

ABSTRACT GSOS RULES AND A COMPOSITIONAL TREATMENT OF RECURSIVE DEFINITIONS

STEFAN MILIUS, LAWRENCE S. MOSS, AND DANIEL SCHWENCKE

Institut für Theoretische Informatik, Technische Universität Braunschweig, Germany
e-mail address: mail@stefan-milius.eu

Department of Mathematics, Indiana University, Bloomington, IN, USA
e-mail address: lsm@cs.indiana.edu

Institut für Theoretische Informatik, Technische Universität Braunschweig, Germany
e-mail address: schwencke@iti.cs.tu-bs.de

ABSTRACT. Terminal coalgebras for a functor serve as semantic domains for state-based systems of various types. For example, formal languages, streams, infinite trees, non-well-founded sets and behaviors of CCS processes form terminal coalgebras. We present a uniform account of the semantics of recursive definitions in terminal coalgebras by combining two ideas: (1) terminal coalgebras are also initial *completely iterative algebras* (cias); (2) an *abstract GSOS rule* ℓ specifies additional algebraic operations on a terminal coalgebra. We first show that an abstract GSOS rule leads to new extended cia structures on the terminal coalgebra. Then we formalize recursive function definitions involving given operations specified by ℓ as recursive program schemes for ℓ , and we prove that unique solutions exist in the extended cias. We illustrate our results by the five concrete terminal coalgebras mentioned above, e.g., a finite stream circuit defines a unique stream function, and we show how to define new process combinators from given ones by sos rules involving recursion.

1. INTRODUCTION

Recursive definitions are a useful tool to specify infinite system behavior. For example, Milner [28] proved that in his calculus CCS, one may specify a process uniquely by the equation

$$P = a.(P|c) + b.$$

More generally, such recursive equations have unique solutions whenever each recursion variable is in the scope of some action prefix. Another example is the shuffle product on streams of real numbers uniquely defined by a *behavioral differential equation* [31]:

$$(\sigma \otimes \tau)_0 = \sigma_0 \cdot \tau_0 \quad (\sigma \otimes \tau)' = (\sigma \otimes \tau' + \sigma' \otimes \tau). \quad (1.1)$$

And as a third example consider non-well-founded sets [2, 13], a framework originating as a semantic basis for circular definitions. Here we can solve recursive function definitions such as

$$g(x) = \{g(\mathcal{P}(x)) \times x, x\} \quad (1.2)$$

Received by the editors 2010-12-08.

Key words and phrases: recursion, semantics, completely iterative algebra, coalgebra, distributive law.

uniquely. It is the aim of this paper to develop abstract tools and results that explain why there exist unique solutions to all the aforementioned equations.

The key observation is that streams, non-well-founded sets and process behaviors constitute *terminal coalgebras* for certain endofunctors on appropriate categories. Furthermore, the structure $c : C \rightarrow HC$ of a terminal coalgebra is an isomorphism [20], and the H -algebra (C, c^{-1}) is the initial *completely iterative algebra* (cia) for H [24]; cias are algebras in which recursive (function) definitions involving the operations given by the algebra structure $c^{-1} : HC \rightarrow C$ can be solved uniquely. However, cia structures for H are not sufficient to yield the existence and uniqueness of solutions in our motivating examples; these involve additional algebraic operations not captured by H . For example, consider $|$ and $+$ in CCS, the stream addition $+$, and powerset and cartesian product \mathcal{P}, \times in the example from non-well-founded set theory. None of these are related to the terminal H -coalgebra.

Additional algebraic operations are often presented by *distributive laws* in various guises. In process algebra one defines operations such as $|$ or $+$ by structural operational semantics (sos) [1]. Plotkin and Turi [29] showed how to capture sos rules as a distributive law of the functor (or monad) M describing the desired algebraic operations over the “behavior” functor H . This distributive law then induces an algebraic structure for M on the terminal H -coalgebra C . Other instances of distributive laws are behavioral differential equations in stream calculus [31, 32] or in infinite tree calculus [33] and definitions of operations on non-well-founded sets as in [13].

Bartels systematically studies definition formats giving rise to distributive laws in his thesis [12] (see also [11]) and shows how to solve parameter-free first order recursive equations involving operations presented by a distributive law; Uustalu et al [35] present the dual of this result.

We base our work in this paper on a categorical formulation of GSOS rules; Turi and Plotkin [29] showed that the GSOS rules of [14] give rise to distributive laws of a free monad over a cofree comonad, and Lenisa et al [22] proved that GSOS rules correspond precisely to distributive laws of a free monad M over a cofree copointed functor on the behavior functor H . We review some basic material on abstract GSOS rules and the solution theorem of Bartels in Section 2, and we provide a variant of this result. In Section 3 we extend these solution theorems to equations with parameters, thereby combining them with our previous work in [3, 24]. More concrete, we prove (Theorems 3.4 and 3.5) that the terminal H -coalgebra carries the structure of a cia for HM and for MHM . These results show how to construct new structures of cias on C out of the initial one using an abstract GSOS rule. This improves Bartels’ result in the sense that first order recursive definitions may employ constant parameters in the terminal coalgebra, and so this explains why recursively defined objects may be used in subsequent recursive definitions as given constants.

In Section 4 we obtain new ways to provide the semantics of recursive definitions by applying the existing solution theorems from [3, 24, 25] with the new cia structures. In addition, we prove a compositionality result for solutions of recursive program schemes as studied in [25]. In Theorem 4.6 we prove that the unique interpreted solution of a recursive program scheme in a given cia extends the structure of that cia.

However, the recursive program schemes from [25] do not capture functional recursive definitions like the above shuffle product (1.1) or the above function g from (1.2) on non-well-founded sets. So in Section 5 we provide results that do capture those examples. We introduce for an abstract GSOS rule ℓ the notion of a recursive program scheme w.r.t. ℓ (ℓ -rps, for short). In Theorem 5.14 we consider a variant of ℓ -rps’s that allows specifications going beyond the format of abstract GSOS rules. We prove that any ℓ -rps has a unique solution in the terminal coalgebra C . Moreover, we

show that these solutions extend the cia structure of C , which means that they can be used in subsequent recursive definitions. This compositionality of taking solutions of recursive equations does not appear in any previous work in this generality.

Finally, in Section 6 we demonstrate the value of our results by instantiating them in five different concrete applications: (1) CCS-processes—we explain how Milner’s solution theorem from [28] arises as a special case of Theorem 3.5, and we also show how to define new process combinators recursively from given ones; (2) streams of real numbers—here we prove that every finite stream circuit defines a unique stream function, we obtain the result from [31] that behavioral differential equations uniquely specify operations on streams as a special instance of our Theorem 5.3, and we show how to uniquely solve recursive equations that cannot be captured by behavioral differential equations by applying Theorem 5.14; (3) infinite trees—we obtain the result from [33] that behavioral differential equations have unique solutions as a special case; (4) non-well-founded sets—we prove that operations on non-well-founded sets are uniquely determined as solutions of ℓ -rps’s; (5) formal languages—here we show how operations on formal languages like union, concatenation, complement, etc. arise step-by-step using the compositionality of unique solutions of ℓ -rps’s, and how languages generated by grammars arise as the unique solutions of flat equation morphisms in cias.

Related Work. Turi’s and Plotkin’s work [29] was taken further by Lenisa, Power and Watanabe in [21, 22]. Jacobs [19] shows how to apply Bartels’ result to obtain the (first order) solution theorems from [3, 24]. Capretta et al. [15] work in a dual setting and generalize the results of [11] beyond terminal coalgebras and they also obtain the (dual of) the solution theorem from [3, 24] by an application of their general results. Our Theorems 3.4 and 3.5 are similar to results in [15], but we extend the work in [11] in a different direction by considering parameters in recursive definitions. So our results in the present paper go beyond what can be accomplished with previous work. For example, while [22] gives an abstract explanation of adding operations to a process calculus, it gives no account of the kind of compositionality we have in our results.

The present paper is a completely revised version containing full proofs of the conference paper [27].

2. ABSTRACT GSOS RULES AND DISTRIBUTIVE LAWS

We shall assume some familiarity with basic notions from category theory such as functors, (initial) algebras and (terminal) coalgebras, monads, see e.g. [23, 30, 4].

Suppose we are given an endofunctor H on some category \mathcal{A} describing the behavior type of a class of systems. In our work we shall be interested in additional algebraic operations on the terminal coalgebra C for H . The type of these algebraic operations is given by an endofunctor K on \mathcal{A} , and the algebraic operations are given by an *abstract GSOS rule* (cf. [29, 11, 12]). Our goal is to provide a setting in which recursive equations with operations specified by abstract GSOS rules have unique solutions. We now review the necessary preliminaries.

Assumption 2.1. Throughout the rest of this paper we assume that \mathcal{A} is a category with binary products and coproducts, $H : \mathcal{A} \rightarrow \mathcal{A}$ is a functor, and that $c : C \rightarrow HC$ is its terminal coalgebra. We also assume that $K : \mathcal{A} \rightarrow \mathcal{A}$ is a functor such that free K -algebras $\widehat{K}X$ on X exist for every object of \mathcal{A} .

Remark 2.2. (1) We denote by $\varphi_X : K\widehat{K}X \rightarrow \widehat{K}X$ and $\eta_X : X \rightarrow \widehat{K}X$ the structure and universal morphism of the free K -algebra $\widehat{K}X$ on X . Recall that the corresponding universal property states

that for every K -algebra $a : KA \rightarrow A$ and every morphism $f : X \rightarrow A$ there exists a unique K -algebra homomorphism $h : (\widehat{K}X, \varphi_X) \rightarrow (A, a)$ such that $h \cdot \eta_X = f$.

(2) Free algebras for the functor K exist under mild assumptions on K . For example, whenever K is an accessible endofunctor on Set it has all free algebras $\widehat{K}X$, see e.g. [8].

(3) As proved by M. Barr [10], free algebras yield free monads. Indeed, \widehat{K} is the object assignment of a monad with the unit given by η_X from item (1) and the multiplication $\mu_X : \widehat{K}\widehat{K}X \rightarrow \widehat{K}X$ given as the unique homomorphic extension of $\text{id}_{\widehat{K}X}$. Also $\varphi : K\widehat{K} \rightarrow \widehat{K}$ is a natural transformation and

$$\kappa = (K \xrightarrow{K\eta} K\widehat{K} \xrightarrow{\varphi} \widehat{K})$$

is the universal natural transformation of the free monad \widehat{K} . We shall abuse notation and write φ , η , κ and μ for different functors later.

Notation 2.3. For any K -algebra $a : KA \rightarrow A$, let $\widehat{a} : \widehat{K}A \rightarrow A$ be the unique K -algebra homomorphism with $\widehat{a} \cdot \eta_A = \text{id}_A$. Then \widehat{a} is the structure of an Eilenberg-Moore algebra for \widehat{K} . Notice also that

$$a = (KA \xrightarrow{\kappa_A} \widehat{K}A \xrightarrow{\widehat{a}} A). \quad (2.1)$$

Example 2.4. In our applications the functor K will be a polynomial functor on Set most of the time. More detailed, let $\mathcal{A} = \text{Set}$ and let $\Sigma = (\Sigma_n)_{n \in \mathbb{N}}$ be a signature of operation symbols with prescribed arity. The polynomial functor K_Σ associated to Σ is given by

$$K_\Sigma X = \coprod_{n \in \mathbb{N}} \Sigma_n \times X^n.$$

Algebras for the functor K_Σ are precisely the usual Σ -algebras and the free monad \widehat{K}_Σ assigns to a set X the Σ -algebra $\widehat{K}_\Sigma X$ of Σ -terms (or finite Σ -trees) on X , where a Σ -tree on X is a rooted and ordered tree with leafs labelled by constant symbols from Σ or elements of X and with inner nodes with n children labelled in Σ_n .

Definition 2.5. [11, 12] An *abstract GSOS rule* is a natural transformation

$$\ell : K(H \times \text{Id}) \rightarrow H\widehat{K}.$$

Remark 2.6. (1) The name abstract GSOS is motivated by the fact that for H the finite power set functor and K a polynomial functor a natural transformation ℓ as in Definition 2.5 corresponds precisely to the GSOS rules of Bloom, Istrail and Meyer [14], see [22].

(2) Abstract GSOS rules are in one-to-one correspondence with distributive laws, see e.g. Bartrès [12], Lemma 3.5.2. More precisely, recall from loc. cit. that a *copointed functor* (D, ε) is a functor $D : \mathcal{A} \rightarrow \mathcal{A}$ equipped with a natural transformation $\varepsilon : D \rightarrow \text{Id}$ and notice that $H \times \text{Id}$ together with the projection $\pi_1 : H \times \text{Id} \rightarrow \text{Id}$ is the cofree copointed functor on H . Further recall that a distributive law of a monad (M, η, μ) over a copointed functor (D, ε) is a natural transformation $\lambda : MD \rightarrow DM$ such that

$$\begin{array}{ccc} \begin{array}{c} D \\ \eta_D \swarrow \quad \searrow D\eta \\ MD \xrightarrow{\lambda} DM \end{array} & \begin{array}{c} MD \xrightarrow{\lambda} DM \\ M\varepsilon \searrow \quad \swarrow \varepsilon M \\ M \end{array} & \begin{array}{c} MMD \xrightarrow{M\lambda} MDM \xrightarrow{\lambda M} DMM \\ \mu_D \downarrow \quad \downarrow D\mu \\ MD \xrightarrow{\lambda} DM \end{array} \end{array} \quad (2.2)$$

Then to give an abstract GSOS rule $\ell : K(H \times \text{Id}) \rightarrow H\widehat{K}$ is equivalent to giving a distributive law λ of the monad \widehat{K} over the copointed functor $H \times \text{Id}$.

Theorem 2.7. [12, 29] Let ℓ be an abstract GSOS rule. There is a unique structure $b : KC \rightarrow C$ of a K -algebra on the terminal H -coalgebra C such that the square below commutes:

$$\begin{array}{ccc} KC & \xrightarrow{K\langle c, \text{id}_C \rangle} & K(HC \times C) & \xrightarrow{\ell_C} & H\widehat{K}C \\ b \downarrow & & & & \downarrow H\widehat{b} \\ C & \xrightarrow{c} & HC & & \end{array} \quad (2.3)$$

Remark 2.8. (1) In the terminology of [12, 29] the triple (C, b, c) is a *model* of the abstract GSOS rule ℓ ; in fact, it is the terminal one.

(2) For the distributive law $\lambda : \widehat{K}(H \times \text{Id}) \rightarrow (H \times \text{Id})\widehat{K}$ from Remark 2.6(2) corresponding to the abstract GSOS rule ℓ we have the following commutative diagram (see e.g. [12], Lemma 3.5.2, where this is formulated for $\mathcal{A} = \text{Set}$):

$$\begin{array}{ccc} \widehat{K}C & \xrightarrow{\widehat{K}\langle c, \text{id}_C \rangle} & \widehat{K}(HC \times C) & \xrightarrow{\lambda_C} & (H \times \text{Id})\widehat{K}C \\ \widehat{b} \downarrow & & & & \downarrow (H \times \text{Id})\widehat{b} \\ C & \xrightarrow{\langle c, \text{id}_C \rangle} & HC \times C & & \end{array} \quad (2.4)$$

(3) In some of the examples below an abstract GSOS rule ℓ will arise from a natural transformation $\ell' : K(H \times \text{Id}) \rightarrow HK$ as

$$\ell = (K(H \times \text{Id}) \xrightarrow{\ell'} HK \xrightarrow{H\kappa} K\widehat{K}).$$

In this case Diagram (2.3) can be simplified; indeed, $b : KC \rightarrow C$ is then the unique morphism such that the diagram below commutes.

$$\begin{array}{ccc} KC & \xrightarrow{K\langle c, \text{id}_C \rangle} & K(HC \times C) & \xrightarrow{\ell'_C} & H\widehat{K}C \\ b \downarrow & & & & \downarrow Hb \\ C & \xrightarrow{c} & HC & & \end{array}$$

Definition 2.9. For every abstract GSOS rule ℓ we will call the algebra structure $b : KC \rightarrow C$ from Theorem 2.7 the ℓ -interpretation in C .

Examples 2.10. We review a couple of examples of interest in this paper where $\mathcal{A} = \text{Set}$. We shall elaborate these examples in Section 6 and present two more examples.

(1) Formal languages. Consider the endofunctor $HX = X^A \times 2$, where $2 = \{0, 1\}$. Coalgebras for H are precisely the (possibly infinite) deterministic automata over the set A (as an alphabet). The terminal coalgebra $c : C \rightarrow HC$ consists of all formal languages with $c(L) = (\lambda a. L^a, i)$ with $i = 1$ iff the empty word ε is in L and where $L^a = \{w \mid aw \in L\}$.

To specify e.g. the intersection of formal languages by an abstract GSOS rule, let $KX = X \times X$ and let $\ell : K(H \times \text{Id}) \rightarrow H\widehat{K}$ be

$$\ell = (K(H \times \text{Id}) \xrightarrow{K\pi_0} KH \xrightarrow{\ell'} HK \xrightarrow{H\kappa} H\widehat{K})$$

where $\ell' : KH \rightarrow HK$ is given by $\ell'_X((f, i), (g, j)) = (\langle f, g \rangle, i \wedge j)$ where \wedge denotes the “and”-operation on $\{0, 1\}$. Then the ℓ -interpretation is easily verified to be the intersection of formal languages.

(2) Streams. Streams have been studied in a coalgebraic setting by Rutten [31]. Here we take the functor $HX = \mathbb{R} \times X$ whose terminal coalgebra (C, c) is carried by the set \mathbb{R}^ω of all streams over \mathbb{R} and $c = \langle \text{hd}, \text{tl} \rangle : \mathbb{R}^\omega \rightarrow \mathbb{R} \times \mathbb{R}^\omega$ is given by the usual head and tail functions on streams.

