Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 325 (2016) 147–168

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Complete Elgot Monads and Coalgebraic Resumptions[†]

Sergey Goncharov[1,⋆] Stefan Milius[2] Christoph Rauch[3,⋆]

Lehrstuhl für Theoretische Informatik, Friedrich-Alexander Universität Erlangen-Nürnberg, Germany

Abstract

Monads are used to abstractly model a wide range of computational effects such as nondeterminism, statefulness, and exceptions. *Complete Elgot monads* are monads that are equipped with a *(uniform) iteration operator* satisfying a set of natural axioms, which allows to model *iterative computations* just as abstractly. It has been shown recently that extending complete Elgot monads with free effects (e.g. operations of sending/receiving messages over channels) canonically leads to *generalized coalgebraic resumption monads*, which were previously used as semantic domains for non-wellfounded guarded processes. In this paper, we continue the study of complete Elgot monads and their relationship with generalized coalgebraic resumption monads. We give a characterization of the Eilenberg-Moore algebras of the latter. In fact, we work more generally with Uustalu's *parametrized monads*; we introduce complete Elgot algebras for a parametrized monad and we prove that they form an Eilenberg-Moore category. This is further used for establishing a characterization of complete Elgot monads as those monads whose algebras are coherently equipped with the structure of complete Elgot algebras for the parametrized monads obtained from generalized coalgebraic resumption monads.

Keywords: Complete Elgot monad, complete Elgot algebra, resumption monad, uniform iteration

1 Introduction

One traditional use of monads in computer science, stemming from the seminal thesis of Lawvere [20], is as a tool for algebraic semantics where monads arise as a high-level metaphor for (clones of) equational theories. More recently, Moggi proposed to associate monads with *computational effects* and use them as a generic tool for denotational semantics [22], which later had a considerable impact on the design of functional programming languages, most prominently Haskell [1]. Finally, in the first decade of the new millennium, Plotkin and Power reestablished the

[†] Full version is available at http://arxiv.org/abs/1603.02148

[⋆] Supported by Deutsche Forschungsgemeinschaft (DFG) under project SCHR 1118/8-1

[1] Email: Sergey.Goncharov@fau.de

[2] Email: mail@stefan-milius.eu Supported by Deutsche Forschungsgemeinschaft (DFG) under project MI 717/5-1

[3] Email: Christoph.Rauch@fau.de

http://dx.doi.org/10.1016/j.entcs.2016.09.036
1571-0661/© 2016 The Author(s). Published by Elsevier B.V.

connection between computational monads and algebraic theories in their theory of *algebraic effects* [23, 24].

We use the outlined view to study the notion of *iteration* as a concept that has a well-established algebraic meaning and which is very relevant in the context of computational effects. On the technical level our present work can be viewed as a continuation of the previous extensive work on monads with iteration [2,5,7] having its roots in the work of Elgot [12] and Bloom and Ésik [10] on iteration theories.

More specifically, we are concerned with a particular construction on monads: given a monad $\mathbb{T}$ and a functor Σ, we assume the existence of the coalgebra

$$T_\Sigma X = \nu\gamma.\, T(X + \Sigma\gamma) \tag{$\bigstar$}$$

for each object X (these final coalgebras exist under mild assumptions on T, Σ, and the base category). It is known [28] that T_Σ extends to a monad $\mathbb{T}_\Sigma$ and we call the latter the *generalized coalgebraic resumption monad.*

Intuitively, ($\bigstar$) is a generic semantic domain for systems combining *extensional* (via $\mathbb{T}$) and *intensional* (via Σ) features with iteration. To make this intuition more precise, consider the following simplistic

Example 1.1 Let $A = \{a, b\}$ be an alphabet of *actions.* Then the following system of equations specifies *processes* x_1, x_2, x_3 of *basic process algebra (BPA)*:

$$x_1 = a \cdot (x_2 + x_3) \qquad\qquad x_2 = a \cdot x_1 + b \cdot x_3 \qquad\qquad x_3 = a \cdot x_1 + \checkmark$$

We can think of this specification as a map $P \to T(\{\checkmark\} + \Sigma P)$ where $P = \{x_1, x_2, x_3\}$, $\Sigma = A \times \text{–}$ and $T = \mathcal{P}_\omega$ is the finite powerset monad. Using the standard approach [26] we can *solve* this specification by finding a map $P \to T_\Sigma\{\checkmark\}$ that assigns to every x_i the corresponding semantics over the domain of possibly non-wellfounded trees $T_\Sigma\{\checkmark\} = \nu\gamma.\mathcal{P}_\omega(\{\checkmark\} + A \times \gamma)$. The crucial fact here is that the original system is *guarded*, i.e. every recursive call of a variable x_i is preceded by an action. This implies that the given recursive system has a unique solution.

If the guardedness assumption is dropped, solutions may fail to be unique, but it is possible to introduce a notion of *canonical solution* if the Kleisli category of the monad $\mathbb{T}$ is enriched in the category of complete partial orders, or more generally, if $\mathbb{T}$ is a *complete Elgot monad.* A monad $\mathbb{T}$ is a complete Elgot monad if it is equipped with an iteration operator that assigns to every morphism of the form $f : X \to T(Y + X)$ a *solution* $f^\dagger : X \to TY$ satisfying a certain well-established set of equational axioms of iteration and also *uniformity* [27] (e.g. $\mathcal{P}_\omega$ is not a complete Elgot monad, but the countable powerset monad $\mathcal{P}_{\omega_1}$ is). The central result of the recent work [14] is that whenever $\mathbb{T}$ is a complete Elgot monad then so is the transformed monad ($\bigstar$). In particular, this allows for canonical solutions of recursive equations over processes (in the sense of Example 1.1) whenever recursive equations over $\mathbb{T}$ are solvable.

In the present paper we investigate the relationship between guarded and unguarded iteration, which are implemented via generalized coalgebraic resumption monads and complete Elgot monads, respectively. As an auxiliary abstraction device, we use the notion of a *parametrized monad* introduced by Uustalu [28], i.e. a bifunctor $\# : \mathbf{C}\times\mathbf{C} \to \mathbf{C}$ such that for every object X, the functor $\text{–} \# X$ is a monad. For example, the bifunctor $X \# Y = T(X+\Sigma Y)$ in ($\bigstar$) yields a parametrized monad. Following [5], we introduce *complete Elgot algebras* for a parametrized monad $\#$, which are algebras for $\#$ equipped with an iteration operator satisfying simplified versions of the axioms of complete Elgot monads. However, in contrast to the latter complete Elgot $\#$-algebras omit any form of the Bekić law that states how mutually recursive definitions are solved. We then prove that for every object X the final coalgebra $\nu\gamma.\, X \# \gamma$ is equivalently a free complete Elgot $\#$-algebra on X and that the category of Eilenberg-Moore algebras for the ensuing monad $\nu\gamma.\, \text{–} \# \gamma$ is isomorphic to the category of complete Elgot $\#$-algebras.

Furthermore, we show that for every complete Elgot monad $\mathbb{T}$, every free $\mathbb{T}$-algebra TZ canonically extends to a complete Elgot $\#$-algebra for $X \# Y = T(X+Y)$. This situation can be roughly summarized as follows:

$$(\nu\gamma.\, \text{–} \# \gamma)\text{-algebras} \quad \cong \quad \text{complete Elgot } \#\text{-algebras} \quad \supseteq \quad \text{free } \mathbb{T}\text{-algebras}$$

From the perspective of Example 1.1 this connection can be regarded as follows. Since $\Sigma = \mathsf{Id}$, the set of guards consists of only one action, which can be understood as *delaying*. Now, the inclusion of $\mathbb{T}$-algebras into $(\nu\gamma.\, \text{–} \# \gamma)$-algebras essentially means that complete Elgot monads interpret staged, possibly infinite, guarded processes over $\mathbb{T}$ by forgetting the guards.

Suppose that, conversely, we have any monad $\mathbb{T}$ such that the above inclusion holds in the sense that $\mathbb{T}$-algebras are coherently equipped with structures of complete Elgot $\#$-algebras. Then $\mathbb{T}$ is equipped with an iteration operator satisfying a set of axioms that are weaker than the axioms of complete Elgot monads; the ensuing notion is that of a *weak complete Elgot monad*.

The paper is organized as follows. After categorical preliminaries (Section 2) we present and discuss complete Elgot monads in Section 3. In Section 4 we introduce algebras and complete Elgot algebras for a parametrized monad $\#$; next, in Section 5, we show that the category of complete Elgot $\#$-algebras is isomorphic to the Eilenberg-Moore category of the monad $\nu\gamma.\, \text{–} \# \gamma$ (Theorem 5.7); furthermore, we show that a free complete Elgot $\#$-algebra on X is equivalent to the final coalgebra $\nu\gamma.\, X \# \gamma$ (Theorem 5.9). Finally, in Section 6 we apply the developed results to characterize complete Elgot monads as those whose algebras are coherently equipped with complete Elgot algebra structures (Theorem 6.4 and 6.6).

Further Related Work. Algebras for parametrized monads were introduced in [4,6] albeit for the special case of a *base*, i.e. a finitary parametrized monad on a locally finitely presentable category. Loc. cit. also introduces iterative base algebras which are algebras for a base having unique solutions of finitary recursive equations. Complete Elgot algebras for an endofunctor H were introduced in [5], and it was proved that they form the Eilenberg-Moore category of the monad T obtained by taking the final coalgebras $TX = \nu\gamma.(X + H\gamma)$; this is the free completely iterative monad on H (see [2]). Since $X \# Y = X + HY$ is a parametrized monad, our notion of complete Elgot algebras generalizes the previous notion to the level of parametrized monads and it extends iterative algebras by considering an iteration operation subject to certain axioms in lieu of unique solutions. Our Theorem 5.7 generalizes [5, Theorem 5.8].

