Available online at www.sciencedirect.com
SCIENCE DIRECT®
Electronic Notes in Theoretical Computer Science 155 (2006) 87–109
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Elgot Algebras† (Extended Abstract)

Jiří Adámek[1,2] Stefan Milius[2]

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

Jiří Velebil[1,3]

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

Abstract

Iterative algebras, i. e., algebras A in which flat recursive equations e have unique solutions $e^\dagger$, are generalized to Elgot algebras, where a choice $e \longmapsto e^\dagger$ of solutions of all such equations e is specified. This specification satisfies two simple and well motivated axioms: functoriality (stating that solutions are "uniform") and compositionality (stating how to perform simultaneous recursion). These two axioms stem canonically from Elgot's iterative theories: We prove that the category of Elgot algebras is the Eilenberg–Moore category of the free iterative monad.

Keywords: Elgot algebra, rational monad, coalgebra, iterative theories

If you are not part of the solution,
you are part of the problem.

Eldridge Cleaver, *speech in San Francisco*, 1968

† The full version of this paper containing all proofs can be found at the URL http://www.iti.cs.tu-bs.de/~milius

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

2 Email:{adamek,milius}@iti.cs.tu-bs.de

3 Email:velebil@math.feld.cvut.cz

1571-0661/$ – see front matter
doi:10.1016/j.entcs.2005.11.053

1 Introduction

This paper studies Elgot algebras, a new notion of algebra useful for application in the semantics of recursive computations. In programming, functions are often specified by a *recursive applicative program scheme* such as

$$\begin{aligned} \varphi(x) &\approx F(x, \varphi(Gx)) \\ \psi(x) &\approx F(\varphi(Gx), GGx) \end{aligned} \tag{1.1}$$

where F and G are given functions and φ and ψ are recursively defined in terms of the given ones by (1.1). We are interested in the semantics of such schemes. Actually, one has to distinguish between *uninterpreted* and *interpreted* semantics. In the uninterpreted semantics the givens are not functions but merely function symbols from a signature Σ. In the present paper we prepare a basis for the interpreted semantics in which a program scheme comes together with a suitable Σ-algebra A, which gives an interpretation to all the given function symbols. The actual application of Elgot algebras to semantics will be dealt with in [19]. By "suitable algebra" we mean, of course, one in which recursive program schemes can be given a semantics. For example, for the recursive program scheme (1.1) we are only interested in those Σ-algebras A, where $\Sigma = \{ F, G \}$, in which the program scheme (1.1) has a *solution*, i. e., we can canonically obtain new operations φ^A and ψ^A on A so that the formal equations (1.1) become valid identities. The question we address is:

(1.2) What Σ-algebras are suitable for semantics?

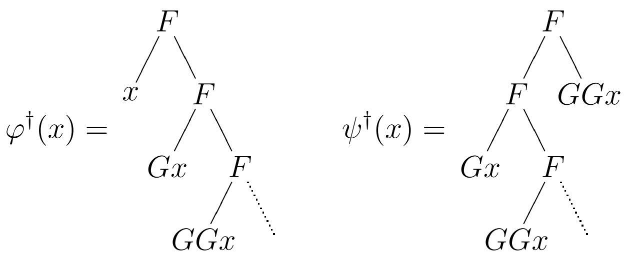
Several answers have been proposed in the literature. One well-known approach is to work with complete posets (CPO) in lieu of sets, see e.g. [14]. Here algebras have an additional CPO structure making all operations continuous. Another approach works with complete metric spaces, see e.g. [6]. Here we have an additional complete metric making all operations contracting. In both of these approaches one imposes extra structure on the algebra in a way that makes it possible to obtain the semantics of a recursive computation as a join (or limit, respectively) of finite approximations.

It was the idea of Calvin Elgot to try and work in a purely algebraic setting avoiding extra structure like order or metric. In [10] he introduced iterative theories which are algebraic theories in which certain systems of recursive equations have *unique* solutions. Later Evelyn Nelson [21] and Jerzy Tiuryn [25] studied iterative algebras, which are algebras for a signature Σ with unique solutions of recursive equations. While avoiding extra structure, these are still not the unifying concept one would hope for, since they do not subsume continuous algebras—least fixed points are typically not unique.

However, analyzing all the above types of algebras we find an interesting common feature which make continuous, metrizable and iterative algebras fit for use in semantics of recursive program schemes: these algebras allow for an interpretation of infinite Σ-trees. Let us make this more precise. For a given signature Σ consider the algebra

$$T_\Sigma X$$

of all (finite and infinite) Σ-trees over X, i. e., rooted ordered trees where inner nodes with n children are labelled by n-ary operation symbols from Σ, and leaves are labelled by constants or elements from X. The algebra $T_\Sigma X$ is the free continuous Σ-algebra on X and also the free metrizable Σ-algebra on X. Consequently, for any continuous or metrizable algebra A we obtain a canonical map $T_\Sigma A \longrightarrow A$ which provides for any Σ-tree over A its result of computation in A. It is then easy to give semantics to recursive program schemes in A. For example, for (1.1) one can simply take the tree unfolding which yields the infinite trees



and then for any argument $x \in A$ compute these infinite trees in A.

Actually, we do not need to be able to compute all infinite trees: all recursive program schemes unfold to *algebraic trees*, see [8] (we mention these in the Summary shortly). Another important subclass are *rational trees*, which are obtained as all solutions of guarded finitary recursive equations. They were characterized in [13] as those Σ-trees having up to isomorphism finitely many subtrees only. We denote by

$$R_\Sigma X$$

the subalgebra of all rational trees in $T_\Sigma X$. With this in mind, we can restate problem (1.2) more formally:

(1.3) What Σ-algebras have a suitable computation of all trees? Or all rational trees?

This means, one further step more formally: what is the largest category of Σ-algebras in which $T_\Sigma X$, or $R_\Sigma X$, respectively, act as free algebras on

X? The answer in case of $T_\Sigma X$ is: complete Elgot algebras. These are Σ-algebras A with an additional operation "dagger" assigning to every system e of recursive equations in A a solution $e^\dagger$. Two (surprisingly simple) axioms are put on $(-)^\dagger$ which stem from the internal structure of $T_\Sigma X$: the functor T_Σ given by $X \longmapsto T_\Sigma X$ is part of a monad in Set, and this is the free completely iterative theory on Σ, as proved in [11]. We will prove that the monadic algebras of this monad (i. e., the Eilenberg–Moore category of T_Σ) is precisely the category of complete Elgot algebras. Basic examples: continuous algebras or metrizable algebras are Elgot algebras. Analogously, the largest category of Σ-algebras in which each $R_\Sigma X$ acts as a free algebra are Elgot algebras. They are defined precisely as the complete Elgot algebras, except that the systems e of recursive equations considered there are required to be finite. For example, every iterative algebra is an Elgot algebra.

Related Work: Solutions of recursive equations are a fundamental part of a number of models of computation, e. g., iterative theories of C. Elgot [10], iteration theories of S. Bloom and Z. Ésik [7], traced monoidal categories of A. Joyal, R. Street and D. Verity [16], fixed-point theories for domains, see S. Eilenberg [9] or G. Plotkin [22], etc. In some of these models the assignment of a solution $e^\dagger$ to a given type of recursive equation e is unique (e. g., in iterative theories every ideal system has a unique solution, or in domains given by a complete metric space there are unique solutions of fixed-point equations, see [6]). The operation $e \longmapsto e^\dagger$ then satisfies a number of equational properties. In other models, e. g., in iteration theories or in the traced cartesian categories, see [15], a specific choice of a solution $e^\dagger$ is assumed, and certain properties (inspired by the models with unique solutions) are formulated as axioms.