Operations on streams can be defined by so-called *behavioral differential equations* [31]. Here one uses for every stream σ the notation $\sigma_0 = \text{hd}(\sigma)$ and $\sigma' = \text{tl}(\sigma)$. Then, for example, the function zip merging two streams is specified by

$$\text{zip}(x, y)_0 = x_0 \quad (\text{zip}(x, y))' = \text{zip}(y, x').$$

This gives rise to an abstract GSOS rule as follows. Let $KX = X \times X$ (representing the binary operation zip) and $\ell : K(H \times \text{Id}) \rightarrow H\widehat{K}$ be

$$\ell = (K(H \times \text{Id}) \xrightarrow{\ell'} HK \xrightarrow{H\kappa} H\widehat{K})$$

where ℓ' is given by

$$\ell'_X((r, x', x), (s, y', y)) = (r, (y, x')).$$

Notice that in a triple $(r, x', x) \in HX \times X$, x is a variable for a stream with head r and tail referred to by the variable x' . It is now straightforward to work out that the ℓ -interpretation $b : KC \rightarrow C$ is the operation zip .

(3) Processes. We shall be interested in Milner's CCS [28]. Let κ be a regular cardinal and \mathcal{P}_κ be the functor assigning to the set X the set of all subsets Y with $|Y| < \kappa$. Here we consider the functor $HX = \mathcal{P}_\kappa(A \times X)$ where A is some fixed alphabet of actions. Following Milner [28], we assume that for every $a \in A$ we also have a complement $\bar{a} \in A$ (so that $\bar{\bar{a}} = a$) and a special silent action $\tau \in A$.

The terminal coalgebra for the finite power set functor \mathcal{P}_{fin} was described by Worrell [36]: it is carried by the set of all strongly extensional finitely branching trees, where an unordered tree t is called *strongly extensional* if two subtrees rooted at distinct children of some node of t are never bisimilar as trees. Similarly, the terminal coalgebra for the countable power set functor \mathcal{P}_c is carried by the set of all strongly extensional countably branching trees, see [34]. The technique by which this result is obtained in loc. cit. generalizes to the functor $\mathcal{P}_\kappa(A \times X)$ from above: its terminal coalgebra C turns out to consist of all strongly extensional κ -branching trees with edges labelled in A ; *strong extensionality* has the analogous meaning as above: two subtrees rooted at distinct children of some node are never bisimilar as trees if both edges to the children carry the same label. The elements of C can be considered as (denotations of) CCS-agents modulo strong bisimilarity.

Notice that the inverse $c^{-1} : \mathcal{P}_\kappa(A \times C) \rightarrow C$ assigns to a set $\{(a_i, E_i) \mid i < \kappa\}$ of pairs of actions and agents the agent $\sum_{i < \kappa} a_i \cdot E_i$. The usual process combinators “ $a.$ —” (prefixing), “ $|$ ” (parallel composition), “ $\sum_{i < \kappa}$ ” (sum), “ $-[f]$ ” (relabelling) and “ $- \setminus L$ ” (restriction) are given by SOS rules. Let E, E', F, F' be agents and $a \in A$ some action, then these rules are:

$$\begin{array}{cccc} \frac{E \xrightarrow{a} E'}{E | F \xrightarrow{a} E' | F} & \frac{F \xrightarrow{a} F'}{E | F \xrightarrow{a} E | F'} & \frac{E \xrightarrow{a} E' \quad F \xrightarrow{\bar{a}} F'}{E | F \xrightarrow{\tau} E' | F'} \quad (a \neq \tau) \\ \frac{}{a \cdot E \xrightarrow{a} E} & \frac{E_j \xrightarrow{a} E'_j}{(\sum_{i < \kappa} E_i) \xrightarrow{a} E'_j} \quad (j < \kappa) & \frac{E \xrightarrow{a} E'}{E[f] \xrightarrow{f(a)} E'[f]} & \frac{E \xrightarrow{a} E'}{E \setminus L \xrightarrow{a} E' \setminus L} \quad (a, \bar{a} \notin L) \end{array}$$

Now let K be the polynomial functor for the signature given by taking these combinators as operation symbols. It easily follows from the work in [12] and [22] that the above rules give an abstract GSOS rule $\ell : K(H \times \text{Id}) \rightarrow H\widehat{K}$, and the ℓ -interpretation $b : KC \rightarrow C$ in C provides the desired operations on CCS-agents (modulo strong bisimilarity). Further details are presented in Section 6.1.

Our first result (Theorem 2.14) improves a result from [11, 12] that we now recall. For the rest of this section we assume that an abstract GSOS rule $\ell : K(H \times \text{Id}) \rightarrow H\widehat{K}$ is given.

Definition 2.11. An ℓ -equation is an $H\widehat{K}$ -coalgebra; that is, a morphism of the form $e : X \rightarrow H\widehat{K}X$. A *solution* of e in the terminal coalgebra C is a morphism $e^\dagger : X \rightarrow C$ such that the diagram below commutes:

$$\begin{array}{ccc} X & \xrightarrow{e^\dagger} & C \\ e \downarrow & & \uparrow c^{-1} \\ H\widehat{K}X & \xrightarrow{H\widehat{K}e^\dagger} & H\widehat{K}C \\ & \uparrow H\widehat{b} & \\ & \widehat{K}C & \end{array} \quad (2.5)$$

Theorem 2.12. [11, 12] For every ℓ -equation there exists a unique solution in C .

This result follows from Corollaries 4.3.6 and 4.3.8 and Lemma 4.3.9 in [12]. The first of these results is the dual of a result obtained independently and at the same time by Uustalu, Vene and Pardo (see [35], Theorem 1), and Capretta, Uustalu and Vene [15], Theorems 19 and 28 generalize this work further.

We shall need a variant of Theorem 2.12 for equations of the form $e : X \rightarrow \widehat{K}H\widehat{K}X$:

Definition 2.13. A *sandwiched ℓ -equation* is a $\widehat{K}H\widehat{K}$ -coalgebra; that is, a morphism of the form $e : X \rightarrow \widehat{K}H\widehat{K}X$. A *solution* of e in the terminal coalgebra C is a morphism $e^\dagger : X \rightarrow C$ such that the diagram below commutes:

$$\begin{array}{ccc} X & \xrightarrow{e^\dagger} & C \\ e \downarrow & & \uparrow \widehat{b} \\ H\widehat{K}X & \xrightarrow{H\widehat{K}e^\dagger} & H\widehat{K}C \\ & \uparrow \widehat{K}c^{-1} & \\ & \widehat{K}HC & \end{array} \quad (2.6)$$

Theorem 2.14. For every sandwiched ℓ -equation there exists a unique solution in C .

Proof. Given a sandwiched ℓ -equation $e : X \rightarrow \widehat{K}H\widehat{K}X$, we form the following (ordinary) ℓ -equation:

$$\bar{e} = (H\widehat{K}X \xrightarrow{H\widehat{K}e} H\widehat{K}\widehat{K}H\widehat{K}X \xrightarrow{H\mu_{H\widehat{K}X}} H\widehat{K}H\widehat{K}X).$$

From Theorem 2.12 we know that \bar{e} has a unique solution $\bar{e}^\dagger : H\widehat{K}X \rightarrow C$. Thus, we are finished if we can show that solutions of e and \bar{e} are in one-to-one correspondence.

Firstly, from the solution \bar{e}^\dagger of \bar{e} we obtain

$$e^\dagger = (X \xrightarrow{e} \widehat{K}H\widehat{K}X \xrightarrow{\widehat{K}\bar{e}^\dagger} \widehat{K}C \xrightarrow{\widehat{b}} C),$$

and we will now verify that e^\dagger is a solution of e . To this end, consider the diagram below:

$$\begin{array}{ccccc}
 X & \xrightarrow{e^\dagger} & C & & \\
 \downarrow e & & & & \uparrow \hat{b} \\
 \widehat{K}H\widehat{K}X & \xrightarrow{\widehat{K}\bar{e}^\dagger} & \widehat{K}C & & \\
 \downarrow \widehat{K}H\widehat{K}e & \searrow \widehat{K}\bar{e} & & & \uparrow \widehat{K}c^{-1} \\
 \widehat{K}H\widehat{K}\widehat{K}H\widehat{K}X & \xrightarrow{\widehat{K}H\mu_{H\widehat{K}X}} & \widehat{K}H\widehat{K}H\widehat{K}X & \xrightarrow{\widehat{K}H\widehat{K}\bar{e}^\dagger} & \widehat{K}H\widehat{K}C \xrightarrow{\widehat{K}H\widehat{b}} \widehat{K}HC \\
 \downarrow \widehat{K}H\widehat{K}\widehat{K}\bar{e}^\dagger & \nearrow \widehat{K}H\mu_C & & & \uparrow \widehat{K}H\widehat{b} \\
 \widehat{K}H\widehat{K}\widehat{K}C & \xrightarrow{\widehat{K}H\widehat{K}\widehat{b}} & & & \uparrow \widehat{K}H\widehat{K}C \\
 & & \text{(iii)} & &
 \end{array}$$

All its inner parts commute: part (i) and (iii) commute by the definition of e^\dagger , for part (ii) use that \bar{e}^\dagger is a solution of \bar{e} (i.e. apply \widehat{K} to Diagram (2.5) with \bar{e} in lieu of e), and the remaining parts commute by the definition of \bar{e} , naturality of μ and the multiplication law for the Eilenberg-Moore algebra \widehat{b} . Thus, the outside commutes proving e^\dagger to be a solution of e .

Secondly, suppose we are given any solution e^\dagger of e . Then we form

$$H\widehat{K}X \xrightarrow{H\widehat{K}e^\dagger} H\widehat{K}C \xrightarrow{H\widehat{b}} HC \xrightarrow{c^{-1}} C ,$$

and we now prove that this is a solution of \bar{e} . Indeed, in the diagram

$$\begin{array}{ccccc}
 H\widehat{K}X & \xrightarrow{H\widehat{K}e^\dagger} & H\widehat{K}C & \xrightarrow{H\widehat{b}} & HC \xrightarrow{c^{-1}} C \\
 \downarrow H\widehat{K}e & & \uparrow H\widehat{K}\widehat{b} & \searrow H\widehat{b} & \uparrow c^{-1} \\
 \bar{e} & & H\widehat{K}\widehat{K}C & & HC \\
 \downarrow H\mu_{H\widehat{K}X} & \xrightarrow{H\widehat{K}\widehat{K}H\widehat{K}e^\dagger} & \downarrow H\widehat{K}\widehat{K}c^{-1} & \nearrow H\mu_C & \uparrow H\widehat{b} \\
 H\widehat{K}\widehat{K}H\widehat{K}X & \xrightarrow{H\widehat{K}\widehat{K}H\widehat{K}e^\dagger} & H\widehat{K}\widehat{K}H\widehat{K}C & \xrightarrow{H\widehat{K}\widehat{K}H\widehat{b}} & H\widehat{K}HC \xrightarrow{H\widehat{K}c^{-1}} H\widehat{K}C
 \end{array}$$

all inner parts commute: for the big left-hand square apply $H\widehat{K}$ to Diagram (2.6), the left-hand part is the definition of \bar{e} , the two lower squares and the lower right triangle commute due to naturality of μ , the upper right triangle is trivial, and the remaining part commutes by the multiplication law for the Eilenberg-Moore algebra \widehat{b} . Thus, the outside commutes proving $c^{-1} \cdot H\widehat{b} \cdot H\widehat{K}e^\dagger$ to be a solution of \bar{e} . Since \bar{e} has a unique solution, we have

$$c^{-1} \cdot H\widehat{b} \cdot H\widehat{K}e^\dagger = \bar{e}^\dagger .$$

Lastly, both constructions are inverses to each other: starting with the unique solution \bar{e}^\dagger of \bar{e} , it is clear that applying the first construction and then the second one results in \bar{e}^\dagger again. Starting with

any solution e^\dagger of e , application of the second construction results in the solution $c^{-1} \cdot \widehat{Hb} \cdot \widehat{HK}e^\dagger = \bar{e}^\dagger$ of \bar{e} . The application of the first construction to that solution gives the solution e^\dagger of e :

$$\widehat{b} \cdot \widehat{K}\bar{e}^\dagger \cdot e = \widehat{b} \cdot \widehat{K}(c^{-1} \cdot \widehat{Hb} \cdot \widehat{HK}e^\dagger) \cdot e = \widehat{b} \cdot \widehat{K}c^{-1} \cdot \widehat{KH}\widehat{b} \cdot \widehat{KH}\widehat{K}e^\dagger \cdot e = e^\dagger$$

where the last step uses Diagram (2.6). We conclude that e has a unique solution e^\dagger . \square

3. COMPLETELY ITERATIVE ALGEBRAS

It is our aim in this section to extend Theorems 2.12 and 2.14 so as to obtain several new structures of completely iterative algebras (for functors other than H) on C . We briefly recall the basic definitions and some examples; more details and examples can be found in [24, 6, 25].

Definition 3.1. [24] A *flat equation morphism* in an object A (of parameters) is a morphism $e : X \rightarrow HX + A$. An H -algebra $a : HA \rightarrow A$ is called *completely iterative* (or a *cia*, for short) if every flat equation morphism in A has a unique solution, i. e., for every $e : X \rightarrow HX + A$ there exists a unique morphism $e^\dagger : X \rightarrow A$ such that

$$e^\dagger = (X \xrightarrow{e} HX + A \xrightarrow{He^\dagger + A} HA + A \xrightarrow{[a, A]} A).$$

Examples 3.2. We recall some examples from previous work.

- (1) Let TX denote a terminal coalgebra for $H(-) + X$ (assuming that TX exists). Its structure is an isomorphism by Lambek's Lemma [20], and so its inverse yields (by composing with the coproduct injections) an H -algebra $\tau_X : HTX \rightarrow TX$ and a morphism $\eta_X : X \rightarrow TX$. Then (TX, τ_X) is a free cia on X with the universal arrow η_X , see [24]. So in particular, the inverse of the structure $c : C \rightarrow HC$ of the terminal coalgebra for H is, equivalently, an initial cia for H .
- (2) Let H_Σ be a polynomial functor (cf. Example 2.4). The terminal coalgebra for $H_\Sigma(-) + X$ is carried by the set $T_\Sigma X$ of all (finite and infinite) Σ -trees on X . According to the previous item, this is a free cia for H_Σ on X .
- (3) The algebra of addition on $\bar{\mathbb{N}} = \{1, 2, 3, \dots\} \cup \{\infty\}$ is a cia for $HX = X \times X$, see [7].
- (4) Let $\mathcal{A} = \text{CMS}$ be the category of complete metric spaces with distances in $[0, 1]$ and with non-expanding maps as morphisms, and let H be a contracting endofunctor of CMS, see e. g. [9]. Then any non-empty algebra for H is a cia, see [24] for details. For example, let A be the set of non-empty compact subsets of the unit interval $[0, 1]$ equipped with the Hausdorff metric [16]. This complete metric space can be turned into a cia such that the Cantor set arises as the unique solution of a flat equation morphism, see [25], Example 3.3(v).
- (5) Unary algebras of Set. Here we take $\mathcal{A} = \text{Set}$ and $H = \text{Id}$. An algebra $\alpha : A \rightarrow A$ is a cia iff α has a fixed point a_0 and there is no infinite sequence a_1, a_2, a_3, \dots with $a_i = \alpha(a_{i+1})$, $i = 1, 2, 3, \dots$, except for the one all of whose members are a_0 . The second part of this condition can be put more vividly as follows: the graph with node set $A \setminus \{a_0\}$ and with an edge from $\alpha(a) \neq a_0$ to a for all a is well-founded. Since any well-founded graph induces a well-founded (strict) order on its node set, we have yet another formulation: there is a well-founded order on $A \setminus \{a_0\}$ for which α is strictly increasing in the sense that $\alpha(a) \neq a_0$ implies $a < \alpha(a)$ for all $a \in A$.
- (6) Classical algebras are seldom cias. For example a group or a semilattice is a cia (for $HX = X \times X$) iff they contain one element only (consider the unique solution of $x = x \cdot 1$ or $x = x \vee x$, respectively).

The following results show that abstract GSOS rules induce further structures of completely iterative algebras on C .

Assumption 3.3. For the rest of this section we shall write (M, η, μ) for the free monad \widehat{K} on K to simplify notation, and we assume that $\ell : K(H \times \text{Id}) \rightarrow HM$ is an abstract GSOS rule.

Theorem 3.4. Consider the algebra

$$k = (HMC \xrightarrow{H\widehat{b}} HC \xrightarrow{c^{-1}} C),$$

where $b : KC \rightarrow C$ is the ℓ -interpretation in C . Then (C, k) is a cia for the functor HM .

Proof. Let $e : X \rightarrow HMX + C$ be a flat equation morphism. We must prove that there exists a unique morphism $e^\dagger : X \rightarrow C$ such that the following square commutes:

$$\begin{array}{ccc} X & \xrightarrow{e^\dagger} & C \\ e \downarrow & & \uparrow [k, C] \\ HMX + C & \xrightarrow{HMe^\dagger + C} & HMC + C \end{array}$$

We start by forming the ℓ -equation

$$\bar{e} = (X + C \xrightarrow{[e, \text{inr}]} HMX + C \xrightarrow{HMX + H\eta_C \cdot c} HMX + HMC \xrightarrow{\text{can}} HM(X + C)).$$

By Theorem 2.12, there exists a unique morphism $s : X + C \rightarrow C$ such that the square below commutes:

$$\begin{array}{ccc} X + C & \xrightarrow{s} & C \\ \bar{e} \downarrow & & \downarrow c \\ & & HC \\ & & \uparrow H\widehat{b} \\ HM(X + C) & \xrightarrow{HM s} & HMC \end{array} \tag{3.1}$$

We will now prove that the morphism $e^\dagger = s \cdot \text{inr} : X \rightarrow C$ is the desired unique solution of e . We begin by proving that the equation $s \cdot \text{inr} = \text{id}_C$ holds. Indeed, consider the diagram below:

$$\begin{array}{ccccc} C & \xrightarrow{\text{inr}} & X + C & \xrightarrow{s} & C \\ c \downarrow & & \bar{e} \downarrow & & \uparrow k \\ & HMC & \xrightarrow{HM \text{inr}} & HM(X + C) & \xrightarrow{HM s} HMC \\ & \nearrow H\eta_C & & & \uparrow H\eta_C \\ HC & \xrightarrow{H(s \cdot \text{inr})} & & & HC \end{array} \tag{3.2}$$

This diagram commutes: the upper right-hand square is Diagram (3.1) above, the left-hand upper square commutes by the definition of \bar{e} , the lower part commutes by the naturality of η , and the right-hand part follows from the definition of $k = c^{-1} \cdot H\widehat{b}$ and the unit law $\widehat{b} \cdot \eta_C = \text{id}_C$ of the Eilenberg-Moore algebra (C, \widehat{b}) , cf. Notation 2.3. Hence, we see that $s \cdot \text{inr}$ is a coalgebra homomorphism from the terminal coalgebra (C, c) to itself. Thus, $s \cdot \text{inr}$ must be the identity as desired.

