Equational Properties of Iterative Monads

Jiří Adámek[a,1], Stefan Milius[a,1], Jiří Velebil[b,2]

[a] *Institut für Theoretische Informatik, Technische Universität Braunschweig, Germany*
[b] *Faculty of Electrical Engineering, Czech Technical University, Prague, Czech Republic*

Abstract

Iterative monads of Calvin Elgot were introduced to treat the semantics of recursive equations purely algebraically. They are Lawvere theories with the property that all ideal systems of recursive equations have unique solutions. We prove that the unique solutions in iterative monads satisfy all the equational properties of iteration monads of Stephen Bloom and Zoltán Ésik, whenever the base category is hyper-extensive and locally finitely presentable. This result is a step towards proving that functorial iteration monads form a monadic category over sets in context. This shows that functoriality is an equational property when considered w. r. t. sets in context.

Key words: iterative monad, iteration monad, Elgot monad, extensive category

Here thou must all distrust behind thee leave; here be vile fear extinguish'd.
Dante Alighieri, *The Divine Comedy*

1. Introduction

This paper is part of a line of research in which we deal with category-theoretic tools for the semantics of recursive specifications. It is based on an invited lecture of the middle author at the workshop on Coalgebraic Methods in Computer Science 2008. In that lecture the survey contained in this introduction was presented, and the technical results of Sections 2–7 were mentioned.

Email addresses: adamek@iti.cs.tu-bs.de (Jiří Adámek), mail@stefan-milius.eu (Stefan Milius), velebil@math.feld.cvut.cz (Jiří Velebil)

[1] The first and second author acknowledge support by the German Research Foundation (DFG) under the project "Rekursion in der koalgebraischen Semantik".

[2] The third author acknowledges the support of the Grant Agency of the Czech Republic under the Grant No. 201/06/664.

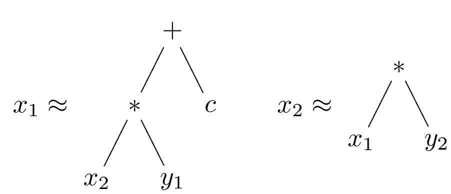
The mathematical tools and structures classically used for semantics of recursion are usually based on domains. More precisely, one works with sets or algebras of data endowed with some extra structure such as a complete partial order or a complete metric. In order to obtain the semantics of some recursive specification one then invokes either the Knaster-Tarski fixed point theorem or Banach's fixed point theorem, respectively. In this paper we follow a different purely algebraic approach that we now recall.

1.1. Iterative Theories

It was the idea of Calvin Elgot [E] to study the semantics of recursion on a level of abstraction that does not employ any concrete extra structure. He used the methods of general algebra and studied iterative algebraic theories which are those algebraic theories in the sense of Bill Lawvere [L] having the property that recursive specifications have unique solutions. More precisely, we consider systems of recursive equations of the form

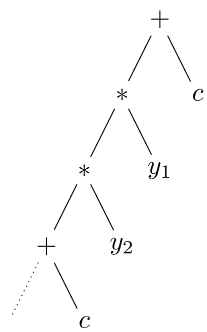
$$\begin{aligned} x_1 &\approx t_1 \\ &\vdots \\ x_n &\approx t_n \end{aligned} \tag{1.1}$$

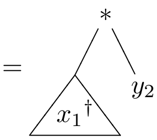
for a finite set $X = \{ x_1, \dots, x_n \}$ of variables, where each t_i is a term of the algebraic theory containing variables and parameters from a finite set $Y = \{ y_1, \dots, y_p \}$. Actually, not all recursive equations are assumed to have a unique solution but only those satisfying a mild syntactic condition—a guardedness condition excluding unproductive equations like $x \approx x$. Important examples of iterative theories are given by trees over a signature. Recall that Σ-trees for a signature Σ are all the (finite and infinite) rooted and ordered trees labelled in Σ so that leaves are labelled by constant symbols or elements of some set of generators, and inner nodes with $n > 1$ children are labelled by an n-ary operation symbol from Σ. The theory $\mathcal{T}_\Sigma$ of all Σ-trees is an iterative theory; here the right-hand sides of a system (1.1) are Σ-trees and the unique solution is obtained by tree unfolding. For example, let Σ be the signature with two binary operation symbols $+$ and $*$ and a constant c. Then the system



(1.2)

has as solution for x_1 and x_2 the infinite trees

$x_1{}^\dagger =$ 

and $x_2{}^\dagger =$ 

(1.3)

The subtheory $\mathcal{R}_\Sigma$ of $\mathcal{T}_\Sigma$ containing all *rational* trees, where a tree is called rational if it has (up to tree isomorphism) only a finite number of subtrees (see [G]), is also an iterative theory. In fact, $\mathcal{R}_\Sigma$ is the free iterative theory on the signature Σ, see [E]. Later a simpler approach to iterative theories was given independently by Evelyn Nelson [N] and Jerzy Tiuryn [T]. Both authors studied iterative *algebras*, which are algebras having the property that guarded recursive specifications have a unique solution. Denote by $T_\Sigma X$ the algebra of all Σ-trees on a set X of generators and consider the subalgebra $R_\Sigma X$ formed by all rational trees on X. Nelson [N] proved that $R_\Sigma X$ is the free iterative algebra on X, and, as already proved by Elgot et al. [EBT], the associated algebraic theory of rational Σ-trees is a free iterative theory on Σ. So it turns out that the algebraic theory of free iterative algebras is the same as the free iterative theory.

1.2. Iteration Theories

While iterative theories capture those structures in which one can obtain unique solutions of recursive equations, e. g. complete metric spaces, they do not capture cpos in which one typically considers the least or the greatest fixed point as providing a canonical solution of a recursive specification. This is the reason why Stephen Bloom and Zoltan Ésik introduced and studied iteration theories, see the monograph [BÉ]. An iteration theory is an algebraic theory endowed with an operation of taking solutions of recursive specifications, and this operation is required to satisfy certain natural axioms. These axioms are justified by the completeness theorem stating that they capture precisely all the identities valid for a (parametric) fixed-point operator in domains. More precisely, Completeness Theorem 8.2.5 of [BÉ] states that an equation follows from the axioms of iteration theories precisely if it holds in every continuous theory. This connection also appears for example in the work of Alex Simpson and Gordon Plotkin [SP], where the conditions under which the iteration theory axioms allow such a completeness theorem are studied.

Iterative theories still play an important rôle for iteration theories. Firstly, as we prove in the present paper, iterative theories are iteration theories, and secondly, every free iteration theory is a (free) iterative theory, see [BÉ].

1.3. Category-theoretic semantics

An important theme of our research is to replace the structures and methods of general algebra by category-theoretic ones. Our goal is to achieve in this way results which are at the same time more general and also conceptually clearer. The first step in this direction was done by Eric Badouel [Bd]. He proved that the assignment $X \mapsto T_\Sigma X$ of the algebra of all Σ-trees on X forms a monad on Set. A further conceptual step was the realization that in lieu of sets, signatures and trees one can work with categories, endofunctors and final coalgebras, respectively. In fact, $T_\Sigma X$ is the final coalgebra for the functor $H_\Sigma(-) + X$ where H_Σ is the polynomial endofunctor associated to the signature Σ. More generally, consider a category $\mathscr{A}$ with binary coproducts and an endofunctor H having enough final coalgebras in the sense that for each functor $H(-) + X$ there exists a final coalgebra TX. Then TX is the object assignment of a monad on $\mathscr{A}$. This has been proved by Larry Moss [Mo$_1$], and also independently and almost at the same time by Neil Ghani, Christoph Lüth, Federico DiMarchi and John Power [GLMP] and by our group [AAV, AAMV]. Moss' and our work also give an account on how to solve recursive specifications in this setting. Indeed, the notions of equation and solution can be understood abstractly as follows: a system such as (1.1) is a map $e : X \longrightarrow T(X + Y)$ and a solution of e is a map $e^\dagger : X \longrightarrow TY$ such that the square

$$\begin{tikzcd}
X \arrow[r, "e^\dagger"] \arrow[d, "e"'] & TY \\
T(X+Y) \arrow[r, "{[e^\dagger, \eta_Y]}"'] & TTY \arrow[u, "\mu_Y"']
\end{tikzcd}$$

commutes, where η_Y and μ_Y are the unit and multiplication of the monad T. The notion of guardedness of an equation morphism needs some more care, see Definitions 3.7 and 3.11. In [Mo$_1$] and [AAMV] is was proved that every guarded equation morphism $e : X \longrightarrow T(X + Y)$ has a unique solution. This makes T an iterative monad. In recent years we have further followed this line of research. For example, in [AMV$_1$] we gave a category-theoretic account of rational trees. More concretely, we proved that every finitary endofunctor H of a locally presentable category generates a free iterative monad. This monad R_H, called the *rational monad of* H, is moreover given object-wise by free iterative algebras for H.[3] In this way we generalized the classical results of Elgot and Nelson, respectively.

However, there is an important gap in our whole line of research. So far there is no analog of iterat*ion* theories in the category theoretic setting that we consider. This means that the question about the "essential" equational

[3] Recently, an alternative characterization of the initial iterative H-algebra was given by Marcello Bonsangue, Jan Rutten and Alexandra Silva [BRS]: for a Set endofunctor H this algebra is the final locally finite coalgebra.

properties of the operation assigning to a recursive equation its unique solution in a free iterative monad is not answered yet. Of course, we have to make precise what we mean by "essential" here. And this pertains to another question: what about an analog of the completeness theorem of Bloom and Ésik with respect to all the valid identities of fixed-point operators in domains? In the setting of iteration theories it turns out that the completeness theorem boils down to the monadicity of the category of iteration theories over the category of signatures. More precisely, denote for a signature Σ by $\mathcal{R}_\Sigma$ the free iterative theory of rational Σ-trees, and let $\Sigma_\perp$ denote the signature Σ extended by a new constant symbol $\perp$. Bloom and Ésik proved that for each signature Σ the theory $\mathcal{R}_{\Sigma_\perp}$ is a free iteration theory on Σ. To put this more categorically denote by $U : \mathsf{ITh} \longrightarrow \mathsf{Sgn}$ the forgetful functor from the category of iteration theories to the category of signatures, which assigns to a theory T the signature $(T(1,n))_{n\in\mathbb{N}}$. Then the above description of free iteration theories states that U has a left adjoint. This yields an induced monad Rat on Sgn. Furthermore, we have recently proved the following:

Theorem 1.1. [AMV$_2$] *The forgetful functor $U : \mathsf{ITh} \longrightarrow \mathsf{Sgn}$ is monadic, i. e., iteration theories are precisely the Eilenberg-Moore algebras for the monad* Rat.

This theorem states that iteration theories define the largest category of structures in which the theory of rational trees on a signature is free on that signature. How does this relate to the Completeness Theorem of Bloom and Ésik? Firstly, recall that $\mathcal{R}_{\Sigma_\perp}$ can also be characterized as the free rational theory [ADJ]; those are theories in which solutions of recursive equations are obtained by taking least fixed points in a suitable order structure. Secondly, the results of Kelly and Power [KP] state the every finitary monad on a locally finitely presentable category can be presented by operations and equations, so that the category of algebras for the monad is equivalent to the category of algebras with such operations satisfying the given equations. For example, a consequence of Theorem 1.1 is that the presentation of the monad Rat is precisely the presentation of iteration theories using one operation of taking solutions subject to the iteration theory axioms. Now if an identity holds for least fixed points in continuous theories (equivalently, the identity holds in rational theories, see our discussion in [AMV$_2$]), then it holds, in particular, in the free rational theory $\mathcal{R}_{\Sigma_\perp}$. Since $\mathcal{R}_{\Sigma_\perp}$ is also the free iteration theory this identity can only hold if it follows from the axioms of iteration theories.

Now we want to work towards establishing a result similar to Theorem 1.1 in our more general category theoretic setting. To make this more precise, consider the category $\mathsf{Fin}[\mathscr{A}, \mathscr{A}]$ of finitary endofunctors on a hyper-extensive locally finitely presentable category $\mathscr{A}$. Hyper-extensivity, introduced in [ABMV], means that $\mathscr{A}$ has very well-behaved coproducts (see Definition 2.9 below); for example sets, posets, graphs, unary algebras and presheaves form hyper-extensive categories. We would like to obtain the largest category of monads in which the rational monad of an endofunctor of $\mathscr{A}$ is free on that endofunctor. We call the objects of the category we are looking for *Elgot monads* to honor Calvin Elgot, whose work has been a great inspiration for us. In analogy to iteration theories, Elgot monads are defined as monads in which there is an operation of taking solutions of all recursive equations, and this operation is required to satisfy certain natural axioms, which are related but not identical to the iteration theory axioms. Our aim is to eventually prove that Elgot monads are monadic over $\mathsf{Fin}[\mathscr{A}, \mathscr{A}]$. More precisely, denote by $\mathsf{Elgot}(\mathscr{A})$ the category of Elgot monads. There is a canonical forgetful functor $V : \mathsf{Elgot}(\mathscr{A}) \longrightarrow \mathsf{Fin}[\mathscr{A}, \mathscr{A}]$. Our long term goal is to prove that the free Elgot monad on an endofunctor H is the rational monad of $H_\perp = H + C_1$, where C_1 is the constant functor on value 1. In fact, we will prove more:

Theorem 1.2. *The forgetful functor $V : \mathsf{Elgot}(\mathscr{A}) \longrightarrow \mathsf{Fin}[\mathscr{A}, \mathscr{A}]$ has a left adjoint given on objects by $H \mapsto R_{H_\perp}$. Moreover, V is monadic, i. e., the category of Eilenberg-Moore algebras induced by the above adjunction is isomorphic to the category of Elgot monads.*

Although easily stated, the proof of this theorem requires several technically involved steps:

(1) The operation of taking the unique solution of recursive equations in a rational monad R_H has to be extended from all guarded recursive equations to all equations.
(2) We need to establish that the extended operation satisfies all the equational properties required of an Elgot monad, that is, we have to prove that rational monads are Elgot monads.
(3) We prove that V has a left adjoint as stated in the above theorem.
(4) Finally, we prove that V is monadic.

Step (1) is the topic of our joint work with Reinhard Börger [ABMV]. There it is proved that if an iterative monad S is equipped with a global element $\perp : 1 \longrightarrow S0$ (such monads are called strict), then every equation morphism e has a unique strict solution $e^\dagger$. And step (2) is treated in the present paper: in all hyper-extensive locally finitely presentable categories we prove that the operation $e \mapsto e^\dagger$ satisfies all axioms of iteration theories, and in addition it satisfies a property called functoriality[4]. Thus, the implication

$$\text{iterative theory} \Longrightarrow \text{iteration theory}$$

holds.

Our results here are related to the work of Larry Moss [Mo_2] who proved (1) and (2) for the monad of final coalgebras TX for $H(-) + X$, where H is an endofunctor of Set. To keep the current paper at a reasonable length we decided to treat the steps (3) and (4) in a subsequent publication [AMV_4].

1.4. Contents of the paper.

We begin in Section 2 with some technical preliminaries, and we continue in Section 3 by recalling iterative monads and the results of [ABMV] which establish the above step (1): for a strict iterative monad every equation morphism has

[4] Functoriality is called functorial dagger implication in [BÉ].

a unique strict solution. In Section 4 we introduce iteration monads in analogy to iteration theories as those monads with an operation of taking solutions of equations, where this operation satisfies exactly the equational properties of iteration theories. We also introduce the stronger notion of an Elgot monad, and we provide some examples of Elgot monads. Every Elgot monad is, of course, an iteration monad, but the former have a simpler axiomatics. In Section 5 and 6 we prove that every strict iterative monad is an Elgot monad. We discuss conclusions and future work in Section 7.

2. Preliminaries

In this section we recall the basic technical machinery necessary for our work. Let us now explain our general category-theoretic setting. Informally, we need to work in a category $\mathscr{A}$ in which the notion of "finite object" makes sense—analogously to a finite set, a finitely presentable algebra etc. In addition, we need coproducts in $\mathscr{A}$ which are as "nicely behaved" as disjoint union in sets. We now make all of this precise.

2.1. Locally finitely presentable categories and finitary functors

A functor is called *finitary* if it preserves filtered colimits. An object A is *finitely presentable* if its hom-functor $\mathscr{A}(A, -)$ is finitary. A category $\mathscr{A}$ is *locally finitely presentable*, see [GU] or [AR], if it is cocomplete and has a set of finitely presentable objects whose closure under filtered colimits is all of $\mathscr{A}$.

For example, the categories Set, Pos (posets and order-preserving maps), Gra (graphs and homomorphisms), Group (groups and homomorphisms), and, more generally, every finitary variety of algebras are locally finitely presentable. The finitely presentable objects are finite sets, posets, graphs, and those groups (resp. algebras) which are presented by finitely many generators and equations, respectively. In contrast, the category CPO of complete partial orders and continuous maps is not locally finitely presentable; for example, no non-trivial CPO is finitely presentable.

2.2. Monads and Algebraic Theories

Recall that a monad is a triple $\mathbb{S} = (S, \eta, \mu)$ where $\eta : Id \longrightarrow S$ (the unit) and $\mu : SS \longrightarrow S$ (the multiplication) are natural transformations such that

$$\mu \cdot \eta S = id = \mu \cdot S\eta \qquad \text{and} \qquad \mu \cdot S\mu = \mu \cdot \mu S$$

hold (see e. g. [ML]). A monad is called *finitary* if its underlying functor is finitary. A *monad morphism* m from a monad $\mathbb{S} = (S, \eta, \mu)$ to $\overline{\mathbb{S}} = (\overline{S}, \overline{\eta}, \overline{\mu})$ is a natural transformation $m : S \longrightarrow \overline{S}$ such that $m \cdot \eta = \overline{\eta}$ and $\mu \cdot (m * m) = m \cdot \overline{\mu}$, where $m * m : \overline{SS} \longrightarrow SS$ is the parallel composition of natural transformations.

Notation 2.1. Let $\mathbb{S}$ be a monad on $\mathscr{A}$. The *Kleisli category* $\mathscr{A}_{\mathbb{S}}$ of $\mathbb{S}$ has the same objects as $\mathscr{A}$, and a morphism f from A to B in $\mathscr{A}_{\mathbb{S}}$ is a morphism $f : A \longrightarrow SB$ in $\mathscr{A}$; we write

$$f : A \multimap B.$$

Given $f : A \longrightarrow\!\!\circ\!\!\longrightarrow B$ and $g : B \longrightarrow\!\!\circ\!\!\longrightarrow C$, the composite in the Kleisli category is

$$A \xrightarrow{f} SB \xrightarrow{Sg} SSC \xrightarrow{\mu_C} SC.$$

There is a canonical identity-on-objects functor $J : \mathscr{A} \longrightarrow \mathscr{A}_{\mathbb{S}}$ given on morphisms $k : A \longrightarrow B$ by

$$Jk = (\,A \xrightarrow{k} B \xrightarrow{\eta_B} SB\,).$$

We will call every morphism Jk a *base morphism*, and we usually drop J and write Jk as if it were the morphism $k : A \longrightarrow B$ in $\mathscr{A}$. So, for example, given a Kleisli morphism $f : B \longrightarrow\!\!\circ\!\!\longrightarrow C$ and base morphisms $k : A \longrightarrow B$ and $h : C \longrightarrow D$ then we have the composite $h \cdot f \cdot k : A \longrightarrow\!\!\circ\!\!\longrightarrow D$ which is just the composite

$$A \xrightarrow{k} B \xrightarrow{f} SC \xrightarrow{Sh} SD$$

in the base category $\mathscr{A}$. Observe that the Kleisli category has the same coproducts as the base category, thus, no notational distinction is needed.

Algebraic theories were introduced by Lawvere [L] in order to capture universal algebra by category-theoretic means. It is well known that finitary monads on Set and algebraic theories form equivalent categories, see [Li]. In fact, recall that an algebraic theory is a category with the set of natural numbers as objects and with coproducts given by the sum of natural numbers. Let $\mathbb{S} = (S, \eta, \mu)$ be a finitary monad of Set. Then its Kleisli category restricted to the natural numbers $n = \{\,0, \ldots, n{-}1\,\}$ forms an algebraic theory T where $T(n, m) = \mathsf{Set}_{\mathbb{S}}(n, m)$. Conversely, for any algebraic theory T we obtain a finitary monad by left Kan extension of the functor $T(1, -)$ along the canonical inclusion $J : \mathbb{N} \longrightarrow \mathsf{Set}$ where $\mathbb{N}$ is considered as the category of sets $n = \{\,0, \ldots, n-1\,\}$ with all functions between them. These two constructions extend to the level of homomorphisms (of monads and algebraic theories, respectively), and it is not difficult to prove that they are mutually inverse (up to isomorphism).

2.3. Extensive and hyper-extensive categories

A crucial step in the proof of our main result in this paper is the *groundedness* analysis. For an equation system (1.1) this amounts to identifying those variables with a non-productive recursive definition such as $x \approx y$, $y \approx x$ etc.

We introduced hyper-extensive categories in [ABMV] as the appropriate notion of a category in which coproducts are well-behaved enough to enable us to make the groundedness analysis for an abstract equation morphism $e : X \longrightarrow T(X + Y)$. Every hyper-extensive category is an extensive category in the sense of [CLW]. Since we will make heavy use of extensivity of our base category $\mathscr{A}$ we recall the basic definitions, and, for the convenience of the reader, we prove in this section all the properties of extensive categories we shall need later. On first reading one may skip this part and come back later to it when we make use of those properties in Sections 5 and 6.

Throughout the paper we denote the injections of any coproduct $A+B$ by inl and inr, then $\nabla = [\mathsf{inl}, \mathsf{inr}] : A + A \longrightarrow A$ is the codiagonal, and $\mathsf{can} = [S\mathsf{inl}, S\mathsf{inr}] : SA + SB \longrightarrow S(A+B)$ is the canonical morphism, where S is any endofunctor.

Definition 2.2. [CLW] A category $\mathscr{C}$ is called *extensive*, if it has binary coproducts, and for each pair A, B of objects the canonical functor

$$\mathscr{C}/A \times \mathscr{C}/B \longrightarrow \mathscr{C}/(A+B) \tag{2.1}$$

given by formation of coproducts is an equivalence of categories.