The study of monads with an iteration operator is inspired by Bloom and Ésik's iteration theories [10]. Extending this from Lawvere theories (i.e. finitary monads on **Set**) to monads on more general categories has led to the notion of Elgot monad introduced in [7]. While iteration theories and Elgot monads study an iteration operator for recursive equations with finitely many recursion variables, complete Elgot monads [14] are equipped with an iteration operator for all (finitary and infinitary) recursive equations.

2 Preliminaries

We assume that readers are familiar with basic category theory [21]; we write $|\mathbf{C}|$ for the class of objects of a category $\mathbf{C}$ and $f : X \to Y$ for morphisms in $\mathbf{C}$. We often omit indexes, e.g. on natural transformations, if they are clear from the context.

In this paper we work with an ambient category $\mathbf{C}$ with finite coproducts. We denote by inl and inr the left- and right-hand coproduct injections from X and Y to $X + Y$, and $[f, g] : X + Y \to Z$ the is the *copair* of $f : X \to Z$ and $g : Y \to Z$, i.e. the unique morphism with $[f, g]\,\mathsf{inl} = f$ and $[f, g]\,\mathsf{inr} = g$. The codiagonal is denoted by $\nabla = [\mathsf{id}, \mathsf{id}] : X + X \to X$ as usual.

We consider *monads* on $\mathbf{C}$ given in the form of *Kleisli triples* $\mathbb{T} = (T, \eta, \text{-}^\star)$ where T is an endomap on $|\mathbf{C}|$, η, called *monad unit*, is a family of morphisms $\eta_X : X \to TX$ indexed over $|\mathbf{C}|$, and *(Kleisli) lifting* assigns to each $f : X \to TY$ a morphism $f^\star : TX \to TY$ such that the following laws hold:

$$\eta^\star = \mathsf{id}, \qquad\qquad f^\star\,\eta = f, \qquad\qquad (f^\star\,g)^\star = f^\star\,g^\star.$$

This is equivalent to the definition of a monad as a triple (T, η, μ) that consists of a functor T and two natural transformations, the monad unit $\eta : \mathsf{Id} \to T$ and the *monad multiplication* $\mu : TT \to T$ [21]. In particular, given a Kleisli triple, $\mu = \mathsf{id}^\star$ yields the monad multiplication, η extends to a natural transformation, and T to an endofunctor with morphism mapping $Tf = (\eta\,f)^\star$. The *Kleisli category* $\mathbf{C}_\mathbb{T}$ of $\mathbb{T}$ is formed by *Kleisli morphisms* $X \to TY$, i.e. $\mathbf{C}_\mathbb{T}(X, Y) = \mathbf{C}(X, TY)$ with η_X as identity morphism on X and *Kleisli composition*: given Kleisli morphisms $f : X \to TY$ and $g : Y \to TZ$ we have

$$f \diamond g = \left(X \xrightarrow{f} TY \xrightarrow{g^\star} TZ\right).$$

We write $f : X \mathrel{\to\!\!\!\!\diamond\!\!\to} Y$ for a Kleisli morphism $f : X \to TY$.

The forgetful functor from $\mathbf{C}_\mathbb{T}$ to $\mathbf{C}$ has a left adjoint sending any $f : X \to Y$ to $\underline{f} = \eta\,f : X \to TY$. Like any left adjoint, this functor preserves colimits, and in particular coproducts. Since $|\mathbf{C}| = |\mathbf{C}_\mathbb{T}|$, this implies that coproducts in $\mathbf{C}_\mathbb{T}$ exist and are lifted from $\mathbf{C}$. Explicitly, $\underline{\mathsf{inl}} = \eta\,\mathsf{inl} : X \mathrel{\to\!\!\!\!\diamond\!\!\to} X + Y$, $\underline{\mathsf{inr}} = \eta\,\mathsf{inr} : Y \mathrel{\to\!\!\!\!\diamond\!\!\to} X + Y$

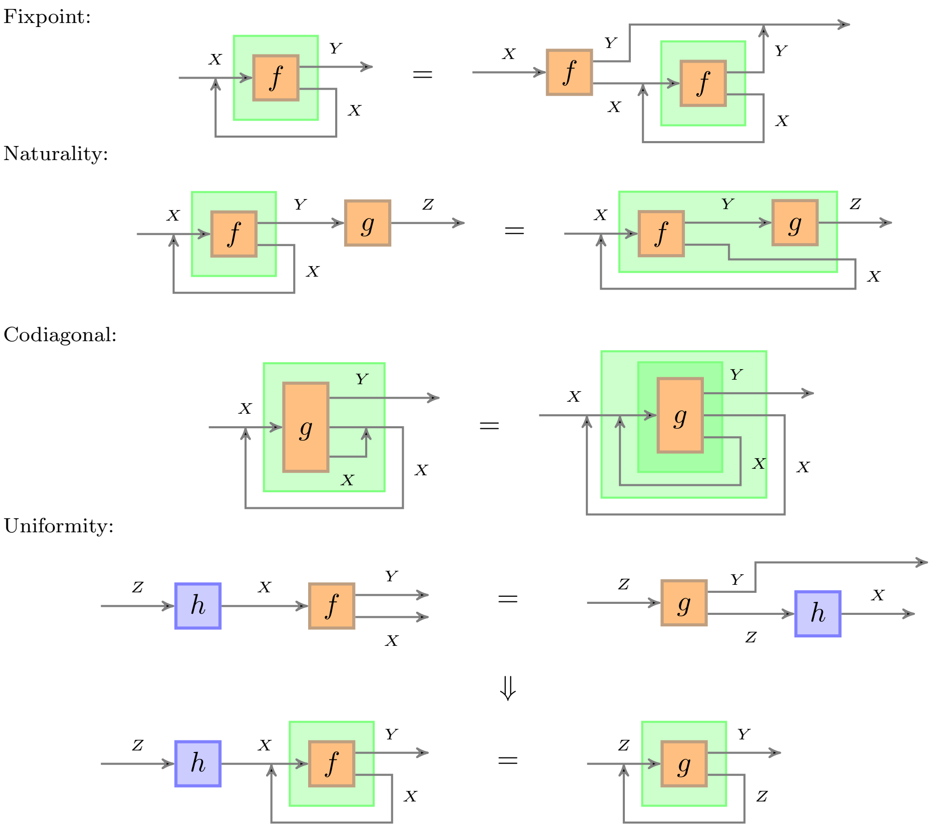

Fig. 1. Axioms of complete Elgot monads.

are the coproduct injections in $\mathbf{C}_{\mathbb{T}}$ and $[f, g] : A + B \mathrel{-\!\!\diamond\!\!\!\to} C$ (formed in $\mathbf{C}$) is the copair of $f : A \mathrel{-\!\!\diamond\!\!\!\to} C$ and $g : B \mathrel{-\!\!\diamond\!\!\!\to} C$ in $\mathbf{T}_{\mathbb{T}}$. We denote by $f \oplus g : A + B \mathrel{-\!\!\diamond\!\!\!\to} A' + B'$ the coproduct of morphisms $f : A \mathrel{-\!\!\diamond\!\!\!\to} A'$ and $g : B \mathrel{-\!\!\diamond\!\!\!\to} B'$ in $\mathbf{C}_{\mathbb{T}}$. Besides $\mathbf{C}_{\mathbb{T}}$, we consider the category $\mathbf{C}^{\mathbb{T}}$ of *(Eilenberg-Moore) algebras* for $\mathbb{T}$, whose objects are pairs $(A, a : TA \to A)$, satisfying two laws: $a\,\eta = \mathsf{id}$ and $a\,(Ta) = a\,\mu$; a $\mathbb{T}$-algebra morphism f from (A, a) to (A, b) is a morphism $f : A \to B$ such that $f\,a = b\,Tf$. See [21] for more details.

We will make use of standard facts on *coalgebras* for an endofunctor [25]. Given an endofunctor $F : \mathbf{C} \to \mathbf{C}$, an F-coalgebra is a pair (X, c) where X is an object of $\mathbf{C}$ called the *carrier* of the coalgebra and $c : X \to FX$ is a morphism called the *(transition) structure*. A coalgebra morphism f from (X, c) to (Y, d) is a morphism $f : X \to Y$ such that $d\,f = (Ff)\,c$. Coalgebras and their morphisms form a category. The final F-coalgebra, if it exists, is the terminal object in that category and is denoted by

$$\nu F \xrightarrow{\mathsf{out}} F(\nu F).$$

By Lambek's lemma, out is an isomorphism, whose inverse $\mathsf{out}^{-1} : F(\nu F) \to \nu F$ can be obtained as $\mathsf{coit}(F\,\mathsf{out})$, where for any coalgebra $(X, f : X \to FX)$ we denote by $\mathsf{coit}\,f$ the unique coalgebra morphism $X \to \nu F$ from X to the final coalgebra νF.

3 Complete Elgot Monads for Iteration

Complete Elgot monads are a slight generalization of Elgot monads from [7,8], which in turn, for the base category being **Set**, correspond precisely to those iteration theories of Bloom and Ésik [10] that satisfy the functorial dagger implication for base morphisms. In the following definition cited from [14] (for simplicity, we do not consider *strong* monads here because the possible presence of a strength has no bearing on our results), we follow the terminology of [9, 27] where the same axioms were considered in the dual setting of generic parametrized recursion.