Next we prove that e^\dagger is a solution of e . To this end we verify that the following diagram commutes:

$$\begin{array}{ccccc}
 & & e^\dagger & & \\
 & X & \xrightarrow{\text{inl}} & X + C & \xrightarrow{s} C \\
 & \downarrow e & & \downarrow \bar{e} & \uparrow k \\
 & & HM(X + C) & \xrightarrow{HMs} HMC & \\
 & & \uparrow \text{can} & \nearrow [HMe^\dagger, HMC] & \\
 & HMX + HMC & \xrightarrow{HMX + H\eta_C \cdot c} & & \\
 & \downarrow HMe^\dagger + C & & & \\
 HMX + C & \xrightarrow{HMX + H\eta_C \cdot c} & & & \xrightarrow{HMe^\dagger + C} HMC + C
 \end{array} \tag{3.3}$$

The upper part is the definition of e^\dagger , the left-hand part commutes by the definition of \bar{e} , the upper right-hand part is Diagram (3.1), and that the inner triangle commutes follows from the definition of e^\dagger and the fact that $s \cdot \text{inr} = \text{id}_C$. Finally, we consider the two coproduct components of the lower part separately; the left-hand component trivially commutes, and for the right-hand one we compute as follows:

$$\begin{aligned}
 k \cdot H\eta_C \cdot c &= c^{-1} \cdot \widehat{Hb} \cdot H\eta_C \cdot c && \text{(by the definition of } k) \\
 &= c^{-1} \cdot c && \text{(since } b \cdot \eta_C = \text{id}_C) \\
 &= \text{id}_C.
 \end{aligned}$$

To complete our proof we show that $e^\dagger = s \cdot \text{inl} : X \rightarrow C$ is the unique solution of e . So suppose that we are given any solution e^\dagger of e . Now form the morphism $s = [e^\dagger, \text{id}_C]$. We are finished if we show that for this morphism s the Diagram (3.1) commutes. We verify the two coproduct components separately: the right-hand component is checked using Diagram (3.2) and the equation $s \cdot \text{inr} = \text{id}_C$ to see that the outside of (3.2) commutes. The commutativity of the left-hand component is established using Diagram (3.3); indeed, since the outside and all other parts of that diagram commute for our morphism s so does the desired upper right-hand square composed with inl . \square

Theorem 3.5. (Sandwich Theorem) *Consider the algebra*

$$k' = (MHC \xrightarrow{Mk} MC \xrightarrow{\widehat{b}} C),$$

where $b : KC \rightarrow C$ is the ℓ -interpretation in C and $k = c^{-1} \cdot \widehat{Hb}$ as in Theorem 3.4. Then (C, k') is a cia for the functor MHM .

Proof. The proof is a slight modification of the proof of Theorem 3.4: now we are given a flat equation morphism $e : X \rightarrow MHMX + C$ and form the morphism

$$\begin{array}{c}
 \bar{e} \equiv X + C \xrightarrow{[e, \text{inr}]} MHMX + C \\
 \downarrow MHMX + MH\eta_C \cdot \eta_{HC} \cdot c \\
 MHMX + MHC \xrightarrow{\text{can}} MHM(X + C).
 \end{array}$$

This morphism \bar{e} is a sandwiched ℓ -equation and we invoke Theorem 2.14 to see that it has a unique solution s . The rest of this proof is left to the reader since it is very close to the one of Theorem 3.4.

□

Remark 3.6. (1) Theorems 3.4 and 3.5 hold more generally for an arbitrary monad M in lieu of the free one \widehat{K} . More detailed, every distributive law λ of a monad M over the cofree copointed functor $H \times \text{Id}$ induces a λ -interpretation, i.e., a unique Eilenberg-Moore algebra $\widehat{b} : MC \rightarrow C$ such that Diagram (2.4) commutes with \widehat{K} replaced by M , cf. Theorem 2.7 and Remark 2.8(2). The version of Theorem 2.12 presented in [11, 12] and dually in [35] states that for every $e : X \rightarrow HMX$ there is a unique solution $e^\dagger : X \rightarrow C$, i.e., e^\dagger is such that (2.5) commutes with M in lieu of \widehat{K} . Then Theorem 3.4 shows that (C, k) is a cia for HM . Similarly, Theorem 2.14 clearly holds for an arbitrary monad M . Thus, Theorem 3.5 shows that (C, k') is a cia for MHM .

(2) Observe that our proof of Theorem 3.4 only makes use of the unit $\eta : \text{Id} \rightarrow M$ of the monad M . In fact, there are versions of Theorem 3.4 and 3.5 that hold for a pointed functor M in lieu of a monad and for a given distributive law of M over the cofree copointed functor $H \times \text{Id}$ or the functor H , respectively. However, in this case we need to assume that the category \mathcal{A} is cocomplete. The technical details are somewhat different than what we have seen and we discuss them in detail in the appendix.

Theorems 3.4 and 3.5 extend Theorems 2.12 and 2.14 in two important ways. Firstly, the structure of a cia allows one to reuse solutions of a given recursive specification by using constants in C on the right-hand sides of recursive equations. This gives a clear explanation of why it is possible to use recursively defined objects (streams, processes, etc.) in subsequent recursive definitions. This kind of compositionality of the unique solutions is a useful and desired property often employed in specifications.

Secondly, Theorem 3.5 permits the right-hand sides of recursive specifications to be from a wider class. For example, Milner's solution theorem for CCS (see [28], Chapter 4, Proposition 14) allows recursion over process terms E in which the recursion variables occur within the scope of some prefixing combinator $a.-$. This combinator can occur *anywhere within* E , not necessarily at the head of that term. Hence, Theorem 3.5 allows us to obtain Milner's result as a special case, directly. This will be explained in detail in Section 6.1.

Remark 3.7. The property of compositionality is made explicit by the following property that is true in every cia, see [6]. Suppose we have two flat equation morphisms $e : X \rightarrow HX + Y$ and $f : Y \rightarrow HY + A$. We form

$$f^\dagger \bullet e = (X \xrightarrow{e} HX + Y \xrightarrow{HX + f^\dagger} HX + A)$$

and

$$f \blacksquare e = (X + Y \xrightarrow{[e, \text{inr}]} HX + Y \xrightarrow{HX + f} HX + HY + A \xrightarrow{\text{can} + A} H(X + Y) + A).$$

Then

$$(f \blacksquare e)^\dagger = (X + Y \xrightarrow{[(f^\dagger \bullet e)^\dagger, f^\dagger]} A).$$

4. SOLUTION THEOREMS FOR FREE

Using the new cia structures obtained from Theorems 3.4 and 3.5, the existing body of results on the semantics of recursion in cias [3, 24, 25] now gives us further theorems.

We begin with a terse review of some terminology from the area [3, 24, 25]. We assume that, in addition to the terminal H -coalgebra C , for every object X the terminal coalgebra $T^H X$ for $H(-) + X$ exists, i. e., in the terminology of loc. cit., H is *iteratable*. Our examples in 2.10 are all iterable endofunctors of Set.

As explained in Example 3.2(1), the structure of the terminal coalgebra $T^H X$ yields the free cia $(T^H X, \tau_X^H)$ on X with the universal arrow η_X^H . From this it easily follows that T^H is the object assignment of a monad and that η^H and τ^H are natural transformations. Denote by κ^H the natural transformation $\tau^H \cdot H\eta^H : H \rightarrow T^H$.

Let (A, a) be a cia for H . Then there is a unique homomorphism $\tilde{a} : T^H A \rightarrow A$ of H -algebras such that $\tilde{a} \cdot \eta_A^H = \text{id}_A$. We call \tilde{a} the *evaluation morphism* associated to A . Notice that $\tilde{a} \cdot \kappa_A^H = a$.

Remark 4.1. Note that (A, \tilde{a}) is an Eilenberg-Moore algebra for the monad T^H . However, it is important to observe that not every Eilenberg-Moore algebra for T^H arises from a cia by forming \tilde{a} . Indeed, the category of Eilenberg-Moore algebras for T^H is isomorphic to the larger category of complete Elgot algebras for H , see [6].

In our previous work we have shown how to obtain unique solutions of more general (first-order) recursive equations than the flat ones appearing in the definition of a cia:

Definition 4.2. [3, 24] An *equation morphism* is a morphism of the form $e : X \rightarrow T^H(X + A)$. It is called *guarded* if there exists a factorization $f : X \rightarrow HT^H(X + A) + A$ such that

$$e = (X \xrightarrow{f} HT^H(X + A) + A \xrightarrow{[\tau_{X+A}^H, \eta_{X+A}^H \cdot \text{inr}]} T^H(X + A)).$$

A *solution* of an equation morphism e in a cia (A, a) is a morphism $e^\dagger : X \rightarrow A$ such that the following equation holds:

$$e^\dagger = (X \xrightarrow{e} T^H(X + A) \xrightarrow{T^H([e^\dagger, \text{id}_A])} T^H A \xrightarrow{\tilde{a}} A).$$

Theorem 4.3. [24] Let (A, a) be a cia for H . Then every guarded equation morphism has a unique solution in A .

An even more general property of cias was proved in [25]; one can solve recursive function definitions uniquely in a cia. We recall the respective result.

Definition 4.4. Let V be an endofunctor such that $H + V$ is iterable. A *recursive program scheme* (rps, for short) is a natural transformation $e : V \rightarrow T^{H+V}$. It is called *guarded* if there exists a natural transformation $f : V \rightarrow HT^{H+V}$ such that

$$e = (V \xrightarrow{f} HT^{H+V} \xrightarrow{\text{inl} T^{H+V}} (H + V)T^{H+V} \xrightarrow{\tau^{H+V}} T^{H+V}),$$

where $\text{inl} : H \rightarrow H + V$.

Now let (A, a) be a cia for H . An *interpreted solution* of e in A is a V -algebra structure $e_A^\ddagger : VA \rightarrow A$ giving rise to an Eilenberg-Moore algebra structure $\beta : T^{H+V}A \rightarrow A$ (i. e., $\beta \cdot \kappa_A^{H+V} = [a, e_A^\ddagger]$) such that we have

$$e_A^\ddagger = (VA \xrightarrow{e_A} T^{H+V}A \xrightarrow{\beta} A). \quad (4.1)$$

Theorem 4.5. [25] In a cia, every guarded rps has a unique interpreted solution.

We are now able to prove more:

Theorem 4.6. Let $e : V \rightarrow T^{H+V}$ be a guarded rps, and let $a : HA \rightarrow A$ be a cia. Then the interpreted solution $e_A^\ddagger : VA \rightarrow A$ extends the cia structure on A ; more precisely, the algebra $[a, e_A^\ddagger] : (H + V)A \rightarrow A$ is a cia for $H + V$.

Remark 4.7. For the proof we need to recall some technical details. Recall that any guarded rps $e : V \rightarrow T^{H+V}$ as in Definition 4.4 induces a natural transformation

$$\bar{e} : T^{H+V} \rightarrow HT^{H+V} + \text{Id}$$

(see [25], Lemma 6.9). The component \bar{e}_A of this natural transformation at A is a flat equation morphism with parameters in A . Its unique solution in the cia (A, a) is the Eilenberg-Moore algebra structure $\beta : T^{H+V}A \rightarrow A$ in (4.1) satisfying $[a, e_A^\ddagger] = \beta \cdot \kappa_A^{H+V}$ (this follows from [25], see Lemma 7.4 and the proof of Theorem 7.3).

Proof of Theorem 4.6. Let $m : X \rightarrow (H + V)X + A$ be a flat equation morphism. We need to prove that m has a unique solution s . As shortcut notations we shall write T' for T^{H+V} , $\tau'_X : (H + V)T'X \rightarrow T'X$ for the corresponding structure of a free cia for $H + V$ as well as μ' for the multiplication of the monad T' and $\kappa' = \tau' \cdot H\eta'$ (cf. the introduction to Section 4).

(1) Existence of a solution. Since $T'A$ is the terminal coalgebra for $(H + V)(-) + A$ we have a unique homomorphism $h : X \rightarrow T'A$. We show that

$$s = (X \xrightarrow{h} T'A \xrightarrow{\beta} A)$$

is a solution of m in the algebra $(A, [a, e_A^\ddagger])$. To see this, consider the diagram

$$\begin{array}{ccccc} X & \xrightarrow{h} & T'A & \xrightarrow{\beta} & A \\ m \downarrow & & \uparrow [\tau'_A, \eta'_A] & & \uparrow [a, e_A^\ddagger, A] \\ (H+V)X+A & \xrightarrow{(H+V)h+A} & (H+V)T'A+A & \xrightarrow{(H+V)\beta+A} & (H+V)A+A \end{array} \quad (4.2)$$

The left-hand square commutes since h is a coalgebra homomorphism, and for the right-hand component of the right-hand square use the unit law $\beta \cdot \eta'_A = \text{id}$ of the Eilenberg-Moore algebra β . It remains to prove the commutativity of the left-hand component. This is established by inspecting the diagram below:

$$\begin{array}{ccccc} & & \tau'_A & & \\ & \swarrow & & \searrow & \\ (H+V)T'A & \xrightarrow{\kappa'_{T'A}} & T'T'A & \xrightarrow{\mu'_A} & T'A \\ (H+V)\beta \downarrow & & T'\beta \downarrow & & \downarrow \beta \\ (H+V)A & \xrightarrow{\kappa'_A} & T'A & \xrightarrow{\beta} & A \\ & \curvearrowleft [a, e_A^\ddagger] & & \curvearrowright & \end{array}$$

An easy calculation shows the commutativity of the upper part, for the lower one see Remark 4.7, the left-hand inner square commutes due to naturality of κ' , and the right-hand inner square by one of the laws for the Eilenberg-Moore algebra β .

(2) Uniqueness of solutions. Suppose that e is a guarded rps with factor $f : V \rightarrow HT'$ as in Definition 4.4. From f and m we form a flat equation morphism $g : X + T'X \rightarrow H(X + T'X) + A$ w.r.t. H as follows. The left hand component of g is

$$g \cdot \text{inl} = (X \xrightarrow{m} HX + VX + A \xrightarrow{HX + f_X + A} HX + HT'X + A \xrightarrow{\text{can} + A} H(X + T'X) + A),$$

and the right-hand component of g is

$$g \cdot \text{inr} = (T'X \xrightarrow{\bar{e}_X} HT'X + X \xrightarrow{[\text{inl}, H\text{inr}, g \cdot \text{inl}]} H(X + T'X) + A).$$

Since A is a cia for H there exists a unique solution $g^\dagger : X + T'X \rightarrow A$. Now let $s : X \rightarrow A$ be any solution of the flat equation morphism m in the algebra $[a, e_A^\dagger] : (H + V)A \rightarrow A$. We will show below that $[s, \beta \cdot T's] : X + T'X \rightarrow A$ is a solution of g in the H -algebra (A, a) . So since (A, a) is a cia we have the following equation:

$$g^\dagger = [s, \beta \cdot T's] : X + T'X \rightarrow A. \quad (4.3)$$

Then s is uniquely determined by g^\dagger .

In order to prove Equation (4.3) we need to verify that the following square commutes:

$$\begin{array}{ccc} X + T'X & \xrightarrow{[s, \beta \cdot T's]} & A \\ g \downarrow & & \uparrow [a, A] \\ H(X + T'X) + A & \xrightarrow{H[s, \beta \cdot T's] + A} & HA + A \end{array} \quad (4.4)$$

We shall verify the commutativity of the two coproduct components separately. For the left-hand component we consider the diagram below:

$$\begin{array}{ccccc} & X & \xrightarrow{s} & A & \\ & m \downarrow & & \uparrow [a, e_A^\dagger, A] & \\ & HX + VX + A & \xrightarrow{Hs + Vs + A} & HA + VA + A & \\ & HX + f_X + A \downarrow & & HA + [e_A^\dagger, A] \downarrow & \\ & HX + HT'X + A & & & \\ & \text{can} + A \downarrow & & [Hs, H(\beta \cdot T's)] + A \searrow & \\ & H(X + T'X) + A & \xrightarrow{H[s, \beta \cdot T's] + A} & HA + A & \end{array} \quad (4.5)$$

The left-hand part commutes by the definition of g , the right-hand part commutes trivially, the upper square commutes since s is a solution of m and the lower triangle commutes trivially, again. It remains to verify that the middle part commutes. We check the commutativity of this part componentwise: the left-hand and right-hand components commute. We do not claim that the middle component commutes. However, in order to prove that the overall outside of (4.5) commutes, we need only show that this middle component commutes when extended by $[a, A] : HA + A \rightarrow A$.

To see this consider the next diagram:

$$\begin{array}{ccccc}
 & VX & \xrightarrow{Vs} & VA & \xrightarrow{e_A^\dagger} A \\
 e_X \downarrow & & & e_A \downarrow & \nearrow \beta \\
 T'X & \xrightarrow{T's} & T'A & & \\
 \tau'_X \cdot \text{inl}_{T'X} \uparrow & & \tau'_A \cdot \text{inl}_{T'A} \uparrow & & \\
 \rightarrow HT'X & \xrightarrow{HT's} & HT'A & \xrightarrow{H\beta} & HA
 \end{array}$$

This diagram commutes: the left-hand part commutes since e is a guarded rps; the upper and lower left squares commute due to the naturality of e and of $\tau' : (H + V)T' \rightarrow T'$ and $\text{inl}T' : HT' \rightarrow (H + V)T'$, respectively; the right-hand triangle commutes since e_A^\dagger is an interpreted solution of the rps e . Finally, to see that the lower right-hand part commutes recall from Diagram (4.2) that

$$[a, e_A^\dagger] \cdot (H + V)\beta = \beta \cdot \tau'_A.$$

Compose this last equation with the coproduct injection $\text{inl}_{T'A} : HT'A \rightarrow (H + V)T'A$ to obtain the desired commutativity.