The approach of the present paper is more elementary in asking for solutions $e \longmapsto e^\dagger$ in a concrete algebra A. Here we work with flat equations e in A, i. e., morphisms of the form $e : X \longrightarrow HX + A$, but flatness is just a technical restriction: in future research we will prove that more general non-flat equations obtain solutions "automatically". The fact that we work with a fixed algebra A (and let only X and e vary) is partly responsible for the simplicity of our axioms in comparison to the work on theories (where A varies as well), see e. g. [7] or [23]. Iterative algebras of Evelyn Nelson [21] and Jerzy Tiuryn [25], where solutions $e^\dagger$ are required to be unique, are a similar approach. And iteration algebras of Zoltan Ésik [12] are another one. Unfortunately, the number of axioms (seven) and their complexity make the question of the relationship of that notion to Elgot algebras a nontrivial one. We intend to study this question in the future.

We work with two variations: Elgot algebras, related to $R_\Sigma X$, where the function $(-)^\dagger$ assigns a solution only to finitary flat recursive systems, and complete Elgot algebras, related to $T_\Sigma X$, where the function $(-)^\dagger$ assigns solutions to all flat recursive systems. This is related to our previous research [1,18,2,3] in which we proved that every finitary endofunctor H generates a free iterative monad R, and a free completely iterative monad T. In the present paper we then study the Eilenberg–Moore categories of the monads R and T. Here H is an endofunctor of a category satisfying some rather mild conditions (not only Set): this generality does not make the proofs any more complex, and later we use other categories than Set (see Summary).

We omit some proofs for lack of space, the reader can find them in the full version of this paper [4].

2 Iterative Algebras and CIAs

Assumption 2.1 Throughout the paper H denotes an endofunctor of a category $\mathcal{A}$ having binary coproducts. We denote by $\mathsf{inl} : A \longrightarrow A + B$ and $\mathsf{inr} : B \longrightarrow A + B$ the corresponding injections. At some stage we assume that $\mathcal{A}$ is locally finitely presentable and that H is finitary, i. e., preserves filtered colimits, but we then make these assumptions explicitly.

Recall that an object X is called *finitely presentable* iff the hom-functor $\mathcal{A}(X, -) : \mathcal{A} \longrightarrow \mathsf{Set}$ is finitary. (In Set, these are precisely the finite sets. In equational classes of algebras these are precisely the finitely presentable algebras in the usual sense.) Recall further that a category $\mathcal{A}$ is called *locally finitely presentable* if it has colimits and a small collection of finitely presentable objects whose closure under filtered colimits is all of $\mathcal{A}$, see [5].

Definition 2.2 Let $\alpha : HA \longrightarrow A$ be an H-algebra. By a *flat equation morphism* in A we understand a morphism $e : X \longrightarrow HX + A$ in $\mathcal{A}$. We call e *finitary* provided that X is finitely presentable. A *solution* of e is a morphism $e^\dagger : X \longrightarrow A$ such that the square

$$\begin{array}{ccc} X & \xrightarrow{\quad e^\dagger \quad} & A \\ {\scriptstyle e}\big\downarrow & & \big\uparrow{\scriptstyle [\alpha, A]} \\ HX + A & \xrightarrow[\quad He^\dagger + A \quad]{} & HA + A \end{array} \tag{2.1}$$

commutes.

If every finitary flat equation morphism has a unique solution, then A is said to be an *iterative algebra*. And A is called a *completely iterative algebra* (CIA) if every flat equation morphism has a unique solution.

Remark 2.3 Iterative algebras of polynomial endofunctors of Set were introduced and studied by Evelyn Nelson [21]. She proved that the algebras $R_\Sigma X$ of rational Σ-trees on X form free iterative algebras, and that the theory obtained from them is a free iterative theory of Calvin Elgot [10]. We have recently studied iterative algebras in a much more general setting; working with a finitary endofunctor of a locally finitely presentable category. Completely iterative algebras were studied by Stefan Milius in [18].

Example 2.4 Consider algebras in Set with one binary operation $*$, i. e., the functor is $HX = X \times X$. A flat equation morphism e in an algebra A assigns to every variable x either a flat term $y * z$ (y and z are variables) or an element of A. A solution $e^\dagger : X \longrightarrow A$ assigns to $x \in X$ either the same element as e, in case $e(x) \in A$, or the result of $e^\dagger(y) * e^\dagger(z)$, in case $e(x) = y * z$. For example, the following recursive equation

$$x \approx x * x\,,$$

represented by the obvious morphism $e : \{\, x \,\} \longrightarrow \{\, x \,\} \times \{\, x \,\} + A$, has as solution $e^\dagger$ an element $a = e^\dagger(x)$ which is idempotent. Consequently, every iterative algebra has a unique idempotent. If A is even completely iterative, then it has, for each sequence $a_0, a_1, a_2, \ldots$ of elements, a unique interpretation of $a_0 * (a_1 * (a_2 \cdots)))$, i. e., a unique sequence $b_0, b_1, b_2, \ldots$ with $b_0 = a_0 * b_1$, $b_1 = a_1 * b_2$, etc. In fact, we consider here the equations

$$x_n \approx a_n * x_{n+1} \qquad (n \in \mathbb{N})\,.$$

Iterative algebras have unique solutions of many non-flat equations because we can flatten them. For example the following recursive equations

$$x_1 \approx (x_2 * a) * b \qquad x_2 \approx x_1 * b$$

are not flat. But they can be easily flattened to obtain a system

$$\begin{array}{ll} x_1 \approx z_1 * z_2 & x_2 \approx x_1 * z_2 \\ z_1 \approx x_2 * z_3 & z_2 \approx b \\ z_3 \approx a & \end{array}$$

represented by a morphism $e : X \longrightarrow X \times X + A$, where $X = \{\, x_1, x_2, z_1, z_2, z_3 \,\}$. Its solution is a map $e^\dagger : X \longrightarrow A$ yielding a pair of elements $s = e^\dagger(x_1)$ and $t = e^\dagger(x_2)$ satisfying $s = (t * a) * b$ and $t = s * a$.

Example 2.5 *Iterative Σ-algebras.* For every finitary signature $\Sigma = (\Sigma_n)_{n \in \mathbb{N}}$ we can identify Σ-algebras with algebras of the *polynomial endofunctor* H_Σ of

Set defined on objects X by

$$H_\Sigma X = \Sigma_0 + \Sigma_1 \times X + \Sigma_2 \times X \times X + \dots$$

A Σ-term which has the form $\sigma(x_1, \dots, x_k)$ for some $\sigma \in \Sigma_k$ and for variables $x_1, \dots, x_k$ from X is called *flat*. Then a flat equation morphism $e : X \longrightarrow H_\Sigma X + A$ in an algebra A represents a system

$$x \approx t_x$$

of recursive equations, one for every variable $x \in X$, where each t_x is either a flat term in X, or an element of A. A solution $e^\dagger$ assigns to every variable x with $t_x = a$, $a \in A$, the element a, and if $t_x = \sigma(x_1, \dots, x_k)$ then $e^\dagger(x) = \sigma_A(e^\dagger(x_1), \dots, e^\dagger(x_k))$.