Proposition 2.3. [CLW] *A category $\mathscr{C}$ is extensive iff it has pullbacks along coproduct injections and every commutative diagram*

$$\begin{tikzcd}
A_1 \arrow[r, "a_1"] \arrow[d, "f_1"'] & A \arrow[d, "f"] & A_2 \arrow[l, "a_2"'] \arrow[d, "f_2"] \\
B_1 \arrow[r, "\mathsf{inl}"'] & B_1 + B_2 & B_2 \arrow[l, "\mathsf{inr}"]
\end{tikzcd} \tag{2.2}$$

consists of a pair of pullback squares iff the top row is a coproduct diagram with $f = f_1 + f_2$.

Proposition 2.4. [CLW] *A category with binary coproducts and pullbacks along their injections is extensive iff coproducts are*

(1) *disjoint, i. e. coproduct injections are monomorphic and the pullback of* inl *and* inr *has domain* 0 *(the initial object), and*
(2) *universal, i. e. pullbacks of coproduct diagrams are again coproduct diagrams (cf.* (2.2)*).*

Remark 2.5. Notice that in an extensive category a coproduct $f+g : C+D \longrightarrow A+B$ is an isomorphism iff each of the components f and g are. This follows from the fact that the equivalence (2.1) preserves and reflects isomorphisms.

Proposition 2.6. *In an extensive category $\mathscr{C}$ coproducts commute with pullbacks. More precisely, if the two squares*

$$\begin{tikzcd}
A_i \arrow[r, "q_i"] \arrow[d, "p_i"'] & B_i \arrow[d, "g_i"] \\
C_i \arrow[r, "f_i"'] & D_i
\end{tikzcd} \qquad i = 1, 2,$$

are pullbacks then so is their coproduct

$$\begin{tikzcd}
A_1 + A_2 \arrow[r, "q_1+q_2"] \arrow[d, "p_1+p_2"'] & B_1 + B_2 \arrow[d, "g_1+g_2"] \\
C_1 + C_2 \arrow[r, "f_1+f_2"'] & D_1 + D_2
\end{tikzcd}$$

Proof. Observe that $(A_i, f_i \cdot p_i)$, $i = 1, 2$, is the product of (C_i, f_i) and (D_i, g_i) in $\mathscr{C}/D_i$. Since the canonical functor (2.1) is an equivalence, it preserves products and so $(A_1 + A_2, (f_1 + f_2) \cdot (p_1 + p_2))$ is a product of $(C_1 + C_2, f_1 + f_2)$ and $(D_1 + D_2, g_1 + g_2)$ in $\mathscr{C}/(D_1 + D_2)$. Equivalently, the desired square is a pullback. $\square$

Lemma 2.7. *In an extensive category, if the two squares*

$$\begin{tikzcd} A \arrow[r, "x"] \arrow[d, "a"'] & X \arrow[d, "e"] \\ C \arrow[r, "c"'] & Z \end{tikzcd} \quad \text{and} \quad \begin{tikzcd} B \arrow[r, "y"] \arrow[d, "b"'] & Y \arrow[d, "f"] \\ C \arrow[r, "c"'] & Z \end{tikzcd}$$

are pullbacks, then so is the square

$$\begin{tikzcd} A + B \arrow[r, "x+y"] \arrow[d, "{[a,b]}"'] & X + Y \arrow[d, "{[e,f]}"] \\ C \arrow[r, "c"'] & Z \end{tikzcd}$$

Proof. First we prove that the square

$$\begin{tikzcd} C + C \arrow[r, "c+c"] \arrow[d, "\nabla"'] & Z + Z \arrow[d, "\nabla"] \\ C \arrow[r, "c"'] & Z \end{tikzcd}$$

is a pullback. Indeed, for a pair $d : D \longrightarrow C$, $d' : D \longrightarrow Z+Z$ of morphisms with $c \cdot d = \nabla \cdot d'$ we have, by extensivity, that $D = D_1 + D_2$ such that $d' = d_1' + d_2'$ for some objects D_1, D_2 and some morphisms $d_i' : D_i \longrightarrow Z$, $i = 1, 2$, and therefore $d = [d_1, d_2] : D_1 + D_2 \longrightarrow C$ for some d_1 and d_2. Then $d_1 + d_2$ is the desired unique mediating morphism. The result now follows from Proposition 2.6 by composing two pullback squares to obtain the diagram

$$\begin{tikzcd} A + B \arrow[r, "x+y"] \arrow[d, "a+b"'] \arrow[dd, bend right=60, "{[a,b]}"'] & X + Y \arrow[d, "e+f"'] \arrow[dd, bend left=60, "{[e,f]}"] \\ C + C \arrow[r, "c+c"'] \arrow[d, "\nabla"] & Z + Z \arrow[d, "\nabla"'] \\ C \arrow[r, "z"'] & Z \end{tikzcd}$$

$\square$

Lemma 2.8. *In an extensive category, an object $A + B$ is finitely presentable iff A and B both are.*

Proof. It suffices to prove that A is finitely presentable, if $A + B$ is. Given a filtered colimit $C = \operatorname{colim}_{i\in I} C_i$ with the colimit cocone $c_i : C_i \longrightarrow C$, $i \in I$, and given a morphism $p : A \longrightarrow C$, we need to prove that (a) there exists an $i \in I$ and a morphism $p' : A \longrightarrow C_i$ such that $c_i \cdot p' = p$ and (b) given two factorizations $p', p'' : A \longrightarrow C_i$ with $c_i \cdot p' = c_i \cdot p'' = p$, then there exists a $j \in I$ and a connecting morphism $p_{ij} : C_i \longrightarrow C_j$ in the diagram with $p_{ij} \cdot p' = p_{ij} \cdot p''$. Indeed, item (b) is clear: we have the filtered colimit $C + B = \operatorname*{colim}_{i\in I} C_i + B$ with injections $c_i + id_B$. So given p' and p'' as above then $p' + id_B$ and $p'' + id_B$ are two factorizations of $p + id_B$ through $c_i + id_B$ and so, since $A + B$ is finitely presentable, these two morphisms are equalized by $c_{ij} + id_B$ for some $j \in I$ and some connecting morphism c_{ij}.

For (a) we can take some $i \in I$ and some factorization $p' : A + B \longrightarrow C_i + B$ of $p + id_B$ through $c_i + id_B$, since $A + B$ is finitely presentable. Now consider the diagram

$$\begin{tikzcd}
A \arrow[r, "\mathsf{inl}"] \arrow[d, dashed, "p''"] \arrow[dd, bend right=60, "p"'] & A + B \arrow[d, "p'"] \arrow[dd, bend left=60, "p+B"] \\
C_i \arrow[r, "\mathsf{inl}"'] \arrow[d, "c_i"] & C_i + B \arrow[d, "c+B"] \\
C \arrow[r, "\mathsf{inl}"'] & C + B
\end{tikzcd}$$

The outside and lower squares are pullback squares by extensivity, and from the universal property of the lower pullback we obtain the dashed morphism p'' as indicated. This is the desired factorization. $\square$

It is not difficult to prove that in an extensive category the following holds: given disjoint subobjects $a_i : A_i \longrightarrow B$, $i = 1, 2$, if a_1 and a_2 are coproduct injections, then so is $[a_1, a_2] : A_1 + A_2 \longrightarrow B$. We need the following generalization to countably many subobjects:

Definition 2.9. ([ABMV]) A category is called *hyper-extensive* if it has countable coproducts which are (a) disjoint, (b) universal and (c) given pairwise disjoint subobjects $a_i : A_i \longrightarrow B$, $i \in \mathbb{N}$, each of which is a coproduct injection, then $[a_i] : \coprod_{i\in\mathbb{N}} A_i \longrightarrow B$ is also a coproduct injection.

Remark 2.10. We will not use this definition, but instead, the following characterization proved in [ABMV]. Recall that an object A is called *connected* if the functor $\mathscr{A}(A, -)$ preserves coproducts.

Proposition 2.11. [ABMV] *A locally finitely presentable category $\mathscr{A}$ is hyper-extensive iff every object of A is a coproduct of connected objects (called* components *of A).*

Example 2.12. Recall that in Set a connected object is a singleton set, and connected posets or graphs are the usual concepts. The categories Set, Pos, Gra and $[\mathscr{C}^{op}, \mathsf{Set}]$ (presheaves) are hyper-extensive locally finitely presentable. We already mentioned that the category CPO is not locally finitely presentable. And the category of Jónsson-Tarski algebras (i. e., algebras with a binary operation which is an isomorphism) is extensive and locally finitely presentable but not hyper-extensive.

Remark 2.13. (1) It is not difficult to show that in an extensive category an object is connected iff it is non-initial and *indecomposable*, i. e., whenever it is a binary coproduct, then one of the coproduct components is initial. See [ABMV] for details.

(2) In an extensive category, initial objects are *strict*, i. e., each object A with a morphism $A \longrightarrow 0$ is an initial object, see [CLW].

Proposition 2.14. *Let A be a finitely presentable object of a hyper-extensive locally finitely presentable category. Then A has finitely many components (n, say), and every decomposition of A into a coproduct of non-initial objects has at most n summands.*

Proof. We start by writing $A = \coprod_{i \in I} A_i$ as a coproduct of connected objects. Then A is the filtered colimit of all subcoproducts $\coprod_{i \in J} A_i$, where J is a finite subset of I. We denote by in_J the corresponding colimit injection. Since A is finitely presentable we see that there exists a finite $J \subseteq I$ and a morphism $p : A \longrightarrow \coprod_{i \in J} A_i$ such that $\mathsf{in}_J \cdot p = id_A$. Thus, in_J is a split epimorphism; it is also a monomorphism since it is the left-hand injection of the coproduct $\coprod_{i \in J} A_i + \coprod_{i \in I \setminus J} A_i = A$. We have proved that A is a finite coproduct of connected objects.

Now let $A = A_1 + \cdots + A_n$ be a finite decomposition of A into connected objects with coproduct injections $\mathsf{in}_1, \ldots, \mathsf{in}_n$. Suppose we have another decomposition $A = \coprod_{i \in I} B_i$ with injections in'_i, where each B_i is non-initial. Then the j-th injection $\mathsf{in}_j : A_j \longrightarrow \coprod_{i \in I} B_i$ factorizes, since A_j is connected, through $\mathsf{in}'_{i(j)}$ for some $i(j) \in I$. It is sufficient to prove that $I = \{ i(1), \ldots, i(n) \}$. Indeed, given $i \in I$ form pullbacks

$$\begin{tikzcd} P_j \arrow[r] \arrow[d] & A_j \arrow[d, "\mathsf{in}_j"] \\ B_i \arrow[r, "\mathsf{in}'_i"'] & A \end{tikzcd} \qquad (j = 1, \ldots, n)$$

to obtain $B_i = P_1 + \cdots + P_n$. If $i \neq i(j)$ we see that P_j is initial; indeed, the above pullback is obtained by composing the two pullbacks

$$\begin{tikzcd} P_j \arrow[r] \arrow[d] & A_j \arrow[d] \arrow[dd, bend left=60, "\mathsf{in}_j"] \\ 0 \arrow[r] \arrow[d] & B_{i(j)} \arrow[d, "\mathsf{in}'_{i(j)}"'] \\ B_i \arrow[r, "\mathsf{in}'_i"'] & A, \end{tikzcd}$$

where the lower square is a pullback by disjointness, and then the upper pullback shows P_j to be initial by the strictness of the initial object (cf. 2.13(2)). Since all B_i are assumed to be non-initial, there must be at least one $j = 1, \dots, n$ with $i = i(j)$ for each $i \in I$. $\square$

3. Iterative Monads

In the present section we recall the notion of an iterative monad, see [E, AMV$_1$]. In iterative monads ideal equation morphisms have unique solutions. In [ABMV] we worked with strict iterative monads, which are iterative monads S equipped with a global element $\perp : 1 \longrightarrow S0$. This element can serve as a unique solutions of "ambiguous" equations such as $x \approx x$. This leads to the notion of a strict solution, and, as shown in loc. cit., *every* equation morphism has a unique strict solution. In the subsequent sections we then study the equational properties of unique strict solutions.

Assumption 3.1. Throughout the rest of the paper we assume that a finitary monad $\mathbb{S} = (S, \eta, \mu)$ is given on a hyper-extensive, locally finitely presentable category $\mathscr{A}$.

Recall that we work with the Kleisli category $\mathscr{A}_{\mathbb{S}}$ where morphisms are denoted by $f : X \longrightarrow\!\!\!\circ\!\!\!\longrightarrow Y$.

Definition 3.2. An *equation morphism* is a Kleisli morphism $e : X \multimap X + A$, where X is a finitely presentable object (of variables) and A an arbitrary object (of parameters).

A *solution* of e is a Kleisli morphism $e^\dagger : X \multimap A$ such that the triangle

$$\begin{tikzcd} X \arrow[r, "e^\dagger"] \arrow[d, "e"'] & A \\ X + A \arrow[ur, "{[e^\dagger, id_A]}"'] & \end{tikzcd} \tag{3.1}$$

commutes (in the Kleisli category of $\mathbb{S}$).

Remark 3.3. For a given object A of parameters, the equation morphisms $e : X \longrightarrow S(X + A)$ are precisely the coalgebras for the endofunctor $S(- + A)$. Given another equation morphism $f : Y \longrightarrow S(Y + A)$, then coalgebra homomorphisms

$$\begin{tikzcd} X \arrow[r, "h"] \arrow[d, "e"'] & Y \arrow[d, "f"] \\ S(X + A) \arrow[r, "S(h+id_A)"'] & S(Y + A) \end{tikzcd} \tag{3.2}$$

are called *homomorphisms of equations*.

Example 3.4. The monad on Set given by

$$SX = X + 1$$

has pointed sets as algebras. An equation morphism can be viewed as a partial function e from X to $X + A$ (in Set), and a solution as a partial function $e^\dagger$ from X to A with

(i) $e^\dagger(x) = a$ whenever $e(x) = a \in A$,

(ii) $e^\dagger(x) = e^\dagger(y)$ whenever $e(x) = y \in X$

and

(iii) $e^\dagger(x)$ undefined whenever $e(x)$ is undefined.

Consequently, $e^\dagger$ is uniquely determined on all variables except for the *ungrounded* ones for which there exists a cycle $x = x_0$, x_1, $\dots$, $x_n = x$ in the sense that $e(x_i) = x_{i+1}$, $i = 0, 1, \dots, n-1$.

Notation 3.5. Given an equation morphism $e : X \multimap X + A$ and a morphism $h : A \multimap B$ then $h \bullet e : X \multimap X + B$ denotes the equation morphism

$$h \bullet e \equiv \; X \overset{e}{\multimap} X + A \overset{X+h}{\multimap} X + B.$$

Observe that in Example 3.4 given a solution $e^\dagger$ of e, then $h \cdot e^\dagger$ is a solution of $h \bullet e$.

Example 3.6. Let $\mathbb{T}$ denote the monad given by the functor

$TX =$ all (finite and infinite) binary trees with leaves labelled in X

where the action of T on mappings is the relabeling of leaves, the monad unit η sends $x \in X$ to the root-only tree $\eta_X(x)$ labelled by x, and $\mu_X : TTX \longrightarrow TX$ is the canonical function interpreting a tree labelled in TX as a tree labelled in X.

(1) An equation morphism $e : X \multimap X + A$ where $X = \{x_1, \dots, x_n\}$ corresponds to a system of equations (1.1), where each t_i is a tree labelled with leaves labelled in $X + A$. It is very easy to see that if every variable x_i is *guarded*, i.e., no right-hand tree t_i is a single variable, then there is a unique solution $e^\dagger : X \multimap A$. The tree $e^\dagger(x_i)$ is then given by unfolding the variable x_i according to the equations above: the unfolding stops whenever a parameter is encountered, and it can be finite or infinite, but it is always uniquely determined. For a concrete example see (1.2) and (1.3).

(2) More generally: let us call a variable x_i *grounded* if among the given equations there are k equations of the form $x_i \approx x_{j_1}, x_{j_1} \approx x_{j_2}, \dots, x_{j_{k-1}} \approx x_{j_k}$ having a variable on both sides and where x_{j_k} is guarded. If all the variables x_1, $\dots$, x_n are grounded, there is a unique solution. (In contrast, the system $x_1 \approx x_2$, $x_2 \approx x_1$ does not have a unique solution.)

While the notion of equation and solution can be expressed for any monad (c.f. Definition 3.2), the notion of a guarded variable requires us to be able to speak about non-variables in a monad. Elgot's notion of an ideal theory allows exactly this. We now recall the corresponding notion for a monad.

Definition 3.7. (See [AAMV].) A monad $\mathbb{S}$ is called *ideal* provided that $\eta : Id \longrightarrow S$ is a coproduct injection of a coproduct

$$S = S' + Id$$

with the second injection denoted by $\sigma : S' \longrightarrow S$, and there exists a restriction of $\mu : SS \longrightarrow S$ to a natural transformation $\mu' : S'S \longrightarrow S'$ in the sense that the square

$$\begin{tikzcd} S'S \arrow[r, "\mu'"] \arrow[d, "\sigma S"'] & S' \arrow[d, "\sigma"] \\ SS \arrow[r, "\mu"'] & S \end{tikzcd}$$

commutes.

Example 3.8.

(1) The monad $X + 1$ of Example 3.4 is ideal, here S' is the constant functor with value 1.
(2) The monad $\mathbb{T}$ of binary trees of Example 3.6 is ideal, here S' is the functor assigning to X the set $TX \setminus \eta_X[X]$.
(3) The submonad of (2) above of all *rational* binary trees on X, which means trees having up to isomorphism only finitely many subtrees, is ideal.
(4) Similarly, the submonad of (2) of all finite binary trees on X is ideal.
(5) Free monads are ideal. Let H be a finitary endofunctor of $\mathscr{A}$. As proved by Michael Barr [Ba], the free monad $\mathbb{F}$ on H is given object-wise by

$$FZ = \text{a free } H\text{-algebra on } Z.$$

Let $\eta_Z : Z \longrightarrow FZ$ be the universal arrow of the free H-algebra and let $\sigma_Z : HFZ \longrightarrow FZ$ be its structure morphism. Then we have $FZ = HFZ + Z$ with coproduct injections σ_Z and η_Z. It follows that $\mathbb{F}$ is an ideal monad with $F' = HF$ and $\mu' = H\mu : HFF \longrightarrow HF$.
(6) The monads of free semigroups, $SX = X^+$, free unary algebras $SX = \mathbb{N} \times X$ and free commutative binary algebras are all ideal.
(7) "Classical" varieties often fail to be ideal: for example, take the free group monad S and define $S'X = SX \setminus \eta[X]$, then S' is not a subfunctor of S. In fact, for $x \neq y$ in X take a map $f : X \longrightarrow Z$ with $f(x) = f(y) = z$. Then the term $x \cdot y^{-1} \cdot x \in S'X$ has as its image under Sf the element $z \in \eta[Z]$.

(8) Let $\mathscr{F}$ denote the category of finite sets and all maps between them. The the presheaf category $\mathsf{Set}^{\mathscr{F}}$ can be interpreted as the category of "sets in context", see [FPT]. This is used for the semantics of untyped λ-calculus. The functor $HX = X \times X + X^V$, where $V : \mathscr{F} \longrightarrow \mathsf{Set}$ is the canonical embedding, has a free monad $\mathbb{F}$ which is ideal. It is proved in loc. cit. that $\mathbb{F}(V)$ assigns to a context the set of λ-terms (up to α-equivalence) in that context. The presheaf $\mathbb{F}(V)$ is a monoid in $\mathsf{Set}^{\mathscr{F}}$, and this yields, equivalenly, a finitary monad on Set, and this monad is ideal, too.

Remark 3.9. Let $\mathbb{S}$ be an ideal monad on $\mathscr{A}$. Then the following are pullbacks:

$$
\begin{tikzcd}
S'SX + S'X \arrow[rr, "{[\sigma_{SX}, \eta_{SX} \cdot \sigma_X]}"] \arrow[d, "{[\mu'_X, id]}"'] & & SSX \arrow[d, "\mu_X"] & & X \arrow[ll, "\eta_{SX} \cdot \eta_X"'] \arrow[d, equal] \\
S'X \arrow[rr, "\sigma_X"'] & & SX & & X. \arrow[ll, "\eta_X"]
\end{tikzcd}
\tag{3.3}
$$

Also for any morphism $h : X \longrightarrow Y$ we have the following pullbacks:

$$
\begin{tikzcd}
S'X \arrow[r, "\sigma_X"] \arrow[d, "S'h"'] & SX \arrow[d, "Sh"] & X \arrow[l, "\eta_X"'] \arrow[d, "h"] \\
S'Y \arrow[r, "\sigma_Y"'] & SY & Y \arrow[l, "\eta_Y"]
\end{tikzcd}
\tag{3.4}
$$

Indeed, this follows from the extensivity of $\mathscr{A}$ since both diagrams commute and in both diagrams the top and bottom rows comprise coproduct diagrams. Furthermore, the diagram

$$
\begin{tikzcd}
S'X + S'B \arrow[r, "\sigma_X + \sigma_B"] \arrow[d, "\mathsf{can}"'] & SX + SB \arrow[d, "\mathsf{can}"'] & X + B \arrow[l, "\eta_X + \eta_B"'] \arrow[d, equal] \\
S'(X+B) \arrow[r, "\sigma_{X+B}"'] & S(X+B) & X + B \arrow[l, "\eta_{X+B}"]
\end{tikzcd}
\tag{3.5}
$$

comprises a pair of pullback squares; to see this, apply Lemma 2.7.