Finally, we verify that the right-hand component of (4.4) commutes. Indeed, consider the diagram below:

$$\begin{array}{ccccc}
 & T'X & \xrightarrow{T's} & T'A & \xrightarrow{\beta} A \\
 \bar{e}_X \downarrow & & & \bar{e}_A \downarrow & \\
 HT'X + X & \xrightarrow{HT's+s} & HT'A + A & & \\
 [\text{inl}, H\text{inr}, g \cdot \text{inl}] \downarrow & & & \searrow H\beta + A & \\
 \rightarrow H(X + T'X) + A & \xrightarrow{H[s, \beta, T's] + A} & HA + A & & [a, A]
 \end{array}$$

The left-hand part commutes by the definition of g ; the upper middle square commutes by the naturality of \bar{e} , the right-hand part of the above diagram commutes since β is the solution of \bar{e}_A in the cia (A, a) (see Remark 4.7), and for the lower middle part we consider the components separately: the left-hand component clearly commutes by the functoriality of H , and for the right-hand component observe that it commutes when extended by $[a, A] : HA + A \rightarrow A$ (see Diagram (4.5)). Thus, the outside of the above diagram commutes, and this completes the proof. \square

The last result implies that operations obtained as solutions of recursive program schemes can be used in subsequent recursive function definitions, which will still have unique solutions. For the special case of interpreted rps solutions in cias this strengthens the results in [26].

Coming back to our setting in Section 3, let $\ell : K(H \times \text{Id}) \rightarrow HM$ be an abstract GSOS rule, where M is the free monad \widehat{K} (or, more generally, let λ be a distributive law of an arbitrary monad over the cofree copointed functor $H \times \text{Id}$). Assume furthermore that the composite HM is iterable. By applying the two Theorems 4.3 and 4.5 and also Theorem 4.6 to the cia $k : HMC \rightarrow C$ from Theorem 3.4 we get two more solutions theorems for free:

Corollary 4.8. *Every guarded equation morphism $e : X \rightarrow T^{HM}(X + C)$ has a unique solution in the cia (C, k) .*

Corollary 4.9. *Every guarded rps $e : V \rightarrow T^{HM+V}$ has a unique interpreted solution in the cia (C, k) , and this solution extends the cia structure on C .*

Assuming that MHM is iteratable two similar theorems hold for the cia $k' : MHMC \rightarrow C$ obtained from Theorem 3.5:

Corollary 4.10. *Every guarded equation morphism $e : X \rightarrow T^{MHM}(X+C)$ has a unique solution in the cia (C, k') .*

Corollary 4.11. *Every guarded rps $e : V \rightarrow T^{MHM+V}$ has a unique interpreted solution in the cia (C, k') , and this solution extends the cia structure on C .*

5. RECURSIVE FUNCTION DEFINITIONS OVER THE BEHAVIOR

Even with all the results we have seen so far, we are still not able to obtain functions such as the shuffle product \otimes on streams (see introduction) as a unique solution since its definition refers to the behavior of the arguments of the function. Notice also that the specification of \otimes makes use of the stream addition $+$, so this operation is assumed as given or previously specified and the specification of \otimes is built on top of the specification of $+$.

In this section we start with an abstract GSOS rule $\ell : K(H \times \text{Id}) \rightarrow H\widehat{K}$ specifying all given operations (such as the stream addition). We introduce a special form of rule called *recursive program scheme w.r.t. ℓ* (or, ℓ -rps, for short) specifying new operations in terms of given ones (such as the shuffle product of streams). We prove that every ℓ -rps has a unique solution in the terminal coalgebra C , and this solution extends the cia structure for $H\widehat{K}$ on C given by Theorem 3.4—this is a compositionality result similar to the one given in Theorem 4.6 for ordinary rps.

We show that every ℓ -rps easily gives rise to a “composed” abstract GSOS rule, and so the results in this section are essentially an application of the work in [11, 12] and our results in Section 3.

Assumption 5.1. In addition to our assumptions in 2.1 we assume that an abstract GSOS rule $\ell : K(H \times \text{Id}) \rightarrow H\widehat{K}$ is given. We shall also use free monads of other functors than K , and we follow the convention that whenever we write \widehat{F} for a functor F we assume that the free monad \widehat{F} exists and is given objectwise by free algebras for F , cf. Remark 2.2(2).

Definition 5.2. A recursive program scheme w.r.t. ℓ (shortly, ℓ -rps) is a natural transformation

$$e : V(H \times \text{Id}) \rightarrow H\widehat{K} + V.$$

An interpreted solution of e in C is a V -algebra structure $s : VC \rightarrow C$ such that

$$s = (VC \xrightarrow{V\langle c, \text{id}_C \rangle} V(HC \times C) \xrightarrow{e_C} H\widehat{K} + VC \xrightarrow{H[\widehat{b}, s]} HC \xrightarrow{c^{-1}} C), \quad (5.1)$$

where $b : KC \rightarrow C$ is the ℓ -interpretation in C , cf. Definition 2.9.

Theorem 5.3. *For every ℓ -rps there exists a unique interpreted solution s in C . In addition, s extends the cia structure on C , i.e., the following is the structure of a cia for $H\widehat{K} + V$ on C :*

$$H\widehat{K} + V(C) \xrightarrow{H[\widehat{b}, s]} HC \xrightarrow{c^{-1}} C. \quad (5.2)$$

Before we proceed to the proof of this theorem we set up some notation, provide the construction of the unique solution s and establish a technical lemma.

Notation 5.4. (1) Let F and G be endofunctors of \mathcal{A} . The coproduct injections $\text{inl} : F \rightarrow F + G \leftarrow G : \text{inr}$ lift to monad morphisms on the corresponding free monads, and we denote those monad morphisms by $\widehat{\text{inl}} : \widehat{F} \rightarrow \widehat{F+G} \leftarrow \widehat{G} : \widehat{\text{inr}}$.

(2) Recall from Remark 2.2(3) that for a functor F on \mathcal{A} we write $\varphi : F\widehat{F} \rightarrow \widehat{F}$ and $\eta : \text{Id} \rightarrow \widehat{F}$ for (the natural transformations given by) the structures and universal morphisms of the free F -algebras as well as $\mu : \widehat{F}\widehat{F} \rightarrow \widehat{F}$ and $\kappa : F \rightarrow \widehat{F}$ for the multiplication and universal natural transformation of the free monad \widehat{F} .

Remark 5.5. (1) Notice that the monad morphism $\widehat{\text{inl}} : \widehat{F} \rightarrow \widehat{F+G}$ is uniquely determined by the commutativity of the following square of natural transformations:

$$\begin{array}{ccc} F & \xrightarrow{\kappa} & \widehat{F} \\ \text{inl} \downarrow & & \downarrow \widehat{\text{inl}} \\ F + G & \xrightarrow{\kappa} & \widehat{F+G} \end{array}$$

(2) Recall from Notation 2.3 that for every F -algebra (A, a) we have the corresponding Eilenberg-Moore algebra $\widehat{a} : \widehat{F}A \rightarrow A$ and that $a = \widehat{a} \cdot \kappa_A$. Moreover, the categories of F -algebras and of Eilenberg-Moore algebras for \widehat{F} are isomorphic; more precisely, $a \mapsto \widehat{a}$ and precomposition with κ_A extend to mutually inverse functors.

(3) Combining parts (1) and (2) of this remark, we see that for every algebra $a : (F + G)A \rightarrow A$ the equation $\widehat{a} \cdot \widehat{\text{inl}}_A = \widehat{a \cdot \text{inl}_A}$ holds. Indeed, both sides are equal when precomposed with κ_A :

$$\widehat{a} \cdot \widehat{\text{inl}}_A \cdot \kappa_A = \widehat{a} \cdot \kappa_A \cdot \text{inl}_A = a \cdot \text{inl}_A = \widehat{a \cdot \text{inl}_A} \cdot \kappa_A.$$

If we make the coproduct algebra structure explicit as in $a = [a_0, a_1]$, we obtain

$$\widehat{[a_0, a_1]} \cdot \widehat{\text{inl}}_A = \widehat{a_0} \quad \text{and} \quad \widehat{[a_0, a_1]} \cdot \widehat{\text{inr}}_A = \widehat{a_1}.$$

Construction 5.6. Throughout the rest of this section we are going to write F for $K + V$.

Let $e : V(H \times \text{Id}) \rightarrow H\widehat{F}$ be an ℓ -rps. This gives an abstract GSOS rule $n : F(H \times \text{Id}) \rightarrow H\widehat{F}$ defined componentwise as displayed below:

$$\begin{array}{ccc} V(H \times \text{Id}) & & \\ \text{inr}(H \times \text{Id}) \downarrow & \searrow e & \\ F(H \times \text{Id}) & \xrightarrow{n} & H\widehat{F} \\ \text{inl}(H \times \text{Id}) \uparrow & & \uparrow H\widehat{\text{inl}} \\ K(H \times \text{Id}) & \xrightarrow{\ell} & H\widehat{K} \end{array}$$

We write $a : FC \rightarrow C$ for the n -interpretation in C . Define

$$s = (VC \xrightarrow{\text{inr}} FC \xrightarrow{a} C)$$

We shall prove that s is the unique solution of e in C .

Lemma 5.7. Let $b : KC \rightarrow C$ be the interpretation of $\ell : K(H \times \text{Id}) \rightarrow H\widehat{K}$. Then for $a : FC \rightarrow C$ from Construction 5.6 we have

$$b = (KC \xrightarrow{\text{inl}_C} FC \xrightarrow{a} C)$$

Proof. Consider the diagram below:

$$\begin{array}{ccccc}
 KC & \xrightarrow{K(c,\text{id}_C)} & K(H \times \text{Id})C & \xrightarrow{\ell_C} & H\widehat{K}C \\
 \text{inl}_C \downarrow & & \text{inl}_{(H \times \text{Id})C} \downarrow & & \downarrow H\widehat{\text{inl}}_C \\
 FC & \xrightarrow{F(c,\text{id}_C)} & F(H \times \text{Id})C & \xrightarrow{n_C} & H\widehat{F}C \\
 a \downarrow & & c & & \downarrow H\widehat{a} \\
 C & & & & HC
 \end{array}
 \quad \boxed{H(\widehat{a \cdot \text{inl}_C})}$$

The lower square commutes by Theorem 2.7 and the upper right-hand one by the definition of n (cf. Construction 5.6). The left-hand square commutes by the naturality of $\text{inl} : K \rightarrow F$, and for the right-hand part we remove H and use Remark 5.5(3).

Now recall that b is uniquely determined by the commutativity of the diagram in Theorem 2.7. Thus, $a \cdot \text{inl}_C = b$ holds as desired. \square

Corollary 5.8. The n -interpretation in C is $a = [b, s] : FC \rightarrow C$ and so we have

$$\widehat{a} = \widehat{[b, s]} : \widehat{FC} \rightarrow C.$$

Proof of Theorem 5.3. We are ready to prove that s in Construction 5.6 is the unique solution of the ℓ -rps $e : V(H \times \text{Id}) \rightarrow H\widehat{F}$ in C .

(1) We prove that s is a solution of e in C . Indeed, consider the commutative diagram

$$\begin{array}{ccccc}
 VC & \xrightarrow{V(c,\text{id}_C)} & V(H \times \text{Id})C & & \\
 \text{inr}_C \downarrow & & \text{inr}_{(H \times \text{Id})C} \downarrow & & \searrow e_C \\
 s \left\{ \begin{array}{c} \downarrow \\ FC \xrightarrow{F(c,\text{id}_C)} F(H \times \text{Id})C \xrightarrow{n_C} H\widehat{F}C \\ \downarrow a \\ C \xrightarrow{c} HC \end{array} \right. & & & & \downarrow H\widehat{a} \\
 & & & &
 \end{array} \tag{5.3}$$

The lower square commutes by Theorem 2.7, and since all other parts also commute, we see that s is a solution of e since $\widehat{a} = \widehat{[b, s]}$ holds by Corollary 5.8.

(2) We now prove that s is unique. Suppose that t is any solution of e . We will prove that

$$[b, t] = a, \tag{5.4}$$

which implies the desired equation $s = t$.

In order to prove (5.4) we have to verify the commutativity of the following diagram (cf. Theorem 2.7):

$$\begin{array}{ccc}
 FC & \xrightarrow{F(c,\text{id}_C)} & F(H \times \text{Id})C \xrightarrow{n_C} H\widehat{F}C \\
 [b,t] \downarrow & & \downarrow H\widehat{[b,t]} \\
 C & \xrightarrow{c} & HC
 \end{array}$$

We verify this for the two coproduct components of $FC = KC + VC$ separately.

For the right-hand component we obtain the above diagram (5.3) with s replaced by t and a by $[b, t]$, which commutes since t is a solution of e . For the left-hand component we obtain the diagram below:

$$\begin{array}{ccccccc}
 KC & \xrightarrow{K\langle c, \text{id}_C \rangle} & K(H \times \text{Id})C & \xrightarrow{\text{inl}_{(H \times \text{Id})C}} & F(H \times \text{Id})C & \xrightarrow{n_C} & H\widehat{F}C \\
 b \downarrow & & \searrow \ell_C & & \nearrow H\widehat{\text{inl}}_C & & \downarrow H[b, t] \\
 C & \xrightarrow{c} & H\widehat{K}C & & H\widehat{C} & \xrightarrow{H\widehat{b}} & HC
 \end{array}$$

The big left-hand part commutes by Theorem 2.7, the upper triangle commutes by the definition of n (see Construction 5.6), and for the right-hand triangle remove H and notice that

$$\widehat{[b, t]} \cdot \widehat{\text{inl}}_C = \widehat{b}$$

by Remark 5.5(3).

(3) To complete the proof we will show that $c^{-1} \cdot H\widehat{[b, s]} : H\widehat{F}C \rightarrow C$ is the structure of a cia for $H\widehat{F}$. But this is a consequence of Theorem 3.4; indeed, recall that $[b, s]$ is the interpretation of the abstract GSOS rule $n : F(H \times \text{Id}) \rightarrow H\widehat{F}$ in C , see Corollary 5.8. \square

Remark 5.9. (1) Notice that the fact that the unique solution s of an ℓ -rps extends the cia structure on C means that the operations on C defined in this way may be part of recursive definitions according to the Corollaries 4.8, 4.9 (where $M = \widehat{K + V}$).

(2) In addition, we have a compositionality principle for solutions of ℓ -rps's—the operations provided by the unique solution $s : VC \rightarrow C$ may occur as givens in subsequent functional recursive definitions (and we will make use of this feature in our applications in Section 6). More precisely, we have seen in Construction 5.6 that the ℓ -rps e gives rise to the abstract GSOS rule $n : F(H \times \text{Id}) \rightarrow H\widehat{F}$, where $F = K + V$, with the interpretation $[b, s] : FC \rightarrow C$. And by Theorem 5.3 every n -rps $W(H \times \text{Id}) \rightarrow H\widehat{F + W}$ has a unique solution $WC \rightarrow C$.

Theorem 5.10. Let $e_i : V_i(H \times \text{Id}) \rightarrow H\widehat{K + V_i}$, $i = 1, 2$, be two ℓ -rps's. Then the cia structure on C extended by the unique solutions $s_i : V_iC \rightarrow C$ of the e_i is independent of the order of extension.

Remark 5.11. More precisely, we may first take $s_1 : V_1C \rightarrow C$ to obtain an extended cia structure as in (5.2), and then take the solution of $s_2 : V_2C \rightarrow C$ in the new cia, or vice versa. Either way, the resulting extended cia structure is

$$H(K + \widehat{V_1 + V_2})C \xrightarrow{H\widehat{[b, s_1, s_2]}} HC \xrightarrow{c^{-1}} C. \quad (5.5)$$

Proof of Theorem 5.10. It is sufficient to prove that the cia structure on C obtained by extending $k : H\widehat{K}C \rightarrow C$ first by s_1 and then by s_2 is (5.5).

So take s_1 and extend the cia structure (C, k) to obtain the cia $c^{-1} \cdot H\widehat{[b, s_1]} : H\widehat{K + V_1}C \rightarrow C$ (cf. (5.2)). Recall from the proof of Theorem 5.3 that this cia structure is obtained as follows: one first forms the abstract GSOS rule

$$n = [H\widehat{\text{inl}} \cdot \ell, e_1] : (K + V_1)(H \times \text{Id}) \rightarrow H\widehat{K + V_1}$$

whose interpretation is $b' = [b, s_1] : K + V_1(C) \rightarrow C$, cf. Corollary 5.8, and then one applies Theorem 3.4.