Definition 3.1 (Complete Elgot monads) *A* complete Elgot monad *is a monad $\mathbb{T}$ equipped with an operator $-^\dagger$, called* iteration*, that assigns to each morphism $f : X \to\!\!\!\!\diamond\!\!\to Y + X$ a morphism $f^\dagger : X \to\!\!\!\!\diamond\!\!\to Y$ such that the following axioms hold:*

- fixpoint: $f^\dagger = [\eta, f^\dagger] \diamond f$, *for any* $f : X \to\!\!\!\!\diamond\!\!\to Y + X$*;*
- naturality: $g \diamond f^\dagger = ((g \oplus \eta) \diamond f)^\dagger$ *for any* $f : X \to\!\!\!\!\diamond\!\!\to Y + X$ *and* $g : Y \to\!\!\!\!\diamond\!\!\to Z$*;*
- codiagonal[4] :$([\eta, \underline{\mathsf{inr}}] \diamond g)^\dagger = g^{\dagger\dagger}$ *for any* $g : X \to\!\!\!\!\diamond\!\!\to (Y + X) + X$*;*
- uniformity: $f \diamond \underline{h} = (\eta \oplus \underline{h}) \diamond g$ *implies* $f^\dagger \diamond \underline{h} = g^\dagger$ *for any* $f : X \to\!\!\!\!\diamond\!\!\to Y + X$*,* $g : Z \to\!\!\!\!\diamond\!\!\to Y + Z$ *and* $h : Z \to X$*.*

The above axioms of iteration can be comprehensibly represented in a flowchart-style as in Fig. 1. Here the feedback loops correspond to iteration and the coloured frames indicate the scope of the constructs being iterated. We believe that this presentation is rather well in touch with the intuition. For example, the naturality axiom expresses the fact that the scope of the iteration can be stretched to embrace a function post-processing the output of the terminating branch.

There is an obvious similarity between the axioms in Fig. 1 and the axioms of *traced monoidal categories* [18]. In fact, Hasegawa [16] proved that there is an equivalent presentation of a dagger operation satisfying the above axioms in terms of a uniform trace operator w.r.t. coproducts (actually, Hasegawa worked in the dual setting with products). Note that the present axioms make use of coproduct injections and the codiagonal morphism, while the trace axioms can be formulated more generally for any monoidal product.

One standard source of examples for complete Elgot monads is a suitable enrichment of the Kleisli category $\mathbf{C}_\mathbb{T}$ over complete partial orders.

Example 3.2 (ω-continuous monads) An ω*-continuous* monad is a monad $\mathbb{T}$ such that the Kleisli category $\mathbf{C}_\mathbb{T}$ is enriched over the category **Cppo** of ω-complete partial orders with bottom $\bot$ and (nonstrict) continuous maps; moreover, composition in $\mathbf{C}$ is required to be left strict and composition in $\mathbf{C}_\mathbb{T}$ right strict: $\bot f = \bot$, $f \diamond \bot = \bot$; equivalently, $\bot$ is a *constant* of $\mathbb{T}$. We also assume that coproducts in $\mathbf{C}_\mathbb{T}$ are **Cppo**-enriched; for this it suffices that copairing is monotone in both arguments. It then follows that it is also continuous; for $\bigsqcup_i [f_i, g]$ is a morphism

[4] The codiagonal axiom is often written as $((\eta \oplus \underline{\nabla}) \diamond g)^\dagger = g^{\dagger\dagger}$ implicitly alluding to the canonical isomorphism $Y + (X + X) \cong (Y + X) + X$.

satisfying $(\bigsqcup_i [f_i, g])\,\mathsf{inl} = \bigsqcup_i f_i$ and $(\bigsqcup_i [f_i, g])\,\mathsf{inr} = g$ by continuity of composition, whence $\bigsqcup_i [f_i, g] = [\bigsqcup_i f_i, g]$. Similarly, one shows continuity in the second argument. (Note that monotonicity is used only so that the $[f_i, g]$ form an ω-chain provided that the f_i do.)

It is shown in [14] that an ω-continuous monad is a complete Elgot monad with $e^\dagger$ calculated as the least fixed point of the map $f \mapsto [\eta, f] \diamond e$. This yields the powerset monad $\mathcal{P}$, the *Maybe-monad* $(\text{-} + 1)$, or the nondeterministic state monad $\mathcal{P}(\text{-} \times S)^S$ as examples of complete Elgot monads on **Set**. The *lifting monad* $(\text{-})_\perp$ is a complete Elgot monad on the category of complete partial orders without bottom.

Another principal source of examples are *free complete Elgot monads* for which the iteration of guarded morphisms is uniquely defined.

Example 3.3 (Free complete Elgot monads) Suppose $\mathbb{T}$ is the initial complete Elgot monad. It is shown in [14] that whenever the functor $\mathbb{T}_\Sigma$ defined by (★) exists, it yields the *free complete Elgot monad on* Σ (note that the original $\mathbb{T}$ is the free complete Elgot monad on the constant functor on the initial object of **C**). In **Set** (more generally, in any *hyperextensive category* [3]) the initial complete Elgot monad $\mathbb{T}$ is the Maybe-monad $\text{-} + 1$.

Example 3.4 (Capretta's partiality monad) One instructive example, not covered by the above cases is the coalgebraic resumption monad $\nu\gamma.\ \text{-} + \gamma$, studied by Capretta for modeling partiality in the intensional type theory [11]. Note that this example is not covered by Example 3.3, for that only states that $\nu\gamma.\ T(X + \gamma)$ is a complete Elgot monad, provided $\mathbb{T}$ is one (e.g. it follows that $\nu\gamma.\ X + 1 + \gamma$ is a complete Elgot monad on **Set**), but $T = \mathsf{Id}$ is not a complete Elgot monad in any of the relevant examples.

We conjecture that $T = \nu\gamma.\ \text{-} + \gamma$ can be shown to be a complete Elgot monad over any sufficiently rich (type-theoretic) universe; in particular, this can be easily seen in **Set**: Here TX explicitly evaluates to $X \times \mathbb{N} + \{\perp\}$, hence every TX can be ordered as a flat domain, i.e. $x \sqsubseteq y$ iff $x = \perp$ or $x = y$; this easily extends to a **Cppo**-enrichment as required in Example 3.2 and hence gives rise to a complete Elgot monad structure on $\mathbb{T}$. Intuitively, every function $f : X \to T(Y + X)$ either diverges or delivers a result together with the number of steps needed to compute it. The iteration $f^\dagger$ sums the numbers occurring across the loop and in case of convergence delivers the sum together with the result value. Note that the number of unfoldings of $f^\dagger$ in this process does not contribute to the result, which explains why the fixpoint identity indeed holds for $\mathbb{T}$.

In comparison to the previous work [14], Definition 3.1 remarkably drops the axiom of *dinaturality* (see Fig. 2). The reason is that this axiom turns out to be derivable, which is a fact that was recently discovered and formalized on the level of iteration theories [13]. Corollary 6 from *op. cit.* can be couched in present terms as follows (modulo the terminological change: *parameter identity* instead of *naturality*, *double dagger* instead of *codiagonal* and *dagger implication for base morphisms* instead of *uniformity*):

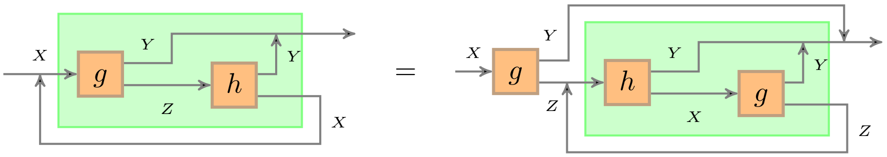

Fig. 2. Dinaturality axiom.

Proposition 3.5 (Dinaturality) *Given* $g : X \mathrel{-\!\!\diamond\!\!\!\to} Y + Z$ *and* $h : Z \mathrel{-\!\!\diamond\!\!\!\to} Y + X$*, then*

$$([\underline{\mathsf{inl}}, h] \diamond g)^{\dagger} = [\eta, ([\underline{\mathsf{inl}}, g] \diamond h)^{\dagger}] \diamond g.$$

The codiagonal axiom in Definition 3.1 can equivalently be replaced by a form of the well-known *Bekić identity*, see [10].

Proposition 3.6 (Bekić identity) *A complete Elgot monad* $\mathbb{T}$ *is, equivalently, a monad satisfying the* fixpoint, naturality *and* uniformity *axioms (as in Definition 3.1), and the* Bekić identity

$$(T\alpha\,[f, g])^{\dagger} = [\eta, h^{\dagger}] \diamond [\underline{\mathsf{inr}}, g^{\dagger}],$$

where $g : X \mathrel{-\!\!\diamond\!\!\!\to} (Z + Y) + X$, $f : Y \mathrel{-\!\!\diamond\!\!\!\to} (Z + Y) + X$, $h = [\eta, g^{\dagger}] \diamond f : Y \mathrel{-\!\!\diamond\!\!\!\to} Z + Y$, *with* $\alpha : (A + B) + C \to A + (B + C)$ *being the obvious associativity isomorphism.*

4 Parametrized Monads for Complete Elgot Algebras

In order to study complete Elgot monads and their algebras it is helpful to make a further abstraction step and generalize from monads to *parametrized monads* [28] (finitary parametrized monads are also called *bases* [4]), which are of independent interest.

Definition 4.1 (Parametrized monad) A *parametrized monad* over $\mathbf{C}$ is a functor from $\mathbf{C}$ to the category of monads over $\mathbf{C}$ and monad morphisms. More explicitly, a parametrized monad is a bifunctor $\# : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$ such that for any $X \in |\mathbf{C}|$, $\text{--} \# X : \mathbf{C} \to \mathbf{C}$ is a monad, and for any $f : X \to Y$, $\mathsf{id} \# f : Z \# X \to Z \# Y$ is the Z-component of a monad morphism from $\text{--} \# X$ to $\text{--} \# Y$.

Remark 4.2 The order of arguments in $X \# Y$ is in agreement with [28] and differs from [4] where the notation $Y \mathbin{\Box} X$ equivalent to the present $X \# Y$ is used. We chose the order of arguments to ensure agreement with the type profile of the iteration operator $\text{--}^{\dagger}$, which is in turn in agreement with the expression (★).

Following [4] we will from now on denote the unit and multiplication of the monads $\text{--} \# X$ by $u^X_A : A \to A \# X$ and $m^X_A : (A \# X) \# X \to A \# X$, respectively.

Example 4.3 (Parametrized monads) We recall some standard examples of parametrized monads from [28]; further examples can be found e.g. in [6].