Definition 3.10. Let $\mathbb{S}$ and $\overline{\mathbb{S}}$ be ideal monads. An *ideal monad morphism* from $\mathbb{S}$ and $\overline{\mathbb{S}}$ is a monad morphism $m : S \longrightarrow \overline{S}$ such that there exists a (necessarily unique) restriction $m' : S' \longrightarrow \overline{S}'$, i. e., the square below commutes:

$$
\begin{tikzcd}
S' \arrow[r, "m'"] \arrow[d, "\sigma"'] & \overline{S}' \arrow[d, "\overline{\sigma}"] \\
S \arrow[r, "m"'] & \overline{S}
\end{tikzcd}
$$

Definition 3.11. (See [AMV$_1$].) Let $\mathbb{S}$ be an ideal monad. An equation morphism $e : X \longrightarrow\!\!\circ\!\!\longrightarrow X + A$ is called *ideal* if it is disjoint from η_{X+A}, that is, if e factorizes in $\mathscr{A}$ as follows:

$$\begin{tikzcd} X \arrow[r, "e"] \arrow[dr, dotted] & S(X+A) \\ & S'(X+A) \arrow[u, "\sigma_{X+A}"'] \end{tikzcd}$$

The monad $\mathbb{S}$ is called *iterative* if every ideal equation morphism has a unique solution.

Remark 3.12. The formulation in [AMV$_1$] uses guarded (rather than ideal) equation morphisms, but as proved in [ABMV] this is equivalent due to extensivity. Also we can restrict ourselves to equation morphisms where (not only X but also) the object A is finitely presentable, see [ABMV].

Notation 3.13. In the category of all finitary monads on $\mathscr{A}$ we denote by

$$\mathsf{FM}_{\mathrm{id}}(\mathscr{A})$$

the (non-full) subcategory of ideal monads and ideal monad morphisms. The full subcategory of $\mathsf{FM}_{\mathrm{id}}(\mathscr{A})$ given by iterative monads is denoted by

$$\mathsf{IM}(\mathscr{A}).$$

For the latter category the choice of morphisms is appropriate as demonstrated by the next result:

Proposition 3.14. *An ideal monad morphism between iterative monads preserves solutions.*

Remark. More detailed: let $\mathbb{S}$ and $\overline{\mathbb{S}}$ be iterative monads and let m be an ideal monad morphism from $\mathbb{S}$ to $\overline{\mathbb{S}}$. Then for every ideal equation morphism $e : X \longrightarrow S(X + A)$ the equation morphism

$$X \xrightarrow{\ e\ } S(X+A) \xrightarrow{\ m_{X+A}\ } \overline{S}(X+A)$$

is ideal and the triangle below commutes:

$$\begin{tikzcd} X \arrow[rr, "m_{X+A}\cdot e"] \arrow[dr, "e^\dagger"'] & & \overline{S}A \\ & SA \arrow[ur, "m_A"'] & \end{tikzcd}$$

We omit the proof of Proposition 3.14 since it is identical to the proof of Proposition 5.9 in [M].

Examples 3.15.

(1) The monad $X + 1$ is iterative as seen in Example 3.4: an ideal equation morphism has no ungrounded variables.

(2) The monad $\mathbb{T}$ of binary trees, see Example 3.6, is iterative. Also its submonad $\mathbb{R}$ of rational binary trees, see Example 3.8, is iterative. In fact, whenever the right-hand sides of the equation system (1.1) are rational trees, it is easy to see that the unfolding of every variable x_i also yields a rational tree. However, the ideal monad $\mathbb{F}$ of finite binary trees is not iterative because the unfolding of a variable in an equation system (1.1) is often an infinite tree.

(3) Trees over a signature. We describe a slightly more general example than in (2) above. Let Σ be a signature, i. e., a ranked alphabet of operation symbols with prescribed arities. Let $T_\Sigma X$ denote the Σ-algebra of all Σ-trees over X, i. e., rooted and ordered trees such that inner nodes with $n > 0$ children are labelled by operation symbols of arity n and leaves are labelled by constant symbols or by elements of X. Then T_Σ gives rise to an iterative monad $\mathbb{T}_\Sigma$. Similarly, the subalgebras $R_\Sigma X$ of rational Σ-trees over X, where again a tree is called rational if it has (up to isomorphism) only finitely many subtrees, give rise to an iterative monad $\mathbb{R}_\Sigma$.

(4) More generally, let H be an endofunctor on $\mathscr{A}$ with enough final coalgebras, i. e., there exists a final coalgebra $T_H X$ for each functor $H(-) + X$. Then T_H is the object assignment of an iterative monad $\mathbb{T}_H$ on $\mathscr{A}$, see [AAMV, M].

(5) In [AMV_1] we discussed a categorical generalization of rational trees to our present setting. In fact, we showed that every finitary functor H generates the "rational" monad $\mathbb{R}_H$ of free iterative H-algebras and this monad is characterized as the free iterative monad on H.

(6) Unordered binary trees. Consider the finitary endofunctor H of Set assigning to every set X the set of unordered pairs. Then $\mathbb{T}_H$ is the monad of all unordered binary trees, more precisely, each $T_H X$ consists of binary trees with leaves labelled in X where for each inner node the order of children is not specified. And $\mathbb{R}_H$ is the monad of rational unordered binary trees.

(7) Strongly extensional trees. Take the finite power set functor $H = \mathcal{P}_f$. Then $\mathbb{T}_H$ is the monad of finitely branching strongly extensional trees, i. e., finitely branching trees where subtrees defined by two distinct children of a node are not bisimilar (when considered as $\mathcal{P}_f$-coalgebras in the obvious sense), see [W].

(8) Free-semigroup monad. Here we take the monad $X \mapsto X^+$ assigning to every set X the set of non-empty finite lists (or words) on X. Add an absorbing element $\perp$ (that means that the binary operation of concatenation is extended by $w \cdot \perp = \perp = \perp \cdot w$ for all words w). Then the resulting monad $SX = X^+ + \{ \perp \}$ is iterative, see [AMV_3].

Next we shall need to consider ungrounded variables in our category-theoretic setting. The corresponding notion of an ungrounded subobject of the object of variables of an equation morphism is introduced in Definition 3.19 below. Later we shall see that all equations can be uniquely solved when we solve all ungrounded variables "by force", i. e., in an iterative monad we choose some global element $\perp$ in each object SA as the forced solution. For this we need the following

Definition 3.16. By a *strict* monad is understood a monad $\mathbb{S}$ together with a choice of a global element in $S0$:

$$\perp : 1 \longrightarrow S0.$$

Notation 3.17. We use the same symbol $\perp$ to denote the composite

$$1 \xrightarrow{\perp} S0 \xrightarrow{Sf_A} SA \quad (f_A : 0 \longrightarrow A \text{ unique})$$

for every object A, and again the same symbol for all the composites of the above morphism with any $g : X \longrightarrow 1$. Also, for every algebra $a : SA \longrightarrow A$ the composite $a \cdot \perp : 1 \longrightarrow A$ is denoted by $\perp$.

Remark 3.18.

(1) Observe that homomorphisms of algebras for the monad $\mathbb{S}$ preserve $\perp : 1 \longrightarrow A$. In fact, let h be a homomorphism from the algebra (A, a) to the algebra (B, b). Then we obtain the equation $h \cdot \perp = \perp$ by verifying that the diagram below commutes:

$$\begin{tikzcd}
 & & SA \arrow[r, "a"] \arrow[dd, "Sh"] & A \arrow[dd, "h"] \\
1 \arrow[r, "\perp"] \arrow[urr, bend left, "\perp"] \arrow[drr, bend right, "\perp"'] \arrow[urrr, bend left=60, "\perp"] \arrow[drrr, bend right=60, "\perp"'] & S0 \arrow[ur, "Sf_A"] \arrow[dr, "Sf_B"'] & & \\
 & & SB \arrow[r, "b"'] & B
\end{tikzcd}$$

In fact, the right-hand square commutes because h is a homomorphism of algebras for $\mathbb{S}$, the middle triangle does by the uniqueness of the morphism f_B, and all other inner parts of the diagram commute by the notation we have just introduced.

(2) Strictness of a monad is not a "hard" restriction. In fact, in many cases there is already a canonical candidate for the global element $\perp$. For the monads T_Σ and R_Σ of (rational) Σ-trees the choice of the global element $\perp$ can for example be achieved by choosing some constant symbol from Σ. More generally, the iterative monads $\mathbb{T}_H$ and $\mathbb{R}_H$ obtained from an endofunctor H are strict if H admits some global element $1 \longrightarrow H0$,

which is then inherited by those monads. If H does not admit a global element, one can freely add one and work with $H_\perp = H + C_1$, where C_1 is the constant functor on 1, in lieu of H.

Definition 3.19. (See [ABMV].) Let $e : X \longrightarrow S(X + A)$ be an equation morphism. A subobject $m : M \longrightarrow X$ is called *ungrounded* provided that e has a restriction to an endomorphism e' on M. More explicitly, let $i_0 = \eta_{X+A} \cdot \mathsf{inl} : X \longrightarrow S(X+A)$ denote the second coproduct injection of $S(X+A) = S'(X + A) + X + A$, then the diagram below commutes:

$$\begin{array}{ccc} M & \xrightarrow{\quad m \quad} & X \\ {\scriptstyle e'}\downarrow & & \downarrow{\scriptstyle e} \\ M \xrightarrow[m]{} X & \xrightarrow[i_0]{} & S(X+A) \end{array}$$

Example 3.20. In the case of the monad $\mathbb{T}$ from Example 3.6 an ungrounded subobject is one containing only ungrounded variables.

Construction 3.21. For any equation morphism $e : X \longrightarrow S(X + A)$ we construct the subobjects $i_n : X_n \longrightarrow X_{n-1}$ and the *n-th restriction* e_n of e for $n = 1, 2, 3, \ldots$ by forming the following pullbacks where $X_0 = X$ and $i_0 = \eta_{X+A} \cdot \mathsf{inl} : X_0 \longrightarrow S(X + A)$:

$$\begin{array}{ccccccccc} \cdots & \longrightarrow & X_3 & \xrightarrow{i_3} & X_2 & \xrightarrow{i_2} & X_1 & \xrightarrow{\quad i_1 \quad} & X \\ & & {\scriptstyle e_3}\downarrow & & {\scriptstyle e_2}\downarrow & & {\scriptstyle e_1}\downarrow & & \downarrow{\scriptstyle e} \\ \cdots & \longrightarrow & X_2 & \xrightarrow[i_2]{} & X_1 & \xrightarrow[i_1]{} & X_0 & \xrightarrow[i_0]{} & S(X+A) \end{array}$$

We put $i_n^* = i_1 \cdot i_2 \cdot \cdots \cdot i_n : X_n \longrightarrow X$ and call this the *n-th derived subobject* of e.

Theorem 3.22. (See [ABMV], Lemma 6.4.) *Every equation morphism has a greatest ungrounded subobject, which is the least derived subobject. More precisely, there exists n such that $i_{n+1} : X_{n+1} \longrightarrow X_n$ is an isomorphism. Then this least derived subobject $i_n^* : X_n \longrightarrow X$ is the greatest ungrounded one.*

Remark 3.23. Theorem 3.22 was proved in [ABMV] in the special case of a rational monad generated by an endofunctor, see Example 3.15(5). However, the proof is valid for any ideal monad as stated above.

Example 3.24. Coming back to our running example, the iterative monad $\mathbb{T}$ from Example 3.6, we see that X_1 consists of all variables which are unguarded, i. e., $e(x)$ is a single variable, X_2 consists of all variables x which are unguarded after two steps, i. e., there are equations $x \approx x', x' \approx x''$ etc. And the least derived subobject $i_n^* : X_n \longrightarrow X$ contains precisely all the variables that are not grounded.

Definition 3.25. (See [ABMV].) Let $\mathbb{S}$ be a strict ideal monad. A solution $e^\dagger : X \longrightarrow SA$ of an equation morphism $e : X \longrightarrow S(X + A)$ is called *strict* provided that for any ungrounded subobject $m : M \longrightarrow X$ we have $e^\dagger \cdot m = \perp : M \longrightarrow SA$.

Remark 3.26. By Theorem 3.22 we have that, equivalently, a solution $e^\dagger$ is strict iff for the least derived subobject $i_n^* : X_n \longrightarrow X$ the triangle below commutes:

$$\begin{tikzcd} X \arrow[r, "e^\dagger"] & SA \\ X_n \arrow[u, "i_n^*"] \arrow[ur, "\perp"'] & \end{tikzcd}$$

Theorem 3.27. (See [ABMV].) *Let $\mathbb{S}$ be a strict iterative monad. Then every equation morphism has a unique strict solution.*

We would like to form the category of strict iterative monads. In order to do so we need an appropriate notion of morphism:

Definition 3.28. Let $\mathbb{S}$ and $\overline{\mathbb{S}}$ be strict monads. A monad morphism $m : \mathbb{S} \longrightarrow \overline{\mathbb{S}}$ is called *strict*, if it preserves the chosen global element, i. e., we have

$$\begin{tikzcd} & 1 \arrow[dl, "\perp"'] \arrow[dr, "\perp"] & \\ S0 \arrow[rr, "m_0"'] & & \overline{S}0 \end{tikzcd}$$

Notation 3.29. We denote by

$$\mathsf{IM}_\perp(\mathscr{A})$$

the category of strict iterative monads and strict ideal monad morphisms. This is a (non-full) subcategory of $\mathsf{IM}(\mathscr{A})$.

4. Iteration Monads

In this section we recall the concept of iteration theory of Stephen Bloom and Zoltán Ésik [BÉ]. Then we formulate the concept of iteration monad in the present generality of hyper-extensive locally finitely presentable categories. We also formulate the property of functoriality (called "functorial dagger implication" in [BÉ]) and mention the simplification it brings to the axioms of iteration monads. We are going to use the name *Elgot monads* for iteration monads satisfying that additional property. In the subsequent sections we then verify that every strict iterative monad is an Elgot monad.

Remark 4.1. We have explained the connection between theories and monads in 2.2. This shows that our notion of an iteration monad in Definition 4.2 below is, for $\mathscr{A} = \mathsf{Set}$, exactly the notion of an iteration theory of Bloom and Ésik [BÉ]. While there is a notion of a theory (of a monad) in the present generality we will not work with that notion in the current paper. Instead, we continue to work, equivalently, with finitary monads throughout.

Definition 4.2. ([BÉ]) An *iteration monad* is a pair consisting of a finitary monad $\mathbb{S} = (S, \eta, \mu)$ and a function $(-)^\dagger$ assigning to every equation morphism $e : X \mathrel{-\!\circ\!\!\rightarrow} X + A$ with X and A finitely presentable a solution $e^\dagger : X \mathrel{-\!\circ\!\!\rightarrow} A$ (see Definition 3.2) so that the following axioms hold:

(1) *Parameter Identity*: Given an equation morphism $e : X \mathrel{-\!\circ\!\!\rightarrow} X + A$ and a morphism $h : A \mathrel{-\!\circ\!\!\rightarrow} B$ with B finitely presentable, the triangle

$$\begin{tikzcd} X \arrow[r, "e^\dagger", "\circ" marking] \arrow[dr, "(h\bullet e)^\dagger"', "\circ" marking] & A \arrow[d, "h", "\circ" marking] \\ & B \end{tikzcd} \tag{4.1}$$

commutes (see Notation 3.5).

(2) *Simplified Composition Identity*: Given morphisms

$$f : X \mathrel{-\!\circ\!\!\rightarrow} Y \quad \text{and} \quad g : Y \mathrel{-\!\circ\!\!\rightarrow} X + A$$

with X, Y and A finitely presentable, we form equation morphisms

$$X \xrightarrow[\circ]{g\cdot f} X + A \qquad \text{and} \qquad Y \xrightarrow[\circ]{g} X + A \xrightarrow{f+A} Y + A.$$

Then the triangle

$$\begin{tikzcd} X \arrow[r, "(g\cdot f)^\dagger", "\circ" marking] \arrow[d, "f"', "\circ" marking] & A \\ Y \arrow[ur, "((f+A)\cdot g)^\dagger"', "\circ" marking] & \end{tikzcd} \tag{4.2}$$

commutes.

(3) *Double-Dagger Identity*: Given an equation morphism

$$e : X \mathrel{-\!\circ\!\!\rightarrow} X + X + A$$

then the solution $e^\dagger : X \mathrel{-\!\circ\!\!\rightarrow} X + A$ is also an equation morphism; the codiagonal $\nabla : X + X \longrightarrow X$ yields another equation morphism

$$\nabla \circ e = X \xrightarrow[\circ]{e} X + X + A \xrightarrow[\circ]{\nabla + A} X + A\,.$$

Their solutions are equal:

$$(\nabla \circ e)^\dagger = (e^\dagger)^\dagger. \tag{4.3}$$

(4) *Commutative Identity*: This is a collection of identities indexed by an arbitrary object X with two decompositions

$$X = X^1 + \cdots + X^r = Y + \cdots + Y \quad (k \text{ summands})$$

where X^i and Y are finitely presentable objects, and by morphisms $\rho^1, \ldots, \rho^r : X \longrightarrow X$ in $\mathscr{A}$ satisfying, for the codiagonal $\nabla : X \longrightarrow Y$,

$$\nabla \cdot \rho^i = \nabla \quad \text{for } i = 1, \ldots, r \text{ (in } \mathscr{A}). \tag{4.4}$$

The statement then concerns an arbitrary morphism

$$f : Y \longrightarrow\!\!\!\!\!\circ\;\; X + A.$$

It defines an equation morphism $\nabla \circ f = (\nabla + A) \cdot f : Y \multimap Y + A$ and another equation morphism

$$\widehat{f} : X \multimap X + A$$

defined by components of $X = X^1 + \cdots + X^r$ as follows: the t-th component, $\widehat{f} \cdot \mathsf{in}_t$, is

$$X^t \xrightarrow{\mathsf{in}_t} X = Y + \cdots + Y \xrightarrow{\nabla} Y \overset{f}{\multimap} X + A \xrightarrow{\rho_t + A} X + A.$$

Then the triangle

$$\begin{tikzcd} X \arrow[r, "(\widehat{f})^\dagger"] \arrow[d, "\nabla"'] & A \\ Y \arrow[ur, "(\nabla \circ f)^\dagger"'] & \end{tikzcd} \tag{4.5}$$

commutes.

Remark 4.3. The above definition is the "B Group" of axioms in [BÉ], except that in the commutative identity the above morphism $\widehat{f}$ is denoted by $\nabla \cdot f \parallel (\rho^1, \ldots, \rho^r)$, and in place of ∇ an arbitrary surjective base morphism is used. However, ∇ is sufficient as proved in [És].

Remark 4.4. In applications, the axiom that is often difficult to deal with is the commutative identity. However, there is an easier, "natural" property which is often fulfilled, and the verification of the commutative identity is not needed then. It states that $(-)^\dagger$ is a functor from the category of coalgebras for the functor $S(- + A)$ (i. e., equation morphisms and their homomorphisms, cf. Remark 3.3) to the comma category $\mathscr{A}/SA$. We call it simply "functoriality", in [BÉ] the name "functorial dagger implication" is used. Examples of applications where commutative identity is circumvented by functoriality can be found in [Mo$_2$], [SP] as well as [BÉ].

Definition 4.5. An *Elgot monad* is a functorial iteration monad. This means that for every pair of equation morphisms $e : X \multimap\!\!\!\longrightarrow X + A$ and $f : Y \multimap\!\!\!\longrightarrow Y + A$, given a homomorphism h of equations (cf. Remark 3.3), then the triangle below commutes:

$$\begin{tikzcd}
X \arrow[r, "e^\dagger", "\circ" marking] \arrow[d, "h"'] & A \\
Y \arrow[ur, "f^\dagger"', "\circ" marking] &
\end{tikzcd} \tag{4.6}$$

Remark 4.6. Functoriality implies the commutative identity in Definition 4.2, see Proposition 3.3.5 in [BÉ]. This follows from the fact that $\nabla : X \longrightarrow Y$ is a homomorphism from $\widehat{f}$ to $\nabla \circ f$. Indeed, to see that the square

$$\begin{tikzcd}
X \arrow[r, "\widehat{f}", "\circ" marking] \arrow[d, "\nabla"'] & X + A \arrow[d, "\nabla+A"] \\
Y \arrow[r, "\nabla\circ f"', "\circ" marking] & Y + A
\end{tikzcd}$$

commutes we consider the components of the coproduct $X = X^1 + \cdots X^r$, separately. For every number t the desired square commutes when extended by the coproduct injection $\mathsf{in}_t : X^t \longrightarrow X$:

$$\begin{tikzcd}
X^t \arrow[r, "\mathsf{in}_t"] \arrow[dr, "\mathsf{in}_t"'] & X \arrow[r, "\nabla"] & Y \arrow[r, "f", "\circ" marking] & X + A \arrow[dl, "\rho^t+A"] \arrow[ddl, bend left, "\nabla+A"] \\
& X \arrow[r, "\widehat{f}", "\circ" marking] \arrow[d, "\nabla"'] & X + A \arrow[d, "\nabla+A"] & \\
& Y \arrow[r, "\nabla\circ f", "\circ" marking] \arrow[d, "f"', "\circ" marking] & Y + A & \\
& X + A \arrow[ur, "\nabla+A"'] & &
\end{tikzcd}$$

Indeed, the upper part of the diagram above commutes by the definition of $\widehat{f}$, the right-hand one commutes by (4.4) and the lower part commutes by the definition of $\nabla \circ f$. The outside of the diagram clearly commutes, and, thus, so does the desired inner square.

Example 4.7. We list some examples of Elgot monads on Set.

(1) Partial functions. The monad $\mathbb{S}$ with $S = Id + 1$ from Example 3.4 has as its Kleisli category the category of sets and partial functions. Its dagger operation is defined as in Example 3.4 on grounded variables and on ungrounded variables $e^\dagger$ is undefined.