Now we form the following n -rps

$$V_2(H \times \text{Id}) \xrightarrow{e_2} H\widehat{K} + \widehat{V}_2 \xrightarrow{H[\text{inl,inr}]} H(K + \widehat{V}_1 + V_2)$$

Its unique solution is easily seen to be s_2 ; indeed, consider the following diagram (and notice that the right-hand arrow is $H[b', s_2]$):

$$\begin{array}{ccccc} V_2C & \xrightarrow{V_2(c, \text{id}_C)} & V_2(H \times \text{Id})C & \xrightarrow{(e_2)_C} & H\widehat{K} + \widehat{V}_2C \xrightarrow{H[\text{inl,inr}]_C} H(K + \widehat{V}_1 + V_2)C \\ s_2 \downarrow & & & & \searrow H[b', s_2] \\ C & \xleftarrow{c^{-1}} & & & HC \downarrow H[b, s_1, s_2] \end{array}$$

The left-hand part commutes since s_2 is the unique solution of e_2 and for the right-hand part we remove H and then precompose with $\kappa_C^{K+V_2}$ to obtain

$$\begin{aligned} [b, s_1, s_2] \cdot [\text{inl, inr}]_C \cdot \kappa_C &= [b, s_1, s_2] \cdot \kappa_C \cdot [\text{inl, inr}]_C && \text{cf. Remark 5.5(1)} \\ &= [b, s_1, s_2] \cdot [\text{inl, inr}]_C && \text{see Remark 5.5(2)} \\ &= [b, s_2] \\ &= \widehat{[b, s_2]} \cdot \kappa_C && \text{see Remark 5.5(2)} \end{aligned}$$

The desired equality now follows since precomposition with κ is an isomorphism of categories, see Remark 5.5(2). \square

Remark 5.12. Notice that we can always consider the algebraic operation obtained from $c^{-1} : HC \rightarrow C$ as a given operation in any ℓ -rps for an abstract GSOS rule $\ell : K(H \times \text{Id}) \rightarrow H\widehat{K}$. More precisely, we can assume that $K = K' + H$ and that the ℓ -interpretation $b : KC \rightarrow C$ has the form $b = [b', c^{-1}]$. Indeed, given any abstract GSOS rule $\ell' : K'(H \times \text{Id}) \rightarrow H\widehat{K}'$ with ℓ' -interpretation b' , we define the ℓ' -rps

$$e = (H(H \times \text{Id}) \xrightarrow{H\pi_0} HH \xrightarrow{H\text{inr}} H(K' + H) \xrightarrow{H\kappa} H\widehat{K}' + \widehat{H}).$$

It is easy to verify that c^{-1} is its solution; and, as we see from the proof of Theorem 5.3, we obtain a new abstract GSOS rule $n : (K' + H)(H \times \text{Id}) \rightarrow H\widehat{K}' + \widehat{H}$ defined as in Construction 5.6 and having the n -interpretation $[b', c^{-1}]$. According to Theorem 5.10 we can do this construction at any step when defining operations, the result being always a GSOS rule $\ell : K(H \times \text{Id}) \rightarrow H\widehat{K}$ with $K = K' + H$ and an ℓ -interpretation $[b', c^{-1}]$ containing the algebraic structure c^{-1} .

We will now prove a version of the Sandwich Theorem 3.5 for ℓ -rps's. The goal is to be able to uniquely solve specifications from a wider class. Let us explain the idea with the help of an example—streams (cf. Example 2.10(2)). Here we have $HX = \mathbb{R} \times X$ on Set. Suppose that K and V both are polynomial functors associated to a signature of givens and newly defined operations on the terminal coalgebra $C = \mathbb{R}^\omega$. The inverse of the coalgebra structure $c = \langle \text{hd}, \text{tl} \rangle$ yields the family of prefix operations $r.(-)$ prepending the number r to a stream. The format of an ℓ -rps means that the new operations of type V are always defined by an equation with a prefix operation as a guard at its head, see e. g. the following specification of the shuffle product reformulated from the behavioral differential equation of the introduction:

$$r.x \otimes s.y = rs.((x \otimes s.y) + (r.x \otimes y)).$$

This guard is sufficient to ensure a unique solution, however its position at the head of the term on the right-hand side is not necessary. We shall provide a result that makes this precise.

Definition 5.13. A *sandwiched recursive program scheme w.r.t. ℓ* (shortly, ℓ -srps) is a natural transformation $e : V(H \times \text{Id}) \rightarrow \widehat{K}H\widehat{K} + \widehat{V}$.

An *interpreted solution* of e in C is a V -algebra structure $s : VC \rightarrow C$ such that

$$s = (VC \xrightarrow{V\langle c, \text{id}_C \rangle} V(HC \times C) \xrightarrow{e_C} \widehat{K}H\widehat{K} + \widehat{V}C \xrightarrow{\widehat{K}H[\widehat{b}, s]} \widehat{K}HC \xrightarrow{\widehat{K}c^{-1}} \widehat{K}C \xrightarrow{\widehat{b}} C).$$

Theorem 5.14. (Sandwich Theorem for ℓ -srps's) *For every ℓ -srps there exists a unique interpreted solution s in C . In addition, s extends the cia structure on C , more precisely, the following is the structure of a cia for $\widehat{K} + \widehat{V}H\widehat{K} + \widehat{V}$ on C :*

$$\widehat{K} + \widehat{V}H\widehat{K} + \widehat{V}(C) \xrightarrow{\widehat{K} + \widehat{V}H[\widehat{b}, s]} \widehat{K} + \widehat{V}HC \xrightarrow{\widehat{K} + \widehat{V}c^{-1}} \widehat{K} + \widehat{V}C \xrightarrow{[b, s]} C. \quad (5.6)$$

Proof. In order to simplify notation we will write (M, η, μ) for the free monad $\widehat{K} + \widehat{V}$ throughout the proof. Recall from Remark 5.12 that we can assume $K = K' + H$ and that the ℓ -interpretation has the form $[b', c^{-1}]$. Furthermore recall from Remark 2.6(2) that ℓ gives rise to a distributive law $\lambda : \widehat{K}(H \times \text{Id}) \rightarrow (H \times \text{Id})\widehat{K}$ of the free monad \widehat{K} over the cofree copointed functor $H \times \text{Id}$.

Given an ℓ -srps $e : V(H \times \text{Id}) \rightarrow \widehat{K}HM$, we form the (ordinary) ℓ -rps $\bar{e} : V(H \times \text{Id}) \rightarrow HM$ by defining

$$\begin{aligned} \bar{e} = (V(H \times \text{Id}) &\xrightarrow{e} \widehat{K}HM \xrightarrow{\widehat{K}\langle HM, \text{inl } M \rangle} \widehat{K}(HM \times (K' + H + V)M) \\ &\quad \underbrace{\qquad\qquad\qquad}_{\widehat{K}(HM \times \varphi)} \\ &\quad \downarrow \\ &= \widehat{K}(H \times \text{Id})M \xrightarrow{\lambda M} (H \times \text{Id})\widehat{K}M \xrightarrow{\pi_0 \widehat{K}M} H\widehat{K}M \xrightarrow{H\text{inl } M} HMM \xrightarrow{H\mu} HM). \end{aligned}$$

Then we verify that solutions of e and \bar{e} are in one-to-one correspondence. Consider the following diagram (we drop the indices denoting components of natural transformations):

$$\begin{array}{ccccc}
& & s & & \\
& \nearrow V\langle c, \text{id} \rangle & & \searrow \widehat{b} & \\
VC & \xrightarrow{\quad} & C & \xleftarrow{\quad} & \\
\downarrow & & \downarrow & & \downarrow \\
V(HC \times C) & \xrightarrow{\quad \text{(i)} \quad} & & & \\
\downarrow e & & & & \\
\widehat{K}HMC & \xrightarrow{\widehat{K}H[\widehat{b}, s]} & \widehat{K}HC & \xrightarrow{\widehat{K}c^{-1}} & \widehat{K}C \\
\downarrow \widehat{K}\langle \text{id}, \varphi \cdot \text{inm } M \rangle & & \downarrow \widehat{K}\langle c, \text{id} \rangle & & \downarrow \langle c, \text{id} \rangle \\
\widehat{K}(H \times \text{Id})MC & \xrightarrow{\widehat{K}(H \times \text{Id})[\widehat{b}, s]} & \widehat{K}(H \times \text{Id})C & & \\
\downarrow \lambda M & & \downarrow \lambda & & \downarrow \text{(v)} \\
(H \times \text{Id})\widehat{K}MC & \xrightarrow{(H \times \text{Id})\widehat{K}[\widehat{b}, s]} & (H \times \text{Id})\widehat{K}C & \xrightarrow{(H \times \text{Id})\widehat{b}} & (H \times \text{Id})C \\
\downarrow \pi_0 \widehat{K}M & & \downarrow \pi_0 \widehat{K} & & \downarrow \pi_0 \\
H\widehat{K}MC & \xrightarrow{H\widehat{K}[\widehat{b}, s]} & H\widehat{K}C & & \text{(viii)} \\
\downarrow H\widehat{\text{inl}} M & & \downarrow H\widehat{\text{inl}} & & \downarrow \pi_0 \\
HMMC & \xrightarrow{HM[\widehat{b}, s]} & HMC & & \text{(ix)} \\
\downarrow H\mu & & \downarrow H\widehat{b} & & \downarrow H\widehat{b} \\
\rightarrow HMC & \xrightarrow{\quad \text{(xi)} \quad} & H\widehat{[b, s]} & \xrightarrow{\quad H\widehat{[b, s]} \quad} & HC \leftarrow
\end{array}$$

(5.7)

We first show that all inner parts except part (i) commute: for part (ii) use the definition of \bar{e} , part (iv) is the commutative diagram in (2.4), part (v) trivially commutes, the parts (vi), (vii), (viii) and (ix) commute by the naturality of λ , $\pi_0 : H \times \text{Id} \rightarrow H$ and $\text{inl} : \widehat{K} \rightarrow M$, respectively, for part (x) use Remark 5.5(3), and part (xi) commutes since $[b, s]$ is the structure of an Eilenberg-Moore algebra for M . It remains to verify the commutativity of part (iii); we remove \widehat{K} and consider the product components separately: the left-hand component commutes using $c \cdot c^{-1} = \text{id}_C$ and for the right-hand one we have the diagram

$$\begin{array}{ccccc}
& & H[\widehat{b,s}] & & \\
HMC & \xrightarrow{\hspace{10cm}} & & & HC \\
\downarrow \text{inm}M & & & & \downarrow c^{-1} \\
(K' + H + V)MC & \xrightarrow{(K' + H + V)[\widehat{b,s}]} & (K' + H + V)C & \xleftarrow{\hspace{10cm}} & C \\
\downarrow \varphi & & & & \downarrow [b,s] \\
MC & \xrightarrow{\hspace{10cm}} & \widehat{[b,s]} & \xrightarrow{\hspace{10cm}} & C
\end{array}$$

Its upper part commutes by the naturality of $\text{inm} : H \rightarrow K' + H + V$, for the lower part recall from Notation 2.3 the definition of $\widehat{[b, s]}$ as a homomorphism of algebras for $K' + H + V$, and the right-hand triangle commutes since $b = [b', c^{-1}]$, see Remark 5.12.

Now if s is a solution of \bar{e} , then the outside of Diagram (5.7) commutes and therefore part (i) commutes proving s to be a solution of e . Conversely, if s is a solution of e part (i) commutes and then s is also a solution of \bar{e} . By Theorem 5.3, \bar{e} has a unique solution; thus, e has a unique solution, too.

We still need to prove that (5.6) is the structure of a cia for MHM . But this follows from Theorem 3.5. Indeed, from the ℓ -rps \bar{e} we form the abstract GSOS rule $n : (K+V)(H \times \text{Id}) \rightarrow HM$ analogously as in Construction 5.6, and the n -interpretation is $[b, s]$ for the unique solution s of \bar{e} (or, equivalently, of e). Now (5.6) is the structure k' from the statement of Theorem 3.5. \square

6. APPLICATIONS

In this section we present five applications illustrating how to use our results from Section 3–5 to obtain unique solutions of recursive equations in five different areas of theoretical computer science.

6.1. Process Algebras. Recall Example 2.10(3) where $HX = \mathcal{P}_\kappa(A \times X)$. We shall first explain more in detail how the abstract GSOS rule ℓ is obtained. Recall that K is the polynomial functor corresponding to the types of the CCS combinators, i.e., KX is a coproduct of the following components:

- $A \times X$ for agent expressions $a.x$, where $a \in A$,
- $\coprod_{n < \kappa} X^n$ for agent expressions $\sum_{i=1}^n x_i$, where $n < \kappa$,
- $X \times X$ for agent expressions $x_1|x_2$,
- $\coprod_f X$ for relabelling $x[f]$, where f ranges over functions on the action set $A \setminus \{\tau\}$ with $\overline{f(a)} = f(\bar{a})$, and
- $\coprod_{L \subseteq A \setminus \{\tau\}} X$ for restriction $x \setminus L$.

The abstract GSOS rule $\ell : K(H \times \text{Id}) \rightarrow H\widehat{K}$ is given by the sos rules in Example 2.10(3) in terms of the components of the coproduct $K(H \times \text{Id})$, i.e., for each combinator separately:

- $\ell_X(a, S, x) = \{(a, x)\}$ for prefixing $a.x$, where $S \subseteq A \times X$,
- $\ell_X((S_i, x_i)_{i < n}) = \bigcup_{i < n} S_i$ for summation $\sum_{i=1}^n x_i$ for every $n < \kappa$, where $(S_i)_{i < n}$ is a family of sets $S_i \subseteq A \times X$,
- $\ell_X(S_1, x_1, S_2, x_2)$ is given by the union of the three sets

$$\{(a, x|x_2) \mid (a, x) \in S_1\}, \quad \{(a, x_1|x) \mid (a, x) \in S_2\} \quad \text{and}$$

$$\{(\tau, x|y) \mid (a, x) \in S_1, (\bar{a}, y) \in S_2 \text{ for some } a \in A \setminus \{\tau\}\}$$

for parallel composition $x_1|x_2$, where $S_1, S_2 \subseteq A \times X$,

- $\ell_X(S, x) = \{(f(a), y[f]) \mid (a, y) \in S\}$ for relabelling $x[f]$ (here we mean $f(\tau) = \tau$), and
- $\ell_X(S, x) = \{(a, y \setminus L) \mid (a, y) \in S, a, \bar{a} \notin L\}$ for restriction $x \setminus L$.

The form of these definitions is very similar to the ones given by Aczel [2] in the setting of non-well-founded set theory. We already mentioned the ℓ -interpretation $b : KC \rightarrow C$ giving the desired operations on agents, and this gives the two new cia structures for $H\widehat{K}$ and $\widehat{K}H\widehat{K}$ as in Theorems 3.4 and 3.5.

Remark 6.1. If we replaced the second component $\coprod_{n < \kappa} X^n$ of K by $\mathcal{P}_\kappa X$ we still have an abstract GSOS rule. Furthermore, in both cases the induced (binary) operation of summation is automatically commutative, associative and idempotent: these three laws that have to be proved in process theory come “for free” by encoding them in the abstract GSOS rule using the union operation.

Now let us recall Milner's solution theorem for CCS agents from [28]. Suppose that $E_i, i \in I$, are agent expressions with the free variables $x_i, i \in I$. Suppose further that each variable x_j in each $E_i, i, j \in I$ is *weakly guarded*, i. e., it only occurs within the scope of some prefix combinator $a.-$. Then there is a unique solution of the recursive system $x_i = E_i$ of equations. More precisely, let \sim denote strong bisimilarity, and let $E_i[\vec{P}/\vec{x}]$ denote simultaneous substitution of P_j for x_j for every j . Then we have

Theorem 6.2. [28] *There exist, up to \sim , unique CCS agents P_i such that $P_i \sim E_i[\vec{P}/\vec{x}]$ holds for each $i \in I$.*

It is easy to see that this theorem is a consequence of our Theorem 2.14; a system $x_i = E_i$ where each variable is weakly guarded is essentially the same as a map $X \rightarrow \widehat{KHKX}$, where $X = \{x_i \mid i \in I\}$. Now our Theorem 3.5 generalizes Theorem 2.14 to flat equation morphisms $X \rightarrow \widehat{KHKX} + C$. These again have unique solutions in C . The extra summand C allows us to use constant agents in recursive specifications. So, for example, we can obtain the agent P as the unique solution of

$$x = a.(x|c) + b$$

in the introduction and then use it in a system like

$$x = b.(x + y) \quad y = P$$

which has a unique solution by Theorem 3.5.

Moreover, suppose we want to define the binary combinator “alt” which performs alternation of two processes. For its definition we shall need another binary combinator, sequential composition of two processes (denoted by the infix “;”). Here we suppose that the latter combinator is already included in our basic calculus—more precisely, we could have added a sixth coproduct component $X \times X$ to K for sequential composition in the above definition and could have completed the above ℓ by

$$\ell_X(S_1, x_1, S_2, x_2) = \begin{cases} \{(a, x; x_2) \mid (a, x) \in S_1\} & \text{if } S_1 \neq \emptyset \\ S_2 & \text{if } S_1 = \emptyset \end{cases}$$

for this coproduct component. This gives indeed the desired combinator for sequential composition. Observe in particular that Theorem 6.2 still holds for the calculus including this sixth combinator. Now for this extended ℓ we give a sandwiched ℓ -rps $e : V(H \times \text{Id}) \rightarrow \widehat{KHK} + \widehat{V}$, where $VX = X \times X$, in order to define the combinator alt:

$$e_X(S_1, x_1, S_2, x_2) = \begin{cases} S_1; \{(a, x; \text{alt}(x_1, x_2)) \mid (a, x) \in S_2\} & \text{if } S_2 \neq \emptyset \\ \{(a, x; \text{alt}(x_2, x_1)) \mid (a, x) \in S_1\} & \text{if } S_1 \neq \emptyset, S_2 = \emptyset \\ \emptyset & \text{if } S_1 = S_2 = \emptyset. \end{cases}$$

Notice that the term in the first line of this definition does not lie in $\widehat{H} \widehat{K} + \widehat{V}$, so Theorem 5.3 cannot be applied. But it does lie in $\widehat{KHK} + \widehat{V}$, so Theorem 5.14 tells us that e has a unique solution $s : C \times C \rightarrow C$. It is not difficult to see that this is alternation of processes indeed. Furthermore, Theorem 5.14 tells us that s extends the cia structure from Theorem 3.5 applied to the extended abstract GSOS rule ℓ . This means that Theorem 6.2 remains true for the calculus extended by sequential composition and alternation of processes, without further work.

Finally, suppose we want to define two unary combinators op_1 and op_2 by the rule

$$\frac{E \xrightarrow{a} F}{\text{op}_1(E) \xrightarrow{a} F|\text{op}_2(F+E) \quad \text{op}_2(E) \xrightarrow{a} F + \text{op}_1(F|E)}.$$

Then Theorem 5.3 tells us that this rule uniquely determines the two combinators. Indeed, we translate the rule into an ℓ -rps: let $V = \text{Id} + \text{Id}$ (two unary combinators are specified) and let $e : V(H \times \text{Id}) \rightarrow \widehat{HK + V}$ be given by

$$e(S, x) = \{(a, y | \text{op}_2(y + x)) \mid (a, y) \in S\}$$

on the first component and

$$e(S, x) = \{(a, y + \text{op}_1(y|x)) \mid (a, y) \in S\}$$

on the second one. The unique solution of e gives us two new unary combinators on C extending its cia structure. This means that Theorem 6.2 remains true for the extended calculus, without further work.