Observe that every iterative Σ-algebra A has, for every $\sigma \in \Sigma_k$, a unique idempotent (i.e., a unique element $a \in A$ with $\sigma(a, \dots, a) = a$). In fact, consider the flat equation $x \approx \sigma(x, \dots, x)$. More generally, every Σ-polynomial has a unique idempotent in A. For example, for a polynomial of depth 2, $\sigma(\tau_1, \dots, \tau_k)$, where $\sigma \in \Sigma_k$ and $\tau_1, \dots, \tau_k \in \Sigma_n$ consider the recursive equations

$$\begin{aligned} x_0 &\approx \sigma(x_1, x_2, \dots, x_k) \\ x_i &\approx \tau_i(x_0, x_0, \dots, x_0) \qquad (i = 1, \dots, k)\,. \end{aligned}$$

An example of an iterative Σ-algebra is the algebra T_Σ of all (finite and infinite) Σ-trees. Also the subalgebra R_Σ of T_Σ of all rational Σ-trees is iterative, see [21].

Example 2.6 In particular, for unary algebras ($H = Id$), an algebra $\alpha : A \longrightarrow A$ is iterative iff α^k has a unique fixed point ($k \geq 1$), see [3]. And A is a CIA iff, moreover, there exists no infinite sequence $(a_n)_{n\in\mathbb{N}}$ in $\mathcal{A}$ with $\alpha a_{n+1} = a_n$, see [18].

Remark 2.7 In [3] we have proved that for every finitary functor H of a locally finitely presentable category $\mathcal{A}$ a free iterative algebra RY exists on every object Y. Furthermore, we have given a canonical construction of RY as a colimit of all coalgebras $X \longrightarrow HX + Y$ carried by finitely presentable objects, in other words, for every object Y of $\mathcal{A}$, RY is a colimit of all finitary flat equations in Y. For example, for a polynomial functor H_Σ of Set the free iterative algebra on a set Y is the algebra $R_\Sigma Y$ of all rational Σ-trees over Y. In general, we call the monad $\mathbb{R}$ of free iterative algebras the *rational monad* generated by H. We have proved in [3] that the rational monad $\mathbb{R}$ is a free iterative monad on H.

Example 2.8 *Completely metrizable algebras.* Complete metric spaces are well-known to be a suitable basis for semantics. The first categorical treatment of complete metric spaces for semantics is due to P. America and J. Rutten [6]. Let

$$\mathsf{CMS}$$

denote the category of all complete metric spaces (i.e., such that every Cauchy sequence has a limit) with metrics in the interval $[0, 1]$. The morphisms are nonexpanding maps $f : (X, d_X) \longrightarrow (Y, d_Y)$, i. e., the inequality $d_Y(f(x), f(x')) \leq d_X(x, x')$ holds for all x, x' in X.

Given complete metric spaces X and Y, the hom-set $\mathsf{CMS}(X, Y)$ carries the pointwise metric $d_{X,Y}$ defined as follows:

$$d_{X,Y}(f, g) = \sup_{x \in X} d_Y(f(x), g(x))$$

America and Rutten call a functor $H : \mathsf{CMS} \longrightarrow \mathsf{CMS}$ *contracting* if there exists a constant $\varepsilon < 1$ such that for arbitrary morphisms $f, g : X \longrightarrow Y$ we have

$$d_{HX,HY}(Hf, Hg) \leq \varepsilon \cdot d_{X,Y}(f, g).$$

Lemma 2.9 *If $H : \mathsf{CMS} \longrightarrow \mathsf{CMS}$ is a contracting functor, then every nonempty H-algebra is a CIA.*

Proof. Let $\alpha : HA \longrightarrow A$ be a nonempty H-algebra. Choose an element a of A. For every equation morphism $e : X \longrightarrow HX + A$ define a sequence $e_n^\dagger$ in $\mathsf{CMS}(X, A)$ as follows:

(i) $e_0^\dagger = \mathsf{const}_a$, the constant function of value a.

(ii) Given $e_n^\dagger$ then $e_{n+1}^\dagger$ is defined as follows (compare (2.1)):

$$\begin{array}{ccc} X & \xrightarrow{\;e_{n+1}^\dagger\;} & A \\ {\scriptstyle e}\downarrow & & \uparrow{\scriptstyle [\alpha, A]} \\ HX + A & \xrightarrow[\;He_n^\dagger + A\;]{} & HA + A \end{array} \tag{2.2}$$

We prove that $(e_n^\dagger)$ is a Cauchy sequence in $\mathsf{CMS}(X, A)$. In fact, put

$$z = d({e_0}^\dagger, {e_1}^\dagger),$$

then we prove by induction that

$$d(e_n^\dagger, e_{n+1}^\dagger) \leq z \cdot \varepsilon^n.$$

For the induction step from the above inequality derive $d(He_{n+1}^\dagger, He_n^\dagger) \leq z \cdot \varepsilon^{n+1}$ and then use the definition of $e_n^\dagger$ and $e_{n+1}^\dagger$, see (2.2). Consequently, the sequence $(e_n^\dagger)$ is Cauchy: for every number $\delta > 0$ choose k with $z \cdot \varepsilon^k < \delta \cdot (1 - \varepsilon)$. Then for all n the inequalities

$$d({e_k}^\dagger, {e_{k+n}}^\dagger) \leq \sum_{i=0}^{n-1} d(e_{k+i}^\dagger, e_{k+i+1}^\dagger) \leq z \cdot \sum_{i=0}^{n-1} \varepsilon^{k+i} < z \cdot \varepsilon^k \cdot \sum_{i=0}^{\infty} \varepsilon^i = \frac{z \cdot \varepsilon^k}{1 - \varepsilon} < \delta$$

take place. Consequently, a limit

$$e^\dagger = \lim_{n \to \infty} e_n^\dagger$$

exists in $\mathsf{CMS}(X, A)$. Due to the contractivity of H, it follows that $He^\dagger = \lim\limits_{n \to \infty} He_n^\dagger$ in $\mathsf{CMS}(HX, HA)$ and thus the equality $He^\dagger + id_A = \lim\limits_{n \to \infty} (He_n^\dagger + id_A)$ holds in $\mathsf{CMS}(HX + A, HA + A)$. Thus, $e^\dagger$ is a solution of e:

$$\begin{aligned}[\alpha, A] \cdot (He^\dagger + A) \cdot e &= \lim_{n \to \infty} [\alpha, A] \cdot (He_n^\dagger + A) \cdot e \\ &= \lim_{n \to \infty} e_{n+1}^\dagger \\ &= e^\dagger.\end{aligned}$$

Let $e^* : X \longrightarrow A$ be another solution of e. Put $b = d(e^\dagger, e^*)$. Then $d(He^\dagger, He^*) \leq \varepsilon \cdot b$ which implies $d(He^\dagger + id_A, He^* + id_A) \leq \varepsilon \cdot b$, consequently,

$$b = d(e^\dagger, e^*) = d([\alpha, A] \cdot (He^\dagger + A) \cdot e, [\alpha, A] \cdot (He^* + A) \cdot e) \leq \varepsilon \cdot b.$$

Since $\varepsilon < 1$, this implies $b = 0$. Thus, $e^\dagger$ is the unique solution. □

Remark 2.10 Many set functors H have a lifting to contracting endofunctors H' of CMS. That is, for the forgetful functor $U : \mathsf{CMS} \longrightarrow \mathsf{Set}$ the following square

$$\begin{array}{ccc} \mathsf{CMS} & \xrightarrow{\;H'\;} & \mathsf{CMS} \\ {\scriptstyle U}\downarrow & & \downarrow{\scriptstyle U} \\ \mathsf{Set} & \xrightarrow[\;H\;]{} & \mathsf{Set} \end{array}$$

commutes. For example, if $HX = X^n$, define

$$H'(X, d) = (X^n, \frac{1}{2} \cdot d')$$

(where d' is the maximum metric) which is a contracting functor with $\varepsilon = \frac{1}{2}$. Since coproducts of $\frac{1}{2}$-contracting liftings are $\frac{1}{2}$-contracting liftings of coproducts, we conclude that every polynomial endofunctor has a contracting lifting to CMS.