(i) Whenever $\mathbb{T} = (T, \eta, \text{-}^\star)$ is a monad and Σ is a functor, $A \mathbin{\#} X = T(A + \Sigma X)$ is a parametrized monad with the unit given by

$$u^X_A = \Big(A \xrightarrow{\mathsf{inl}} A + \Sigma X \xrightarrow{\eta_{A+\Sigma X}} T(A + \Sigma X)\Big)$$

and the multiplication by

$$m^X_A = \Big(T(T(A + \Sigma X) + \Sigma X) \xrightarrow{[\mathsf{id}, \eta_{A+\Sigma X}\, \mathsf{inr}]^\star} T(A + \Sigma X)\Big).$$

Specifically, if Σ is the constant functor on an object E then $X \mathbin{\#} Y$ is the exception monad transformer with exceptions from E [22]. Another interesting special case is when $\mathbb{T}$ is the identity monad (cf. Remark 4.8).

(ii) $A \mathbin{\#} X = A \times X^\star$ is a parametrized monad with the unit and multiplication given by

$$u^X_A : a \mapsto (a, \varepsilon) \quad \text{and} \quad m^X_A : (a, w, v) \mapsto (a, wv),$$

where ε denotes the empty word and wv concatenation of words.

(iii) Given a contravariant endofunctor H, $A \mathbin{\#} X = A^{HX}$ is a parametrized monad with the unit and multiplication given by

$$u^X_A : a \mapsto \lambda x.\, a \quad \text{and} \quad m^X_A : (f : HX \to (HX \to A)) \mapsto \lambda x.\, f(x)(x).$$

This is a generalization of the well known *reader monad*, which can be recovered by instantiating H with a constant functor.

The following is a straightforward extension of the notion of an algebra for a base studied in [4] to arbitrary parametrized monads.

Definition 4.4 (#-algebras) Given a parametrized monad $\# : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$, a $\#$-*algebra* is a pair (A, a) consisting of an object A of $\mathbf{C}$, and an algebra for the monad $\text{-} \mathbin{\#} A$, i.e. a morphism $a : A \mathbin{\#} A \to A$ satisfying

$$\begin{tikzcd} A \arrow[r, "u^A_A"] \arrow[dr, equal, "\mathsf{id}"'] & A \mathbin{\#} A \arrow[d, "a"] \\ & A \end{tikzcd} \qquad \begin{tikzcd} (A \mathbin{\#} A) \mathbin{\#} A \arrow[r, "a \# \mathsf{id}"] \arrow[d, "m^A_A"'] & A \mathbin{\#} A \arrow[d, "a"] \\ A \mathbin{\#} A \arrow[r, "a"] & A \end{tikzcd}$$

A morphism between #-algebras (A, a) and (B, b) is a $\mathbf{C}$-morphism $f : A \to B$ such that

$$\begin{tikzcd} A \mathbin{\#} A \arrow[r, "a"] \arrow[d, "f \# f"'] & A \arrow[d, "f"] \\ B \mathbin{\#} B \arrow[r, "b"] & B \end{tikzcd}$$

For our leading example $X \mathbin{\#} Y = T(X + \Sigma Y)$ the category of #-algebras can be described explicitly. Recall that a $\mathbb{T}$-Σ-bialgebra in the sense of Kelly [19] is a triple (A, a, f) where $a : TA \to A$ is a $\mathbb{T}$-algebra and $f : \Sigma A \to A$ is a Σ-algebra.

Proposition 4.5 *Let $X \# Y = T(X + \Sigma Y)$ for a monad $\mathbb{T}$ and a functor Σ on $\mathbf{C}$. Then $\#$-algebras are precisely $\mathbb{T}$-Σ-bialgebras.*

Proof (Sketch). Given a $\#$-algebra $\alpha : T(A + \Sigma A) \to A$ one forms two algebra structures

$$a = \Big(TA \xrightarrow{T\,\mathsf{inl}} T(A + \Sigma A) \xrightarrow{\alpha} A\Big)$$
$$\text{and}\quad b = \Big(\Sigma A \xrightarrow{\mathsf{inr}} A + \Sigma A \xrightarrow{\eta} T(A + \Sigma A) \xrightarrow{\alpha} A\Big).$$

A straightforward calculation then shows that a is a $\mathbb{T}$-algebra structure, whence (A, a, b) is a $\mathbb{T}$-Σ-bialgebra.

Conversely, given any $\mathbb{T}$-Σ-bialgebra $a : TA \to A \leftarrow \Sigma A : b$ one forms

$$\alpha = \Big(T(A + \Sigma A) \xrightarrow{T[\mathsf{id}, b]} TA \xrightarrow{a} A\Big).$$

Another straightforward computation establishes that this is the structure of a $\#$-algebra.

Finally, it is easy to see that the above two constructions are mutually inverse and extend to an (identity on morphisms) isomorphism between the categories of $\#$-algebras and $\mathbb{T}$-Σ-bialgebras. □

Corollary 4.6 *Let $X \# Y = T(X + Y)$ for a monad $\mathbb{T}$ on $\mathbf{C}$. The category $\mathbf{C}^{\mathbb{T}}$ of $\mathbb{T}$-algebras is isomorphic to the full subcategory of those $\#$-algebras $a : T(A + A) \to A$, which factor through $T\nabla : T(A + A) \to TA$.*

Analogously to complete Elgot monads, we introduce $\#$-algebras with iteration. This generalizes the definition of a *complete Elgot algebra for a functor* from [5].

Definition 4.7 (Complete Elgot $\#$-algebras) A *complete Elgot $\#$-algebra* is a $\#$-algebra $a : A \# A \to A$ equipped with an iteration operator

$$\frac{e : X \to A \# X}{e^\dagger : X \to A}$$

satisfying the following axioms:

- *solution:* for every $e : X \to A \# X$ we have

$$\begin{tikzcd}
X \arrow[r, "e^\dagger"] \arrow[d, "e"'] & A \\
A \# X \arrow[r, "A \# e^\dagger"] & A \# A \arrow[u, "a"']
\end{tikzcd}$$

- *functoriality:* for every $e : X \to A \# X$, $f : Y \to A \# X$ and $h : X \to Y$,

$$\begin{tikzcd}
X \arrow[r, "e"] \arrow[d, "h"'] & A \# X \arrow[d, "A \# h"] & & X \arrow[dr, "e^\dagger"] \arrow[dd, "h"'] & \\
Y \arrow[r, "f"'] & A \# Y & \text{implies} & & A \\
& & & Y \arrow[ur, "f^\dagger"'] &
\end{tikzcd}$$

$f\,h = (\mathsf{id} \# h)\,e$ implies $f^\dagger\,h = e^\dagger$;

- *compositionality:* for every $f : Y \to A \# Y$ and $g : X \to Y \# X$ define

$$f^\dagger \bullet g = (X \xrightarrow{g} Y \# X \xrightarrow{f^\dagger \# \mathsf{id}} A \# X)$$

and $f \blacksquare g : Y + X \to A \# (Y + X)$ by

$$\begin{tikzcd}
Y + X \arrow[r, "{[u_Y^X, g]}"] & Y \# X \arrow[r, "f \# \mathsf{id}"] & (A \# Y) \# X \arrow[d, "(\mathsf{id} \# \mathsf{inl}) \# \mathsf{inr}"] \\
A \# (Y + X) & & (A \# (Y + X)) \# (Y + X) \arrow[ll, "m_A^{Y+X}"']
\end{tikzcd}$$

Compositionality states that $(f \blacksquare g)^\dagger\ \mathsf{inr} = (f^\dagger \bullet g)^\dagger : X \to A$.

A *morphism* from a complete Elgot #-algebra $(A, a, \text{-}^\dagger)$ to a complete Elgot #-algebra $(B, b, \text{-}^\ddagger)$ is a $\mathbf{C}$-morphism $f : A \to B$, such that for all $e : X \to A \# X$ we have:

$$\Big(X \xrightarrow{e^\dagger} A \xrightarrow{f} B\Big) = \Big(X \xrightarrow{e} A \# X \xrightarrow{f \# \mathsf{id}} B \# X\Big)^\ddagger.$$

This defines the category of complete Elgot #-algebras $\mathbf{CElg}_\#(\mathbf{C})$.

Remark 4.8 Note that complete Elgot #-algebras for the parametrized monad $A \# X = A + \Sigma X$ (i.e. the parametrized monad of Example 4.3 (i) for $\mathbb{T}$ the identity monad) are precisely the complete Elgot algebras for the functor Σ introduced and studied in [5].

Like in the case of complete Elgot monads, a standard way to obtain complete Elgot #-algebras is by enforcing a suitable enrichment over complete partial orders with bottom.

Example 4.9 (Continuous algebras are complete Elgot algebras) Consider any category $\mathbf{C}$ that is enriched over $\mathbf{Cppo}$ such that composition is left strict, i.e. $\bot\, f = \bot$, and a parametrized monad $\# : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$ that is *locally continuous* in both arguments, i.e.

$$\bigsqcup_i (f_i \# g_i) = \Big(\bigsqcup_i f_i\Big) \# \Big(\bigsqcup_i g_i\Big)$$

holds for any ω-chains $(f_i : X \to Y)_{i<\omega}$ and $(g_i : X' \to Y')_{i<\omega}$. Then every #-algebra becomes a complete Elgot #-algebra when equipped with the operation $\text{-}^\dagger$ assigning to every $e : X \to A \# X$ its least solution. In more detail, let A be a #-algebra. To every $e : X \to A \# X$ we assign $e^\dagger : X \to A$ given by

$$e^\dagger = \bigsqcup_i e_i^\dagger,$$

where $e_0^\dagger = \bot : X \to A$ and $e_{i+1} = a\,(\mathsf{id} \# e_i^\dagger)\,e$. That means that $e^\dagger$ is the least fixed point of the function $s \mapsto a\,(\mathsf{id} \# s)\,e$ on $\mathbf{C}(X, A)$. The verification that this satisfies the axioms of a complete Elgot $\#$-algebra can be found in the full version of our paper.