6.2. Streams. Recall from Example 2.10(2) that here we take $HX = \mathbb{R} \times X$ and we have $C = \mathbb{R}^\omega$ with the structure given by $\langle \text{hd}, \text{tl} \rangle : C \rightarrow \mathbb{R} \times C$.

Further recall that stream operations are defined by behavioral differential equations. For example, the componentwise addition of two streams σ and τ is specified by

$$(\sigma + \tau)_0 = \sigma_0 + \tau_0 \quad (\sigma + \tau)' = (\sigma' + \tau') . \quad (6.1)$$

And the shuffle product from the introduction is specified by

$$(\sigma \otimes \tau)_0 = \sigma_0 \cdot \tau_0 \quad (\sigma \otimes \tau)' = (\sigma \otimes \tau' + \sigma' \otimes \tau) . \quad (6.2)$$

Rutten gives in [31] a general theorem for the existence of the solution of systems of behavioral differential equations. We will now recall this result and show that it is a special instance of our Theorem 5.3. For a system of behavioral differential equations one starts with the signature Σ of all the operations to be specified. One uses an infinite supply of variables, and for each variable x there is also a variable x' and a variable $x(0)$ (also written as x_0). For each operation symbol f from Σ one specifies

$$f(x_1, \dots, x_n)_0 = h_f(x_1(0), \dots, x_n(0)) \quad f(x_1, \dots, x_n)' = t_f , \quad (6.3)$$

where h_f denotes a function from \mathbb{R}^n to \mathbb{R} and t_f is a term built from operation symbols from Σ on variables x_i, x'_i and $x_i(0)$, $i = 1, \dots, n$. Theorem A.1 of [31] asserts that for every f from Σ there exists a unique function $(\mathbb{R}^\omega)^n \rightarrow \mathbb{R}^\omega$ satisfying the equation (6.3) above.

We shall now show that a system such as (6.3) gives rise to an ℓ -rps for a suitable abstract GSOS rule ℓ . To this end let $KX = \mathbb{R}$ be the constant functor and let

$$\ell = (K(H \times \text{Id}) \xrightarrow{\ell'} \widehat{HK} \xrightarrow{H\kappa} \widehat{HK})$$

with ℓ' given by $\ell'_X(r) = (r, 0)$. Then the ℓ -interpretation $b : \mathbb{R} \rightarrow C$ assigns to every $r \in \mathbb{R}$ the stream $b(r) = (r, 0, 0, \dots)$.

Now given the system (6.3) let V be the polynomial functor associated to Σ , cf. Example 2.4. Notice that $K + V$ is the polynomial functor of the signature Σ extended with a constant symbol r for every real number r . We translate the system (6.3) into an ℓ -rps $e : V(H \times \text{Id}) \rightarrow \widehat{HK + V}$ as follows. For every f from Σ the corresponding component of e_X is defined by

$$e_X((r_1, x'_1, x_1), \dots, (r_n, x'_n, x_n)) = (h_f(r_1, \dots, r_n), \overline{t_f}) ,$$

where the term $\overline{t_f} \in \widehat{K + V}(X)$ is obtained by replacing in t_f all variables $x_i(0)$ by the constant r_i . Notice also, that here $h_f(r_1, \dots, r_n)$ is a real number (the value of h_f at (r_1, \dots, r_n)) whereas in (6.3) we have formal application of h_f to the variables $x_i(0)$.

It is now straightforward to verify that a solution of e in C corresponds precisely to a solution of the system (6.3). Thus, we obtain from Theorem 5.3 the

Theorem 6.3. *Every system of behavioral differential equations has a unique solution.*

Example 6.4. For the system given by (6.1) and (6.2) we have $VX = X \times X + X \times X$ and e given componentwise as follows: for the $+$ component we have

$$e_X((r, x', x), (s, y', y)) = (r + s, x' + y') \quad (6.4)$$

and for the \otimes component we have

$$e_X((r, x', x), (s, y', y)) = (r \cdot s, (x \otimes y') + (x' \otimes y)). \quad (6.5)$$

Observe that the systems (6.3) do not distinguish between given operations and newly defined ones. However, our result in Theorem 5.3 allows to make this distinction, and the compositionality principle for solution of an ℓ -rps (cf. Remark 5.9(2)) means that operations specified by behavioural differential equations may be used in subsequent behavioral differential equations as given operations in the terms t_f from (6.3). We believe this compositionality for behavioral differential equations is a new result.

Example 6.5. Take $VX = X \times X$ and the ℓ -rps e given by (6.4) whose solution is the operation of stream addition. As shown in the proof of Theorem 5.3, ℓ and e yield an abstract GSOS rule $n : F(H \times \text{Id}) \rightarrow H\widehat{F}$ for $F = K + V$. Now let $V_1X = X \times X$. Then (6.5) yields an n -rps e_1 whose solution is the shuffle product.

Next, we present an example illustrating Theorem 5.10.

Example 6.6. Continuing the previous example, consider the convolution product of streams specified by

$$(\sigma \times \tau)_0 = \sigma_0 \cdot \tau_0 \quad (\sigma \times \tau)' = (\sigma' \times \tau + \sigma_0 \times \tau'),$$

see [31]. Let $V_2X = X \times X$ and let the n -rps e_2 be given by

$$(e_2)_X((r, x', x), (s, y', y)) = (r \cdot s, (x' \times y) + (r \times y')). \quad (6.6)$$

Notice that this illustrates why we introduced the constants r ; in this way we are able to deal with σ_0 in the equation for $(\sigma \times \tau)'$. Then the unique solution of e_2 is the convolution product as expected. Theorem 5.10 asserts that the extended cia structure for $H\widehat{F}$, $F = K + V + V_1 + V_2$ on C does not depend on the order of taking the solution of (6.5) and (6.6)—either way this is given by the constants coming from b and the operations of stream addition as well as convolution and shuffle product.

Our results also allow to obtain unique solutions of specifications that go beyond behavioral differential equations, and we now provide one example.

Example 6.7. We give a sandwiched rps w.r.t. ℓ defining the binary operation $\sigma \circ \tau$ assigning to two streams σ and τ the “undersampled” zipped stream

$$(0, \sigma_0, 0, \tau_0, 0, \sigma_1, 0, \tau_1, \dots).$$

Indeed, let $KX = 1 + \mathbb{R} + X \times X + X \times X$ be the polynomial functor for the signature having a constant symbol for every $r \in \mathbb{R}$, an extra constant symbol X and two binary symbols $+$ and \times . The abstract GSOS rule $\ell : K(H \times \text{Id}) \rightarrow H\widehat{K}$ is given for each of the coproduct components separately by $\ell_X(r) = (r, 0)$ for the \mathbb{R} component, $\ell_X(*) = (0, 1)$ for the extra constant X and similarly as in (6.4) and (6.6) for $+$ and \times . Now let $VX = X \times X$ (i.e., V corresponds to one binary operation \circ), and let $e : V(H \times \text{Id}) \rightarrow \widehat{KHK+V}$ be the ℓ -srps given by

$$e_X((r, x', x), (s, y', y)) = X \times (r, y \circ x').$$

Then one easily verifies that the interpreted solution $C \times C \rightarrow C$ of e is the desired operation of “undersampled” zipping.

In general one can think of the right-hand sides of an ℓ -srps as terms of givens (represented by K) and newly defined operations (represented by V), where each newly defined operation must be guarded by a prefixing operation $r.-$.

We now turn to another method to define operations on streams—stream circuits [32], which are also called (signal) flow graphs in the literature. We shall demonstrate that specification of operations by stream circuit arises as a special case of our results. Stream circuits are usually defined as pictorial compositions of the following basic stream circuits



The r -multiplier multiplies all elements in a stream by $r \in \mathbb{R}$, the adder performs componentwise addition, the copier yields two copies of a stream, and the register prepends $r \in \mathbb{R}$ to a stream σ to yield $r.\sigma$. The stream circuits are then built from the basic circuits by plugging wires together, and there may also be feedback (loops). For example the following picture shows a simple stream circuit:



It defines the unary operation $f(\sigma) = (1 + \sigma_0, 1 + \sigma_0 + \sigma_1, 1 + \sigma_0 + \sigma_1 + \sigma_2, \dots)$ on streams. For our treatment we shall consider the operations presented by r -multipliers, adders and registers as givens. So let K be the signature functor associated to the signature Σ given by these operations (copying will be implicit via variable sharing). In symbols, $KX = \mathbb{R} \times X + X \times X + \mathbb{R} \times X$. Our given operations are defined by the behavioral differential equations:

$$\begin{array}{lll} (r\sigma)_0 &= r\sigma_0 & (r\sigma)' = r\sigma' \\ (\sigma + \tau)_0 &= \sigma_0 + \tau_0 & (\sigma + \tau)' = \sigma' + \tau' \\ (r.\sigma)_0 &= r & (r.\sigma)' = \sigma \end{array}$$

As explained above these definitions easily give rise to an abstract GSOS rule $\ell : K(H \times \text{Id}) \rightarrow H\widehat{K}$, see also [12], Section 3.5.1. We then get the ℓ -interpretation in C and the corresponding extended cia structures by Theorems 3.4 and 3.5. A stream circuit is called *valid* if every loop passes through at least one register. It is well-known that every finite valid stream circuit with one input and one output defines a unique stream function, see [32]. Of course, a similar result holds for more than one input and output, and we present here a new proof of this result based on our Theorem 5.3.

Theorem 6.8. *Every finite valid stream circuit defines a unique stream function at every output.*

Proof. Let a finite valid stream circuit be given. We explain how to construct an ℓ -rps from the circuit. Notice first that the wires in a circuit can be regarded as directed edges (cf. (6.7)). We take for every register R in our circuit an operation symbol g_R and define its arity as the number of inputs that can be reached by following all possible paths from R backwards through the circuit. Similarly, we take for every output edge O of the circuit an operation symbol f_O with the arity obtained in the same way. Let Σ be the signature of all f_O and g_R , and let V be the corresponding polynomial functor for Σ . To give an ℓ -rps $e : V(H \times \text{Id}) \rightarrow H\widehat{K} + V$ it suffices to give a natural transformation $e' : VH \rightarrow H\widehat{K} + V$ and to define $e = e' \cdot V\pi_0$, where $\pi_0 : H \times \text{Id} \rightarrow H$ is the

projection. To obtain e' , we give for each n -ary symbol s from Σ an assignment

$$s(r_1.x_1, \dots, r_n.x_n) \mapsto (r, t)$$

where $r \in \mathbb{R}$ and t is a term built from symbols of Σ and of the signature Γ of basic circuit operations using the variables x_1, \dots, x_n . Notice that the arguments of s stand for generic elements (r_i, x_i) from HX for some set X and that r may depend on r_i and t may contain operation symbols $r_i.-$. We now show how to define the above assignment for each g_R . Suppose that R has the initial value r . Then

$$g_R(r_1.x_1, \dots, r_n.x_n) \mapsto (r, t_R),$$

and we now explain how to obtain t_R : one follows every possible path in the circuit backwards that ends in R until

- (i) an input edge corresponding to some argument $r_i.x_i$ is met, or
- (ii) some register is met.

More precisely, we construct t_R as a $(\Sigma + \Gamma)$ -tree: we follow the input edge of R backwards until we reach either the output wire of an r -multiplier, the output wire of an adder, an input wire of the whole circuit or the output wire of a register. For an r -multiplier or an adder we add a node to t_R labelled by the corresponding operation symbol and continue this process for each input node of the r -multiplier or adder constructing the corresponding subtrees of t_R . For an input wire corresponding to $r_i.x_i$ add a node labelled by the prefix operation $r_i.-$ and below that a leaf labelled by x_i ; for a register S add the tree (of height 2) given by $g_S(r_{i_1}.x_{i_1}, \dots, r_{i_k}.x_{i_k})$, where the $r_{i_j}.x_{i_j}$ correspond to those input wires of the circuit backwards reachable from the register S . (Notice that these arguments of g_S form a subset of the arguments $\{r_1.x_1, \dots, r_n.x_n\}$ of g_R since every input that is backwards reachable from S is also backwards reachable from R . Also notice that copiers are ignored while forming t_R .) Since the given circuit C is valid we have indeed constructed only a finite tree, whence a term t_R .

We still need to define the assignment corresponding to e' for output symbols f_O :

$$f_O(r_1.x_1, \dots, r_n.x_n) \mapsto (r, t_O).$$

We first form the tree t'_O in essentially the same way as t_R for a register R with the difference that for every input wire and for every register we just insert an unlabelled leaf for the moment. To obtain r , label every leaf of t'_O corresponding to the input $r_i.x_i$ by r_i and every leaf corresponding to a register by its initial value; now evaluate the corresponding term to get r . In order to get t_O one replaces leaves of t'_O corresponding to inputs $r_i.x_i$ by x_i , and register leaves are replaced by the second components t_S from the right-hand sides of the equations for $g_S(r_{i_1}.x_{i_1}, \dots, r_{i_k}.x_{i_k})$.

Finally, the unique solution of e yields a unique operation f_O on streams for every output O . By construction this is the operation computing the stream circuit. \square

Remark 6.9. Suppose we are given a finite valid stream circuit where, in addition, every path from an input to an output passes through a register. Then the construction in the above proof would not need to refer to the behavior of the input $r_i.x_i$. That means that we could assume “structureless” inputs x_1, \dots, x_n , and the above construction then even gives a guarded rps $V \rightarrow \widehat{T^{HK+V}}$. Corollary 4.9 allows us to obtain a unique solution of this rps, and this result even allows for the unique solution of infinite valid stream circuits where every path from an input to an output passes through a register.

The compositionality principle for the unique solution of an ℓ -rps we discussed in Remark 5.9(2) yields an important *modularity* of stream circuits: they can be used as building blocks

as if they were basic operations in subsequent stream circuits. And Theorem 6.8 remains valid for the extended circuits.

Example 6.10. The proof of Theorem 6.8 essentially gives a translation of an arbitrary finite valid stream circuit into an ℓ -rps. We demonstrate this on the circuit given in (6.7) above. First we introduce for the output a function symbol f and for the register output the function symbol g . To determine their arity we count the number of input wires which have a (directed) path to the register and the output, respectively. In both cases the arity is one. Now we must give a definition of $f(r.x)$ and $g(r.x)$ for an abstract input stream with head $r \in \mathbb{R}$. These definitions are each given by a pair (s, t) where $s \in \mathbb{R}$ and t is a term in the one variable x over operations corresponding to the basic circuits and f, g . We define

$$g(r.x) = (1, r.x + g(r.x)) \quad f(r.x) = (r + 1, x + (r.x + g(r.x))).$$

For $g(r.x)$ we take the value 1 of the register as first component, and the right-hand term is obtained as follows: we follow all paths from the register backwards until we find an input or a register. So we get a finite tree or, equivalently, the desired term. For $f(r.x)$ we first follow all paths to inputs and registers backwards to get the term $t' = x_I + x_R$, where x_I represents the input and x_R the register. For the first component of $f(r.x)$ we evaluate t' with the head r of the input and the initial value 1 of the register, and for the second component we replace in t' the input by x and the register by the second component of the right-hand side of the above definition of $g(r.x)$. The two equations above are easily seen to yield an ℓ -rps $e : V(H \times \text{Id}) \rightarrow \widehat{HK + V}$, where $V = \text{Id} + \text{Id}$ is the polynomial functor for the signature with two unary symbols f and g . The unique solution of e gives two unary operations (for f and g) on C , and the one for f is precisely the function computed by the circuit (6.7). By the modularity of stream circuits explained above, we can use f (and also g) as ‘‘black-boxes’’ in subsequent stream circuits.

6.3. Infinite Trees. Rutten and Silva [33] developed behavioral differential equations for infinite trees and proved a unique solution theorem for them. Here we shall show that, similar to Theorem 6.3 for streams, we obtain that theorem as a special instance of our Theorem 5.3.

Let $HX = X \times X$. The terminal coalgebra C for H consists of all infinite node binary trees with nodes labelled in \mathbb{R} , and the terminal coalgebra structure $c : C \rightarrow C \times \mathbb{R} \times C$ assigns to a tree the triple (t_L, r, t_R) where r is the node label of the root of t and t_L and t_R are the trees rooted at the left-hand right-hand child nodes of the root of t . Single trees (constants) and operations on trees can be specified by behavioral differential equations. For example,

$$\begin{aligned} \text{pi}(\varepsilon) &= \pi & \text{pi}_L &= \text{pi} \\ \text{pi}_R &= \text{pi} \end{aligned}$$

specifies the tree with every node labelled by the number π . For every real number r we have the constant $[r]$ specified by

$$\begin{aligned} [r](\varepsilon) &= r & [r]_L &= [0] \\ & & [r]_R &= [0]. \end{aligned}$$

The nodewise addition of numbers stored in the nodes of the trees t and s is defined by

$$\begin{aligned} (t + s)(\varepsilon) &= t(\varepsilon) + s(\varepsilon) & (t + s)_L &= t_L + s_L \\ & & (t + s)_R &= t_R + s_R. \end{aligned}$$

See [33] for further and more exciting examples.

In general a system of behavioral differential equations is specified as follows. Let V be an infinite set of syntactic variables. For every $x \in V$ we have the notational variants x_L, x_R and also

$x(\varepsilon)$. Furthermore, let Σ be a signature of operations to be specified. For each operation symbol f from Σ of arity n we provide equations of the form

initial value	differential equations	
$(f(x_1, \dots, x_n))(\varepsilon) = c_f(x_1(\varepsilon), \dots, x_n(\varepsilon))$	$f(x_1, \dots, x_n)_L = t_1$ $f(x_1, \dots, x_n)_R = t_2$	(6.8)

where c_f denotes a function $\mathbb{R}^n \rightarrow \mathbb{R}$ and t_1 and t_2 are Σ -terms on the variables x_1, \dots, x_n and their three notational variants.