Let us call an H-algebra $\alpha : HA \longrightarrow A$ *completely metrizable* if there exists a complete metric, d, on A such that α is a nonexpanding map from $H'(A,d)$ to (A,d).

Corollary 2.11 *Every completely metrizable algebra A is a CIA.*

In fact, to every equation morphism $e : X \longrightarrow HX + A$ assign the unique solution of $e : (X, d_0) \longrightarrow H'(X, d_0) + (A, d)$, where d_0 is the discrete metric ($d_0(x, x') = 1$ iff $x \neq x'$).

Remark 2.12 Stefan Milius [18] proved that for any endofunctor H of $\mathcal{A}$ a final coalgebra TY of $H(-) + Y$ is a free CIA on Y, and conversely. Furthermore, assuming that the free CIAs exist, it follows that the monad $\mathbb{T}$ of free CIAs is a free completely iterative monad on H. This generalizes and extends the classical result of [11] since for a polynomial functor H_Σ of Set the free completely iterative algebra on a set Y is the algebra $T_\Sigma Y$ of all Σ-trees over Y.

Remark 2.13 We are going to prove two properties of iterative algebras and CIA's: the functoriality and compositionality for solutions. We will use two "operations" on equation morphisms. One, $\bullet$, is just change of parameter names: given a flat equation morphism $e : X \longrightarrow HX + Y$ and a morphism $h : Y \longrightarrow Z$ we obtain the following equation morphism

$$h \bullet e \equiv X \xrightarrow{\;e\;} HX + Y \xrightarrow{HX+h} HX + Z\,.$$

The other operation $\blacksquare$ combines two flat equation morphisms

$$e : X \longrightarrow HX + Y \qquad \text{and} \qquad f : Y \longrightarrow HY + A$$

into the single flat equation morphism $f \blacksquare e : X + Y \longrightarrow H(X + Y) + A$ in a canonical way: put $\mathsf{can} = [H\mathsf{inl}, H\mathsf{inr}] : HX + HY \longrightarrow H(X + Y)$ and define

$$f \blacksquare e \equiv X{+}Y \xrightarrow{[e,\mathsf{inr}]} HX{+}Y \xrightarrow{HX+f} HX{+}HY{+}A \xrightarrow{\mathsf{can}+A} H(X{+}Y){+}A, \tag{2.3}$$

2.14 Functoriality. This states that solutions are invariant under renaming of variables, provided, of course, that the right-hand sides of equations are renamed accordingly. Formally, observe that every flat equation morphism is a coalgebra of the endofunctor $H(-) + A$. Given two such coalgebras e and f, a renaming of the variables (or *morphism of equations*) is a morphism $h : X \longrightarrow Y$ which forms a coalgebra homomorphism:

$$\begin{array}{ccc} X & \xrightarrow{\;e\;} & HX + A \\ {\scriptstyle h}\downarrow & & \downarrow{\scriptstyle Hh+A} \\ Y & \xrightarrow[\;f\;]{} & HY + A \end{array} \tag{2.4}$$

Definition 2.15 Let A be an algebra with a choice $e \longmapsto e^\dagger$ of solutions, for all flat equation morphisms e in A. We say that the choice is *functorial* provided that

$$e^\dagger = f^\dagger \cdot h \tag{2.5}$$

holds for all equation morphisms $h : e \longrightarrow f$. In other words: $(-)^\dagger$ is a functor from the category of all flat equation morphisms in the algebra A into the comma-category of the object A.

Lemma 2.16 *In every CIA the assignment* $(-)^\dagger$ *is functorial.*

Remark. The same holds for every iterative algebra, except that there we restrict X and Y in Definition 2.15 to finitely presentable objects.

2.17 Compositionality. This tells us how to perform simultaneous recursion: given an equation morphism f in A with a variable object Y, we can combine it with any equation morphism e in Y with a variable object X to obtain the equation morphism $f \blacksquare e$ in A of Remark 2.13. The compositionality decrees that the left-hand component of $(f \blacksquare e)^\dagger$ is just the solution of $f^\dagger \bullet e$, i. e., in lieu of solving f and e simultaneously we first solve f, plug in the solution in e and solve the resulting equation morphism.

Definition 2.18 Let A be an algebra with a choice $e \longmapsto e^\dagger$ of solutions, for all flat equation morphisms e in A. We say that the choice is *compositional* if for each pair $e : X \longrightarrow HX + Y$ and $f : Y \longrightarrow HY + A$ of flat equation morphisms the equation below holds.

$$\left(f^\dagger \bullet e\right)^\dagger = (f \blacksquare e)^\dagger \cdot \mathsf{inl} \tag{2.6}$$

Remark 2.19 Notice that the coproduct injection $\mathsf{inr} : Y \longrightarrow X + Y$ is a morphism of equations from f to $f \blacksquare e$. Functoriality then implies that $f^\dagger = (f \blacksquare e)^\dagger \cdot \mathsf{inr}$. Thus, in the presence of functoriality, the compositionality is equivalent to

$$(f \blacksquare e)^\dagger = \left[\left(f^\dagger \bullet e\right)^\dagger, f^\dagger\right]. \tag{2.7}$$

Lemma 2.20 *In every CIA the assignment* $(-)^\dagger$ *is compositional.*

Remark 2.21 The same holds for every iterative algebra, except that here we restrict X and Y in Definition 2.18 to finitely presentable objects.

Remark 2.22 As mentioned in the Introduction, our two axioms, functoriality and compositionality, are not new as ideas of axiomatizing recursion—we believe however, that their concrete form is new, and their motivation strengthened by the results below.

Functoriality corresponds precisely to the "functorial dagger implication" of S. Bloom and Z. Ésik [7], 5.3.3, which states that for every object p of an iterative theory the formation $f \longmapsto f^\dagger$ of solutions for ideal morphisms $f : m \longrightarrow m+p$ is a functor. And the compositionality resembles the "left pairing identity" of [7], 5.3.1, which for $f : n \longrightarrow n+m+p$ and $g : m \longrightarrow n+m+p$ states that

$$[f,g]^\dagger = [f^\dagger \cdot [h^\dagger, id_p], h^\dagger]\,,$$

where

$$h \equiv\ m \xrightarrow{\ g\ } n+m+p \xrightarrow{[f^\dagger, id_{m+p}]} m+p\,.$$

This identity corresponds also to the Bekić-Scott identity, see e. g. [20], 2.1.

