003), 315–336.