Theorem 6.11. ([33], Theorem 2) *Every system (6.8) of behavioral differential equations has a unique solution, i. e., for every f from Σ there exists a unique function $f : C^n \rightarrow C$ satisfying (6.8) (denoted by the same symbol).*

We present a new proof of this result based on Theorem 5.3. Let $KX = \mathbb{R}$ be the constant functor, and let the abstract GSOS rule ℓ be

$$\ell = (K(H \times \text{Id}) \xrightarrow{\ell'} HK \xrightarrow{H\kappa} H\widehat{K})$$

where the natural transformation ℓ' is given by $\ell'_X(r) = (0, r, 0)$. Then the ℓ -interpretation is $b : \mathbb{R} \rightarrow C$ with $b(r) = [r]$. Now every system (6.8) gives an ℓ -rps $e : V(H \times \text{Id}) \rightarrow \widehat{HK + V}$ as follows: let V be the polynomial functor associated to Σ and let e be given on each component corresponding to f from Σ by

$$e_X(((x_1)_L, r_1, (x_1)_R, x_1), \dots, ((x_n)_L, r_n, (x_n)_R, x_n)) = (\overline{t_1}, c_f(r_1, \dots, r_n), \overline{t_2}),$$

where $\overline{t_i}$ is obtained from t_i by replacing each $x_i(\varepsilon)$ by the corresponding constant r_i . The solutions of e in C correspond precisely to solutions of (6.8); thus, Theorem 6.11 follows from Theorem 5.3.

In addition, we have again a compositionality principle (cf. Remark 5.9): operations specified by behavioral differential equations may be used as givens in subsequent behavioral differential equations.

6.4. Non-well-founded Sets. For background on non-well-founded sets, the antifoundation axiom (AFA), and classes, please see the books [2, 13]. We work here on the category $\mathcal{A} = \text{Class}$ of classes. The results of Section 5 hold true for Class since every endofunctor of Class has terminal coalgebras and free algebras, see [5].

Consider $\mathcal{P} : \text{Class} \rightarrow \text{Class}$ taking a class X to the class $\mathcal{P}X$ of subsets of X . AFA is equivalent to the assertion that (V, c) is a terminal coalgebra, where V is the class of all sets, and $c : V \rightarrow \mathcal{P}V$ takes a set and considers it a set of sets. (That is, $c(s) = s$ for all s .) Let us note some natural transformations:

$$\begin{array}{lll} p : \mathcal{P} \rightarrow \mathcal{P}\mathcal{P} & op : \text{Id} \times \text{Id} \rightarrow \mathcal{P}\mathcal{P} & cp : \mathcal{P} \times \mathcal{P} \rightarrow \mathcal{P}(\text{Id} \times \text{Id}) \\ p_X(x) = \mathcal{P}(x) & op_X(x, y) = \{\{x\}, \{x, y\}\} & cp_X(x, y) = x \times y \end{array}$$

Also note that c^{-1} is the operation on V taking a family $x \subseteq V$ of sets to the set $\{y \mid y \in x\}$.

We will now define three additional operations on V : the powerset operation $b_1 : x \mapsto \{y \mid y \subseteq x\}$, the Kuratowski pair $b_2 : (x, y) \mapsto \{\{x\}, \{x, y\}\}$ and the cartesian product $b_3 : (x, y) \mapsto x \times y$. So let K be the functor $\text{Id} + (\text{Id} \times \text{Id}) + (\text{Id} \times \text{Id}) + \mathcal{P} + \mathcal{P}^2$; its first three components represent (the type of) our three desired operations, the fourth component \mathcal{P} represents c^{-1} and the fifth one represents $c^{-1} \cdot \mathcal{P}c^{-1}$ —the latter two are needed for the definition of the former three. We write the coproduct injections of K as $\text{inj}_1, \dots, \text{inj}_5$. We define a natural transformation $\ell' : K\mathcal{P} \rightarrow \mathcal{P}K$

componentwise, using

$$\begin{array}{c}
 \mathcal{P} \xrightarrow{p} \mathcal{P}\mathcal{P} \xrightarrow{\mathcal{P}\text{inj}_4} \mathcal{P}K \\
 \mathcal{P} \times \mathcal{P} \xrightarrow{\text{op}\mathcal{P}} \mathcal{P}\mathcal{P}\mathcal{P} \xrightarrow{\mathcal{P}\text{inj}_5} \mathcal{P}K \\
 \mathcal{P} \times \mathcal{P} \xrightarrow{cp} \mathcal{P}(\text{Id} \times \text{Id}) \xrightarrow{\mathcal{P}\text{inj}_2} \mathcal{P}K
 \end{array}
 \quad
 \begin{array}{c}
 \mathcal{P}\mathcal{P} \xrightarrow{\mathcal{P}\text{inj}_4} \mathcal{P}K \\
 \mathcal{P}\mathcal{P}\mathcal{P} \xrightarrow{\mathcal{P}\text{inj}_5} \mathcal{P}K
 \end{array}$$

Then ℓ' yields an abstract GSOS rule

$$\ell = (K(\mathcal{P} \times \text{Id}) \xrightarrow{K\pi_0} K\mathcal{P} \xrightarrow{\ell'} \mathcal{P}K \xrightarrow{\mathcal{P}\kappa} \mathcal{P}\widehat{K}).$$

Let $b : KV \rightarrow V$ be the ℓ -interpretation in V . Let us write b_1, \dots, b_5 for the components of b , so $b_i = b \cdot (\text{inj}_i)_V$. To obtain explicit formulas for these, we use Diagram (2.3) and the above definitions to write:

$$\begin{aligned}
 c \cdot b_1 &= \mathcal{P}b_4 \cdot p_V \cdot c & c \cdot b_4 &= \mathcal{P}b_4 \cdot \mathcal{P}c \\
 c \cdot b_2 &= \mathcal{P}b_5 \cdot \text{op}\mathcal{P}V \cdot (c \times c) & c \cdot b_5 &= \mathcal{P}b_5 \cdot \mathcal{P}^2c \\
 c \cdot b_3 &= \mathcal{P}b_2 \cdot cp_V \cdot (c \times c)
 \end{aligned}$$

We check easily that $b_4 = c^{-1}$ and $b_5 = c^{-1} \cdot \mathcal{P}c^{-1}$ satisfy the last two equations. From these we see that

$$b_1 = c^{-1} \cdot \mathcal{P}c^{-1} \cdot p_V \cdot c, \quad b_2 = c^{-1} \cdot \mathcal{P}b_5 \cdot \text{op}\mathcal{P}V \cdot (c \times c),$$

and

$$b_3 = c^{-1} \cdot \mathcal{P}b_2 \cdot cp_V \cdot (c \times c).$$

In words, b_4 and b_5 are the identity, and b_1, b_2 and b_3 are as desired.

By Theorem 3.4, we have a cia structure $(V, c^{-1} \cdot \mathcal{P}\widehat{b})$ for the composite $\mathcal{P}\widehat{K}$.

Remark 6.12. We could have obtained the various operations on V in a step by step fashion starting with b_4 and b_5 and then defining b_1, b_2, b_3 by successive applications of Theorem 5.3. We decided against this, to keep the presentation short. But in the next section on formal languages we follow this approach.

Continuing our discussion of non-well-founded sets, we may solve systems of equations which go beyond what one finds in the standard literature on non-well-founded sets [2, 13]. For example, one may solve the system

$$x = \{\mathcal{P}(y)\} \quad y = \{y \times y, z\} \quad z = \emptyset,$$

which gives rise to a flat equation morphism $X \rightarrow \mathcal{P}\widehat{K}X + V$ where $X = \{x, y, z\}$. Further, one may uniquely solve recursive function definitions such as

$$g(x) = \{g(\mathcal{P}(x)) \times x, x\}$$

from the introduction. Indeed, for $W = \text{Id}$ this equation yields an ℓ -rps $e : W(\mathcal{P} \times \text{Id}) \rightarrow \mathcal{P}\widehat{K} + W$ whose unique solution given by Theorem 5.3 is a function $g_V : V \rightarrow V$ behaving as specified.

6.5. Formal Languages. Recall Example 2.10(1); here we have $HX = X^A \times 2$ on Set. A coalgebra $x : X \rightarrow X^A \times 2$ for H is precisely a deterministic automaton with the (possibly infinite) state set X . Here $C = \mathcal{P}(A^*)$, and the unique homomorphism $h : (X, x) \rightarrow (C, c)$ assigns to each state the language it accepts. We shall now show how various operations on formal languages can be defined in a compositional way using Theorem 5.3. It is well-known that such operations can be defined as interpretations of one abstract GSOS rule (or distributive law) in C , see e.g. [18].

However, the previous bialgebraic account does not explain why one may define these operations in a step-by-step fashion by subsequent recursive definitions. This is the added value of Theorem 5.3.

We start with the functor $K_0 = C_\emptyset$ (that means, we start from scratch with no given operations) and with $\ell_0 : C_\emptyset(H \times \text{Id}) \rightarrow H\widehat{C}_\emptyset = H$ given by the empty maps. The corresponding interpretation is the empty map $b : \emptyset \rightarrow C$, and \widehat{b} is then the identity on C . Thus, the cia structure for $H\widehat{K}_0$ on C given by Theorem 3.4 is simply the initial cia (C, c^{-1}) for H . At each subsequent step we are given a functor K_i and an abstract GSOS rule $\ell_i : K_i(H \times \text{Id}) \rightarrow H\widehat{K}_i$ with its interpretation $b_i : K_i C \rightarrow C$. We then give an ℓ_i -rps $e_i : V_i(H \times \text{Id}) \rightarrow H\widehat{K}_i + V_i$, and its unique solution $s_i : V_i C \rightarrow C$ extends the cia structure as follows: let $K_{i+1} = K_i + V_i$ and let $\ell_{i+1} = [H\widehat{\text{inl}} \cdot \ell_i, e_i] : K_{i+1}(H \times \text{Id}) \rightarrow H\widehat{K}_{i+1}$, where $\widehat{\text{inl}} : \widehat{K}_i \rightarrow \widehat{K}_{i+1}$ is the monad morphism induced by $\text{inl} : K_i \rightarrow K_{i+1}$ (cf. Notation 5.4(1)). By induction it is easy to see that the ℓ_{i+1} -interpretation is $b_{i+1} = [s_j]_{j=0,\dots,i} : K_{i+1} C \rightarrow C$. And this gives an extended cia $c^{-1} \cdot H\widehat{b}_{i+1} : H\widehat{K}_{i+1}(C) \rightarrow C$ by Theorem 3.4.

As a first step we define constants in C for \emptyset , $\{\varepsilon\}$, and $\{a\}$, for each $a \in A$, as solutions of an ℓ_0 -rps. We express this as an ℓ_0 -rps as follows: take the functor $V_0 X = 1 + 1 + A$ corresponding to the above constants. We define $e_0 : V_0(H \times \text{Id}) \rightarrow H\widehat{K}_0 + V_0 = H\widehat{V}_0$ componentwise. We write for every set X , \emptyset for $\text{inj}_1(*) \in V_0 X$ and ε for $\text{inj}_2(*) \in V_0 X$. Then $(e_0)_X$ is given by the assignments

$$\begin{aligned} \emptyset &\mapsto ((\emptyset)_{a \in A}, 0) \\ \varepsilon &\mapsto ((\emptyset)_{a \in A}, 1) \\ a &\mapsto ((t_b)_{b \in A}, 0) \end{aligned} \quad \text{where } t_b = \begin{cases} \varepsilon & \text{if } b = a \\ \emptyset & \text{else.} \end{cases} \quad (6.9)$$

It is now straightforward to check that the unique solution s_0 of e_0 yields the desired constants in C extending the cia structure.

Next we add the operations of union, intersection and language complement to the cia structure. Let $K_1 = K_0 + V_0$ and let $\ell_1 = [H\widehat{\text{inl}} \cdot \ell_0, e_0]$ as above with interpretation $b_1 = s_0$. Let $V_1 X = X \times X + X \times X + X$ be the polynomial functor corresponding to two binary symbols \cup and \cap and one unary one $(\overline{-})$. We give the ℓ_1 -rps $e_1 : V_1(H \times \text{Id}) \rightarrow H\widehat{K}_1 + V_1$ componentwise in the form of the three assignments in (6.10) below. We write $((x_a), j, x)$ for elements of $HX \times X$, where (x_a) is an A -tuple, i. e., an element of X^A . We also write elements of $V_2 Z$, $Z = HX \times X$, as flat terms $z_1 \cup z_2$, $z_1 \cap z_2$ and \overline{z} for the three components:

$$\begin{aligned} ((x_a), j, x) \cup ((y_a), k, y) &\mapsto ((x_a \cup y_a), j \vee k) & \overline{((x_a), j, x)} &\mapsto ((\overline{x_a}), \neg j) \\ ((x_a), j, x) \cap ((y_a), k, y) &\mapsto ((x_a \cap y_a), j \wedge k) \end{aligned} \quad (6.10)$$

where \vee , \wedge and \neg are the evident operations on $2 = \{0, 1\}$. The corresponding unique solution $s_1 : V_1 C \rightarrow C$ is easily checked to provide the desired operations extending the cia structure on C .

The next step adds concatenation to the cia structure on C . For this let $V_2 X = X \times X$ and e_2 is given by the assignment

$$((x_a), j, x) \cdot ((y_a), k, y) \mapsto ((t_a), j \wedge k) \quad \text{where } t_a = \begin{cases} (x_a \cdot y) \cup y_a & \text{if } j = 1 \\ x_a \cdot y & \text{else.} \end{cases} \quad (6.11)$$

Its unique solution $s_2 : C \times C \rightarrow C$ is the concatenation operation.

As the final step we add the Kleene star operation by taking $V_3 X = X$ and e_3 given by

$$e_3((x_a), j, x) = ((x_a \cdot x^*), 1).$$

Notice that this definition makes use of concatenation which was a solution at the previous stage and concatenation makes use of union which was a solution at stage 1.

Remark 6.13. There are many further operations on formal languages that are definable by ℓ -rps's, including the following ones:

- prefixing $a.L = \{aw \mid w \in L\}$ for any $a \in A$;
- the operation given by $c^{-1} : C^A \times 2 \rightarrow C$ (see Remark 5.12)

$$((L_a), j) \mapsto \begin{cases} \bigcup_{a \in A} a.L_a & \text{if } j = 0 \\ \bigcup_{a \in A} a.L_a \cup \{\varepsilon\} & \text{else;} \end{cases}$$

- $\text{shuffle}(L_1, L_2) = \bigcup_{w_1 \in L_1, w_2 \in L_2} \text{shuffle}(w_1, w_2)$ where $\text{shuffle}(w_1, w_2)$ is the usual operation merging the words w_1 and w_2 .

We leave it to the reader to work out the details. Notice, however, that there exist operations that cannot be defined by ℓ -rps's (or abstract GSOS rules). An example is the language derivative $L^a = \{w \mid aw \in L\}$ for some $a \in A$. Indeed, if this was definable by an ℓ -rps, Theorem 5.3 would yield a cia structure on C for the functor $\widehat{HK + V}$, where $VX = X$ corresponds to the unary operation $(-)^a$. For a term t in $\widehat{K + V}(X)$, we shall use the notation $a.t$ as a shortcut for $((t_b), 0) \in \widehat{HK + V}(X)$ for the A -tuple (t_b) with $t_a = t$ and $t_b = \emptyset$ for every $b \in A \setminus \{a\}$. Thus, for $X = \{x\}$ the flat equation morphism $e : X \rightarrow \widehat{HK + V}X + C$ given by $e(x) = a.x^a$ would have a unique solution. But this is clearly not the case: every formal language L' whose words all start with a gives a solution $e^\dagger(x) = L'$ of the flat equation morphism e .

Next we show how context-free grammars in Greibach Normal Form and their generated languages are special instances of flat equations $e : X \rightarrow \widehat{KH}\widehat{K}X + C$ and their unique solutions in C for a suitable functor K . Recall (e.g. from [17]) that a *context-free grammar* is a four-tuple $G = (A, N, P, S)$ where A is a non-empty finite set of *terminal symbols*, N a finite set of *non-terminal symbols*, $P \subseteq N \times (A + N)^*$ is a finite relation with elements called *production rules* of G , and $S \in N$ is the starting symbol. As usual we write $n \rightarrow w$ for $(n, w) \in P$.

A context-free grammar G is in *Greibach Normal Form (GNF, for short)* if all its production rules are of the form $n \rightarrow aw$ with $a \in A$ and $w \in N^*$.

The language *generated* by a context-free grammar G is the set of all words over A that arise by starting with the string S and repeatedly substituting substrings according to the production rules of the grammar, and eliminating ε from the string whenever it occurs.

To see that context-free grammars in GNF yield flat equation morphisms we consider the constant \emptyset and the operations of union and concatenation as given operations. More precisely, let $KX = 1 + X \times X + X \times X$ be the polynomial functor corresponding to \emptyset , \cup and \cdot , and let $\ell : K(H \times \text{Id}) \rightarrow H\widehat{K}$ be the abstract GSOS rule given by the corresponding assignments in (6.9), (6.10) and (6.11) with the interpretation $b : KC \rightarrow C$ given as desired. By Theorem 3.5 we obtain the cia structure $k' : \widehat{KH}\widehat{K}C \rightarrow C$. Now observe that a flat equation morphism assigns to each $x \in X$ either an element $e(x) \in C$ or $e(x)$ corresponds to a term of given operations on $H\widehat{K}X$.

We now show how to construct a flat equation morphism for any context-free grammar in GNF.

Construction 6.14. Let G be a context-free grammar in GNF. We now define the flat equation morphism $e_G : X \rightarrow \widehat{KH}\widehat{K}X + C$. The set X is simply the set N of non-terminals of G , and e_G is the following map: for $n \in N$ for which there is no production rule $n \rightarrow aw$ in P we take $e_G(n) = \emptyset$, the constant term in $\widehat{K}(H\widehat{K}X)$. Otherwise for each production rule $n \rightarrow aw$ with n on

the left-hand side we define the term $t_r \in H\hat{K}X$ as

$$t_r = \begin{cases} a.(n_1 \cdot n_2 \cdots \cdot n_k) & w = n_1 n_2 \cdots n_k \\ a.\emptyset & w = \varepsilon \end{cases}$$

using the concatenation operation. Recall from Remark 6.13 the notation $a.t$ to see that $t_r \in H\hat{K}X$. We define $e_G(n)$ as (the term in $\hat{K}(H\hat{K}X)$ representing) the “union” of all right-hand sides of production rules $n \rightarrow r_i$, $i = 1, \dots, l$, in P :

$$e_G(n) = t_{r_1} \cup t_{r_2} \cup \cdots \cup t_{r_l}.$$

Notice that e_G does not make use of the parameters in C .