3 Elgot Algebras

Definition 3.1 Let H be an endofunctor of a category with finite coproducts. An *Elgot algebra* is an H-algebra $\alpha : HA \longrightarrow A$ together with a function $(-)^\dagger$ which to every finitary flat equation morphism

$$e : X \longrightarrow HX + A \qquad (X \text{ finitely presentable})$$

assigns a solution $e^\dagger : X \longrightarrow A$ in such a way that the functoriality (2.5) and the compositionality (2.6) are satisfied.

By a *complete Elgot algebra* we analogously understand an H-algebra together with a function $(-)^\dagger$ assigning to every flat equation e a solution $e^\dagger$ so that functoriality and compositionality are satisfied.

Example 3.2 Every join semilattice A is an Elgot algebra. More precisely: consider the polynomial endofunctor $HX = X \times X$ of Set (expressing one binary operation). Then for every join semilattice A there is a "canonical" structure of an Elgot algebra on A obtained as follows: the algebra RA of all rational binary trees on A has an interpretation on A given by the function $\alpha : RA \longrightarrow A$ forming, for every rational binary tree t the join of all the (finitely many!) labels of leaves of t in A. Now given a finitary flat equation morphism $e : X \longrightarrow X \times X + A$, it has a unique solution $e^\dagger : X \longrightarrow RA$ in the free iterative algebra RA, and composed with α this yields a structure $e \longmapsto \alpha \cdot e^\dagger$ of an Elgot algebra on A. See Example 4.9 for a proof.

Remark 3.3 In contrast, no nontrivial join semilattice is iterative. In fact, in an iterative join semilattice there must be a unique solution of the formal equation $x \approx x \vee x$.

Example 3.4 Continuous algebras on cpos are complete Elgot algebras. Let us work here in the category

$$\mathsf{CPO}$$

of all ω-*complete posets*, i.e., posets having joins of increasing ω-chains; morphisms are the *continuous functions*, i.e., functions preserving joins of ω-chains. A functor $H : \mathsf{CPO} \longrightarrow \mathsf{CPO}$ is called *locally continuous* provided that for arbitrary CPOs, X and Y, the derived function from $\mathsf{CPO}(X, Y)$ to $\mathsf{CPO}(HX, HY)$ is continuous (i.e., $H(\bigsqcup f_n) = \bigsqcup H f_n$ holds for all increasing ω-sequences $f_n : X \longrightarrow Y$). For example, every polynomial endofunctor $X \longmapsto \coprod_n \Sigma_n \times X^n$ of CPO (where Σ_n are cpos) is locally continuous.

Observe that the category CPO has coproducts: they are the disjoint unions with elements of different summands incompatible.

Proposition 3.5 *Let $H : \mathsf{CPO} \longrightarrow \mathsf{CPO}$ be a locally continuous functor and let $\alpha : HA \longrightarrow A$ be an H-algebra with a least element $\perp \in A$. Then $(A, \alpha, (-)^\dagger)$ is a complete Elgot algebra w.r.t. the assignment of the least solution $e^\dagger$ to every flat equation morphism e.*

Notice that the least solution of $e : X \longrightarrow HX + A$ refers to the elementwise order of the hom-set $\mathsf{CPO}(X, A)$. We can actually prove a concrete formula for $e^\dagger$ as a join of the ω-chain

$$e^\dagger = \bigsqcup_{n \in \omega} e_n^\dagger$$

of "approximations": $e_0^\dagger$ is the constant function to $\perp$, the least element of A, and given $e_n^\dagger$, then $e_{n+1}^\dagger$ is defined by the commutativity of (2.2).

Remark 3.6 Many set functors H have a lifting to locally continuous endofunctors H' of CPO. That is, for the forgetful functor $U : \mathsf{CPO} \longrightarrow \mathsf{Set}$ the following square

$$\begin{array}{ccc} \mathsf{CPO} & \xrightarrow{H'} & \mathsf{CPO} \\ {\scriptstyle U}\downarrow & & \downarrow{\scriptstyle U} \\ \mathsf{Set} & \xrightarrow[H]{} & \mathsf{Set} \end{array}$$

commutes. For example, every polynomial functor H_Σ has such a lifting. Let us call an H-algebra $\alpha : HA \longrightarrow A$, CPO-*enrichable* if there exists a CPO-ordering $\sqsubseteq$ with a least element on the set A such that α is a continuous function from $H'(A, \sqsubseteq)$ to $(A, \sqsubseteq)$.

Corollary 3.7 *Every CPO-enrichable H-algebra A in Set is a complete Elgot algebra.*

In fact, to every equation morphism $e : X \longrightarrow HX + A$ assign the least solution of $e : (X, \leq) \longrightarrow H'(X, \leq) + (A, \sqsubseteq)$ where $\leq$ is the discrete ordering of X ($x \leq y$ iff $x = y$).

Example 3.8 *Unary algebras.* Let $H = Id$ as an endofunctor of Set. Given an H-algebra $\alpha : A \longrightarrow A$, if α has no fixed point, then A carries no structure of an Elgot algebra: consider the equation $x \approx \alpha(x)$.

Conversely, every fixed point a_0 of α yields a flat cpo structure with a least element a_0 on A, i. e., $x \leq y$ iff $x = y$ or $x = a_0$. Thus, A is a complete Elgot algebra since it is CPO-enrichable.

Example 3.9 Every complete lattice A is a complete Elgot algebra of $HX = X \times X$. Analogously to Example 3.2 we have a function $\alpha : TA \longrightarrow A$ assigning to every binary tree t in TA the join of all labels of leaves of t in A. Now for every flat equation morphism e in A we have its unique solution $e^\dagger$ in TA and this yields a structure $e \longmapsto \alpha \cdot e^\dagger$ of a complete Elgot algebra. See Example 5.5 for a proof.

4 The Eilenberg-Moore Category of the Monad $\mathbb{R}$

We prove now that the category of all Elgot algebras and solution-preserving morphisms, defined as expected, is the category $\mathcal{A}^{\mathbb{R}}$ of Eilenberg-Moore algebras of the rational monad $\mathbb{R}$ of H, see Remark 2.7.

Throughout this section H denotes a finitary endofunctor of a locally finitely presentable category $\mathcal{A}$. We denote by $\mathcal{A}_{fp}$ a small full subcategory representing all finitely presentable objects of $\mathcal{A}$. Recall the operations $\bullet$ and $\blacksquare$ from Remark 2.13.

Definition 4.1 Let $(A, \alpha, (-)^\dagger)$, and $(B, \beta, (-)^\ddagger)$ be Elgot algebras. We say that a morphism $h : A \longrightarrow B$ in $\mathcal{A}$ *preserves solutions* provided that for every finitary flat equation morphism $e : X \longrightarrow HX + A$ we have the following equation

$$X \xrightarrow{e^\dagger} A \xrightarrow{h} B \equiv X \xrightarrow{(h \bullet e)^\ddagger} B\,. \tag{4.1}$$

Lemma 4.2 *Every solution-preserving morphism between Elgot algebras is a homomorphism of H-algebras, i.e., we have $h \cdot \alpha = \beta \cdot Hh$.*

Example 4.3 The converse of Lemma 4.2 is true for iterative algebras, as proved in [3], but for Elgot algebras in general it is false. In fact, consider the unary algebra $id : A \longrightarrow A$, where $A = \{ 0, 1 \}$. This is an Elgot algebra with the solution structure $(-)^\dagger$ given by the fixed point $0 \in A$, see Example 3.8.