(2) Multifunctions. Here we take the finite powerset monad $\mathcal{P}_f$ whose algebras are upper semilattices with a least element. Its Kleisli category is the category of sets and one-to-finite multifunctions. To every multifunction $a : X \multimap X$ assign its iteration $a^* = id \cup a \cup (a \cdot a) \cup \cdots$. Then the dagger of $e : X \multimap X + A$ is defined as follows: let $a : X \multimap X$ and $b : X \multimap A$ be the multifunctions with $e = a \cup b$, then $e^\dagger = b \cdot a^*$. Observe that (1) is a special case—thus, the axioms of Elgot monads follow from those for $\mathcal{P}_f$. And this example is a special case of the next one:

(3) Matrix theories. In this example, which is taken from [BÉ], see 9.3.10, we make an exception and work with theories in lieu of finitary monads, but both notions are equivalent, see Subsection 2.2. Let $(C, +, \cdot, 0, 1)$ be an ω-complete semiring, that is, $+$ is extended to a summation $\sum_{i\in\mathbb{N}} a_i$ of countable families which is associative and distributive over $\cdot$, that is (finite) product. The matrix theory Mat_C has as morphisms from n to k all $n \times k$-matrices over C. Product of matrices defines composition. For every square matrix $a : n \multimap n$ define its iteration

$$a^* = \sum_{i\in\mathbb{N}} a^i.$$

Then the dagger of $e : n \multimap n + k$ is defined by $e^\dagger = b \cdot a^*$ for e written in the form of the block matrix $e = [\, a \; b \,]$.

Theorem 4.8. (See [Mo$_2$].) *An Elgot monad is precisely a pair consisting of a finitary monad $\mathbb{S}$ and a function $(-)^\dagger$ assigning to every equation morphism $e : X \multimap X + A$ a solution $e^\dagger : X \longrightarrow A$ satisfying*

(1) *Functoriality (cf.* (4.6)*),*
(2) *Parameter Identity (cf.* (4.1)*),*

and

(3) Bekić (or Pairing) Identity*: Given equation morphisms*

$$e : X \multimap X + Y + A \quad \textit{and} \quad f : Y \multimap X + Y + A \tag{4.7}$$

form equation morphisms

$$[e, f] : X + Y \multimap X + Y + A$$

$$e_R \equiv Y \xrightarrow{f} X + Y + A \xrightarrow{[e^\dagger, Y+A]} Y + A, \tag{4.8}$$

and

$$e_L \equiv X \xrightarrow{e} X + Y + A \xrightarrow{X+[{e_R}^\dagger, A]} X + A. \tag{4.9}$$

Then the Bekić Identity states that

$$[e, f]^\dagger = [{e_L}^\dagger, {e_R}^\dagger] : X + Y \multimap A \tag{4.10}$$

The proof that every iteration theory satisfies the pairing identity can be found in [BÉ]. Conversely, the pairing identity together with functoriality and parameter identity imply the remaining axioms of iteration theories: for $\mathscr{A} = \mathsf{Set}$ this is also implicitly contained in [BÉ], and explicitly this is explained in Section 6 of [Mo_2]. Notice also that the respective result was proved for Set but the proof holds in the present generality.

To form the category of Elgot monads we introduce below the appropriate notion of morphisms of Elgot monads.

Definition 4.9. An *Elgot monad morphism* m from an Elgot monad $\mathbb{S} = (S, \eta, \mu, (-)^\dagger)$ to an Elgot monad $\overline{\mathbb{S}} = (\overline{S}, \overline{\eta}, \overline{\mu}, (-)^\ddagger)$ is a monad morphism m from $\mathbb{S}$ to $\overline{\mathbb{S}}$ preserving $(-)^\dagger$. That means that for every equation morphism $e : X \longrightarrow S(X + A)$ we have that

$$m_A \cdot e^\dagger = (m_{X+A} \cdot e)^\ddagger : X \longrightarrow \overline{S}A.$$

Notation 4.10. We denote by

$$\mathsf{Elgot}(\mathscr{A})$$

the category of Elgot monads and their morphisms.

Remark 4.11. Let us summarize the four different notions of monads with solutions of recursive equations mentioned above.

1. Iterative monads have unique solutions of all ideal equation morphisms. In order to reasonably study the equational properties of the operation of taking the unique solution, one first needs to extend this operation to all equation morphisms.

2. Strict iterative monads, which are just iterative monads $\mathbb{S}$ with a global element $\perp : 1 \longrightarrow S0$, make that extension possible. As stated in Theorem 3.22, in a strict iterative monad each equation morphism has a unique strict solution. The purpose of the remaining two notions of monads with solutions of recursive equations is now to summarize the essential equational properties of the strict solution operation in strict iterative monads.

3. In iteration monads of Bloom and Ésik [BÉ] one adds to a monad an extra structure providing solutions of recursive equations rather than having (unique) solutions as a property. So iteration monads are defined as monads with an operation $(-)^\dagger$ that assigns to every equation morphism a solution, and this operation $(-)^\dagger$ satisfies the Parameter, Simplified Composition, Double-Dagger and Commutative Identities. The Completeness Theorem of Bloom and Ésik then shows that these properties do indeed capture the essential equational properties of strict solutions in strict iterative monads (cf. also Theorem 1.1). Notice that in an iteration monad $e^\dagger$ may not be the unique solution of e; this happens for example in continuous theories where $(-)^\dagger$ is the operation of taking the least solution.

4. Elgot monads are those iteration monads, where the operation $(-)^\dagger$ satisfies in addition to the axioms of iteration monads the property of functoriality.

An equivalent axiom system is given in Theorem 4.8. Every Elgot monad is an iteration monad, but not conversely, see [BÉ]. All natural examples of iteration monads are actually Elgot monads, and it is often more easy to establish the axioms of the latter in concrete cases. So, on the one hand, Elgot monads seem to be "more practical" in this sense than iteration monads. On the other hand, functoriality is an implication, and so an axiomatization containing this property might be considered to be awkward. However, our intended result in Theorem 1.2 implies that functoriality is an equational axiom when considered w. r. t. the category $\mathsf{Fin}[\mathscr{A},\mathscr{A}]$ of endofunctors; this is elaborated in [AMV$_4$].

5. Iterative Monads are Iteration Monads

The category $\mathsf{Elgot}(\mathscr{A})$ of Elgot monads and their morphisms is a full subcategory of the category of iteration monads (with the same morphisms). It is our aim in this paper to prove that the category $\mathsf{IM}_\perp(\mathscr{A})$ of strict iterative monads and strict ideal monad morphisms is a subcategory of $\mathsf{Elgot}(\mathscr{A})$.

In the rest of our paper we assume that $\mathbb{S}$ is a strict iterative monad on a hyper-extensive, locally finitely presentable category. By Theorem 3.27 we then have the function $(-)^\dagger$ assigning to every equation morphism e the unique strict solution $e^\dagger$. We will prove that this results in an Elgot monad: in fact, our main result in the present paper is the following

Theorem 5.1. *The category $\mathsf{IM}_\perp(\mathscr{A})$ is a subcategory of $\mathsf{Elgot}(\mathscr{A})$. More detailed:*

(1) *Every strict iterative monad is an Elgot monad,*
(2) *every strict ideal monad morphism is an Elgot monad morphism.*

The proof of this theorem is presented in the rest of this section and Section 6.

5.1. Strict ideal monad morphisms are Elgot monads morphisms

Lemma 5.2. *Let $\mathbb{S}$ and $\overline{\mathbb{S}}$ be strict iterative monads, and let $m : \mathbb{S} \longrightarrow \overline{\mathbb{S}}$ be a strict ideal monad morphism. Then m preserves strict solutions, i. e., for every equation morphism $e : X \longrightarrow S(X+A)$ the morphism $m_A \cdot e^\dagger : X \longrightarrow \overline{S}A$ is the unique strict solution of $(m_{X+A}\cdot e) : X \longrightarrow \overline{S}(X+A)$.*

Proof. That fact that $m_A \cdot e^\dagger$ is a solution of $m_{X+A}\cdot e$ can be verified as in the proof of Proposition 3.14. We only need to check that $m_A \cdot e^\dagger$ is strict. To this end we compute the desired least derived subobject of $m_{X+A}\cdot e$:

$$\begin{tikzcd}
\cdots \arrow[r, "i_3"] & X_2 \arrow[r, "i_2"] \arrow[dd, "e_2"'] & X_1 \arrow[r, "i_1"] \arrow[d, "e_1"'] & X \arrow[d, "e"] \\
& & X \arrow[r, "i_0=\eta\cdot\mathsf{inl}"] \arrow[d, equal] & S(X+A) \arrow[d, "m_{X+A}"] \\
\cdots \arrow[r, "i_2"'] & X_1 \arrow[r, "i_1"'] & X \arrow[r, "\overline{i_0}=\overline{\eta}\cdot\mathsf{inl}"'] & \overline{S}(X+A)
\end{tikzcd}$$

The lower right-hand square is a pullback by extensivity since for the ideal monad morphism m we have $m_{X+A} = m'_{X+A} + id_{X+A}$, and we see that by forming the upper right-hand pullback, the first derived subobject of $m_{X+A} \cdot e$ and e coincide. Now let $i_n^* : X_n \longrightarrow X$ be the least derived subobject of $m_{X+A} \cdot e$. Then we have a commutative diagram

$$\begin{tikzcd}
X_n \arrow[d, "i_n^*"'] \arrow[dr, "\perp"'] \arrow[drr, "\perp"] & & \\
X \arrow[r, "e^\dagger"'] & SA \arrow[r, "m_A"'] & \overline{S}A.
\end{tikzcd}$$

Indeed, the left-hand triangle commutes since $e^\dagger$ is a strict solution of e, and the right-hand one does since m is strict. This proves that $m_A \cdot e^\dagger$ is a strict solution of $m_{X+A} \cdot e$. $\square$

5.2. Strict Iterative monads are Elgot Monads

We now turn to the proof of part (1) in Theorem 5.1. We show that the operation of taking unique strict solutions of equation morphisms in strict iterative monads satisfies the axioms of Elgot monads by verifying the three properties in Theorem 4.8. Functoriality and Parameter Identity are easy and we prove them now. Bekić identity will be proved in Section 6.

Theorem 5.3. *Every strict iterative monad satisfies functoriality.*

Proof. Given a homomorphism (3.2), it is our task to prove that $f^\dagger \cdot h : X \longrightarrow\!\!\!\!\circ\;\; A$ is a strict solution of e. It is indeed a solution because in the Kleisli category we get a commutative diagram

$$\begin{tikzcd}
X \arrow[r, "h"] \arrow[dd, "e"'] & Y \arrow[r, "f^\dagger"] \arrow[d, "f"'] & A \\
& Y + A \arrow[ur, "{[f^\dagger, A]}"'] & \\
X + A \arrow[ur, "h+A"] \arrow[rr, rounded corners, to path={-- (\tikztotarget |- \tikztostart) -- (\tikztotarget)}, "{[f^\dagger \cdot h, A]}"'] & &
\end{tikzcd}$$

To prove the strictness, we relate the derived subobjects $i_n : X_n \longrightarrow X_{n-1}$ of e

to the derived subobjects $j_n : Y_n \longrightarrow Y_{n-1}$ of f:

$$\begin{tikzcd}
X_2 \arrow[rrrrrr, "e_2"] \arrow[dr, "i_2"] \arrow[ddddd, "h_2"'] & & & & & & X_1 \arrow[dl, "i_1"'] \arrow[ddddd, "h_1"] \\
& X_1 \arrow[rrrr, "e_1"] \arrow[dr, "i_1"] \arrow[ddd, "h_1"'] & & & & X \arrow[dl, "i_0=\eta\cdot\mathsf{inl}"'] \arrow[ddd, "h"] & \\
& & X \arrow[rr, "e"] \arrow[d, "h"'] & & S(X+A) \arrow[d, "S(h+A)"] & & \\
& & Y \arrow[rr, "f"'] & & S(Y+A) & & \\
& Y_1 \arrow[ur, "j_1"] \arrow[rrrr, "f_1"'] & & & & Y \arrow[ul, "j_0=\eta\cdot\mathsf{inl}"'] & \\
Y_2 \arrow[ur, "j_2"] \arrow[rrrrrr, "f_2"'] & & & & & & Y_1 \arrow[ul, "j_1"']
\end{tikzcd}$$

The central square commutes since h is a homomorphism of equations. Below the morphism f we have the derived subobjects of f and above e the derived subobjects of e, computed as pullbacks. The square to the right of the central one is a pullback due to extensivity, see (3.3). Thus, the universal property of the pullback with vertex Y_1 gives us the unique morphism $h_1 : X_1 \longrightarrow Y_1$ such that the square to the left of the central one commutes. Continuing, we obtain $h_2 : X_2 \longrightarrow Y_2$, $h_3 : X_3 \longrightarrow Y_3$, etc. By Theorem 3.22 there is a number n such that $i_n^* : X_n \longrightarrow X$ is the greatest ungrounded subobject of e and $j_n^* : Y_n \longrightarrow Y$ that of f. Since $f^\dagger$ is a strict solution, the diagram

$$\begin{tikzcd}
X_n \arrow[r, "h_n"] \arrow[d, "i_n^*"'] & Y_n \arrow[d, "j_n^*"'] \arrow[dr, "\perp"] & \\
X \arrow[r, "h"'] & Y \arrow[r, "f^\dagger"'] & SA
\end{tikzcd}$$

commutes, thus, from $\perp \cdot h_n = \perp$ we conclude that $f^\dagger \cdot h$ is a strict solution, as required. $\square$

Theorem 5.4. *Every strict iterative monad satisfies the parameter identity.*

Proof. Given an equation morphism $h \bullet e : X \longrightarrow\!\!\!\!\!\!\circ\;\; X + B$ (see Notation 3.5), we are to prove that its strict solution is $h \cdot e^\dagger : X \longrightarrow\!\!\!\!\!\!\circ\;\; B$. It is clear that $h \cdot e^\dagger$ is a solution because we have the commutative diagram

$$\begin{tikzcd}
X \arrow[r, "e^\dagger"] \arrow[d, "e"'] & A \arrow[r, "h"] & B \\
X + A \arrow[ur, "{[e^\dagger, A]}"'] \arrow[d, "X+h"'] & & \\
X + B \arrow[uurr, "{[h \cdot e^\dagger, B]}"'] & &
\end{tikzcd}$$

To prove strictness, we first verify that the derived subobjects of e and $h \bullet e$ coincide. This follows from the following diagram in the category $\mathscr{A}$ in which i_n denote the derived subobjects of e:

$$\begin{tikzcd}
X \arrow[d, "e"'] \arrow[dddd, bend right=60, "h \bullet e"'] & & & X_1 \arrow[lll, "i_1"'] \arrow[d, "e_1"] & X_2 \arrow[l, "i_2"'] \arrow[dddd, "e_2"] & \cdots \arrow[l, "i_3"'] \\
S(X+A) \arrow[d, "S(\eta+h)"'] & X+A \arrow[l, "\eta"'] \arrow[d, "\eta+h"'] & X \arrow[l, "\mathsf{inl}"'] \arrow[d, "\eta"] \arrow[r, equal] & X \arrow[d, equal] & & \\
S(SX+SB) \arrow[d, "S\mathsf{can}"'] & SX+SB \arrow[l, "\eta"'] \arrow[d, "\mathsf{can}"'] & SX \arrow[l, "\mathsf{inl}"'] \arrow[d, phantom, "(*)"] & X \arrow[l, "\eta"'] \arrow[d, equal] & & \\
SS(X+B) \arrow[d, "\mu"'] & S(X+B) \arrow[l, "\eta S"'] & X+B \arrow[l, "\eta"'] \arrow[d, equal] & X \arrow[l, "\mathsf{inl}"'] \arrow[d, equal] & & \\
S(X+B) & & X+B \arrow[ll, "\eta"] & X \arrow[l, "\mathsf{inl}"] & X_1 \arrow[l, "i_1"] & \cdots \arrow[l]
\end{tikzcd}$$

All the squares in this diagram are pullbacks: for the upper and right-hand square this follows from the definition of the derived subobjects, for the lower left-hand square recall (3.3), and all the other squares except $(*)$ are pullbacks of coproduct injections by extensivity. To see that the square $(*)$ is a pullback notice that its top row can be rewritten as in the diagram

$$\begin{tikzcd}
SX + SB \arrow[d, "\mathsf{can}"'] & X + B \arrow[l, "\eta+\eta"'] \arrow[d, equal] & X \arrow[l, "\mathsf{inl}"'] \arrow[d, equal] \\
S(X + B) & X + B \arrow[l, "\eta"] & X \arrow[l, "\mathsf{inl}"]
\end{tikzcd}$$

Here the left-hand part is one of the pullbacks (3.5) in Remark 3.9, and the right-hand part is obviously a pullback.

Now let $i_n^* : X_n \longrightarrow X$ be the greatest ungrounded subobject of $h \bullet e$ (or e). Then we have the commutative diagram

$$\begin{tikzcd}
X_n \arrow[d, "i_n^*"'] \arrow[dr, "\perp"] \arrow[rr, "\perp"] & & {} \arrow[d] \\
X \arrow[r, "e^\dagger"'] & A \arrow[r, "h"'] & B
\end{tikzcd}$$

where the right-hand triangle commutes by Remark 3.18(1) since $\mu_B \cdot Sh$ is a homomorphism of algebras for $\mathbb{S}$. □

6. Iterative Monads Satisfy Bekić Identity

Our aim in the present section is to present the most involved part of the proof of our main result in Theorem 5.1(1):

Theorem 6.1. *Every strict iterative monad $\mathbb{S}$ on a hyper-extensive, locally finitely presentable category satisfies the Bekić identity.*

The rest of this section is devoted to a proof of Theorem 6.1. We thus assume that equation morphisms (4.7) are given. We prove, in a series of auxiliary statements, that the morphism

$$[{e_L}^\dagger, {e_R}^\dagger] : X + Y \longrightarrow\!\!\circ\!\!\longrightarrow A$$

is a strict solution of the equation morphism $[e, f]$.

Notation 6.2.

(1) Put, for short,

$$g \equiv [e, f] : X + Y \longrightarrow\!\!\circ\!\!\longrightarrow X + Y + A.$$

(2) Recall that since $\mathbb{S}$ is ideal (Definition 3.7) we have

$$S = S' + Id$$

with η as the right-hand coproduct injection.

We now proceed in several steps:

(1) We show that $[{e_L}^\dagger, {e_R}^\dagger]$ is a solution of g, see Subsection 6.1.
(2) We compute the derived subobjects $(X+Y)_n$ in a way that makes it possible to relate them to the derived subobjects of e_L and e_R. In particular, we show how to decompose $(X+Y)_n$ into X-components (relevant for e_L) and Y-components (relevant for e_R), see Subsection 6.2.
(3) We analyze the least derived subobject $(X + Y)_k$ of g. In particular we decompose the corresponding isomorphism $i_{k+1} : X_{k+1} \longrightarrow X_k$ to obtain isomorphisms on the X- and Y-components from (2), see Subsection 6.3.

(4) We compute the derived subobjects of e_R in terms of the Y-components from (2), see Subsection 6.4.
(5) We show that the right-hand component of $[{e_L}^\dagger, {e_R}^\dagger]$ satisfies the desired strictness, see Subsection 6.5.
(6) We compute the derived subobjects of e_L in terms of the X-components from (2), and we show that the left-hand component of $[{e_L}^\dagger, {e_R}^\dagger]$ satisfies the desired strictness, see Subsection 6.6.
(7) Finally, we conclude that the Bekić identity holds for strict iterative monads, see Subsection 6.7.

6.1. Relating the solutions

Lemma 6.3. *The morphism $[{e_L}^\dagger, {e_R}^\dagger]$ is a solution of the equation morphism $g = [e, f]$; in other words, the triangle*

$$\begin{tikzcd} X+Y \arrow[r, "{[{e_L}^\dagger,{e_R}^\dagger]}"] \arrow[d, "g"'] & A \\ X+Y+A \arrow[ur, "{[{e_L}^\dagger,{e_R}^\dagger,A]}"'] & \end{tikzcd} \tag{6.1}$$

commutes.

Proof. Consider the components of $X+Y$ separately. For the right-hand component Y of (6.1) we have the diagram

$$\begin{tikzcd} Y \arrow[rr, "{e_R}^\dagger"] \arrow[dr, "e_R"] \arrow[dd, "f"'] & & A \\ & Y+A \arrow[ur, "{[{e_R}^\dagger,A]}"] & \\ X+Y+A \arrow[ur, "{[e^\dagger,Y+A]}"] \arrow[rr, "{[{e_L}^\dagger,{e_R}^\dagger,A]}"'] & & A \arrow[u, equal] \end{tikzcd}$$

The upper triangle commutes since ${e_R}^\dagger$ is a solution of e_R, and the left-hand triangle is the definition of e_R. For the right-hand part the middle and right-hand coproduct components obviously commute, and for left-hand component use Parameter identity (4.1): since $e_L = h \bullet e$ for $h = [{e_R}^\dagger, A]$, we know that

$${e_L}^\dagger = h \cdot e^\dagger = [{e_R}^\dagger, A] \cdot e^\dagger.$$

For the left-hand component X of (6.1) consider the diagram

$$\begin{tikzcd} X \arrow[rr, "{e_L}^\dagger"] \arrow[dr, "e_L"] \arrow[dd, "e"'] & & A \\ & X+A \arrow[ur, "{[{e_L}^\dagger,A]}"] & \\ X+Y+A \arrow[ur, "{X+[{e_R}^\dagger,A]}"] \arrow[rr, "{[{e_L}^\dagger,{e_R}^\dagger,A]}"'] & & A \arrow[u, equal] \end{tikzcd}$$

The upper triangle commutes since ${e_L}^\dagger$ solves e_L, the left-hand one commutes due to the definition of e_L, and all components of the right-hand part trivially commute. $\square$

6.2. The derived subobjects of g

Remark 6.4. All we need to verify at this point in order to complete the proof of Theorem 5.1 is that $[{e_L}^\dagger, {e_R}^\dagger]$ is a strict solution of $g = [e, f]$ (see (4.7)). To this end we need to relate the derived subobjects of g those of e_L and e_R. Unfortunately, the analysis turns out to be rather involved. We start by a concrete example explaining the structure of the derived subobjects of g.