Example 4.10 The previous example can easily be generalized as follows. Let $\#$ be a parametrized monad on an arbitrary category $\mathbf{C}$. Suppose that $a : A \# A \to A$ is a $\#$-algebra such that

(i) for every object X, $\mathbf{C}(X, A)$ is a cpo with $\bot$,

(ii) for every morphism $g : X \to Y$, the map $\mathbf{C}(g, A) : \mathbf{C}(Y, A) \to \mathbf{C}(X, A)$ with $f \mapsto fg$ is continuous,

(iii) the map $f \mapsto a\,(\mathsf{id} \# f)$ is a continuous map on $\mathbf{C}(X, A)$.

Then clearly for every $e : X \to A \# X$ the least solution $e^\dagger$ exists; indeed, the map $s \mapsto a\,(\mathsf{id} \# s)e$ is continuous on $\mathbf{C}(X, A)$. And the assignment $e \mapsto e^\dagger$ of a least solution turns A into a complete Elgot $\#$-algebra. The proof of this fact is identical to the proof for the previous example.

Note that if $\mathbf{C} = \mathbf{Set}$ and A is a cpo with $\bot$ then $\mathbf{C}(X, A)$ is equipped with the pointwise cpo structure and then conditions (i) and (ii) follow automatically.

For illustrative purposes we proceed to describe one concrete instance of this scenario. Let $X \# Y = X + Y \times Y$ on $\mathbf{C} = \mathbf{Set}$. Let S be a set and let $S' = S + \{0, \bot\}$. Then

$$(S', \mathsf{seq\_or} : S' + S' \times S' \to S')$$

is a $\#$-algebra under the following assignments:

$$\mathsf{seq\_or}(x) = x \qquad \mathsf{seq\_or}(\bot, x) = \bot \qquad \mathsf{seq\_or}(s, x) = s \qquad \mathsf{seq\_or}(0, x) = x$$

where $s \in S$. Moreover, S' is equipped with the flat cpo structure, i.e. $x \sqsubseteq y$ iff $x = \bot$ or $x = y$, and $\mathsf{seq\_or}$ is continuous. Then all hom-sets $\mathbf{C}(X, S')$ are, of course, cpos with $\bot$ under the pointwise order, and it is then easy to see that our three conditions above are satisfied; for condition (iii) one uses that $S' + S' \times S'$ is also a cpo (without bottom) and that $\mathsf{seq\_or}$ is clearly continuous. Thus S' is a complete Elgot $\#$-algebra.

Suppose we have a predicate $p : S \to 2$ on S and a function $f : X \to S + X \times X$ representing a graph over the set of nodes X (where every vertex has either two outgoing transitions or none and is labeled in S). Let $p? : S \to S'$ be defined by $p?(s) = s$ if $p(s) = 1$ and $p?(s) = 0$ otherwise and consider the map

$$g = \Big(X \xrightarrow{f} S + X \times X \xrightarrow{p? + \mathsf{id}} S' + X \times X\Big).$$

Since $g : X \to S' \# X$, we obtain the function $g^\dagger : X \to S'$, which performs the *depth-first search* of the first element of S satisfying p starting for a given vertex. The results from S' are to be interpreted as follows: $s \in S$ is returned if the element is found, 0 if the element is not found, $\bot$ indicates the divergence.

We revisit these definitions in Example 5.8 after giving a characterization of complete Elgot algebras.

Note that we did not require a morphism of complete Elgot $\#$-algebras to be a morphism of $\#$-algebras. In fact, this follows automatically.

Proposition 4.11 *Let $f : A \to B$ be a complete Elgot $\#$-algebra morphism from $(A, a, \text{-}^{\dagger})$ to $(B, b, \text{-}^{\ddagger})$. Then f is a morphism of $\#$-algebras.*

Proof (Sketch). The idea is to represent a as a loop terminating after the first iteration and then deduce preservation of a by f from preservation of iteration by f guaranteed by definition. More concretely, we take

$$e = (\mathsf{id} \# \mathsf{inr})\,[\mathsf{id}, u_A^A] : (A \# A) + A \to A \# ((A \# A) + A)$$

and show that $e^{\dagger} = [a, \mathsf{id}]$. The remaining proof amounts to deriving $b\,(f \# f) = f\,a$ from $f\,e^{\dagger} = ((f \# \mathsf{id})\,e)^{\ddagger}$. □

5 Complete Elgot Algebras as Algebras for a Monad

In this section we show that complete Elgot $\#$-algebras can be recognized as precisely Eilenberg-Moore algebras of the monad of generalized coalgebraic resumptions on $\#$, which we introduce below.

Recall that it was shown by Uustalu [28] that parametrized monads give rise to monads at least in two different ways:

Proposition 5.1 *Suppose $\#$ is a parametrized monad on $\mathbf{C}$ such that the least fixpoint $\mu\gamma.\, X \# \gamma$ (the greatest fixpoint $\nu\gamma.\, X \# \gamma$) exists for every $X \in |\mathbf{C}|$. Then $\mu\gamma.\text{–} \#\gamma$ ($\nu\gamma.\text{–} \#\gamma$) is the underlying functor of a monad.*

Remark 5.2 For the parametrized monad $X \# Y = T(X + \Sigma Y)$ it is well-known that

$$T_{\Sigma}^{\mu} X = \mu\gamma.\, T(X + \Sigma\gamma)$$

is the object mapping of a monad $\mathbb{T}_{\Sigma}^{\mu}$ (in fact, $\mathbb{T}_{\Sigma}^{\mu}$ is the coproduct of the monad $\mathbb{T}$ amd the free monad on Σ, see [17]). In the following we shall mostly be interested in the case where μ is replaced by ν, i.e. the monad $\mathbb{T}_{\Sigma}$ of (★).

It is known that the initial algebra $\mu\gamma.\ X \# \gamma$ carries the free $\#$-algebra on X; conversely, the free $\#$-algebra is an initial $(X \# \text{–})$-algebra (see [6, Theorem 2.18]). Here we are interested in the final coalgebras $\nu\gamma.\, X \# \gamma$. One of the goals of this section is to establish that the final $(X \#\text{–})$-coalgebra carries the free complete Elgot $\#$-algebra on X, and conversely, assuming a free complete Elgot $\#$-algebra on X, its carrier is a final $(X \# \text{–})$-coalgebra (see Corollary 5.10).

From now on we assume that the final coalgebras $\nu\gamma.\, X \# \gamma$ exist and denote them $F_{\#}X$ (standardly omitting the structure morphisms $\mathsf{out}_X : F_{\#}X \to X \# F_{\#}X$). Recall that $\mathsf{coit}\, f : X \to F_{\#}Y$ is the unique final morphism induced by a coalgebra $(X, f : X \to Y \# X)$. Following [28], in order to introduce and reason about the monad structure of $F_{\#}$, we use a more flexible *primitive corecursion principle*, derived from the standard *coiteration principle* embodied in coit.

Proposition 5.3 ([28]) *For any endofunctor F with a final coalgebra νF, and any $f : X \to F(\nu F + X)$, there is a unique morphism $h : X \to \nu F$ satisfying $\mathsf{out}\, h = F[\mathsf{id}, h]\, f$.*

The morphism h in Proposition 5.3 is said to be defined by primitive corecursion. We use primitive corecursion to slightly generalize the coit construct in the special case of $\mathcal{F}_\#$:

Lemma 5.4 *For any $e : X \to B \# X$ and $f : B \to A \# \mathcal{F}_\# A$, there is a unique morphism h satisfying*

$$\begin{tikzcd} X \arrow[r, "e"] \arrow[d, "h"'] & B \# X \arrow[d, "m_A^{\mathcal{F}_\# A}\,(f \# h)"] \\ \mathcal{F}_\# A \arrow[r, "\mathsf{out}"] & A \# \mathcal{F}_\# A. \end{tikzcd} \tag{1}$$

For any $e : X \to B \# X$ and $f : B \to A \# \mathcal{F}_\# A$ we denote by

$$\mathsf{coit}(e, f) : X \longrightarrow \mathcal{F}_\# A$$

the unique h making diagram (1) commute. Using (1), the monad structure on $\mathcal{F}_\#$ can be given as follows:

$$\begin{aligned} \eta^\nu_X &= \mathsf{out}^{\text{-}1}\, u_X^{\mathcal{F}_\# X} = \mathsf{coit}\, u_X^X \\ f^\star &= \mathsf{coit}((f \# \mathsf{id})\ \mathsf{out}, \mathsf{out}) \qquad\qquad \text{where } f : X \to \mathcal{F}_\# Y \end{aligned}$$

This also defines $\mu^\nu = \mathsf{id}^\star = \mathsf{coit}(\mathsf{out}, \mathsf{out})$. Note that, by Lemma 5.4, $f^\star$ is the unique morphism satisfying the equation

$$\mathsf{out}\, f^\star = m_Y^{\mathcal{F}_\# Y}\, (\mathsf{out}\, f \# f^\star)\ \mathsf{out}\,. \tag{2}$$

Lemma 5.5 *Let $e : X \to B \# X$ and $f : B \to A \# \mathcal{F}_\# A$. Then*

$$\mathsf{coit}(e, f) = (\mathsf{out}^{\text{-}1}\, f)^\star\, (\mathsf{coit}\, e).$$

As an easy corollary of Lemma 5.5 we obtain that $\mathsf{coit}\, e = \mathsf{coit}(e, u_X^{\mathcal{F}_\# X})$; indeed, we have

$$\mathsf{coit}(e, u_X^{\mathcal{F}_\# X}) = (\mathsf{out}^{\text{-}1}\, u_X^{\mathcal{F}_\# X})^\star\, (\mathsf{coit}\, e) = (\eta^\nu_X)^\star\, (\mathsf{coit}\, e) = \mathsf{coit}\, e.$$

We state another useful property in the following lemma:

Lemma 5.6 *Let $e : X \to B \# X$ and $g : B \to C$. Then*

$$\mathcal{F}_\# g\, (\mathsf{coit}\, e) = \mathsf{coit}((g \# \mathsf{id})\, e).$$

The following theorem is our first main result. It establishes an equivalence of complete Elgot $\#$-algebras and $\mathcal{F}_\#$-algebras.