Then $\mathrm{const}_1 : A \longrightarrow A$ is a homomorphism of unary algebras that does not preserve solutions. Indeed, consider the following equation morphism

$$e : \{x\} \longrightarrow \{x\} + A, \quad x \longmapsto x.$$

We have $e^\dagger(x) = 0$, and thus $1 = \mathrm{const}_1 \cdot e^\dagger(x) \neq (\mathrm{const}_1 \bullet e)^\dagger(x) = e^\dagger(x) = 0$.

Notation 4.4 We denote by

$$\mathsf{Alg}^{\dagger}\, H$$

the category of all Elgot algebras and solution-preserving morphisms.

Proposition 4.5 *A free iterative algebra on Y is a free Elgot algebra on Y.*

Remark 4.6 For the two operations $\bullet$ and $\blacksquare$ from Remark 2.13 we list some obvious properties that these operations have for all $e : X \longrightarrow HX + Y$, $f : Y \longrightarrow HY + Z$, $s : Z \longrightarrow Z'$ and $t : Z' \longrightarrow Z''$:

(i) $id_Y \bullet e = e$.

(ii) $t \bullet (s \bullet e) = (t \cdot s) \bullet e$.

(iii) $s \bullet (f \blacksquare e) = (s \bullet f) \blacksquare e$.

Theorem 4.7 *The category $\mathsf{Alg}^{\dagger}\, H$ of Elgot algebras is isomorphic to the Eilenberg-Moore category $\mathcal{A}^{\mathbb{R}}$ of $\mathbb{R}$-algebras for the rational monad $\mathbb{R}$ of H.*

Remark 4.8 The shortest proof we know is based on Beck's Theorem. But it is not very intuitive. A slightly more technical (and much more illuminating) proof has the following sketch: Denote for any object Y by $(RY, \rho_Y, (-)^{\ddagger})$ a free Elgot algebra on Y with a universal arrow $\eta_Y : Y \longrightarrow RY$.

(i) For every $\mathbb{R}$-algebra $\alpha_0 : RA \longrightarrow A$ we have an "underlying" H-algebra

$$\alpha \equiv HA \xrightarrow{H\eta_A} HRA \xrightarrow{\rho_A} RA \xrightarrow{\alpha_0} A,$$

and the following formula for solving equations: given a finitary flat equation morphism $e : X \longrightarrow HX + A$ put

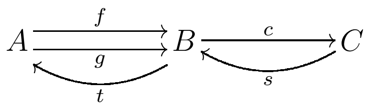
$$e^{\dagger} \equiv X \xrightarrow{(\eta_A \bullet e)^{\ddagger}} RA \xrightarrow{\alpha_0} A\,.$$

It is not difficult to see that this formula indeed yields a choice of solutions satisfying functoriality and compositionality.

(ii) Conversely, given an Elgot algebra $\alpha : HA \longrightarrow A$, define $\alpha_0 : RA \longrightarrow A$ as the unique solution-preserving morphism such that $\alpha_0 \cdot \eta_A = id$. It is easy to see that α_0 satisfies the two axioms of an Eilenberg-Moore algebra.

(iii) It is necessary to prove that the above passages extend to the level of morphisms and they form functors which are inverse to each other.

Proof. [Theorem 4.7] By Proposition 4.5 the natural forgetful functor $U : \mathsf{Alg}^{\dagger}\, H \longrightarrow \mathcal{A}$ has a left adjoint $Y \longmapsto RY$. Thus, the monad obtained by this adjunction is $\mathbb{R}$. We prove that the comparison functor $K : \mathsf{Alg}^{\dagger}\, H \longrightarrow \mathcal{A}^{\mathbb{R}}$ is an isomorphism, using Beck's theorem (see [17], Theorem 1 in VI.7). Thus, we must prove that U creates coequalizers of U-split pairs. Let $(A, \alpha, (-)^\dagger)$ and $(B, \beta, (-)^\ddagger)$ be Elgot algebras, and $f, g : A \longrightarrow B$ be solution-preserving morphisms with a splitting



in $\mathcal{A}$ (where $cs = id$, $ft = id$ and $gt = sc$). Since c is, then, an absolute coequalizer of f and g, c is a coequalizer in $\mathsf{Alg}\, H$ for a unique H-algebra structure $\gamma : HC \longrightarrow C$. In fact, the forgetful functor $\mathsf{Alg}\, H \longrightarrow \mathcal{A}$ creates every colimit that H preserves.

It remains to show that C has a unique structure of an Elgot algebra such that

(1) c preserves solutions, and

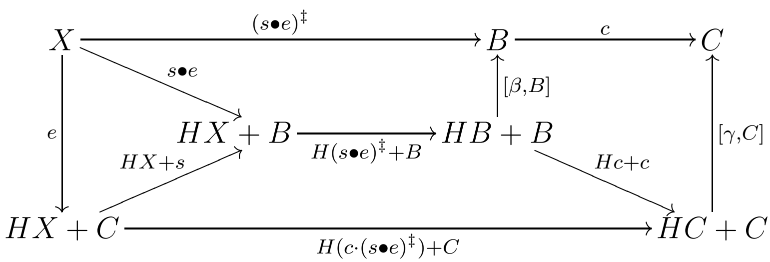
(2) c is a coequalizer in $\mathsf{Alg}^\dagger H$.

We establish (1) and (2) in several steps.

(a) An Elgot algebra on (C, γ). For every finitary flat equation morphism $e : X \longrightarrow HX + C$ we prove that the following morphism

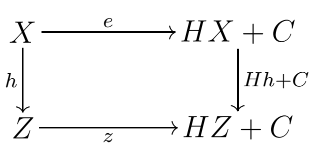
$$e^* \equiv X \xrightarrow{(s \bullet e)^\ddagger} B \xrightarrow{c} C$$

is a solution of e. In fact, the following diagram



clearly commutes.

Functoriality: any coalgebra homomorphism



is, of course, a coalgebra homomorphism

$$h : (X, s \bullet e) \longrightarrow (Z, s \bullet z)\,.$$

Thus,

$$e^* = c \cdot (s \bullet e)^{\ddagger} = c \cdot (s \bullet z)^{\ddagger} \cdot h = z^* \cdot h$$

by the functoriality of $(-)^{\ddagger}$.

Let us prove compositionality: suppose we have finitary flat equation morphisms

$$e : X \longrightarrow HX + Y \quad \text{and} \quad k : Y \longrightarrow HY + C$$