Example 6.5. Let us consider the iterative monad $SX = X + 1$ in Set, see Example 3.4. Given partial maps

$$e : X \multimap X + Y + A$$

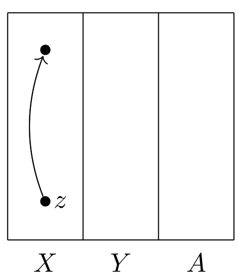
and

$$f : Y \multimap X + Y + A$$

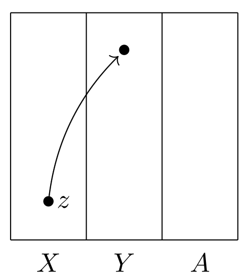
we want to compute the derived subobjects $(X + Y)_i$ of the partial map

$$g = [e, f] : X + Y \multimap X + Y + A.$$

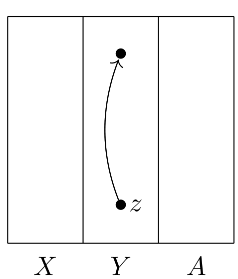
For $i = 1$ we know that $(X + Y)_1$ consists of those variables z (that is, elements of $X + Y$) that are sent to a variable by g. They are of four types (the arrows indicate the action of g):



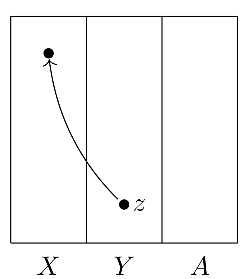
Type X^0



Type X^1

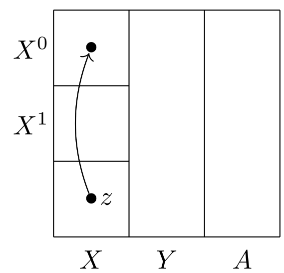


Type Y^0

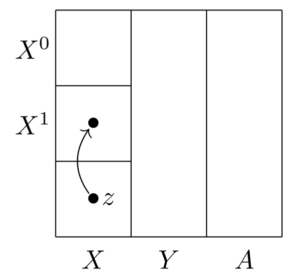


Type Y^1

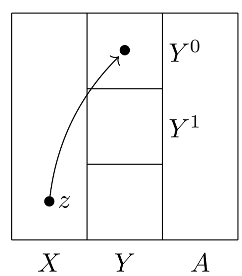
For $i = 2$ the set $(X+Y)_2$ consists of all variables z that g sends to $X^0 + X^1 + Y^0 + Y^1$, where we use X^j and Y^j to denote the subsets of X and Y, respectively containing the four types of variables. The variables z in $(X+Y)_2$ are of eight types: X^{ij} and Y^{ij} for all $i, j = 0, 1$, where X^{0j} are variables of X that e sends to X^j and X^{1j} are those that e sends to Y^j (notice that X^j and Y^j are subsets of X and Y, respectively):



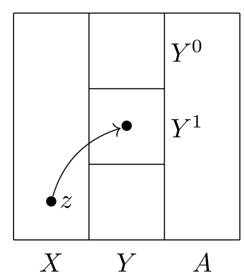
Type X^{00}



Type X^{01}



Type X^{10}



Type X^{11}

Analogously for Y^{ij}: we have the set Y^{0j} of all variables in Y taken by f to X^j and the set Y^{1j} of those taken to Y^j.

In general, the n-th derived subobject of $[e, f]$ is

$$(X+Y)_n = \coprod_{|w|=n} (X^w + Y^w)$$

where the coproduct ranges over binary words w of length n, and for $w = 0v$ we have

$$X^{0v} = \{x \in X \mid e(x) \in X^v\}$$

whereas for $w = 1v$ we have

$$X^{1v} = \{x \in X \mid e(x) \in Y^v\}$$

and analogously with

$$Y^{0v} = \{y \in Y \mid f(y) \in X^v\} \quad \text{and} \quad Y^{1v} = \{y \in Y \mid f(y) \in Y^v\}.$$

So the index w maintains information about the unfolding of the equation of a variable $x \in X^w$ (or Y^w) for as long as this unfolding produces only single variables. For example, for (part of) an equation system

$$x \approx y \qquad y \approx x' \qquad x' \approx x''$$

with $x, x', x'' \in X$ and $y \in Y$ we see that $x \in X^1 \cup X^{10} \cup X^{100}$, $y \in Y^0 \cup Y^{00}$ and $x' \in X^0$.

Remark 6.6. For general iterative monads $\mathbb{S}$ our analysis of the derived subobjects of the equation morphism

$$g = [e, f] : X + Y \longrightarrow S(X + Y + A)$$

proceeds analogously to the example above. First consider the empty word ε and put

$$X^\varepsilon = X, \quad x^\varepsilon = \eta_{X+Y+A} \cdot \mathsf{inl} : X \longrightarrow S(X + Y + A) \quad \text{and} \quad e^\varepsilon = e$$

and analogously

$$Y^\varepsilon = Y, \quad y^\varepsilon = \eta_{X+Y+A} \cdot \mathsf{inm} : Y \longrightarrow S(X + Y + A) \quad \text{and} \quad f^\varepsilon = f,$$

where $\mathsf{inm} : Y \longrightarrow X + Y + A$ denotes the middle coproduct injection. Then use the following four pullbacks to define objects and morphisms with upper index 0 or 1 (straight arrows) from the given morphisms (wavy arrows)

$$\begin{tikzcd}
X^0 \arrow[r, "x^0"] \arrow[d, "e^0"'] & X^\varepsilon \arrow[d, squiggly, "e^\varepsilon"] & X^1 \arrow[l, "x^1"'] \arrow[d, "e^1"] \\
X \arrow[r, squiggly, "x^\varepsilon"] & S(X + Y + A) & Y \arrow[l, squiggly, "y^\varepsilon"'] \\
Y^0 \arrow[r, "y^0"'] \arrow[u, "f^0"] & Y^\varepsilon \arrow[u, squiggly, "f^\varepsilon"'] & Y^1 \arrow[l, "y^1"] \arrow[u, "f^1"']
\end{tikzcd} \tag{6.2}$$

Given these, use the following eight pullbacks to define the objects and the morphisms with upper index ij:

$$\begin{tikzcd}
X^{00} \arrow[r, "x^{00}"] \arrow[d, "e^{00}"'] & X^0 \arrow[d, squiggly, "e^0"] & X^{01} \arrow[l, "x^{01}"'] \arrow[d, "e^{01}"] \\
X^0 \arrow[r, squiggly, "x^0"] & X^\varepsilon & X^1 \arrow[l, squiggly, "x^1"'] \\
Y^{00} \arrow[r, "y^{00}"'] \arrow[u, "f^{00}"] & Y^0 \arrow[u, squiggly, "f^0"'] & Y^{01} \arrow[l, "y^{01}"] \arrow[u, "f^{01}"']
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
X^{10} \arrow[r, "x^{10}"] \arrow[d, "e^{10}"'] & X^1 \arrow[d, squiggly, "e^1"] & X^{11} \arrow[l, "x^{11}"'] \arrow[d, "e^{11}"] \\
Y^0 \arrow[r, squiggly, "y^0"] & Y^\varepsilon & Y^1 \arrow[l, squiggly, "y^1"'] \\
Y^{10} \arrow[r, "y^{10}"'] \arrow[u, "f^{10}"] & Y^1 \arrow[u, squiggly, "f^1"'] & Y^{11} \arrow[l, "y^{11}"] \arrow[u, "f^{11}"']
\end{tikzcd} \tag{6.3}$$

Etc.

Continuing in this way we obtain again the X- and Y-components of $(X + Y)_n$ indexed by binary words w.

We now describe the general case.

Notation 6.7. We define for every binary word w objects X^w and Y^w and morphisms e^w, f^w, x^w and y^w by induction on the length of w.

Length 0*:* $X^\varepsilon = X$ and $Y^\varepsilon = Y$, analogously $e^\varepsilon = e$ and $f^\varepsilon = f$. Finally, we have coproduct injections

$$x^\varepsilon \equiv X \xrightarrow{\mathsf{inl}} X + Y + A \xrightarrow{\eta} S(X + Y + A)$$

and

$$y^\varepsilon \equiv Y \xrightarrow{\mathsf{inm}} X + Y + A \xrightarrow{\eta} S(X + Y + A).$$

Length 1*:* The objects and morphisms with one-letter upper index are defined by the four pullbacks of (6.2).

Length $n+1$*:* Every word w of length $n+1$ ($n \geq 1$) has the form $w = ivj$ where v is a word of length $n-1$. The following eight pullbacks define X^w and Y^w and all the corresponding morphisms:

$$\begin{tikzcd}
X^{0v0} \arrow[r, "x^{0v0}"] \arrow[d, "e^{0v0}"'] & X^{0v} \arrow[d, squiggly, "e^{0v}"] & X^{0v1} \arrow[l, "x^{0v1}"'] \arrow[d, "e^{0v1}"] \\
X^{v0} \arrow[r, squiggly, "x^{v0}"] & X^{v} & X^{v1} \arrow[l, squiggly, "x^{v1}"'] \\
Y^{0v0} \arrow[r, "y^{0v0}"'] \arrow[u, "f^{0v0}"] & Y^{0v} \arrow[u, squiggly, "f^{0v}"'] & Y^{0v1} \arrow[l, "y^{0v1}"] \arrow[u, "f^{0v1}"']
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
X^{1v0} \arrow[r, "x^{1v0}"] \arrow[d, "e^{1v0}"'] & X^{1v} \arrow[d, squiggly, "e^{1v}"] & X^{1v1} \arrow[l, "x^{1v1}"'] \arrow[d, "e^{1v1}"] \\
Y^{v0} \arrow[r, squiggly, "y^{v0}"] & Y^{v} & Y^{v1} \arrow[l, squiggly, "y^{v1}"'] \\
Y^{1v0} \arrow[r, "y^{1v0}"'] \arrow[u, "f^{1v0}"] & Y^{1v} \arrow[u, squiggly, "f^{1v}"'] & Y^{1v1} \arrow[l, "y^{1v1}"] \arrow[u, "f^{1v1}"']
\end{tikzcd}$$

Remark 6.8. By Lemma 2.7, we see that the squares

$$\begin{tikzcd}
X^0 + X^1 \arrow[r, "{[x^0, x^1]}"] \arrow[d, "e^0 + e^1"'] & X \arrow[d, "e"] \\
X + Y \arrow[r, "{i_0 = [x^\varepsilon, y^\varepsilon]}"'] & S(X + Y + A)
\end{tikzcd}
\qquad
\begin{tikzcd}
Y^0 + Y^1 \arrow[r, "{[y^0, y^1]}"] \arrow[d, "f^0 + f^1"'] & Y \arrow[d, "f"] \\
X + Y \arrow[r, "{i_0 = [x^\varepsilon, y^\varepsilon]}"'] & S(X + Y + A)
\end{tikzcd}$$

are pullbacks, and putting these two squares together, another application of Lemma 2.7 yields the pullback

$$\begin{tikzcd}
X^0 + X^1 + Y^0 + Y^1 \arrow[r, "{[x^0, x^1] + [y^0, y^1]}"] \arrow[d, "{[e^0 + e^1, f^0 + f^1]}"'] & X + Y \arrow[d, "{[e, f]}"] \\
X + Y \arrow[r, "i_0"'] & S(X + Y + A)
\end{tikzcd}$$

Consequently, for our equation morphism

$$g = [e, f] : X + Y \longrightarrow S(X + Y + A)$$

the first derived subobject is

$$i_1^* \equiv X^0 + X^1 + Y^0 + Y^1 \xrightarrow{[x^0,x^1]+[y^0,y^1]} X + Y,$$

and the first domain-codomain restriction of g is

$$g_1 = [e^0 + e^1, f^0 + f^1].$$

Notation 6.9.

(1) The length of a binary word w is denoted by $|w|$.
(2) For every nonempty binary word $w = b_1 \cdots b_n$ put

$$x^{w*} \equiv X^w \xrightarrow{x^w} X^{b_1 \ldots b_{n-1}} \xrightarrow{x^{b_1 \ldots b_{n-1}}} \cdots \longrightarrow X^{b_1} \xrightarrow{x^{b_1}} X,$$

and

$$y^{w*} \equiv Y^w \xrightarrow{y^w} Y^{b_1 \ldots b_{n-1}} \xrightarrow{y^{b_1 \ldots b_{n-1}}} \cdots \longrightarrow Y^{b_1} \xrightarrow{y^{b_1}} Y.$$

Lemma 6.10. *For every $n \geq 1$ the n-th derived subobject of $g = [e, f]$ is the morphism*

$$i_n^* \equiv \coprod_{|w|=n} X^w + \coprod_{|w|=n} Y^w \xrightarrow{[x^{w*}]+[y^{w*}]} X + Y$$

with components $[x^{w*}] : \coprod_{|w|=n} X^w \longrightarrow X$ and $[y^{w*}] : \coprod_{|w|=n} Y^w \longrightarrow Y$.

Moreover the corresponding restriction of g

$$g_n : \coprod_{|w|=n} X^w + \coprod_{|w|=n} Y^w \longrightarrow \coprod_{|v|=n-1} X^v + \coprod_{|v|=n-1} Y^v$$

has the components

$$\coprod_{|w|=n} e^w \quad \text{and} \quad \coprod_{|w|=n} f^w.$$

Proof. For $n = 1$ see Remark 6.8. For $n = 2$ we form the pullback

$$\begin{tikzcd}
P_2 \arrow[r, "i_2"] \arrow[d, "g_2"'] & X^0 + X^1 + Y^0 + Y^1 \arrow[d, "{g_1=[e^0+e^1,f^0+f^1]}"] \\
X^0 + X^1 + Y^0 + Y^1 \arrow[r, "{i_1=[x^0,x^1]+[y^0,y^1]}"'] & X + Y
\end{tikzcd}$$

Since the base category is extensive, this pullback is a coproduct of the pullback of $[x^0, x^1]$ along $[e^0, f^0]$ with the pullback of $[y^0, y^1]$ along $[e^1, f^1]$; indeed, reorder the summands in the upper right-hand corner of the square as $X^0+Y^0+X^1+Y^1$ and then apply Proposition 2.6. By applying Notation 6.7 to the case $v = \varepsilon$

we obtain those pullbacks as follows (apply Lemma 2.7 twice to each of the two groups of four pullback squares in (6.3)):

$$\begin{tikzcd}
X^{00}+X^{01}+Y^{00}+Y^{01} \arrow[r, "{[x^{00},x^{01}]+[y^{00},y^{01}]}"] \arrow[d, "{[e^{00}+e^{01},f^{00}+f^{01}]}"'] & X^0+Y^0 \arrow[d, "{[e^0,f^0]}"] \\
X^0+X^1 \arrow[r, "{[x^0,x^1]}"'] & X
\end{tikzcd}$$

and

$$\begin{tikzcd}
X^{10}+X^{11}+Y^{10}+Y^{11} \arrow[r, "{[x^{10},x^{11}]+[y^{10},y^{11}]}"] \arrow[d, "{[e^{10}+e^{11},f^{10}+f^{11}]}"'] & X^1+Y^1 \arrow[d, "{[e^1,f^1]}"] \\
Y^0+Y^1 \arrow[r, "{[y^0,y^1]}"'] & Y
\end{tikzcd}$$

The formation of the coproduct then yields the desired result: we group X^w and Y^w together to obtain

$$P_2 = X^{00}+X^{01}+X^{10}+X^{11}+Y^{00}+Y^{01}+Y^{10}+Y^{11}$$

with

$$g_2 = [e^{00}+e^{01}, e^{10}+e^{11}, f^{00}+f^{01}, f^{10}+f^{11}]$$

and

$$\begin{aligned}
i_2^* &= i_2 \cdot i_1 \\
&= [x^0\cdot x^{00}, x^0\cdot x^{01}] + [x^1\cdot x^{11}, x^1\cdot x^{10}] + [y^0\cdot y^{00}, y^0\cdot y^{01}] \\
&\quad + [y^1\cdot y^{11}, y^1\cdot y^{10}] \\
&= [x^{00*}, x^{01*}] + [x^{11*}, x^{10*}] + [y^{00*}, y^{01*}] + [y^{11*}, y^{10*}].
\end{aligned}$$

Analogously for $n = 3, 4, \ldots$. □

Remark 6.11. The morphisms x^{w*} and y^{w*} are easily seen to be coproduct injections and to form the following pullbacks (obtained by glueing the pullbacks of Notation 6.7)

$$\begin{tikzcd}
X^{0v} \arrow[rr, "x^{0v*}"] \arrow[d, "e^{0v}"'] & & X \arrow[d, "e"] & & X^{1v} \arrow[ll, "x^{1v*}"'] \arrow[d, "e^{1v}"] \\
X^v \arrow[r, "x^{v*}"'] & X \arrow[r, "x^{\varepsilon}"'] & S(X+Y+A) & Y \arrow[l, "y^{\varepsilon}"] & Y^v \arrow[l, "y^{v*}"]
\end{tikzcd}$$

and

$$\begin{tikzcd}
Y^{0v} \arrow[rr, "y^{1v*}"] \arrow[d, "f^{1v}"'] & & Y \arrow[d, "f"] & & Y^{0v} \arrow[ll, "y^{0v*}"'] \arrow[d, "f^{0v}"] \\
Y^v \arrow[r, "y^{v*}"'] & Y \arrow[r, "y^{\varepsilon}"'] & S(X+Y+A) & X \arrow[l, "x^{\varepsilon}"] & X^v \arrow[l, "x^{v*}"]
\end{tikzcd}$$

6.3. The least derived subobject of g

Notation 6.12. Recall from Theorem 3.22 that there exists a natural number k such that $(X+Y)_k$ is the least derived subobject of $g=[e,f]$. More precisely, that means that $i_{k+1}:(X+Y)_{k+1}\longrightarrow(X+Y)_k$ is an isomorphism. We now break this isomorphism and its inverse down to the components X^w and Y^w of $(X+Y)_{k+1}$ and $(X+Y)_k$. Recall that i_{k+1} is a coproduct of the morphisms

$$X^{1w0}+X^{1w1}\xrightarrow{[x^{1w0},x^{1w1}]}X^{1w}\qquad\qquad Y^{1w0}+Y^{1w1}\xrightarrow{[y^{1w0},y^{1w1}]}Y^{1w}$$

$$X^{0w0}+X^{0w1}\xrightarrow{[x^{0w0},x^{0w1}]}X^{0w}\qquad\qquad Y^{0w0}+Y^{0w1}\xrightarrow{[y^{0w0},y^{0w1}]}Y^{0w}$$

ranging over binary words w of length $k-1$. Hence, by extensivity (see Remark 2.5), all these morphisms are isomorphisms, too. The inverses of the above isomorphisms yield an endomorphism on $(X+Y)_k$ which is given componentwise as follows: for every $w\in\{0,1\}^{k-1}$ we have

$$a^{1w}\equiv X^{1w}\xrightarrow{[x^{1w0},x^{1w1}]^{-1}}X^{1w0}+X^{1w1}\xrightarrow{e^{1w0}+e^{1w1}}Y^{w0}+Y^{w1}$$

$$a^{0w}\equiv X^{0w}\xrightarrow{[x^{0w0},x^{0w1}]^{-1}}X^{0w0}+X^{0w1}\xrightarrow{e^{0w0}+e^{0w1}}X^{w0}+X^{w1}$$

$$b^{0w}\equiv Y^{0w}\xrightarrow{[y^{0w0},y^{0w1}]^{-1}}Y^{0w0}+Y^{0w1}\xrightarrow{f^{0w0}+f^{0w1}}X^{w0}+X^{w1}$$

$$b^{1w}\equiv Y^{1w}\xrightarrow{[y^{1w0},y^{1w1}]^{-1}}Y^{1w0}+Y^{1w1}\xrightarrow{f^{1w0}+f^{1w1}}Y^{w0}+Y^{w1}$$

Remark 6.13. Let w be an arbitrary word of length $\geq k-1$.

(1) The triangle

$$\begin{tikzcd} X^{1w} \arrow[d, "a^{1w}"'] \arrow[dr, "e^{1w}"] & \\ Y^{w0}+Y^{w1} \arrow[r, "{[y^{w0},y^{w1}]}"'] & Y^w \end{tikzcd}$$

commutes: just use the definition of a^{1w} and Notation 6.7.

(2) Let $w=b_1\dots b_n$. Using Remark 6.11 and the above triangle we obtain a commutative square

$$\begin{tikzcd} X^{1w} \arrow[rrr, "x^{1w*}"] \arrow[d, "a^{1w}"'] \arrow[dr, "e^{1w}"] & & & X \arrow[d, "e"] \\ Y^{w0}+Y^{w1} \arrow[r, "{[y^{w0},y^{w1}]}"'] & Y^w \arrow[r, "y^{w*}"'] & Y \arrow[r, "y^{\varepsilon}"'] & S(X+Y+A) \end{tikzcd}$$

(3) Analogous commutative squares are obtained for a^{0w}, b^{0w} and b^{1w}.