Theorem 5.7 *For any parametrized monad $\# : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$, the Eilenberg-Moore algebras of $\mathcal{F}_\# = \nu\gamma.\, - \# \gamma$ are exactly the complete Elgot $\#$-algebras. More precisely, $\mathbf{C}^{\mathcal{F}_\#}$ and $\mathbf{CElg}_\#(\mathbf{C})$ are isomorphic categories, witnessed by the following construction (in both directions, morphisms are mapped to themselves):*

- $\mathbf{C}^{\mathcal{F}_\#} \to \mathbf{CElg}_\#(\mathbf{C})$*: for a $\mathcal{F}_\#$-algebra $(A, \chi : \mathcal{F}_\# A \to A)$ we define a $\#$-algebra $(A, \chi\, \mathsf{out}^{-1}\, (\mathsf{id} \# \eta^\nu) : A \# A \to A, -^\dagger)$ with $e^\dagger = \chi\, (\mathsf{coit}\, e) : X \to A$ for any $e : X \to A \# X$.*
- $\mathbf{CElg}_\#(\mathbf{C}) \to \mathbf{C}^{\mathcal{F}_\#}$*: for a $\#$-algebra $(A, a : A \# A \to A, -^\dagger)$ we define an $\mathcal{F}_\#$-algebra $(A, \mathsf{out}^\dagger : \mathcal{F}_\# A \to A)$.*

Proof (Sketch). For the direction from $\mathbf{C}^{\mathcal{F}_\#}$ to $\mathbf{CElg}_\#(\mathbf{C})$ we have to verify the axioms of complete Elgot $\#$-algebras. The hardest case is that of the compositionality identity. We have on the one hand

$$\begin{aligned}
(f^\dagger \bullet g)^\dagger &= \chi\ \mathsf{coit}(f^\dagger \bullet g)\\
&= \chi\ \mathsf{coit}\,((\chi\,(\mathsf{coit}\, f) \# \mathsf{id})\, g)\\
&= \chi\ \mathsf{coit}\,((\chi \# \mathsf{id})\,((\mathsf{coit}\, f) \# \mathsf{id})\, g)\\
&= \chi\,(\mathcal{F}_\#\chi)\ \mathsf{coit}(((\mathsf{coit}\, f) \# \mathsf{id})\, g) && \text{// Lemma 5.6}\\
&= \chi\, \mu^\nu\, \mathsf{coit}(((\mathsf{coit}\, f) \# \mathsf{id})\, g) && \text{// } \chi \text{ is an } \mathcal{F}_\#\text{-algebra}\\
&= \chi\ \mathsf{coit}(((\mathsf{coit}\, f) \# \mathsf{id})\, g, \mathsf{out}), && \text{// Lemma 5.5}
\end{aligned}$$

and on the other hand, by definition,

$$(f \blacksquare g)^\dagger\ \mathsf{inr} = \chi\ \mathsf{coit}(m_A^{Y+X}\ (((\mathsf{id} \# \mathsf{inl})\, f) \# \mathsf{inr})\, [u_Y^X, g])\ \mathsf{inr}\,.$$

Let us denote $m_A^{Y+X}\ (((\mathsf{id} \# \mathsf{inl})\, f) \# \mathsf{inr})\, [u_Y^X, g]$ by h. By Lemma 5.4, it suffices to show the identity $\mathsf{out}\,(\mathsf{coit}\, h)\ \mathsf{inr} = m_A^{\mathcal{F}_\# A}\ (\mathsf{out} \#((\mathsf{coit}\, h)\ \mathsf{inr}))\,(\mathsf{coit}\, f \# \mathsf{id})\, g$. The latter is easy to obtain from the auxiliary equation $(\mathsf{coit}\, h)\ \mathsf{inl} = \mathsf{coit}\, f$ whose proof is a routine.

For the direction from $\mathbf{CElg}_\#(\mathbf{C})$ to $\mathbf{C}^{\mathcal{F}_\#}$, we have to prove the two axioms of Eilenberg-Moore algebras. The harder one is $\mathsf{out}^\dagger\ \mathcal{F}_\#(\mathsf{out}^\dagger) = \mathsf{out}^\dagger\ \mu^\nu$ and it is obtained from the instance of compositionality $(\mathsf{out} \blacksquare \mathsf{out})^\dagger\, \mathsf{inr} = (\mathsf{out}^\dagger \bullet \mathsf{out})^\dagger$ by establishing $\mathsf{out}^\dagger\, [\mathsf{id}, \mu^\nu] = (\mathsf{out} \blacksquare \mathsf{out})^\dagger$ and $\mathsf{out}^\dagger\ \mathcal{F}_\#(\mathsf{out}^\dagger) = (\mathsf{out}^\dagger \bullet \mathsf{out})^\dagger$. Further calculations ensure that the correspondence between $\mathbf{CElg}_\#(\mathbf{C})$ and $\mathbf{C}^{\mathcal{F}_\#}$ is functorial and moreover an isomorphism. □

Let us illustrate Theorem 5.7 by revisiting Example 4.10.

Example 5.8 Recall that we consider the parametrized monad $X \# Y = X + Y \times Y$ so that $\mathcal{F}_\# X$ is the set of finite and infinite binary trees whose leaves are labeled in X. The fact that $(S', \mathsf{seq\_or} : S' + S' \times S' \to S', -^\dagger)$ is a $\#$-algebra means, equivalently, that S' is an $\mathcal{F}_\#$-algebra. In particular, the $\mathcal{F}_\#$-algebra structure is a function $\mathsf{Seq\_or} : \mathcal{F}_\# S' \to S'$ that transforms a given binary tree over S' to a single element of S' calculated by using the depth-first search strategy seeking the first leaf of the given tree that is labeled by S: In case of success, the answer is in S, otherwise the answer is either $0 \in S'$ meaning that no element from S was found and $\bot$ in case of divergence (i.e. the procedure arrived in an infinite branch before any element from S was detected).

Note further that any function $f : X \to S + X \times X = S \# X$ represents a graph as explained in Example 4.10. The unique map $\mathsf{coit}\, f : X \to \mathsf{F}_{\#} S$ into the final coalgebra then computes for every node $x \in X$ its tree unfolding. Now, starting with a predicate $p : S \to 2$, the function $g^\dagger : X \to S'$ defined in Example 4.10 is equal to the composition

$$\begin{tikzcd} X \arrow[r, "\mathsf{coit}\, f"] & \mathsf{F}_{\#} S \arrow[r, "\mathsf{F}_{\#}(p?)"] & \mathsf{F}_{\#} S' \arrow[r, "\mathsf{Seq\_or}"] & S'. \end{tikzcd}$$

Indeed, by Theorem 5.7, $g^\dagger = \mathsf{Seq\_or}\,(\mathsf{coit}\, g) = \mathsf{Seq\_or}\ \mathsf{coit}((p? + \mathsf{id})\, f)$ and the rest follows by Lemma 5.6.

We proceed with the goal of showing that the final coalgebras $\nu\gamma.\, X \# \gamma$ are, equivalently, the free complete Elgot $\#$-algebras. In fact, given a final $(X \# \text{–})$-coalgebra $\mathsf{out}_X : \mathsf{F}_{\#} X \to X \# \mathsf{F}_{\#} X$, it follows from (the proof of) Theorem 5.7 that $\mathsf{F}_{\#} X$ carries the free $\mathsf{F}_{\#}$-algebra on X, and therefore it carries the free complete Elgot $\#$-algebra on X (because the isomorphism of categories preserves freeness of algebras). It is not difficult to work out that the following morphisms

$$\begin{tikzcd} \mathsf{F}_{\#} X \# \mathsf{F}_{\#} X \arrow[r, "\mathsf{out}_X \# \mathsf{id}"] & (X \# \mathsf{F}_{\#} X) \# \mathsf{F}_{\#} X \arrow[r, "m_X^{\mathsf{F}_{\#} X}"] & X \# \mathsf{F}_{\#} X \arrow[r, "\mathsf{out}_X^{-1}"] & \mathsf{F}_{\#} X \end{tikzcd}$$

and

$$\begin{tikzcd} X \arrow[r, "u_X^{\mathsf{F}_{\#} X}"] & X \# \mathsf{F}_{\#} X \arrow[r, "\mathsf{out}_X^{-1}"] & \mathsf{F}_{\#} X \end{tikzcd}$$

form the algebra structure and universal morphism of a free complete Elgot algebra for $\#$ on X. The iteration operator on $\mathsf{F}_{\#} Y$ is obtained as follows. Given $e : X \to \mathsf{F}_{\#} Y \# X$ one forms the following coalgebra $c : \mathsf{F}_{\#} Y + X \to Y \# (\mathsf{F}_{\#} Y + X)$ for $Y \# \text{–}$:

$$\begin{tikzcd} \mathsf{F}_{\#} Y + X \arrow[r, "{[u^X_{\mathsf{F}_{\#} Y}, e]}"] & \mathsf{F}_{\#} Y \# X \arrow[r, "\mathsf{out} \# \mathsf{id}"] & (Y \# \mathsf{F}_{\#} Y) \# X \arrow[d, "(\mathsf{id} \# \mathsf{inl}) \# \mathsf{inr}"] \\ Y \# (\mathsf{F}_{\#} Y + X) & & (Y \# (\mathsf{F}_{\#} Y + X)) \# (\mathsf{F}_{\#} Y + X) \arrow[ll, "m_Y^{\mathsf{F}_{\#} Y + X}"'] \end{tikzcd}$$

Then one puts $e^\dagger = (\mathsf{coit}\, c)\ \mathsf{inr}$.