Then we obtain the desired equation as follows:

$$\begin{aligned}
(k^* \bullet e)^* &= c \cdot (s \bullet (k^* \bullet e))^{\ddagger} && \text{(Definition of } (-)^*\text{)}\\
&= c \cdot (s \bullet (c \cdot (s \bullet k)^{\ddagger} \bullet e))^{\ddagger} && \text{(Definition of } (-)^*\text{)}\\
&= c \cdot ((s \cdot c) \bullet ((s \bullet k)^{\ddagger} \bullet e))^{\ddagger} && \text{(see 4.6(ii))}\\
&= c \cdot ((g \cdot t) \bullet ((s \bullet k)^{\ddagger} \bullet e))^{\ddagger} && (g \cdot t = s \cdot c)\\
&= c \cdot (g \bullet (t \bullet ((s \bullet k)^{\ddagger} \bullet e)))^{\ddagger} && \text{(see 4.6(ii))}\\
&= c \cdot g \cdot (t \bullet ((s \bullet k)^{\ddagger} \bullet e))^{\dagger} && (g \text{ preserves solutions})\\
&= c \cdot f \cdot (t \bullet ((s \bullet k)^{\ddagger} \bullet e))^{\dagger} && (c \cdot f = c \cdot g)\\
&= c \cdot ((f \cdot t) \bullet ((s \bullet k)^{\ddagger} \bullet e))^{\ddagger}\\
&\qquad (f \text{ preserves solutions and 4.6(ii)})\\
&= c \cdot ((s \bullet k)^{\ddagger} \bullet e)^{\ddagger} && (f \cdot t = id \text{ and 4.6(i)})\\
&= c \cdot ((s \bullet k) \,\blacksquare\, e)^{\ddagger} \cdot \mathsf{inl} && \text{(compositionality for } (-)^{\ddagger}\text{)}\\
&= c \cdot (s \bullet (k \,\blacksquare\, e))^{\ddagger} \cdot \mathsf{inl}\\
&\qquad \text{(Since } (s \bullet k) \,\blacksquare\, e = s \bullet (k \,\blacksquare\, e) \text{ by 4.6(iii))}\\
&= (k \,\blacksquare\, e)^* \cdot \mathsf{inl} && \text{(Definition of } (-)^*\text{)}
\end{aligned}$$

(b) The morphism $c : B \longrightarrow C$ is solution-preserving. In fact, for any finitary flat equation morphism

$$e : X \longrightarrow HX + B$$

we have the desired equation:

$$
\begin{aligned}
(c \bullet e)^* &= c \cdot (s \bullet (c \bullet e))^\ddagger && \text{(Definition of } (-)^*\text{)}\\
&= c \cdot ((s \cdot c) \bullet e)^\ddagger && \text{(See 4.6(ii))}\\
&= c \cdot ((g \cdot t) \bullet e)^\ddagger && (g \cdot t = s \cdot c)\\
&= c \cdot (g \bullet (t \bullet e))^\ddagger && \text{(See 4.6(ii))}\\
&= c \cdot g \cdot (t \bullet e)^\dagger && (g \text{ preserves solutions})\\
&= c \cdot f \cdot (t \bullet e)^\dagger && (c \cdot f = c \cdot g)\\
&= c \cdot (f \bullet (t \bullet e))^\ddagger && (f \text{ preserves solutions})\\
&= c \cdot ((f \cdot t) \bullet e)^\ddagger && \text{(See 4.6(ii))}\\
&= c \cdot (id \bullet e)^\ddagger && (f \cdot t = id)\\
&= c \cdot e^\ddagger && \text{(See 4.6(i))}
\end{aligned}
$$

(c) $(-)^*$ is a unique structure of an Elgot algebra such that c is solution-preserving: in fact, for any such solution structure $(-)^*$ and for any finitary flat equation morphism $e : X \longrightarrow HX + B$ we have $c \cdot e^\ddagger = (c \bullet e)^*$. In particular, this is true for any equation morphism of the form

$$(s \bullet e') \equiv X \xrightarrow{\ e'\ } HX + C \xrightarrow{\ HX+s\ } HX + B$$

Thus, we conclude

$$
\begin{aligned}
e^* &= ((c \cdot s) \bullet e)^* && (c \cdot s = id \text{ and } 4.6\text{(iii)})\\
&= (c \bullet (s \bullet e))^* && \text{(See 4.6(ii))}\\
&= c \cdot (s \bullet e)^\ddagger && (c \text{ preserves solutions})
\end{aligned}
$$

(d) c is a coequalizer of f and g in $\mathsf{Alg}^\dagger H$. In fact, let $h : (B, \beta, (-)^\ddagger) \longrightarrow (D, \delta, (-)^+)$ be a solution-preserving morphism with $h \cdot f = h \cdot g$. There is a unique homomorphism $\overline{h} : C \longrightarrow D$ of H-algebras with $\overline{h} \cdot c = h$ (because c is a coequalizer of f and g in $\mathsf{Alg}\, H$). We prove that $\overline{h}$ is solution-preserving.

Let $e : X \longrightarrow HX + C$, be a finitary flat equation morphism. Then we have

$$
\begin{aligned}
\overline{h} \cdot e^* &= \overline{h} \cdot c \cdot (s \bullet e)^\ddagger && \text{(Definition of } (-)^*) \\
&= h \cdot (s \bullet e)^\ddagger && (h = \overline{h} \cdot c) \\
&= (h \bullet (s \bullet e))^+ && (h \text{ preserves solutions}) \\
&= ((h \cdot s) \bullet e)^+ && \text{(See 4.6(ii))} \\
&= ((\overline{h} \cdot c \cdot s) \bullet e)^+ && (h = \overline{h} \cdot c) \\
&= (\overline{h} \bullet e)^+ && (c \cdot s = id)
\end{aligned}
$$

as desired. This completes the proof. □

Example 4.9 Let A be a join semilattice. Recall from Example 3.2 the function $\alpha : RA \longrightarrow A$ assigning to a rational binary tree t in RA the join of the labels of all leaves of t in A. Since joins commute with joins it follows that this is the structure of an Eilenberg-Moore algebra on A. Thus, A is an Elgot algebra as described in Example 3.2.

5 Complete Elgot Algebras

Recall our standing assumptions that H is an endofunctor of a category $\mathcal{A}$ with finite coproducts. Stefan Milius has established in [18] that for every object-mapping T of $\mathcal{A}$ the following three statements are equivalent:

(a) for every object Y, TY is a final coalgebra of $H(-) + Y$

(b) for every object Y, TY is a free completely iterative H-algebra on Y, and

(c) T is the functor part of a free completely iterative monad $\mathbb{T}$ on H.

See also [1] where the monad $\mathbb{T}$ is described and the implication that (a) implies (c) is proved.

We are going to add another equivalent item to the above list, bringing complete Elgot algebras into the picture. The statements (a) to (c) are equivalent to

(d) for every object Y, TY is a free complete Elgot algebra on Y.

Furthermore, recall from [1] that H is *iteratable* if there exist objects TY such that one of the above equivalent statements holds. We will describe for every iteratable endofunctor the category $\mathcal{A}^{\mathbb{T}}$ of Eilenberg–Moore algebras—it is isomorphic to the category of complete Elgot algebras of H.

Example 5.1 For a polynomial endofunctor H_Σ of Set the above monad is the monad T_Σ of all (finite and infinite) Σ-trees.

In the following result the concept of *solution-preserving morphism* is defined for complete Elgot algebras analogously to Definition 4.1: the equation (4.1) holds for *all* flat equation morphisms e. We denote by

$$\mathsf{Alg}_c^\dagger H$$

the category of all complete Elgot algebras and solution-preserving morphisms.

Lemma 5.2 *Every solution-preserving morphism between complete Elgot algebras is a homomorphism of H-algebras.*

Theorem 5.3 *Let Y be an object of $\mathcal{A}$. Then the following are equivalent:*

(1) TY is a final coalgebra of $H(-) + Y$, and

(2) TY is a free complete Elgot algebra on Y.