We have seen that the least derived subobject $(X+Y)_k$ of g is a coproduct with the components X^w and Y^w. Next we shall enlarge k so that the components become as small as possible, and after that no further decomposition of $(X+Y)_k$ into smaller bits is possible:

Lemma 6.14. *There exists a natural number k' such that for every word w of length $\geq k'$ we have $X^w = X^{wi}$ ($i = 0$ or 1) and $Y^w = Y^{wj}$ ($j = 0$ or 1). More precisely:*

$$\textit{either } x^{w0} : X^{w0} \longrightarrow X^w \textit{ or } x^{w1} : X^{w1} \longrightarrow X^w \textit{ is an isomorphism}$$

as well as

$$\textit{either } y^{w0} : Y^{w0} \longrightarrow Y^w \textit{ or } y^{w1} : Y^{w1} \longrightarrow Y^w \textit{ is an isomorphism.}$$

Proof. It is sufficient to prove that there exists k' such that for every word $1w$ of length $\geq k'$ either $x^{1w0} : X^{1w0} \longrightarrow X^{1w}$ or $x^{1w1} : X^{1w1} \longrightarrow X^{1w}$ is an isomorphism. The case with $0w$ of length $\geq k'$ and the two cases involving Y^{0w} and Y^{1w} are proved similarly, and one can choose k' as the maximum of the four constants obtained. Let k be a natural number as in Notation 6.12. Since $(X+Y)_k$ is the least derived subobject of $[e, f]$ we have an isomorphism

$$X^{1w} \cong X^{1w0} + X^{1w1}$$

for all words w of length at least $k-1$. Now let w be a fixed word of length $k-1$ and denote by ℓ the number of components of the finitely presentable object X^{1w}, see Proposition 2.14. We have a diagram of binary coproducts as follows:

$$\begin{tikzcd}
 & & & {} \arrow[dl] & \\
 & & X^{1w00} \arrow[ddl, "x^{1w00}"'] & {} \arrow[l] & \\
 & & & {} \arrow[dl] & \\
 & X^{1w0} \arrow[ddl, "x^{1w0}"'] & X^{1w01} \arrow[l, "x^{1w01}"] & & \\
 & & & {} \arrow[ul] & \\
X^{1w} & & & & \cdots \\
 & & & {} \arrow[dl] & \\
 & X^{1w1} \arrow[uul, "x^{1w1}"] & X^{1w10} \arrow[l, "x^{1w10}"'] & & \\
 & & & {} \arrow[ul] & \\
 & & X^{1w11} \arrow[uul, "x^{1w11}"] & {} \arrow[l] & \\
 & & & {} \arrow[ul] &
\end{tikzcd}$$

After at most $\lceil \log_2 \ell \rceil$ steps in this decomposition no new objects can occur in this diagram. Thus, for words v of length at least $k' = k + \lceil \log_2 \ell \rceil$ we have

$$X^{1v} \cong X^{1v0} + X^{1v1},$$

where one of the two summands is 0. Thus, x^{1v0} or x^{1v1} is an isomorphism. $\square$

Assumption 6.15. Without loss of generality we shall henceforth assume that the constant k from Notation 6.12 has the property of k' from Lemma 6.14.

Remark 6.16. Notice that our choice of k in Assumption 6.15 ensures that every binary word w with $|w| \geq k$ fulfills:

(1) For every binary word $u = d_1 d_2 \dots d_r$ either $X^{wu} \cong 0$ or the composite $x^{wd_1} \cdot x^{wd_1d_2} \dots x^{wu} : X^{wu} \longrightarrow X^w$ is an isomorphism;
(2) There exists a unique infinite binary sequence $p = d_0 d_1 d_2 \dots$ such that $X^w \cong X^{wd_0} \cong X^{wd_0d_1} \cong \dots$

Similar results hold for Y^w.

Example 6.17. Let us illustrate what is going on with the objects X^w and Y^w by a concrete example. Consider the following system of equations:

$$\begin{array}{lclclclclcl}
x_0 & \approx & y_0 & \quad & x_3 & \approx & y_1 & \quad & y_0 & \approx & x_2 \\
x_1 & \approx & y_0 & & x_4 & \approx & y_1 & & y_1 & \approx & x_4 \\
x_2 & \approx & x_1 & & x_5 & \approx & y_2 & & y_2 & \approx & t
\end{array}$$

where t is a parameter and all x_i, y_j are variables. As an illustration consider the following graph:

$$\begin{tikzcd}
x_0 \arrow[r] & y_0 \arrow[ddl] & x_3 \arrow[r] & y_1 \arrow[dl, shift left] & x_5 \arrow[r] & y_2 \arrow[d] \\
x_1 \arrow[ur] \arrow[d, leftrightarrow] & & x_4 \arrow[ur, shift left] & & & t \\
x_2 & & & & &
\end{tikzcd}$$

We compute the sets X^w and Y^w leaving out those which are empty. For $|w| = 0$ we have

$$X = X^\varepsilon = \{ x_0, x_1, x_2, x_3, x_4, x_5 \} \qquad \text{and} \qquad Y = Y^\varepsilon = \{ y_0, y_1, y_2 \}$$

The next step yields

$$\begin{array}{lcl} X^0 & = & \{ x_2 \} \\ X^1 & = & \{ x_0, x_1, x_3, x_4, x_5 \} \end{array} \qquad \text{and} \qquad Y^0 = \{ y_0, y_1 \}$$

For $|w| = 2$ we obtain

$$\begin{array}{lcl} X^{01} & = & \{ x_2 \} \\ X^{01} & = & \{ x_0, x_1, x_3, x_4 \} \end{array} \qquad \text{and} \qquad \begin{array}{lcl} Y^{00} & = & \{ y_0 \} \\ Y^{01} & = & \{ y_1 \} \end{array}$$

At this point we have computed all ungrounded variables, i. e., all non-empty components of $(X+Y)_k$ from Notation 6.12. To obtain the isomorphism from Lemma 6.14 we have to continue one more step and obtain:

$$\begin{array}{rcl} X^{010}=X^{01} & = & \{\,x_2\,\} \\ X^{100} & = & \{\,x_0,x_1\,\} \\ X^{101} & = & \{\,x_3,x_4\,\} \end{array} \qquad \text{and} \qquad \begin{array}{rcl} Y^{001}=Y^{00} & = & \{\,y_0\,\} \\ Y^{010}=Y^{01} & = & \{\,y_1\,\} \end{array}$$

These sets now fulfill the desired two properties of Remark 6.16. For example we have

$$X^{100}=X^{1001}=X^{10010}=X^{100100}=\cdots$$

and

$$Y^{010}=Y^{0101}=Y^{01010}=Y^{010101}=\cdots$$

Corollary 6.18. *For every word w with $|w|\geq k$ the morphism a^{1w} of Notation 6.12 has the following form: either*

$$a^{1w}\equiv X^{1w}\longrightarrow Y^{w0}\xrightarrow{\mathsf{inl}}Y^{w0}+Y^{w1},$$

or

$$a^{1w}\equiv X^{1w}\longrightarrow Y^{w1}\xrightarrow{\mathsf{inr}}Y^{w0}+Y^{w1}.$$

Similarly for a^{0w}, b^{0w} and b^{1w}.

Indeed, either $Y^{w1}\cong 0$ or $Y^{w0}\cong 0$ by Remark 6.16.

Remark 6.19. Let w be a word of length $\geq k$.

(1) We slightly abuse the notation by denoting the restrictions of a^{1w} by $a^{1w}: X^{1w}\longrightarrow Y^{w0}$ or $a^{1w}: X^{1w}\longrightarrow Y^{w1}$, again. This will not lead to confusion since we will only need to deal with these restrictions.

(2) We will need the restricted forms of the commutative diagrams of Remark 6.13, and we will now list those in the two cases arising for a^{1w} (the diagrams for a^{0w}, b^{0w}, and b^{1w} are completely analogous):

$$\begin{tikzcd} X^{1w} \arrow[d, "e^{1w}"'] \arrow[dr, "a^{1w}"] & \\ Y^{w} & Y^{w0} \arrow[l, "y^{w0}"] \end{tikzcd} \qquad \text{or} \qquad \begin{tikzcd} X^{1w} \arrow[d, "e^{1w}"'] \arrow[dr, "a^{1w}"] & \\ Y^{w} & Y^{w1} \arrow[l, "y^{w1}"] \end{tikzcd}$$

and

$$\begin{tikzcd} X^{1w} \arrow[r, "x^{1w*}"] \arrow[d, "a^{1w}"'] & X \arrow[d, "e"] \\ Y^{w0} \arrow[r, "y^{\varepsilon}\cdot y^{w*}\cdot y^{w0}"'] & S(X+Y+A) \end{tikzcd} \qquad \text{or} \qquad \begin{tikzcd} X^{1w} \arrow[r, "x^{1w*}"] \arrow[d, "a^{1w}"'] & X \arrow[d, "e"] \\ Y^{w1} \arrow[r, "y^{\varepsilon}\cdot y^{w*}\cdot y^{w1}"'] & S(X+Y+A) \end{tikzcd}$$

From now on we shall assume that the horizontal arrows are understood to be the composites of coproduct injections of the form x^{w} and y^{w}, and we shall not label these arrows anymore.

Example 6.20. We continue Example 6.17 above. The morphisms a^w and b^w are the domain-codomain restrictions of the equation morphism $[e, f]$ given by the equation system to the sets of variables X^w and Y^w with $|w| = 3$. For example, we have the chain of maps

$$X^{100} \xrightarrow{a^{100}} Y^{001} \xrightarrow{b^{001}} X^{010} \xrightarrow{a^{010}} X^{100} \xrightarrow{a^{010}} Y^{001} \xrightarrow{b^{001}} \cdots \tag{6.4}$$

Notice that this chain is periodic.

In general, when talking about a chain like the above one we do not want to distinguish between X and Y on the one hand, and a and b on the other hand. We now introduce the appropriate notation.

Notation 6.21. Let w be a word of length $\geq k$.

(1) The above morphisms a^{iw} and b^{iw} have domains and codomains with upper indices of the same length, in fact, the domain index is iw and the codomain one is wj for $j = 0, 1$. Let us introduce a "variable" object C to mean

$$C = X \quad \text{or} \quad C = Y$$

so that C^w stands for X^w or Y^w. Analogously, let us introduce a "variable" morphism c^w to mean

$$c^w = a^w \quad \text{or} \quad c^w = b^w.$$

Then we see that for every binary word w_0 of length $\geq k$ and every choice $C = X$ or Y we obtain, by Corollary 6.18 a unique infinite chain of morphisms

$$C^{w_0} \xrightarrow{c^{w_0}} C^{w_1} \xrightarrow{c^{w_1}} C^{w_2} \xrightarrow{c^{w_2}} \cdots \tag{6.5}$$

where all the words w_0, w_1, w_2, ... are of the same length. Necessarily, since there are only finitely many such morphisms c^{w_i} $(i = 0, 1, 2, \dots)$, we obtain after j steps a cycle of length q

$$\begin{tikzcd}
C^{w_0} \arrow[r, "c^{w_0}"] & C^{w_1} \arrow[r, "c^{w_1}"] & \cdots \arrow[r, "c^{w_{j-2}}"] & C^{w_{j-1}} \arrow[r, "c^{w_{j-1}}"] & C^{w_j} \arrow[dr, "c^{w_j}"] & \\
 & & & C^{w_{j+q}} \arrow[ur, "c^{w_{j+q}}"] & & C^{w_{j+1}} \arrow[dl, "c^{w_{j+1}}"] \\
 & & & & \cdots \arrow[ul, "c^{w_{j+q-1}}"] &
\end{tikzcd}$$

for some $j = 0, 1, 2, \dots$ and $q = 1, 2, 3, \dots$

(2) Denote for every $i = 0, 1, 2, \dots$ by d_i the first bit of the word w_i above. Then the given word w_0 is a prefix of the infinite word $d_0 d_1 d_2 \dots$, and we know that this infinite word is eventually periodic (after j steps with period length q), thus, we can write it as

$$d_0 d_1 d_2 \cdots = u\pi \quad u \text{ finite, } \pi \text{ periodic.}$$

(3) For every word w of length $\geq k$ let $d_0 d_1 d_2 \cdots = u\pi$ be the above infinite word for $w = w_0$ and $C^{w_0} = X^w$. We put

$$X^{u\pi} \stackrel{\text{def}}{=} X^w$$

Analogously, if $w = w_0$ and $C^{w_0} = Y^w$ we put

$$Y^{u\pi} \stackrel{\text{def}}{=} Y^w.$$

Remark 6.22. We obtain versions of the diagrams from Remark 6.19 where infinite words appear as superscripts. For example:

$$\begin{tikzcd}
X^{1u\pi} \arrow[r, "x^{1u\pi *}"] \arrow[d, "a^{1u\pi}"'] & X \arrow[d, "e"] \\
Y^{u\pi} \arrow[r] & S(X+Y+A)
\end{tikzcd}$$

Again, the horizontal morphisms are understood to be composites of injections x^v and y^v, respectively.

Analogously for f:

$$\begin{tikzcd}
Y^{1u\pi} \arrow[r, "y^{1u\pi *}"] \arrow[d, "b^{1u\pi}"'] & Y \arrow[d, "f"] \\
Y^{u\pi} \arrow[r] & S(X+Y+A)
\end{tikzcd}$$

Example 6.23. We continue Example 6.20 above. Here we have $X^{\overline{100}} \stackrel{\text{def}}{=} X^{100}$, i. e., $w = 100$, $u = \varepsilon$ and $\pi = \overline{100}$ is a periodic sequence. Similarly, we have $Y^{\overline{001}} \stackrel{\text{def}}{=} Y^{001}$, and the morphisms in (6.4) are equipped with infinite periodic superscripts, too, e. g. $a^{\overline{100}} : X^{\overline{100}} \longrightarrow Y^{\overline{001}}$.

6.4. The derived subobjects of e_R

We are ready to compute the derived subobjects of $e_R : Y \longrightarrow S(Y + A)$, see (4.8). Since this equation morphism is formed with the help of $e^\dagger : X \longrightarrow S(Y + A)$, the next lemma is the first step.

Lemma 6.24. *There exists a constant $\ell \geq k$ such that for the morphisms*

$$p : X^1 + X^{01} + X^{001} + \cdots + X^{0^{\ell-1}1} \xrightarrow{[x^1, x^{01*}, x^{001*}, \ldots, x^{0^{\ell-1}1*}]} X$$

and

$$q : X^1 + X^{01} + X^{001} + \cdots + X^{0^{\ell-1}1} \xrightarrow{[e^1, e^1 \cdot e^{01}, e^1 \cdot e^{01} \cdot e^{001}, \ldots, e^1 \cdot e^{01} \cdot \ldots \cdot e^{0^{\ell-1}1}]} Y$$

the square below is a pullback:

$$\begin{tikzcd}
\coprod_{i=0}^{\ell-1} X^{0^i 1} \arrow[rr, "q"] \arrow[dd, "p"'] & & Y \arrow[d, "\mathsf{inl}"] \\
& & Y+A \arrow[d, "\eta"] \\
X \arrow[rr, "e^\dagger"'] & & S(Y+A)
\end{tikzcd} \tag{6.6}$$

Proof. Let us denote for purposes of the proof the pullback of $e^\dagger$ and $\eta \cdot \mathsf{inl}$ by

$$\begin{tikzcd}
P^0 \arrow[r, "q^0"] \arrow[d, "p^0"'] & Y \arrow[d, "\eta_{Y+A} \cdot \mathsf{inl}"] \\
X \arrow[r, "e^\dagger"'] & S(Y+A)
\end{tikzcd} \tag{6.7}$$

(1) The object P^0 of (6.7) is finitely presentable: since $\eta_{Y+A} \cdot \mathsf{inl}$ is a coproduct injection, so is p^0 by extensivity and we can apply Lemma 2.8.

(2) Define coproduct injections $p^i : P^i \longrightarrow X$ for $i = 0, 1, 2, \dots$ by induction starting with p^0 above and the induction step given by the pullback

$$\begin{tikzcd}
P^{i+1} \arrow[rr, "q^{i+1}"] \arrow[dd, "p^{i+1}"'] & & P^i \arrow[d, "p^i"] \\
& & X \arrow[d, "x^\varepsilon"] \\
X \arrow[rr, "e"'] & & S(X+Y+A)
\end{tikzcd} \tag{6.8}$$

Consequently, we obtain the following pullback

$$\begin{tikzcd}
P^{i+1} + \coprod_{j=1}^{i} X^{0^j 1} + X^1 \arrow[rr, "q^{i+1} + \coprod_{j=1}^{i} e^{0^j 1} + e^1"] \arrow[dd, "{[p^{i+1}, [x^{0^j 1 *}], x^1]}"] & & P^i + \coprod_{j=0}^{i-1} X^{0^j 1} + Y \arrow[d, "{[p^i, [x^{0^j 1 *}]] + \mathsf{inl}}"'] \\
& & X+Y+A \arrow[d, "\eta"] \\
X \arrow[rr, "e"'] & & S(X+Y+A)
\end{tikzcd} \tag{6.9}$$

Indeed, this follows (by applying Lemma 2.7) from the left-hand component being the preceding pullback, whereas the other components are pullbacks by definition of $X^{0^j 1}$.

(3) For every i put

$$\widehat{q}^i = q^0 \cdot q^1 \cdot \ldots \cdot q^i : P^i \longrightarrow Y$$

and

$$\widehat{e}^i = e^1 \cdot e^{01} \cdot \ldots \cdot e^{0^n 1} : X^{0^n 1} \longrightarrow X.$$

We are going to prove by induction on $i = 0, 1, 2, \ldots$ that the square

$$\begin{tikzcd}
P^i + X^{0^{i-1}1} + X^{0^{i-2}1} + \cdots + X^1 \arrow[r, "{[\widehat{q}^i, \widehat{e}^{i-1}, \widehat{e}^{i-2}, \ldots, \widehat{e}^0]}"] \arrow[d, "{[p^i, x^{0^{i-1}1}, x^{0^{i-2}1}, \ldots, x^1]}"'] & Y \arrow[d, "\eta \cdot \mathsf{inl}"] \\
X \arrow[r, "e^\dagger"'] & S(Y+A)
\end{tikzcd}$$

is a pullback. For $i = 0$ this is Diagram (6.7). For the induction step we use that $e^\dagger = \mu_{Y+A} \cdot S[e^\dagger, \eta_{Y+A}] \cdot e$, see Diagram (3.1), so that the desired pullback is a composite of the three pullbacks in the diagram below:

$$\begin{tikzcd}
P^{i+1} + \coprod_{j=1}^{i} X^{0^j 1} + X^1 \arrow[r, "q^{i+1} + \coprod_{j=1}^{i} e^{0^j 1} + e^1"] \arrow[ddd, "{[p^{i+1}, [x^{0^j 1 *}], x^1]}"] & P^i + \coprod_{j=0}^{i-1} X^{0^j 1} + Y \arrow[r, "{[\widehat{q}^i, [\widehat{e}^j], Y]}"] \arrow[dd, "{[p^i, [x^{0^j 1 *}]] + \mathsf{inl}}"] & Y \arrow[r, equal] \arrow[d, "\mathsf{inl}"] & Y \arrow[d, "\mathsf{inl}"] \\
 & & Y+A \arrow[r, equal] \arrow[d, "\eta"] & Y+A \arrow[dd, "\eta"] \\
 & X+Y+A \arrow[r, "{[e^\dagger, \eta_{Y+A}]}"] \arrow[d, "\eta"] & S(Y+A) \arrow[d, "\eta S"] & \\
X \arrow[r, "e"'] \arrow[rrr, bend right, "e^\dagger"'] & S(X+Y+A) \arrow[r, "{S[e^\dagger, \eta_{Y+A}]}"'] & SS(Y+A) \arrow[r, "\mu"'] & S(Y+A)
\end{tikzcd}$$

The right-hand square and the lower middle one are pullbacks by Remark 3.9. The upper middle square is a pullback by Lemma 2.7 and the induction hypothesis (one computes the pullbacks of $\eta_{Y+A} \cdot \mathsf{inl}$ along $e^\dagger$ and along η_{Y+A} separately to obtain the sum in the upper left-hand corner). Finally, see Diagram (6.9) for the remaining left-hand pullback.

(4) We derive from (3) that

$$\begin{aligned}
P^0 &= P^1 + X^1 \\
&= P^2 + X^{01} + X^1 \\
&= P^3 + X^{001} + X^{01} + X^1 \\
&\ \vdots
\end{aligned}$$

Since P^0 is finitely presentable, by Proposition 2.14, there exists some number ℓ such that $P^{\ell+1}$ and P^ℓ are isomorphic. Then by diagram (6.8) we see that

P^ℓ is an ungrounded subobject of X (cf. Definition 3.19). Since $e^\dagger$ is a strict solution, we have a commutative square

$$\begin{tikzcd} P^\ell \arrow[r] \arrow[d, "p^\ell"'] & 1 \arrow[d, "\perp"] \\ X \arrow[r, "e^\dagger"'] & S(Y+A) \end{tikzcd}$$

We also have a commutative diagram

$$\begin{tikzcd} P^\ell \arrow[r] \arrow[dr, "p^\ell"'] & P^0 \arrow[r, "q^0"] \arrow[d, "p^0"] & Y \arrow[d, "\eta\cdot\mathsf{inl}"] \\ & X \arrow[r, "e^\dagger"'] & S(Y+A) \end{tikzcd}$$

Indeed, use diagram (6.7) and the fact that P^ℓ is a coproduct component of P^0. Now since $\perp : 1 \longrightarrow S(Y+A)$ factors through $\sigma_{Y+A} : S'(Y+A) \longrightarrow S(Y+A)$, the pullback of $\perp$ and $\eta_{Y+A} \cdot \mathsf{inl}$ is the initial object by extensivity. Thus, we obtain a morphism $P^\ell \longrightarrow 0$, which implies that P^ℓ is the initial object, too, see Remark 2.13(2). Hence, we have

$$P^0 = X^{0^{\ell-1}1} + \cdots + X^{01} + X^1$$

as well as

$$p^0 = p.$$

The proof of $q^0 = q$ follows from the fact that $\eta \cdot \mathsf{inl}$ is a monomorphism and (6.6) and (6.7) both commute. □

Assumption 6.25. Without any loss of generality we shall assume that our chosen number k, see Assumption 6.15, has the property of ℓ in Lemma 6.24.

Proposition 6.26. *The n-th derived subobject of the equation morphism*

$$e_R : Y \longrightarrow S(Y+A)$$

(see (4.8)*) is the subobject $y_n : Y_n \longrightarrow Y$ where*

$$Y_n = \coprod_{i_1=0}^{k} \coprod_{i_2=0}^{k} \cdots \coprod_{i_n=0}^{k} Y^{0^{i_1}10^{i_2}1\ldots 0^{i_n}1} \tag{6.10}$$

with $y_n : Y_n \longrightarrow Y$ having components $y^{0^{i_1}10^{i_2}1\ldots 0^{i_n}1*}$, see Notation 6.7.

Proof. We will prove the cases $n = 1$ and $n = 2$ in detail and leave the obvious continuation for $n = 3, 4, \ldots$ to the reader.