Conversely, we have the following result.

Theorem 5.9 *Suppose that $\varphi_X : FX \# FX \to FX$ and $\eta_X : X \to FX$ form a free complete Elgot $\#$-algebra on X. Then*

$$\begin{tikzcd} X \# FX \arrow[r, "\eta_X \# \mathsf{id}"] & FX \# FX \arrow[r, "\varphi_X"] & FX \end{tikzcd}$$

is an isomorphism, and its inverse is the structure of a final $(X \# \text{–})$-coalgebra.

The proof of the above bijective correspondence between final $(X \# -)$-coalgebras and free complete Elgot $\#$-algebras is a non-trivial generalization of the proof of [5, Theorem 5.4] from complete Elgot algebras for endofunctors to those for parametrized monads; here we have seen one direction of the bijective correspondence as a consequence of Theorem 5.7 while we outline the proof of Theorem 5.9 in the full version of the paper.

Corollary 5.10 *A free complete Elgot $\#$-algebra on X is equivalently a final coalgebra for $X \# -$.*

To conclude the present section, we show that surprisingly, in any free complete Elgot $\#$-algebra FY the iteration operator always assigns a unique solution to any morphism $e : Y \to FY \# Y$.

Proposition 5.11 *Suppose that $\varphi_Y : FY \# FY \to FY$ and $\eta_Y : Y \to FY$ form a free complete Elgot $\#$-algebra on Y. Then for every $e : X \to FY \# X$, $e^\dagger : X \to FY$ is a unique solution, i.e. a unique morphism satisfying the solution axiom with e.*

Proof. Recall first from Theorem 5.9 that FY is (equivalently) a final $(Y \# -)$-coalgebra with the structure $t : FY \to Y \# FY$ obtained as an inverse of

$$Y \# FY \xrightarrow{\eta_Y \# \mathsf{id}} FY \# FY \xrightarrow{\varphi_Y} FY.$$

Let $e : X \to FY \# X$ and consider the following $(Y \# -)$-coalgebra

$$\begin{tikzcd}
\bar{e} = (FY \# X \arrow[r, "t \# e"] & (Y \# FY) \# (FY \# X) \arrow[d, "(\mathsf{id} \# u^X_{FY}) \# (\mathsf{id} \# \mathsf{id})"] & \\
& (Y \# (FY \# X)) \# (FY \# X) \arrow[r, "m_Y^{FY \# X}"] & Y \# (FY \# X)).
\end{tikzcd}$$

Now let $d : X \to FY$ be any solution of e, i.e. we have $d = \varphi_Y(FY \# d)e$. We will prove below that $\varphi_Y(FY \# d) : FY \# X \to FY$ is a coalgebra homomorphism from $\bar{e}$ to t. Since $\bar{e}$ does not depend on the solution d we then conclude that

$$e^\dagger = \varphi_Y(\mathsf{id} \# e^\dagger)e = \varphi_Y(\mathsf{id} \# d)e = d$$

using finality of FY in the middle step.

To finish the proof consider the following diagram:

$$\begin{tikzcd}
FY \# X \arrow[r, "\mathsf{id} \# d"] \arrow[dd, "t \# e"] & FY \# FY \arrow[r, "\varphi_Y"] \arrow[d, "t \# \mathsf{id}"'] & FY \arrow[dd, "t"] \\
& (Y \# FY) \# FY \arrow[rd, bend left, "m_Y^{FY}"] & \\
(Y \# FY) \# (FY \# X) \arrow[r, "\mathsf{id} \# (\mathsf{id} \# d)"] \arrow[d, "(\mathsf{id} \# u^X_{FY}) \# \mathsf{id}"] & (Y \# FY) \# (FY \# FY) \arrow[u, "\mathsf{id} \# \varphi_Y"'] \arrow[d, "(\mathsf{id} \# u^{FY}_{FY}) \# \mathsf{id}"'] & Y \# FY \arrow[dd, equal] \\
(Y \# (FY \# X)) \# (FY \# X) \arrow[r, "(\mathsf{id} \# (\mathsf{id} \# d)) \# (\mathsf{id} \# d)"'] \arrow[d, "m_Y^{FY \# X}"] & (Y \# (FY \# FY)) \# (FY \# FY) \arrow[uu, bend right=60, "(\mathsf{id} \# \varphi_Y) \# \varphi_Y"'] \arrow[d, "m_Y^{FY \# FY}"'] & \\
Y \# (FY \# X) \arrow[r, "\mathsf{id} \# (\mathsf{id} \# d)"'] & Y \# (FY \# FY) \arrow[r, "\mathsf{id} \# \varphi_Y"'] & Y \# FY
\end{tikzcd}$$

Note first that the left-hand edge is $\bar{e}$. The upper left-hand square commutes since d is a solution of e, for the part below it use that $(\mathsf{id} \# d)\, u^X_{FY} = u^{FY}_{FY}$ holds since $\mathsf{id} \# d$ is a monad morphism, and the lower left- and right-hand part commute by the laws of $\#$. That the upper-right hand part commutes follows from the proof of Theorem 5.9, and the remaining little inner part commutes since $\varphi_Y\, u^{FY}_{FY} = \mathsf{id}_{FY}$, which holds because φ_Y is the structure of a $\#$-algebra. Hence $\varphi_Y\,(\mathsf{id} \# d)$ is a coalgebra homomorphisms as desired, which completes the proof. □

6 Algebras of Complete Elgot Monads

We are now in a position to apply the results on complete Elgot $\#$-algebras developed in the previous sections to explore the connection between complete Elgot monads and complete Elgot algebras.

Recall that given a monad $\mathbb{T}$ and an endofunctor Σ over $\mathbf{C}$, $X \# Y = T(X + \Sigma Y)$ is a parametrized monad and therefore, by Proposition 5.1, $\mathbb{T}_\Sigma$ given by ($\bigstar$) is a monad. We reserve notation $\mathbb{T}_\nu$ for the special case when $\Sigma = \mathsf{Id}$:

$$T_\nu X = \nu\gamma.\, T(X + \gamma).$$

From a computational point of view, $T_\nu X$ can be considered as a type of processes triggering a computational effect formalized by $\mathbb{T}$ at each step and eventually outputting values from X in case of successful termination. The unary operation captured by $\Sigma = \mathsf{Id}$ can be understood as *delaying*. This perspective was previously pursued in [15]. Now, if $\mathbb{T}$ is a complete Elgot monad, or more generally, any monad equipped with an iteration operator, we can define a *collapsing morphism* $\delta_X : T_\nu X \to TX$ as follows:

$$\delta_X = \Big(T_\nu X \xrightarrow{\mathsf{out}_X} T(X + T_\nu X)\Big)^\dagger, \tag{3}$$

which intuitively flattens every possibly infinite sequence of computational steps of $T_\nu X$ into a single step of TX. Let us illustrate this with the following toy example.

Example 6.1 Let $TX = \mathcal{P}_{\omega_1}(A^\star \times X)$ where $\mathcal{P}_{\omega_1}$ is the countable powerset functor and A is some fixed alphabet of *actions* like in Example 1.1. We extend T to a monad $\mathbb{T}$ by putting

$$\eta_X(x) = \{(\varepsilon, x)\} \quad \text{and} \quad f^\star(s \subseteq A^\star \times X) = \{(ww', y) \mid (w, x) \in s, (w', y) \in f(x)\},$$

where $\varepsilon \in A^\star$ is the empty word and $f : X \to \mathcal{P}_{\omega_1}(A^\star \times Y)$. It is easy to see that $\mathbb{T}$ is an ω-continuous monad (see Example 3.2) and hence a complete Elgot monad with the iteration operator defined using least fixed points. An element of TX is intuitively a countably branching process, with results in X, at each step capable of executing a finite series of actions. Now the collapsing morphism (3) for every process $p \in T_\nu\{\checkmark\}$ calculates the set $\mathsf{tr}(p) \subseteq A^\star$ of all sucessful traces of p.

As we will see later (Theorem 6.4 (i)), $\big(TX,\ T_\nu TX \xrightarrow{\delta_{TX}} TTX \xrightarrow{\mu_X} TX\big)$ is a $\mathbb{T}_\nu$-algebra and hence, by Theorem 5.7, a complete Elgot $\#$-algebra. Hence, for a complete Elgot monad $\mathbb{T}$, its free algebras are complete Elgot $\#$-algebras. Our next question concerns the converse: Is it possible to equip a given monad $\mathbb{T}$ with an iteration operator provided that free $\mathbb{T}$-algebras are equipped with structures of complete Elgot $\#$-algebras in a coherent way? It turns out that without any further assumptions on the category of complete Elgot $\#$-algebras almost all laws of complete Elgot monads become derivable. More precisely, we introduce the following class of monads.

Definition 6.2 *A monad $\mathbb{T}$ is called a* weak complete Elgot monad *if it is equipped with an iteration operator $\mathord{-}^\dagger$ that satisfies* fixpoint*,* naturality*, and* uniformity *axioms and the following identity: for any $g : X \mathrel{-\!\!\diamond\!\!\to} Y + X$, $f : Y \mathrel{-\!\!\diamond\!\!\to} Z + Y$ we have*

$$\Big(Y + X \xrightarrow{[\mathsf{inl}, g]} Y + X \xrightarrow{f + \mathsf{id}} Z + Y + X\Big)^\dagger \mathsf{inr} = X \xrightarrow{g^\dagger} Y \xrightarrow{f^\dagger} Z. \tag{4}$$

(See Fig. 3 for the pictorial form.)

It is relatively easy to deduce (4) from the codiagonal identity, hence we obtain

Proposition 6.3 *Any complete Elgot monad is a weak complete Elgot monad.*

We now can establish a tight connection between weak complete Elgot monads and complete Elgot $\#$-algebras.