Theorem 5.4 *If H is an iteratable functor, then the category $\mathsf{Alg}_c^\dagger H$ of complete Elgot algebras is isomorphic to the Eilenberg–Moore category $\mathcal{A}^{\mathbb{T}}$ of monadic $\mathbb{T}$-algebras (for the free completely iterative monad $\mathbb{T}$ of H).*

Proof. By Theorem 5.3, the natural forgetful functor $U : \mathsf{Alg}_c^\dagger H \longrightarrow \mathcal{A}$ has a left adjoint $Y \longmapsto TY$. Thus, the monad obtained by this adjunction is $\mathbb{T}$. To prove that the comparison functor $K : \mathsf{Alg}_c^\dagger H \longrightarrow \mathcal{A}^{\mathbb{T}}$ is an isomorphism use Beck's Theorem. In fact, the argument that U creates coequalizers of U-split pairs is entirely analogous to that of Theorem 4.7. □

Example 5.5 Let A be a complete lattice. Recall from Example 3.9 the function $\alpha : TA \longrightarrow A$ assigning to every binary tree t in TA the join of all labels of leaves of t in A. Since joins commute with joins it follows that $\alpha : TA \longrightarrow A$ is the structure of an Eilenberg-Moore algebra on A. Thus, A is a complete Elgot algebra as described in Example 3.9.

6 Summary and Future Work

The concept of Elgot algebra introduced in our paper formalizes algebras in which finitary flat equation morphisms have solutions satisfying two simple axioms: one for change of parameters and one for simultaneous recursion. And, analogously, complete Elgot algebras are algebras in which flat equation morphisms (not necessarily finitary) have solutions subject to the same two axioms. Such algebras can be used for interpreted semantics of recursive program schemes such as (1.1). In view of the simplicity of the two axioms we consider this is a success. Moreover, the structure of Elgot algebras is provided canonically by Elgot's iterative theories: Elgot algebras are the monadic algebras of the free iterative theory (as described by Calvin Elgot et al. for signatures in [11] and by the authors in [2,3] for general endofunctors). And complete Elgot algebras are the monadic algebras of the free completely iterative monad of Calvin Elgot et al. [11] (generalized by Stefan Milius in [18]).

For the important "in-between" variant of algebraic trees of Bruno Courcelle [8], i. e., precisely all trees obtained by tree unfoldings of recursive program schemes, no abstract treatment has been presented so far. The present authors are planning to work in a setting in which abstract algebraic trees can be treated. The basic category is, however, not Set, but $\mathsf{Fin}(\mathsf{Set})$, the category of all finitary endofunctors of Set. This category is locally finitely presentable, and that was one reason for presenting our theory in such general categories, not only in Set.

The function $e \longmapsto e^\dagger$ which is part of an Elgot algebra extends canonically from the above flat equation morphisms e to a much broader class of "rational" equation morphisms—another topic of our planned future research. In that sense one gets close to iteration algebras of Zoltan Ésik [12]. The relationship of the latter to Elgot algebras needs further investigation.

Finally, this paper can be considered as part of a program proposed by Lawrence Moss to rework the theory of recursive program schemes and their semantics using coalgebraic methods. We believe that our paper contributed by presenting a "suitable" notion of algebra of a functor which can be used for interpreted semantics or recursive program schemes. We do not have the space to treat this semantics in our paper. This is the topic of the forthcoming paper [19], where basic results of a categorical theory of recursive program schemes are presented. In that paper the authors introduce a general notion of recursive program scheme (rps), and they prove that any guarded rps has a unique "uninterpreted" solution in the final coalgebra of the functor describing the given operations. Furthermore, it is proved that an interpreted solution can be given to a recursive program scheme in any complete Elgot algebra, and that this solution is unique in case of a CIA. Finally, the fundamental result that every interpreted solution factors through an uninterpreted one is proved. As applications one obtains the classical theory using continuous algebras or completely metrizable ones as interpretations. New applications include, for example, recursively defined operations satisfying extra conditions like commutativity, or applications in non-well founded sets or measure spaces.

We admit that the whole program is at this point still at a beginning phase and so far has not yet produced many new results in semantics that go beyond what can be done with the well-established classical methods. However, we strongly believe that our approach deepens the understanding of the mechanisms at work in algebraic semantics, with categorical results of great conceptual clarity. We hope that this will eventually lead to new insights and results for the semantics of recursive computations.

References

[1] 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.

[2] J. Adámek, S. Milius and J. Velebil, Free iterative theories: a coalgebraic view, *Math. Structures Comput. Sci.* 13 (2003), 259–320.

[3] J. Adámek, S. Milius and J. Velebil, From Iterative Algebras to Iterative Theories, *Electron. Notes Theor. Comput. Sci.* 106 (2004), 3–24, full version is available from http://www.iti.cs.tu-bs.de/~milius/.

[4] J. Adámek, S. Milius and J. Velebil, Elgot Algebras, full version, available from http://www.iti.cs.tu-bs.de/~milius/.

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

[6] P. America and J. Rutten, Solving Reflexive Domain Equations in a Category of Complete Metric Spaces, in: *Proc. 3rd Workshop on the Mathematical Foundations of Programming Language Semantics*, M. Main, A. Melton, M. Mislove and D. Schmidt, eds., Springer-Verlag, LNCS 298, 254–288.

[7] S. L. Bloom and Z. Ésik, *Iteration Theories: The Equational Logic of Iterative Processes*, EATCS Monograph Series on Theoretical Computer Science, Springer-Verlag, 1993.

[8] B. Courcelle, Fundamental properties of infinite trees, *Theoret. Comput. Sci.* 25 (1983), no. 2, 95–169.

[9] S. Eilenberg, The category $\mathcal{C}$. Unpublished manuscript.

[10] C. C. Elgot, Monadic Computation and Iterative Algebraic Theories, in: *Logic Colloquium '73* (eds: H. E. Rose and J. C. Shepherdson), North-Holland Publishers, Amsterdam, 1975.

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

[12] Z. Ésik, Algebras of iteration theories, *J. Comput. System Sci.* 27 (1983), 291–303.

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

[14] J. Goguen, J. Thatcher, E. Wagner and J. Wright, Initial algebra semantics and continous algebras, *J. ACM* 24 (1977), 68–95.

[15] M. Hasegawa, Recursion from Cyclic Sharing: Traced Monoidal Categories and Models of Cyclic Lambda Calculi, *Proc. 3rd International Conference on Typed Lambda Calculi and Applications*, Springer LNCS 1210, 196–213, 1997.

[16] A. Joyal, R. Street and D. Verity, Traced Monoidal Categories, *Math. Proc. Cambridge Philos. Soc.* 119 (3), 447–468.

[17] S. MacLane, *Categories for the Working Mathematician*, Springer-Verlag, 2^{nd} edition, 1998.

[18] S. Milius, Completely Iterative Algebras and Completely Iterative Monads, accepted for publication in *Inform. and Comput.*

[19] S. Milius and L. Moss, The Category Theoretic Solution of Recursive Program Schemes, extended abstract to appear in Lecture Notes in Comput. Sci., full version available from http://www.iti.cs.tu-bs.de/~milius/

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

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

[22] G. Plotkin, *Domains*, The "Pisa" notes, available from http://www.dcs.ed.ac.uk/home/gdp/, 1983.

[23] A. Simpson and G. Plotkin, Complete Axioms for Categorical Fixed-point Operators, Proceedings of the Fifteenth Annual IEEE Symposium on Logic in Computer Science 2000, 30–41.

[24] M. B. Smyth and G. Plotkin, The Category-Theoretic Solution of Recursive Domain Equations, *SIAM J. Comp.* 11 (4), 1982, 761–783.

[25] J. Tiuryn, Unique Fixed Points vs. Least Fixed Points, *Theoret. Comput. Sci.* 12 (1980), 229–254.