(1) The first derived subobject of e_R is

$$y_1 = [y^1, y^0 \cdot y^{01}, y^0 \cdot y^{00} \cdot y^{001}, \dots] : Y_1 = \coprod_{i=0}^{k} Y^{0^i 1} \longrightarrow Y.$$

To prove this we will first verify that the square

$$\begin{tikzcd}
\coprod_{i=0}^{k-1} X^{0^i 1} + Y \arrow[rr, "p+\mathsf{inl}"] \arrow[d, "{[q,Y]}"'] & & X + Y + A \arrow[d, "{[e^\dagger,\eta]}"] \\
Y \arrow[r, "\mathsf{inl}"'] & Y + A \arrow[r, "\eta"'] & S(Y + A)
\end{tikzcd} \tag{6.11}$$

is a pullback; in fact, due to Lemma 2.7, it is sufficient to consider pullbacks of $\eta \cdot \mathsf{inl}$ along $e^\dagger$ and along η separately. The first one is presented in Lemma 6.24, the latter one is trivial.
The first derived subobject Y_1 of e_R is given by the following pullback (glued from smaller pullbacks):

$$\begin{tikzcd}
\coprod_{i=0}^{k} Y^{0^i 1} = Y^1 + \coprod_{i=1}^{k} Y^{0^i 1} \arrow[rrr, "y_1 \overset{\mathrm{def}}{=} {[y^1, y^0 \cdot y^{01}, y^0 \cdot y^{00} \cdot y^{001}, \dots]}"] \arrow[d, "\coprod_{i=0}^{k} f^{0^i 1} = f^1 + \coprod_{i=1}^{k} f^{0^i 1} \quad (*)"] & & & Y \arrow[d, "f"'] \arrow[ddd, bend left, "e_R"] \\
Y + \coprod_{i=0}^{k-1} X^{0^i 1} = \coprod_{i=0}^{k-1} X^{0^i 1} + Y \arrow[rr, "{[\mathsf{inm}, \mathsf{inl} \cdot p] = p + \mathsf{inl}}"'] \arrow[d, "{[q,Y]}"'] & & X+Y+A \arrow[r, "\eta"] \arrow[d, "{[e^\dagger,\eta]}"'] & S(X+Y+A) \arrow[d, "{S[e^\dagger,\eta]}"'] \\
Y \arrow[r, "\mathsf{inl}"] \arrow[d, equal] & Y+A \arrow[r, "\eta"] \arrow[d, equal] & S(Y+A) \arrow[r, "\eta S"] & SS(Y+A) \arrow[d, "\mu"'] \\
Y \arrow[r, "\mathsf{inl}"'] & Y+A \arrow[rr, "\eta"'] & & S(Y+A)
\end{tikzcd}$$

The only square that needs some work is the upper one; to see that the middle and lower squares are pullbacks apply Lemma 2.7, Remark 3.9 and use (6.11).
We prove that $(*)$ is a pullback by considering the components of $\coprod_{i=0}^{k} Y^{0^i 1}$ separately (using Lemma 2.7). The left-hand component is the definition of Y^1 from Notation 6.7. Also all the other components arising from $\coprod X^{0^i 1}$ are compositions of pullbacks from Notation 6.7: for $i = 0$ we have the pullback square

$$\begin{tikzcd}
Y^{01} \arrow[r] \arrow[d, "f^{01}"'] & Y^0 \arrow[rr] \arrow[d, "f^0"] & & Y \arrow[d, "f"] \\
X^1 \arrow[r, "x^1"'] & X \arrow[r, "\mathsf{inl}"'] & X + Y + A \arrow[r, "\eta"'] & S(X + Y + A)
\end{tikzcd}$$

for $i = 1$ we have

$$\begin{tikzcd}
Y^{001} \arrow[r, "y^{001}"] \arrow[d, "f^{001}"'] & Y^{00} \arrow[r, "y^{00}"] \arrow[d, "f^{01}"] & Y^{0} \arrow[rr, "y^{0}"] \arrow[d, "f^{0}"] & & Y \arrow[d, "f"] \\
X^{01} \arrow[r, "x^{01}"'] & X^{0} \arrow[r, "x^{0}"'] & X \arrow[r, "\mathsf{inl}"'] & X+Y+A \arrow[r, "\eta"'] & S(X+Y+A)
\end{tikzcd}$$

etc. This concludes the proof that $(*)$ is a pullback. Thus, $y_1 : Y_1 \longrightarrow Y$ has the required form.

(2) The second derived subobject of e_R is $y_2^* = y_1 \cdot y_2 : Y_2 \longrightarrow Y_1 \longrightarrow Y$ where

$$y_2 : \coprod_{i=0}^{k} \coprod_{t=0}^{k} Y^{0^t 10^i 1} \longrightarrow \coprod_{t=0}^{k} Y^{0^t 1}$$

is the morphism

$$\left[\coprod_{t=0}^{k} y^{0^t 11}, \coprod_{t=0}^{k} y^{0^t 10} \cdot y^{0^t 101}, \coprod_{t=0}^{k} y^{0^t 10} \cdot y^{0^t 100} \cdot y^{0^t 1001}, \dots \right].$$

We prove this by a series of auxiliary statements.

(2a) We compute a pullback of $y^1 : Y^1 \longrightarrow Y$ along q. From Notation 6.7 we have the following pullbacks

$$\begin{tikzcd}
\vdots & \vdots \\
X^{0011} \arrow[r, "x^{0011}"] \arrow[d, "e^{0011}"'] & X^{001} \arrow[d, "e^{001}"] \\
X^{011} \arrow[r, "x^{011}"] \arrow[d, "e^{011}"'] & X^{01} \arrow[d, "e^{01}"] \\
X^{11} \arrow[r, "x^{11}"] \arrow[d, "e^{11}"'] & X^{1} \arrow[d, "e^{1}"] \\
Y^{1} \arrow[r, "y^{1}"'] & Y = Y^{\varepsilon}
\end{tikzcd}$$

Thus, from Lemma 2.7 we see that the pullback of $y^1 : Y^1 \longrightarrow Y$ along q is

$$\begin{tikzcd}
\coprod_{t=0}^{k-1} X^{0^t 11} \arrow[r, "\coprod_{t=0}^{k-1} x^{0^t 11}"] \arrow[d, "{[e^{11}, e^{11} \cdot e^{011}, e^{11} \cdot e^{011} \cdot e^{0011}, \dots]}"'] & \coprod_{t=0}^{k-1} X^{0^t 1} \arrow[d, "q"] \\
Y^{1} \arrow[r, "y^{1}"'] & Y
\end{tikzcd}$$

(2b) Next we compute a pullback of $y^{01*} = y^0 \cdot y^{01} : Y^{01} \longrightarrow Y$ along q. We use the following diagram of pullbacks

$$
\begin{tikzcd}
\vdots & \vdots & \vdots \\
X^{00101} \arrow[r, "x^{00101}"] \arrow[d, "e^{00101}"'] & X^{0010} \arrow[r, "x^{0010}"] \arrow[d, "e^{0010}"] & X^{001} \arrow[d, "e^{001}"] \\
X^{0101} \arrow[r, "x^{0101}"] \arrow[d, "e^{0101}"'] & X^{010} \arrow[r, "x^{010}"] \arrow[d, "e^{010}"] & X^{01} \arrow[d, "e^{01}"] \\
X^{101} \arrow[r, "x^{101}"] \arrow[d, "e^{101}"'] & X^{10} \arrow[r, "x^{10}"] \arrow[d, "e^{10}"] & X^{1} \arrow[d, "e^{1}"] \\
Y^{01} \arrow[r, "y^{01}"'] & Y^{0} \arrow[r, "y^{0}"'] & Y = Y^{\varepsilon}
\end{tikzcd}
$$

Thus, the pullback of y^{01*} along q is (by another application of Lemma 2.7) the following square

$$
\begin{tikzcd}
\coprod_{t=0}^{k-1} X^{0^t 101} \arrow[rr, "\coprod_{t=0}^{k-1} x^{0^t 10} \cdot x^{0^t 101}"] \arrow[d, "{[e^{101}, e^{101} \cdot e^{0101}, \dots]}"'] & & \coprod_{t=0}^{k-1} X^{0^t 1} \arrow[d, "{q = [e^1, e^1 \cdot e^{01}, \dots]}"] \\
Y^{01} \arrow[r, "y^{01}"'] & Y^{1} \arrow[r, "y^{1}"'] & Y
\end{tikzcd}
$$

(2c) By continuing in the obvious way we obtain the following pullback:

$$
\begin{tikzcd}
\coprod_{i=0}^{k} \coprod_{t=0}^{k-1} X^{0^t 1 0^i 1} \arrow[r, "s"] \arrow[d, "{\coprod_{i=0}^{k} [e^{10^i 1}, e^{10^i 1} \cdot e^{010^i 1}, \dots]}"'] & \coprod_{t=0}^{k-1} X^{0^t 1} \arrow[d, "q"] \\
Y_1 = \coprod_{i=0}^{k} Y^{0^i 1} \arrow[r, "{y_1 = [y^1, y^{01*}, \dots]}"'] & Y
\end{tikzcd}
$$

where s has as components the obvious composites of the morphisms of type x^w:

$$
s = \left[\coprod_{t=0}^{k} x^{0^t 11}, \coprod_{t=0}^{k} x^{0^t 10} \cdot x^{0^t 101}, \coprod_{t=0}^{k} x^{0^t 10} \cdot x^{0^t 100} \cdot x^{0^t 1001}, \dots \right] \tag{6.12}
$$

(2d) Next we compute a pullback of $y_1 = [y^1, y^{01*}, \dots]$ along $f^1 : Y^1 \longrightarrow Y$. We proceed again component-wise, using the pullbacks of Notation 6.7

$$
\begin{tikzcd}
Y^{11} \arrow[r, "y^{11}"] \arrow[d, "f^{11}"'] & Y^{1} \arrow[d, "f^{1}"] \\
Y^{1} \arrow[r, "y^{1}"'] & Y
\end{tikzcd}
\quad \text{and} \quad
\begin{tikzcd}
Y^{101} \arrow[r, "y^{101}"] \arrow[d, "f^{101}"'] & Y^{10} \arrow[r, "y^{10}"] \arrow[d, "f^{10}"] & Y^{1} \arrow[d, "f^{1}"] \\
Y^{01} \arrow[r, "y^{01}"'] & Y^{0} \arrow[r, "y^{0}"'] & Y
\end{tikzcd}
\quad \text{etc.}
$$

The desired pullback is

$$\begin{tikzcd}
\coprod_{i=0}^k Y^{10^i1} \arrow[r, "{[y^{11}, y^{10}\cdot y^{101}, \ldots]}"] \arrow[d, "\coprod_{i=0}^k f^{10^i1}"'] & Y^1 \arrow[d, "f^1"] \\
Y_1 = \coprod_{i=0}^k Y^{0^i1} \arrow[r, "{y_1 = [y^1, y^{01*}, \ldots]}"'] & Y
\end{tikzcd}$$

(2e) Next we compute a pullback of the morphism

$$s : \coprod_{i=0}^{k} \coprod_{t=0}^{k-1} X^{0^t10^i1} \longrightarrow \coprod_{t=0}^{k-1} X^{0^t1}$$

from (6.12) along the morphism

$$\coprod_{t=1}^{k} f^{0^t1} : \coprod_{t=1}^{k} Y^{0^t1} \longrightarrow \coprod_{t=0}^{k-1} X^{0^t1}$$

We consider again the i-components of s separately.
The case $i = 0$ is given by the following coproduct of pullbacks (which, in an extensive category, is a pullback, see Proposition 2.6):

$$\begin{tikzcd}
\coprod_{t=1}^k Y^{0^t11} \arrow[r, "\coprod_{t=1}^k y^{0^t11}"] \arrow[d, "\coprod_{t=1}^k f^{0^t11}"'] & \coprod_{t=1}^k Y^{0^t1} \arrow[d, "\coprod_{t=1}^k f^{0^t1}"] \\
\coprod_{t=0}^{k-1} X^{0^t11} \arrow[r, "\coprod_{t=0}^{k-1} x^{0^t11}"'] & \coprod_{t=0}^{k-1} X^{0^t1}
\end{tikzcd}$$

The lower arrow is the 0-th component of s.
For $i = 1$ we get the composite of (coproducts of) pullbacks below

$$\begin{tikzcd}
\coprod_{t=1}^k Y^{0^t101} \arrow[r, "\coprod_{t=1}^k y^{0^t101}"] \arrow[d, "\coprod_{t=1}^k f^{0^t101}"'] & \coprod_{t=1}^k Y^{0^t10} \arrow[r, "\coprod_{t=1}^k y^{0^t10}"] \arrow[d, "\coprod_{t=1}^k f^{0^t10}"] & \coprod_{t=1}^k Y^{0^t1} \arrow[d, "\coprod_{t=1}^k f^{0^t1}"] \\
\coprod_{t=0}^{k-1} X^{0^t101} \arrow[r, "\coprod_{t=0}^{k-1} x^{0^t101}"'] & \coprod_{t=0}^{k-1} X^{0^t10} \arrow[r, "\coprod_{t=0}^{k-1} x^{0^t10}"'] & \coprod_{t=0}^{k-1} X^{0^t1}
\end{tikzcd}$$

The lower arrow is the first component of s. And so on.
Combining all these pullbacks we obtain the desired pullback

$$\begin{tikzcd}
\coprod_{i=0}^k \coprod_{t=1}^k Y^{0^t10^i1} \arrow[r, "r"] \arrow[d, "\coprod_{i=0}^k \coprod_{t=1}^k f^{0^t10^i1}"'] & \coprod_{t=1}^k Y^{0^t1} \arrow[d, "\coprod_{t=1}^k f^{0^t1}"] \\
\coprod_{i=0}^k \coprod_{t=0}^{k-1} X^{0^t10^i1} \arrow[r, "s"'] & \coprod_{t=0}^{k-1} X^{0^t1}
\end{tikzcd}$$

where r is the following morphism

$$r = \left[\coprod_{t=1}^{k} y^{0^t 11}, \coprod_{t=1}^{k} y^{0^t 10} \cdot y^{0^t 101}, \coprod_{t=1}^{k} y^{0^t 10} \cdot y^{0^t 100} \cdot y^{0^t 1001}, \dots \right]$$

(2f) We are ready to compute Y_2. We know from (1) that the first restriction of e_R is

$$\textstyle\coprod_{t=0}^{k} Y^{0^t 1} \xrightarrow{\coprod_{t=0}^{k} f^{0^t 1}} Y + \coprod_{t=0}^{k-1} X^{0^t 1} \xrightarrow{[Y,q]} Y$$

To compute the second restriction we consider the diagram below:

$$\begin{tikzcd}
Y_2 = \coprod_{i=0}^{k} \coprod_{t=0}^{k} Y^{0^t 10^i 1} \arrow[r, "y_2"] \arrow[d, equal] & \coprod_{t=0}^{k} Y^{0^t 1} = Y_1 \arrow[d, equal] \\
\coprod_{i=0}^{k} Y^{10^i 1} + \coprod_{i=0}^{k} \coprod_{t=1}^{k} Y^{0^t 10^i 1} \arrow[r, "{[y^{11}, y^{10} \cdot y^{101}, \dots] + r}"] \arrow[d, "\coprod_{i=0}^{k} f^{10^i 1} + \coprod_{i=0}^{k} \coprod_{t=1}^{k} f^{0^t 10^i 1}"] & Y^1 + \coprod_{t=1}^{k} Y^{0^t 1} \arrow[d, "\coprod_{t=0}^{k} f^{0^t 1} = f^1 + \coprod_{t=1}^{k} f^{0^t 1}"'] \\
\coprod_{i=0}^{k} Y^{0^i 1} + \coprod_{i=0}^{k} \coprod_{t=0}^{k-1} X^{0^t 10^i 1} \arrow[r, "y_1 + s"] \arrow[d, equal] & Y + \coprod_{t=0}^{k-1} X^{0^t 1} \arrow[d, equal] \\
\coprod_{i=0}^{k} \coprod_{t=0}^{k-1} X^{0^t 10^i 1} + \coprod_{i=0}^{k} Y^{0^i 1} \arrow[r, "s + y_1"] \arrow[d, "{\left[[e^{101}, e^{101} \cdot e^{0101}, \dots], id\right]}"'] & \coprod_{t=0}^{k-1} X^{0^t 1} + Y \arrow[d, "{[q,Y]}"] \\
Y_1 = \coprod_{i=0}^{k} Y^{0^i 1} \arrow[r, "{y_1 = [y^1, y^0 \cdot y^{01}, y^0 \cdot y^{00} \cdot y^{001}, \dots]}"'] & Y
\end{tikzcd}$$

In the lower square we combine the pullback from (2c) with the trivial one of id_Y along y_1. The middle square is the coproduct of the two pullback squares from (2d) and (2e), and in the upper square we just reordered the coproduct. So we see that the topmost morphism is

$$y_2 = \left[\coprod_{t=0}^{k} y^{0^t 11}, \coprod_{t=0}^{k} y^{0^t 10} \cdot y^{0^t 101}, \coprod_{t=0}^{k} y^{0^t 10} \cdot y^{0^t 100} \cdot y^{0^t 1001}, \dots \right],$$

and this proves the desired formula for Y_2 and y_2.

As promised we leave the continuation for $n = 3, 4, \dots$ to the reader. $\square$

Remark 6.27. We know from Theorem 3.22 that there exists n such that Y_n in Proposition 6.26 is the least derived subobject of e_R. Without loss of generality the index k of Assumption 6.15 fulfills

$$n \geq k$$

Consequently, since every w with $|w| = k$ appears as a (prefix of a) summand in (6.10), the least derived subobject Y_n of e_R satisfies

$$Y_n = \coprod_{|w|=k} Y^w \quad \text{with } Y^w = Y^{u\pi}$$

see Notation 6.21(3). All these summands arise from the least derived subobject $(X+Y)_k$ of $[e, f]$.

6.5. The morphism ${e_R}^\dagger$ is "strict"

We now prove that the right-hand component of $[{e_L}^\dagger, {e_R}^\dagger] : X + Y \longrightarrow SA$ satisfies the "strictness" condition that all composites with y^{w*} are $\perp$ provided w has length $\geq k$.

Notation 6.28. For every finite word w we denote by $\bar{w}$ the infinite word obtained from repeatedly concatenating w:

$$\bar{w} = www\ldots$$

In particular, $\bar{0} = 000\ldots$

Remark 6.29. Let us turn to the equation morphism

$$e : X \longrightarrow S(X + Y + A)$$

with the object $Y + A$ of parameters. Its derived subobjects are:

$$\begin{tikzcd}
\cdots & X^{000} \arrow[r, "x^{000}"] \arrow[d, "e^{000}"'] & X^{00} \arrow[r, "x^{00}"] \arrow[d, "e^{00}"] & X^0 \arrow[r, "x^0"] \arrow[d, "e^0"] & X \arrow[d, "e"] \\
\cdots & X^{00} \arrow[r, "x^{00}"'] & X^0 \arrow[r, "x^0"'] & X \arrow[r, "\eta\cdot\mathsf{inl}"'] & S(X+Y+A)
\end{tikzcd}$$

After finitely many steps we obtain the least derived subobject, see Theorem 3.22. Without loss of generality we can assume that for the above constant k the least derived subobject of e is X^{0^k}. The associated chain (6.5) has, in this special case, the form

$$X^{0^k} \xrightarrow{a^{0^k}} X^{0^k} \xrightarrow{a^{0^k}} X^{0^k} \xrightarrow{a^{0^k}} \cdots$$

and so we can write $X^{\bar{0}} = X^{0^k}$ by Notation 6.21(3). Thus, we have a commutative triangle

$$\begin{tikzcd}
X^{\bar{0}} \arrow[d, "x^0\cdot x^{00}\cdot\ldots\cdot x^{0^k}"'] \arrow[dr, "\perp"] & \\
X \arrow[r, "e^\dagger"'] & S(Y+A)
\end{tikzcd}$$

since $e^\dagger$ is a strict solution of e.

Proposition 6.30. *The solution $e_R{}^\dagger$ is strict in the sense that the triangle*

$$\begin{tikzcd}
Y^w \arrow[d, "y^{w*}"'] \arrow[dr, "\perp"] & \\
Y \arrow[r, "e_R{}^\dagger"'] & SA
\end{tikzcd} \tag{6.13}$$

commutes for all binary words w of length at least k.

Proof. Recall from Notation 6.21(3) that $Y^w = Y^{u\pi}$ (u finite and π infinite, periodic).

(1) Assume $\pi = \bar{0}$. We proceed by induction on the length of u.

(1a) For $u = \varepsilon$ we have $Y^w = Y^{\bar{0}}$. Since $e_R{}^\dagger = \mu_A \cdot S[e_R{}^\dagger, \eta_A] \cdot e_R$ and $\mu_A \cdot S[e_R{}^\dagger, \eta_A] \cdot \perp = \perp$ it is sufficient to verify the commutativity of the following diagram

$$\begin{tikzcd}
Y^{\bar{0}} \arrow[r, "b^{\bar{0}}"] \arrow[dd, "y^{\bar{0}*}"'] & X^{\bar{0}} \arrow[d, "x^{\bar{0}*}"'] \arrow[dr, "\perp"] & & \\
& X \arrow[r, "e^\dagger"] \arrow[d, "\eta\cdot\mathsf{inl}"] & S(Y+A) \arrow[d, "\eta S"] \arrow[dr, equal] & \\
Y \arrow[r, "f"'] \arrow[rrr, bend right, "e_R"'] & S(X+Y+A) \arrow[r, "{S[e^\dagger,\eta]}"'] & SS(Y+A) \arrow[r, "\mu"'] & S(Y+A)
\end{tikzcd}$$

The only inner part that needs explanation is the left-hand square. For $Y^{\bar{0}}$, which abbreviates Y^{0^ℓ} where $\ell \geq k$, the associated chain (6.5) is

$$Y^{0^\ell} \xrightarrow{b^{0^\ell}} X^{0^\ell} \xrightarrow{a^{0^\ell}} X^{0^\ell} \xrightarrow{a^{0^\ell}} \cdots$$

We conclude (by Notation 6.21(3)) that $b^{\bar{0}}$ has the domain $Y^{\bar{0}}$ and the commutativity of the left-hand square above now follows from Remark 6.22.