Theorem 6.4 *Let $\mathbb{T}$ be a monad on $\mathbf{C}$ and let $X \# Y = T(X + Y)$.*

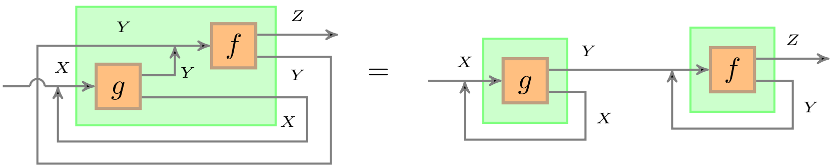

Fig. 3. The additional axiom for weak complete Elgot monads.

(i) *If* $\mathbb{T} = (T, \eta, \mathord{-}^{\star}, \mathord{-}^{\dagger})$ *is a weak complete Elgot monad then* $\mathbf{C}^{\mathbb{T}}$ *is isomorphic to the full subcategory of* $\mathbf{CElg}_{\#}(\mathbf{C})$ *formed by those complete Elgot* $\#$*-algebras* $(A, a : T(A + A) \to A, \mathord{-}^{\ddagger})$ *which factor through* $T\nabla : T(A + A) \to TA$ *and for which* $e^{\ddagger} = a\,(T\,\mathsf{inl})\,e^{\dagger}$ *for every* $e : X \to T(A + X)$*.*

(ii) *Conversely, any functor* $J : \mathbf{C}^{\mathbb{T}} \to \mathbf{CElg}_{\#}(\mathbf{C})$ *sending a* $\mathbb{T}$*-algebra* $a : TA \to A$ *to* $a\,(T\nabla) : T(A+A) \to A$ *and identical on morphisms induces a weak complete Elgot monad structure on* $\mathbb{T}$ *as follows:*

$$\frac{e : X \to T(Y + X)}{e^{\dagger} = (T(\eta + \mathsf{id})\,e)^{\ddagger} : X \to TY} \tag{5}$$

where $\mathord{-}^{\ddagger}$ *is the iteration operator on* $J(TY, \mu)$ *(by Clause (i),* J *is then full and faithful).*

Remark 6.5 Note that Theorem 6.4(i) can be seen as an analogue of Corollary 4.6 for complete Elgot $\#$-algebras.

If $\mathbf{CElg}_{\#}(\mathbf{C})$ additionally satisfies a variant of the codiagonal identity, the construction from Clause (ii) of Theorem 6.4 produces precisely complete Elgot monads.

Theorem 6.6 *Let* $\mathbb{T}$ *be a monad on* $\mathbf{C}$*, let* $X \# Y = T(X + Y)$ *and let* $J : \mathbf{C}^{\mathbb{T}} \to \mathbf{CElg}_{\#}(\mathbf{C})$ *be a functor as in Clause (ii) of Theorem 6.4. Then* $\mathbb{T}$ *is equipped with the structure of a weak complete Elgot monad given by* (5)*, and moreover* $\mathbb{T}$ *is a complete Elgot monad iff every* $(A, a, \mathord{-}^{\ddagger})$ *in the image of* J *satisfies the equations*

$$(m^{X}_{A \# X}\, e)^{\ddagger} = (e^{\ddagger})^{\ddagger} \tag{6}$$

for every $e : X \to (A \# X) \# X$ *(this uses the fact that* $A \# X = T(A + X)$ *is a free* $\mathbb{T}$*-algebra and hence a complete Elgot* $\#$*-algebra).*

7 Conclusions and Further Work

We introduced the notion of complete Elgot algebra for a parametrized monad, based on the previous work [4, 28]. We showed that the category of complete Elgot algebras for a parametrized monad $\#$ is isomorphic to the category of Eilenberg-Moore algebras for the monad $\nu\gamma.\, \mathord{-} \# \gamma$ whenever the latter exists. As the category of complete Elgot $\#$-algebras is given axiomatically, this can be considered as a form of soundness and completeness result, specifically, it indicates that algebras for $\nu\gamma.\, \mathord{-} \# \gamma$ are subject to a lightweight theory of (uniform) iteration.

We explored the connection between complete Elgot $\#$-algebras for $X \# Y = T(X+Y)$ and Eilenberg-Moore algebras of complete Elgot monads, i.e. monads from [14] supporting a uniform iteration operator satisfying standard axioms of iteration. Specifically, we showed that monads $\mathbb{T}$ whose algebras are coherently equipped with the structure of a complete Elgot $\#$-algebra are precisely complete Elgot monads with the codiagonal axiom replaced by its weakened form (Theorem 6.4). Moreover, if the category of complete Elgot $\#$-algebras satisfies a variant of the codiagonal law, such monads $\mathbb{T}$ are complete Elgot monads (Theorem 6.6).

As an open problem we leave the question whether assumption (6) on complete Elgot algebras in Theorem 6.6 can be lifted. If this was the case, then the notions of weak complete Elgot monads and complete Elgot monads would be equivalent.

We believe that the results we obtained are potentially useful for facilitating constructions over complete Elgot monads, in particular we seek a conceptual simplification for the sophisticated proofs underlying the main result of [14] stating that ($\bigstar$) is a complete Elgot monad whenever $\mathbb{T}$ is. Also we are interested in applications of our results to semantics of abstract side-effecting processes in the style of [15] under equivalences coarser than the behavioral equivalence.

References

[1] "Haskell 98 Language and Libraries — The Revised Report," Cambridge University Press, 2003, also: J. Funct. Prog. **13** (2003).

[2] Aczel, P., J. Adámek, S. Milius and J. Velebil, *Infinite trees and completely iterative theories: a coalgebraic view*, Theoretical Computer Science **300** (2003), pp. 1–45.

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

[4] Adámek, J., S. Milius and J. Velebil, *Iterative algebras for a base*, Electr. Notes Theor. Comput. Sci. **122** (2005), pp. 147–170.

[5] Adámek, J., S. Milius and J. Velebil, *Elgot algebras*, Log. Methods Comput. Sci. **2** (2006), pp. 1–31.

[6] Adámek, J., S. Milius and J. Velebil, *Bases for parametrized iterativity*, Inform. and Comput. **206** (2008), pp. 966–1002.

[7] Adámek, J., S. Milius and J. Velebil, *Equational properties of iterative monads*, Information and Computation **208** (2010), pp. 1306 – 1348.

[8] Adámek, J., S. Milius and J. Velebil, *Elgot theories: a new perspective of the equational properties of iteration*, Math. Structures Comput. Sci. **21** (2011), pp. 417–480.

[9] Benton, N. and M. Hyland, *Traced premonoidal categories*, ITA **37** (2003), pp. 273–299.

[10] Bloom, S. L. and Z. Ésik, "Iteration theories: the equational logic of iterative processes," Springer-Verlag New York, Inc., New York, NY, USA, 1993.

[11] Capretta, V., *General recursion via coinductive types*, Logical Methods in Computer Science **1** (2005).

[12] Elgot, C. C., *Monadic computation and iterative algebraic theories**, in: H. Rose and J. Shepherdson, editors, *Logic Colloquium '73Proceedings of the Logic Colloquium*, Studies in Logic and the Foundations of Mathematics **80**, Elsevier, 1975 pp. 175–230.

[13] Ésik, Z. and S. Goncharov, *Some remarks on Conway and iteration theories* (2016), arXiv preprint: http://arxiv.org/abs/1603.00838.

[14] Goncharov, S., C. Rauch and L. Schröder, *Unguarded recursion on coinductive resumptions*, in: *Proc. Mathematical Foundations of Programming Semantics XXXI, MFPS 2015*, ENTCS, 2015.

[15] Goncharov, S. and L. Schröder, *A coinductive calculus for asynchronous side-effecting processes*, Information and Computation **231** (2013), pp. 204 – 232.

[16] Hasegawa, M., *Recursion from cyclic sharing: Traced monoidal categories and models of cyclic lambda calculi*, in: *Proc. 3rd International Conference on Typed Lambda Calculi and Applications*, Lecture Notes Comput. Sci. **1210** (1997), pp. 196–213.

[17] Hyland, M., G. Plotkin and J. Power, *Combining effects: Sum and tensor*, Theoret. Comput. Sci. **357** (2006), pp. 70–99.

[18] Joyal, A., R. Street and D. Verity, *Traced monoidal categories*, Mathematical Proceedings of the Cambridge Philosophical Society **119** (1996), pp. 447–468.

[19] Kelly, G., *A unified treatment of transfinite constructions for free algebras, free monoids, colimits, associated sheaves, and so on*, Bulletin of the Australian Mathematical Society **22** (1980), pp. 1–83.

[20] Lawvere, W., *Functorial semantics of algebraic theories*, Proc. Natl. Acad. Sci. USA **50** (1963), pp. 869–872.

[21] Mac Lane, S., “Categories for the Working Mathematician,” Springer, 1971.

[22] Moggi, E., *Notions of computation and monads*, Inf. Comput. **93** (1991), pp. 55–92.

[23] Plotkin, G. and J. Power, *Adequacy for algebraic effects*, in: *FoSSaCS'01*, LNCS **2030**, 2001, pp. 1–24.

[24] Plotkin, G. and J. Power, *Notions of computation determine monads*, in: *FoSSaCS'02*, LNCS **2303** (2002), pp. 342–356.

[25] Rutten, J., *Universal coalgebra: a theory of systems*, Technical report, Amsterdam, The Netherlands, The Netherlands (1996).

[26] Rutten, J. and D. Turi, *Initial algebra and final coalgebra semantics for concurrency*, Springer-Verlag, 1994 pp. 530–582.

[27] Simpson, A. and G. Plotkin, *Complete axioms for categorical fixed-point operators*, in: *In Proceedings of 15th Annual Symposium on Logic in Computer Science*, 2000, pp. 30–41.

[28] Uustalu, T., *Generalizing substitution*, ITA **37** (2003), pp. 315–336.