(1b) Induction step: we distinguish $u = 0u'$ and $u = 1v$.

(1b1) $u = 0u'$ and u' contains 1 at least once. Consequently, there is a finite word v with

$$u\pi = 0^{i+1}1v\bar{0}.$$

Let us choose a word v' of length at least k with $a^{1v\bar{0}} = a^{1v'}$ so that v is a prefix of v'. Then the diagram below commutes:

$$
\begin{tikzcd}
X^{0^i1v\bar{0}} = X^{0^i1v'} \arrow[r] \arrow[d, "a^{0^i1v\bar{0}}=a^{0^i1v'}"'] & X^{0^i1} \arrow[r] \arrow[d, "e^{0^i1}"] & X \arrow[dddddd, "e^{\dagger}"] \\
X^{0^{i-1}1v\bar{0}} = X^{0^{i-1}1v'} \arrow[r] \arrow[d, "a^{0^{i-1}1v\bar{0}}=a^{0^{i-1}1v'}"'] & X^{0^{i-1}1} \arrow[d, "e^{0^{i-1}1}"] & \\
X^{0^{i-2}1v\bar{0}} = X^{0^{i-2}1v'} \arrow[r] \arrow[d, "a^{0^{i-2}1v\bar{0}}=a^{0^{i-2}1v'}"'] & X^{0^{i-2}1} \arrow[d, "e^{0^{i-2}1}"] & \\
\vdots & \vdots & \\
{} \arrow[d] & {} \arrow[d] & \\
X^{1v\bar{0}} = X^{1v'} \arrow[r] \arrow[d, "a^{1v\bar{0}}=a^{1v'}"'] & X^{1} \arrow[d, "e^{1}"] & \\
Y^{v\bar{0}} = Y^{v'0} \arrow[r, "y^{v'0*}"'] & Y \arrow[r, "\eta\cdot\mathsf{inl}"'] & S(Y+A)
\end{tikzcd}
\tag{6.14}
$$

The right-hand square commutes by Lemma 6.24 (in fact, consider the i-th component of diagram (6.6) for i as occuring in the decomposition of $u\pi$) and the squares on the left-hand side are composites of the triangles and the squares from Remark 6.19. Since v' is chosen such that $|v'| \geq k$, the arrows on the left-hand edge are a part of the chain (6.5). This proves that (6.13) commutes due to the following commutative diagram:

$$
\begin{tikzcd}
Y^{00^i1v\bar{0}} \arrow[rrr, "y^{00^i1v\bar{0}*}"] \arrow[d, "b^{00^i1v\bar{0}}"] & & & Y \arrow[rr, "{e_R}^\dagger"] \arrow[d, "f"] & & SA \arrow[dddd, equal] \\
X^{0^i1v\bar{0}} \arrow[rr, "x^{0^i1v\bar{0}*}"] \arrow[d] & & X \arrow[r, "x^\varepsilon=\eta\cdot\mathsf{inl}"] \arrow[d, "e^\dagger"] & S(X+Y+A) \arrow[d, "{S[e^\dagger,\eta]}"] & & \\
Y^{v\bar{0}} \arrow[r, "y^{v\bar{0}*}"] \arrow[dr, "y^{v\bar{0}*}"'] \arrow[ddrrrrr, rounded corners, "\perp"'] & Y \arrow[r, "\eta\cdot\mathsf{inl}"'] \arrow[d, equal] & S(Y+A) \arrow[r, "\eta S"'] \arrow[dr, equal] & SS(Y+A) \arrow[d, "\mu"] & & \\
 & Y \arrow[rr, "\eta\cdot\mathsf{inl}"] \arrow[drrrr, "{e_R}^\dagger"'] & & S(Y+A) \arrow[r, "{S[{e_R}^\dagger,\eta]}"] & SSA \arrow[uuur, "\mu"'] & \\
 & & & & & SA \arrow[ul, "\eta S"']
\end{tikzcd}
$$

The upper right-hand part commutes since ${e_R}^\dagger$ solves e_R, the upper left-hand part does by Remark 6.22, and the left-hand square below it commutes by (6.14). The lowest part commutes by the induction hypothesis since v can be chosen as a proper postfix of u. All the other parts clearly commute.

(1b2) $u = 1v$. This case is proved by the following diagram:

$$
\begin{tikzcd}
Y^{1v\bar{0}} \arrow[rr, "y^{1v\bar{0}*}"] \arrow[d, "b^{1v\bar{0}}"'] & & Y \arrow[rr, "{e_R}^\dagger"] \arrow[d, "f"] & & SA \arrow[dddd, equal] \\
Y^{v\bar{0}} \arrow[r, "y^{v\bar{0}*}"] \arrow[dddd, "y^{v\bar{0}*}"'] & Y \arrow[r, "y^\varepsilon=\eta\cdot\mathsf{inm}"] \arrow[d, "\eta\cdot\mathsf{inl}"] & S(X+Y+A) \arrow[d, "{S[e^\dagger,\eta]}"] & & \\
 & S(Y+A) \arrow[r, "\eta S"'] \arrow[dr, equal] & SS(Y+A) \arrow[d, "\mu"] & & \\
 & & S(Y+A) \arrow[r, "{S[{e_R}^\dagger,\eta]}"] & SSA \arrow[uuur, "\mu"'] & \\
Y \arrow[urr, "\eta\cdot\mathsf{inl}"] \arrow[rrrr, "{e_R}^\dagger"'] & & & & SA \arrow[ul, "\eta S"']
\end{tikzcd}
$$

The middle parts clearly commute. The upper right-hand square commutes by the definition of solutions, the upper left-hand square does by Remark 6.22. By induction hypothesis the lower passage from $Y^{v\bar{0}}$ to

SA is $\perp$. Since all the other parts of the diagram commute, this proves that (6.13) commutes.

(2) Assume $\pi \neq \bar{0}$. Then $\pi = \bar{v}$ where v is a finite binary word of the form $v = 0^i 1 v'$. We will show that $Y^{u\pi}$ is a summand of the least derived subobject Y_n, see Remark 6.27. Let ℓ be the number of "bits" 1 in u. Without loss of generality we may assume that the index n fulfills $\ell \geq n \geq k$ (in fact, simply choose u to be long enough to contain at least n "bits" 1—this is possible because $\bar{v}$ contains infinitely many such "bits"). Then we have

$$Y^{u\pi} = Y^{0^{i_1}1 \ldots 0^{i_\ell}10^{i_{\ell+1}}\bar{v}}.$$

Since $\ell \geq n \geq k$, we know that $Y^{u\pi}$ is isomorphic to each Y^w where w is a finite prefix of $u\pi$ of length at least k. So in particular we have

$$Y^{u\pi} \cong Y^{0^{i_1}1 \ldots 0^{i_n}1}$$

which is one of the summands of Y_n, see Proposition 6.26. Since ${e_R}^\dagger$ is a strict solution of e_R, this proves that (6.13) commutes.

□

6.6. The derived subobjects of e_L

To complete our proof that $[{e_L}^\dagger, {e_R}^\dagger] : X + Y \longrightarrow SA$ is a strict solution of $[e, f]$, we still need to show that the left-hand component ${e_L}^\dagger$ satisfies the desired strictness condition: all composites with $x^{w*} : X^w \longrightarrow X$ are $\perp$ provided that w has length $\geq k$. This is the purpose of the present subsection. We analyse the derived subobjects of the equation morphism $e_L : X \longrightarrow S(X + A)$ and relate them to the components X^w of the least derived subobject $(X + Y)_k$ of $[e, f]$.

Proposition 6.31. *The derived subobjects of $e_L : X \longrightarrow S(X + A)$ are*

$$x^{0^i*} : X^{0^i} \longrightarrow X \quad (i = 1, 2, 3, \ldots) \tag{6.15}$$

and the solution ${e_L}^\dagger$ is strict in the sense that the triangle

$$\begin{tikzcd} X^w \arrow[d, "x^{w*}"'] \arrow[dr, "\perp"] & \\ X \arrow[r, "{e_L}^\dagger"'] & SA \end{tikzcd} \tag{6.16}$$

commutes for all binary words w of length at least k.

Proof. 1. The derived subobjects can be seen from the following diagram, where the pullback of $[{e_R}^\dagger, \eta_A] : Y + A \longrightarrow SA$ along $\eta_A : A \longrightarrow SA$ is denoted by P:

$$
\begin{tikzcd}
\cdots \arrow[r, "x^{000}"] & X^{00} \arrow[r, "x^{00}"] \arrow[dddd, "e^{00}"'] & X^0 \arrow[rrr, "x^0"] \arrow[d, "e^0"] & & & X \arrow[d, "e"] \arrow[dddd, bend left, "e_L"] \\
& & X \arrow[r, "\mathsf{inl}"'] \arrow[rr, bend left, "\mathsf{inl}"] \arrow[d, equal] & X+P \arrow[r, "X+q"'] \arrow[d, "X+p"'] & X+Y+A \arrow[r, "\eta"] \arrow[d, "\eta+[{e_R}^\dagger,\eta]"'] & S(X+Y+A) \arrow[d, "S(\eta+[{e_R}^\dagger,\eta])"'] \\
& & X \arrow[r, "\mathsf{inl}"] \arrow[d, equal] & X+A \arrow[r, "\eta+\eta"] \arrow[d, equal] & SX+SA \arrow[r, "\eta"] \arrow[d, "\mathsf{can}"] & S(SX+SA) \arrow[d, "S\mathsf{can}"] \\
& & X \arrow[r, "\mathsf{inl}"] \arrow[d, equal] & X+A \arrow[r, "\eta"] \arrow[d, equal] & S(X+A) \arrow[r, "\eta S"] & SS(X+A) \arrow[d, "\mu"] \\
\cdots \arrow[r, "x^0"'] & X^0 \arrow[r, "x^0"'] & X \arrow[r, "\mathsf{inl}"'] & X+A \arrow[rr, "\eta"'] & & S(X+A)
\end{tikzcd}
$$

Notice that all squares in the above diagram are pullbacks: for the upper square see Notation 6.7 and for the left-hand ones see Remark 6.6 and Notation 6.7. For the remaining squares use Remark 3.9, Proposition 2.6 and Lemma 2.7.

2. Let us prove ${e_L}^\dagger \cdot x^{w*} = \bot$. Recall from Notation 6.21(3) that every object X^w with $|w| \geq k$ can be written as $X^{u\pi}$ for a finite binary word u and an infinite periodic one π.

(i) Let $u\pi = \bar{0}$. Then X^w is a least derived subobject of e_L, and we use the fact that ${e_L}^\dagger$ is a strict solution of e_L.

(ii) Let $u\pi = 0^i 1v$. We prove the desired statement by induction on i.

(iia) For $i = 0$ consider the diagram below:

$$
\begin{tikzcd}
X^{1v} \arrow[r, "a^{1v}"] \arrow[ddd, "x^{1v*}"'] & Y^s \arrow[d, "y^{s*}"'] \arrow[dr, "\bot"] & & & \\
& Y \arrow[r, "{e_R}^\dagger"'] \arrow[dd, "\eta\cdot\mathsf{inm}"] & SA \arrow[d, "\eta S"'] \arrow[dr, equal] & & \\
& & SSA \arrow[r, "\mu"] \arrow[d, "S\mathsf{inr}"'] \arrow[dr, "SS\mathsf{inr}"] & SA \arrow[dr, "S\mathsf{inr}"] & \\
X \arrow[r, "e"'] \arrow[rrrr, bend right, "e_L"'] & S(X+Y+A) \arrow[r, "S(\eta+[{e_R}^\dagger,\eta])"'] & S(SX+SA) \arrow[r, "S\mathsf{can}"'] & SS(X+A) \arrow[r, "\mu"'] & S(X+A)
\end{tikzcd}
$$

The left-hand square commutes (either for $s = v0$ or $s = v1$) by Notation 6.21(3) and Remark 6.19, the upper triangle does by Proposition 6.30, and all other parts clearly commute. The commutativity of the desired triangle now follows easily since $\mu_A \cdot S[{e_L}^\dagger, \eta_A]$, which is a homomorphism of algebras for the monad

$\mathbb{S}$ by (3.2), preserves $\perp$ (see Remark 3.18):

$$
\begin{tikzcd}
X^{1v} \arrow[r, "x^{1v*}"] \arrow[dr, "\perp"'] & X \arrow[r, "{e_L}^{\dagger}"] \arrow[d, "e_L"'] & SA \\
& S(X+A) \arrow[r, "{S[{e_L}^{\dagger},\eta]}"'] & SSA \arrow[u, "\mu"']
\end{tikzcd}
$$

(iib) For the induction step consider the following diagram

$$
\begin{tikzcd}
X^{0^{i+1}1v} \arrow[rrr, "x^{0^{i+1}1v*}"] \arrow[d, "a^{0^{i+1}1v}"'] & & & X \arrow[rr, "{e_L}^{\dagger}"] \arrow[d, "e"] & & SA \arrow[ddddd, equal] \\
X^{0^{i}1v} \arrow[r, "x^{0^{i}1v*}"] \arrow[rrrrr, rounded corners, to path={|- (\tikztotarget)}, "\perp"'] \arrow[ddd, "x^{0^{i}1v*}"'] & X \arrow[rr, "x^{\varepsilon}=\eta\cdot\mathsf{inl}"] \arrow[d, "\eta"] \arrow[ddd, bend right, "id"'] & & S(X+Y+A) \arrow[d, "{S(\eta+[{e_R}^{\dagger},\eta])}"] & & \\
& SX \arrow[r, "\eta S"] \arrow[dr, equal] & SSX \arrow[r, "S\mathsf{inl}"] \arrow[d, "\mu"] \arrow[dr, "SS\mathsf{inl}"] & S(SX+SA) \arrow[d, "S\mathsf{can}"] & & \\
& & SX \arrow[dr, "S\mathsf{inl}"] & SS(X+A) \arrow[d, "\mu"] & & \\
& X \arrow[rr, "\eta\cdot\mathsf{inl}"] \arrow[drrrr, "{e_L}^{\dagger}"] & & S(X+A) \arrow[r, "{S[{e_L}^{\dagger},\eta]}"] & SSA \arrow[uuuur, "\mu"] & \\
& & & & & SA \arrow[ul, "\eta S"]
\end{tikzcd}
$$

The upper right-hand part commutes by the definition of solutions, the upper left-hand square commutes due to Notation 6.21(3) and Remark 6.19, the lowest part commutes by the induction hypothesis, and all other parts clearly commute. □

6.7. Bekić Identity for Strict Iterative Monads

Theorem 6.32 (Bekić identity)**.** *The unique strict solution of the equation morphism $g = [e, f]$ is $[{e_L}^{\dagger}, {e_R}^{\dagger}]$.*

Proof. We know from Lemma 6.3 that this is a solution. It remains to prove the strictness. In Notation 6.12 we have chosen k so that

$$i_k^* : \coprod_{|w|=k} X^w + \coprod_{|w|=k} Y^w \longrightarrow X + Y$$

is the least derived subobject of g, and we know from Lemma 6.10 that the components of i_k^* are x^{w*} and y^{w*}. Thus Propositions 6.30 and 6.31 yield strictness:

$$[{e_L}^{\dagger}, {e_R}^{\dagger}] \cdot i_k^* = [{e_L}^{\dagger} \cdot x^{w*}, {e_R}^{\dagger} \cdot y^{w*}] = \perp.$$

$\square$

Corollary 6.33. *Every strict iterative monad is an Elgot monad.*

7. Conclusions and Further Research

The aim of this paper was to prove that every strict iterative monad is an iteration monad in the sense of Stephen Bloom and Zoltán Ésik. For monads on Set this has already been proved in the monograph [BÉ], our generalization concerns monads on every hyper-extensive locally finitely presentable category, see [ABMV]. The technical proof presented here makes it possible to apply this result to other base categories than Set , e. g. to the base category $\mathsf{Set}^{\mathscr{F}}$ of sets in context, where $\mathscr{F}$ is the category of finite sets and functions.

In the subsequent paper [AMV_4] we show that the category of Elgot monads (or, as called in [SP], all iteration theories with parametrized uniformity) is monadic over $\mathsf{Set}^{\mathscr{F}}$. Moreover, the corresponding monad on $\mathsf{Set}^{\mathscr{F}}$ is the monad of free iteration theories assigning to a set X in context the rational monad generated by $X_\perp$, which is a strict iterative monad. This shows a further deep relationship between iterative and iteration theories.

References

[AAV] P. Aczel, J. Adámek and J. Velebil, A coalgebraic view of infinite trees and iteration, *Electron. Notes Theor. Comput. Sci.* 44 (2001), no. 1, 26 pp.

[AAMV] P. Aczel, J. Adámek, S. Milius and J. Velebil, Infinite trees and completely iterative theories: A coalgebraic view, *Theoret. Comput. Sci*, 300 (2003), 1–45.

[ABMV] J. Adámek, R. Börger, S. Milius and J. Velebil, Iterative algebras: How iterative are they?, *Theory Appl. Categ.*, Vol. 19 (2008), 61–92.

[ADJ] J. B. Wright, J. W. Thatcher, E. G. Wagner and J. A. Goguen, Rational algebraic theories and fixed-point solutions, in: *Proc. 17th IEEE Symposium on Foundations of Computing, Houston, Texas*, IEEE Computer Society Press, Los Alamitos, 1976, 147–158.

[AMV_1] J. Adámek, S. Milius and J. Velebil, Iterative algebras at work, *Math. Structures Comput. Sci.*, 16 (2006), no. 6, 1085–1131.

[AMV_2] J. Adámek, S. Milius and J. Velebil, What are iteration theories?, in: *Proceedings of MFCS 2007* (L. Kučera and A. Kučera, eds.), Lecture Notes in Comput. Sci. 4708, Springer 2007, 240–252.

[AMV_3] J. Adámek, S. Milius and J. Velebil, Iterative reflection of monads, accepted for publication in *Math. Structures Comput. Sci.*

[AMV_4] J. Adámek, S. Milius and J. Velebil, Elgot theories: a new perspective of iteration theories, in: *Proceedings of MFPS XXV*, Electron. Notes Theor. Comp. Sci. 249 (2009), 407–427.

[AR] J. Adámek and J. Rosický, *Locally presentable and accessible categories*, Cambridge University Press, 1994.

[Bd] E. Badouel, Terms and Infinite Trees as Monads Over a Signature, TAPSOFT, Vol. 1, *Lecture Notes in Comput. Sci.* 351 (1989), 89–103.

[Ba] M. Barr, Coequalizers and free triples, *Math. Z.* 116 (1970), 307–322.

[BÉ] S. L. Bloom and Z. Ésik, *Iteration theories: The equational logic of iterative processes*, EATCS Monograph Series on Theoretical Computer Science, Springer-Verlag, 1993.

[BRS] M. Bonsangue, J. Rutten and A. Silva, An algebra for Kripke polynomial coalgebras, in: Proc. 24th Annual Symposium on Logic in Computer Science (LICS'09), IEEE Computer Society Press, 49–58.

[CLW] A. Carboni, S. Lack and R. F .C. Walters, Introduction to extensive and distributive categories, *J. Pure Appl. Algebra* 84 (1993), 145–158.

[E] C. C. Elgot, Monadic computation and iterative algebraic theories, in: *Logic Colloquium '73* (H. E. Rose and J. C. Shepherdson, eds.), North Holland Publishers, Amsterdam, 1975.

[EBT] C. C. Elgot, S. L. Bloom and R. Tindell, On the Algebraic Structure of Rooted Trees, *J. Comput. System Sci.* 16 (1978), 361–399.

[Ës] Z. Ésik, Axiomatizing iteration categories, *Acta Cybenet.* 14 (1999), no. 1, 65–82.

[FPT] M. Fiore, G. D. Plotkin and D. Turi, Abstract syntax and variable binding, in: *Proceedings of the 14th Symposium on Logic in Computer Science*, 1999, 193–202.

[GU] P. Gabriel and F. Ulmer, *Lokal präsentierbare Kategorien*, Lecture N. Math 221, Springer-Verlag, Berlin 1971.

[GLMP] N. Ghani, C. Lüth, F. DeMarchi and A. J. Power, Algebras, coalgebras, monads and comonads, *Electron. Notes Theor. Comput. Sci.* 44 (2001), no. 1, 18 pp.

[G] S. Ginali, Regular trees and the free iterative theory, *J. Comput. System Sci.* 18 (1979), 228–242.

[KP] G. M. Kelly, and A. J. Power. Adjunctions whose counits are coequalizers, and presentations of finitary enriched monads, *J. Pure Appl. Algebra* 89 (1993), 163–179.

[L] F. W, Lawvere, Functorial semantics of algebraic theories, PhD thesis, Columbia University, 1963. Republished in *Reprints in Theory Appl. Categ.* 5 (2004), 1–121.

[Li] F. E. J. Linton, Some aspects of equational theories, Proc. Conf. on Categorical Algebra at La Jolla (1966), 84–95.

[ML] S. Mac Lane, *Categories for the working mathematician*, 2nd edition, Springer-Verlag, 1998.

[M] S. Milius, Completely iterative algebras and completely Iterative Monads, *Inform. and Comput.* 196 (2005), 1–41.

[Mo_1] L. Moss, Parametric Corecursion, *Theoret. Comput. Sci.* 260 (2001), no. 1–2, 139–163.

[Mo_2] L. Moss, Recursion and corecursion have the same equational logic, *Theoret. Comput. Sci*, 294 (2003), 233–267.

[N] E. Nelson, Iterative Algebras, *Theoret. Comput. Sci.* 25 (1983), 67–94.

[SP] A. K. Simpson and G. D. Plotkin, Complete axioms for categorical fixed-point operators, in: *Proceedings of the 15th Symposium on Logic in Computer Science*, 2000, 30–44.

[T] J. Tiuryn, Unique fixed points vs. least fixed points, *Theoret. Comput. Sci.* 12 (1980), 229–254.

[W] J. Worrell, On the final sequence of a finitary set functor, *Theoret. Comput. Sci.* 338 (2005), 184–199.