It is not difficult to see that the language generated by the grammar G is precisely the language $e_G^\dagger(S)$, where S is the starting symbol of G . So as a consequence of Theorem 3.5 we see that the language generated by G arises as the unique solution of the flat equation morphism e_G .

Analogously, it is also possible to translate right-linear grammars (which are a special case of context-free grammars generating regular languages) into flat equation morphisms using the constant empty and empty-word languages as well as union as the given operations. Again Theorem 3.5 implies that there is a unique solution which yields the language generated by the given grammar by the translation.

Remark 6.15. We have seen that by defining operations via ℓ -rps’s (or ℓ -srps’s) we obtain cia structures for $H\hat{K}$ (or $\hat{K}H\hat{K}$) on C . It is interesting to ask what formal languages can arise as solutions of flat equation morphisms $e : X \rightarrow H\hat{K}X + C$ (or $e : X \rightarrow \hat{K}H\hat{K} + C$) if X is *finite*. Not surprisingly one obtains precisely the regular languages for stages $i = 0, 1, 2$ in our above definition process, i.e., when we add the constant languages \emptyset , $\{\varepsilon\}$ and $\{a\}$ for every $a \in A$, and the operations \cup , \cap and $\overline{(-)}$ (of course, each $e(x) \in C$ needs to be regular here). But adding concatenation one obtains non-regular languages: if for $i = 3$ one restricts to using union and concatenation in the terms in $\hat{K}_3 H \hat{K}_3$, the flat equation morphisms essentially correspond to context-free grammars in GNF. So if each $e(x) \in C$ is context-free then so is the solution of any such flat equation morphism. This remains true even if $(-)^*$ is added and used in flat equations. However, using intersection and/or complement allows one to obtain non context-free languages as solutions. Precisely what class of languages can be defined by flat equation morphisms using different combinations of operations remains the subject of future work.

7. CONCLUSIONS

In many areas of theoretical computer science, one is interested in recursive definitions of functions on terminal coalgebras C for various functors H . This paper provides a more comprehensive foundation for recursive definitions than had been presented up until now. The overall idea is to present operations in terms of an abstract GSOS rule $\ell : K(H \times \text{Id}) \rightarrow H\hat{K}$. We proved that ℓ induces new completely iterative algebra structures for $H\hat{K}$ and $\hat{K}H\hat{K}$ on C . As a result, we are able to define operations with useful algebraic properties such as commutativity or associativity “for free”. We also introduced the notion of an ℓ -rps and showed how to uniquely solve recursive function definitions in C which are given by an ℓ -rps. Our results explain why taking unique solutions of such equations is a compositional process. And we have seen that our results can be applied to provide the semantics of recursive specifications in a number of different areas of theoretical computer science.

REFERENCES

- [1] L. Aceto, W. Fokkinga, and C. Verhoef. *Handbook of Process Algebra*, chapter Structural Operational Semantics. Elsevier, 2001.
- [2] P. Aczel. *Non-Well-Founded Sets*, volume 14 of *CLSI Lecture Notes*. CLSI Publications, Stanford, 1988.
- [3] P. Aczel, J. Adámek, S. Milius, and J. Velebil. Infinite trees and completely iterative theories: A coalgebraic view. *Theoret. Comput. Sci.*, 300:1–45, 2003.
- [4] J. Adámek. Introduction to coalgebra. *Theory Appl. Categ.*, 14:157–199, 2005.
- [5] J. Adámek, S. Milius, and J. Velebil. On coalgebras based on classes. *Theoret. Comput. Sci.*, 316:3–23, 2004.
- [6] J. Adámek, S. Milius, and J. Velebil. Elgot algebras. *Log. Methods Comput. Sci.*, 2(5:4):31 pp., 2006.
- [7] J. Adámek, S. Milius, and J. Velebil. Iterative algebras at work. *Math. Structures Comput. Sci.*, 16(6):1085–1131, 2006.
- [8] J. Adámek and V. Trnková. *Automata and Algebras in Categories*, volume 37 of *Mathematics and its Applications*. Kluwer Academic Publishers, 1990.
- [9] P. America and J. J. M. M. Rutten. Solving reflexive domain equations in a category of complete metric spaces. *J. Comput. System Sci.*, 39:343–375, 1989.
- [10] M. Barr. Coequalizers and free triples. *Math. Z.*, 116:307–322, 1970.
- [11] F. Bartels. Generalized coinduction. *Math. Structures Comput. Sci.*, 13(2):321–348, 2003.
- [12] F. Bartels. *On generalized coinduction and probabilistic specification formats*. PhD thesis, Vrije Universiteit Amsterdam, 2004.
- [13] J. Barwise and L. S. Moss. *Vicious circles*. CLSI publications, Stanford, 1996.
- [14] B. Bloom, S. Istrail, and A. R. Meyer. Bisimulation can't be traced. *J. ACM*, 42(1):232–268, 1995.
- [15] V. Capretta, T. Uustalu, and V. Vene. Recursive coalgebras from comonads. *Inform. and Comput.*, 204:437–468, 2006.
- [16] F. Hausdorff. *Grundzüge der Mengenlehre*. Von Veit & Co., Leipzig, 1914.
- [17] J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley Longman, Amsterdam, 3rd edition, 2007.
- [18] B. Jacobs. A bialgebraic review of deterministic automata, regular expressions and languages. In *Goguen Festschrift*, volume 4060 of *Lecture Notes Comput. Sci.*, 2006.
- [19] B. Jacobs. Distributive laws for the coinductive solution of recursive equations. *Inform. and Comput.*, 204(4):561–587, 2006.
- [20] J. Lambek. A fixpoint theorem for complete categories. *Math. Z.*, 103:151–161, 1968.
- [21] M. Lenisa, A. J. Power, and H. Watanabe. Distributivity for endofunctors, pointed and co-pointed endofunctors, monads and comonads. In H. Reichel, editor, *Proc. Coalgebraic Methods in Computer Science*, volume 33 of *Electron. Notes Theor. Comput. Sci.*, Amsterdam, 2000. Elsevier.
- [22] M. Lenisa, A. J. Power, and H. Watanabe. Category theory for operational semantics. *Theoret. Comput. Sci.*, 327:135–154, 2004.
- [23] S. MacLane. *Categories for the working mathematician*. Springer, 2nd edition, 1998.
- [24] S. Milius. Completely iterative algebras and completely iterative monads. *Inform. and Comput.*, 196:1–41, 2005.
- [25] S. Milius and L. S. Moss. The category theoretic solution of recursive program schemes. *Theoret. Comput. Sci.*, 366:3–59, 2006.
- [26] S. Milius and L. S. Moss. Equational properties of recursive program scheme solutions. *Cah. Topol. Géom. Différ. Catég.*, 50:23–66, 2009.
- [27] S. Milius, L. S. Moss, and D. Schwencke. CIA structures and the semantics of recursion. In C.-H. L. Ong, editor, *Proc. Foundations of Software Science and Computation Structures (FoSSaCS)*, volume 6014 of *Lecture Notes Comput. Sci.*, pages 312–327. Springer, 2010.
- [28] R. Milner. *Communication and Concurrency*. International Series in Computer Science. Prentice Hall, 1989.
- [29] G. D. Plotkin and D. Turi. Towards a mathematical operational semantics. In *Proc. Logic in Computer Science (LICS)*, 1997.
- [30] J. J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theoret. Comput. Sci.*, 249(1):3–80, 2000.
- [31] J. J. M. M. Rutten. A coinductive calculus of streams. *Math. Structures Comput. Sci.*, 15(1):93–147, 2005.
- [32] J. J. M. M. Rutten. A tutorial on coinductive stream calculus and signal flow graphs. *Theor. Comput. Sci.*, 343(3):443–481, 2005.
- [33] J. J. M. M. Rutten and A. Silva. Behavioural differential equations and coinduction for binary trees. In D. Leivant and R. de Queiroz, editors, *Proc. WoLLIC 2007*, volume 4576 of *Lecture Notes Comput. Sci.*, page 2007. Springer, 322–336.

- [34] D. Schwencke. Coequational logic for accessible functors. *Inform. and Comput.*, 208:1469–1489, 2010.
- [35] T. Uustalu, V. Vene, and A. Pardo. Recursion schemes from comonads. *Nordic J. Comput.*, 8(3):366–390, 2001.
- [36] J. Worrell. On the final sequence of a finitary set functor. *Theoret. Comput. Sci.*, 338:184–199, 2005.

APPENDIX A. RESULTS FOR POINTED FUNCTORS

We mentioned in Remark 3.6 that Theorems 3.4 and 3.5 hold more generally for pointed endofunctors M in lieu of a free monad $M = \widehat{K}$. However, in this case we need our base category to be cocomplete. We will now provide the details.

Assumption A.1. We assume that \mathcal{A} is a cocomplete category, that $H : \mathcal{A} \rightarrow \mathcal{A}$ is a functor and that (M, η) is a *pointed functor* on \mathcal{A} , i.e., $M : \mathcal{A} \rightarrow \mathcal{A}$ is a functor and $\eta : \text{Id} \rightarrow M$ is a natural transformation. As before $c : C \rightarrow HC$ is a terminal coalgebra for H .

Definition A.2. (1) An *algebra for (M, η)* is a pair (A, a) where A is an object of \mathcal{A} and $a : MA \rightarrow A$ is a morphism satisfying the *unit law* $a \cdot \eta_A = \text{id}_A$.

(2) A *distributive law of M over H* is a natural transformation $\lambda : MH \rightarrow HM$ such that the diagram

$$\begin{array}{ccc} & H & \\ \eta H \swarrow & & \searrow H\eta \\ M\bar{H} & \xrightarrow{\lambda} & HM \end{array} \quad (\text{A.1})$$

commutes.

(3) Let (D, ε) be a copointed endofunctor on \mathcal{A} . A *distributive law of (M, η) over (D, ε)* is a distributive law $\lambda : MD \rightarrow DM$ of (M, η) over the functor D that makes, in addition to (A.1) with H replaced by D , the diagram

$$\begin{array}{ccc} MD & \xrightarrow{\lambda} & DM \\ M\varepsilon \searrow & & \swarrow \varepsilon M \\ & M & \end{array}$$

commutes.

Remark A.3. (1) Every distributive law $\lambda : MH \rightarrow HM$ gives one of the cofree copointed functor $H \times \text{Id}$ via

$$\begin{array}{ccc} MH & \xrightarrow{\lambda} & HM \\ M\pi_0 \uparrow & & \uparrow \pi_0 \\ M(H \times \text{Id}) & \dashrightarrow & (H \times \text{Id})M \\ & M\pi_1 \searrow & \downarrow \pi_1 \\ & & M \end{array} \quad (\text{A.2})$$

but not conversely.

(2) Analogously to Theorem 2.7, we have for any distributive law λ of M over the cofree copointed functor $H \times \text{Id}$ a unique λ -*interpretation*, i.e., a unique morphism $b : MC \rightarrow C$ such that the diagram below commutes

$$\begin{array}{ccccc} MC & \xrightarrow{M(c, \text{id}_C)} & M(HC \times C) & \xrightarrow{\lambda_C} & HMC \times MC \xrightarrow{\pi_0} HMC \\ b \downarrow & & c & & \downarrow Hb \\ C & \xrightarrow{c} & HC & & \end{array}$$

and (C, b) is an algebra for the pointed functor M , see [12]. Notice that b here corresponds to $\widehat{b} : \widehat{K}C \rightarrow C$ in Theorem 2.7. If we have a distributive law $\lambda : MH \rightarrow HM$, then we obtain

one of M over the copointed functor $H \times \text{Id}$ as in (A.2). We again call the resulting morphism $b : MC \rightarrow C$ the λ -interpretation in C . In this case, the above diagram simplifies to

$$\begin{array}{ccccc} MC & \xrightarrow{Mc} & MHC & \xrightarrow{\lambda_C} & HMC \\ b \downarrow & & & & \downarrow Hb \\ C & \xrightarrow{c} & HC & & \end{array} \quad (\text{A.3})$$

Next we shall need a version of Theorem 2.12 for a given distributive law λ of M over H (or over the cofree copointed functor $H \times \text{Id}$). This is a variation of Theorem 4.2.2 of Bartels [12] (see also Lemma 4.3.2 in loc. cit.) using the cocompleteness of \mathcal{A} . Since one part of the proof in [12] is only presented for Set we give a full proof here for the convenience of the reader.

Theorem A.4. *Let $\lambda : MH \rightarrow HM$ be a distributive law of the pointed functor M over the functor H . Then for every λ -equation $e : X \rightarrow HMX$ there exists a unique solution, i. e., a unique morphism $e^\dagger : X \rightarrow C$ such that the diagram below commutes:*

$$\begin{array}{ccccc} X & \xrightarrow{e} & HMX & & \\ e^\dagger \downarrow & & \downarrow HMe^\dagger & & \\ C & \xrightarrow{c} & HC & \xleftarrow{Hb} & HMC \end{array} \quad (\text{A.4})$$

Before we proceed to the proof of the statement we need some auxiliary constructions and lemmas. We begin by defining an endofunctor S on our cocomplete category \mathcal{A} as a colimit. We denote by M^n , $n \in \mathbb{N}$, the n -fold composition of M with itself. Now we consider the diagram D in the category of endofunctors on \mathcal{A} given by the natural transformations in the picture below:

$$\begin{array}{ccccccc} & & & & \eta MM & & \\ & & \eta M & & \nearrow M\eta & & \\ \text{Id} & \xrightarrow{\eta} & M & \xrightarrow{MM} & M & \xrightarrow{M\eta M} & \dots \\ & & M\eta & & \swarrow MM\eta & & \end{array}$$

More formally, the diagram D is formed by all natural transformations

$$M^{i+j} \xrightarrow{M^i \eta M^j} M^{i+1+j} \quad i, j \in \mathbb{N}.$$

Let S be a colimit of this diagram D :

$$S = \text{colim } D \quad \text{with injections } \text{inj}^i : M^i \rightarrow S.$$

Then S is a pointed endofunctor with the point $\text{inj}^0 : \text{Id} = M^0 \rightarrow S$.

Recall that colimits in the category of endofunctors of \mathcal{A} are formed objectwise. So for any object X , SX is a colimit of the diagram D at that object X with colimit injections $\text{inj}_X^n : M^n X \rightarrow SX$, $n \in \mathbb{N}$. This implies that, for any endofunctor F of \mathcal{A} the functor SF is a colimit with injections $\text{inj}^n F : M^n F \rightarrow SF$.

The above definition of S appears in Bartels [12]. Next we define additional data using the universal property of the colimits SM and SH :

- (1) a natural transformation $\chi : SM \rightarrow S$ uniquely determined by the commutativity of the triangles below:

$$\begin{array}{ccc} M^{n+1} & & \\ \downarrow \text{inj}^n M & \searrow \text{inj}^{n+1} & \\ SM & \xrightarrow{\chi} & S \end{array} \quad \text{for all } n \in \mathbb{N}.$$

- (2) a natural transformation $\varepsilon : SM \rightarrow MS$ uniquely determined by the commutativity of the triangles below:

$$\begin{array}{ccc} M^{n+1} & & \\ \downarrow \text{inj}^n M & \searrow M\text{inj}^n & \\ SM & \xrightarrow{\varepsilon} & MS \end{array}$$

- (3) a natural transformation $\lambda^* : SH \rightarrow HS$; indeed, define first $\lambda^n : M^n H \rightarrow HM^n$ recursively as follows:

$$\begin{aligned} \lambda^0 &= \text{id}_H : H \rightarrow H; \\ \lambda^{n+1} &= M^{n+1} H = MM^n H \xrightarrow{M\lambda^n} MHM^n \xrightarrow{\lambda M^n} HMM^n = HM^{n+1}. \end{aligned}$$

Then λ^* is uniquely determined by the commutativity of the squares below:

$$\begin{array}{ccc} M^n H & \xrightarrow{\lambda^n} & HM^n \\ \downarrow \text{inj}^n H & & \downarrow H\text{inj}^n \\ SH & \xrightarrow{\lambda^*} & HS \end{array} \quad \text{for all } n \in \mathbb{N}.$$

Observe that λ^* is a distributive law of the pointed endofunctor S over H ; the unit law is the above square for the case $n = 0$.

We now need to verify that the three natural transformations above are well-defined. More precisely, we need to prove that those natural transformations are induced by appropriate cocones. For $\chi : SM \rightarrow S$ and $\lambda^* : SH \rightarrow HS$, this follows from Lemma 4.3.2 in Bartels' thesis [12]. Hence, we make the explicit verification only for ε and leave the details for the other two natural transformations for the reader. To verify that the natural transformations $M\text{inj}^n : M^{n+1} \rightarrow MS$ form a cocone for the appropriate diagram with colimit SM consider the triangles below:

$$\begin{array}{ccc} M^{1+i+j} & \xrightarrow{MM^i \eta M^j} & M^{1+i+1+j} \\ & \searrow M\text{inj}^n & \swarrow M\text{inj}^{n+1} \\ & MS & \end{array} \quad \text{for all } n \in \mathbb{N}, n = i + j.$$

These triangles commute since $\text{inj}^n : M^n \rightarrow S$ form a cocone.

Next, notice that in the definition of λ^* above there are two possible canonical choices for λ^{n+1} . We now show that these two choices are equal:

Lemma A.5. *For all natural numbers n we have the commutative square below:*

$$\begin{array}{ccc} M^{n+1}H & \xrightarrow{M^n\lambda} & M^nHM \\ M\lambda^n \downarrow & & \downarrow \lambda^nM \\ MHM^n & \xrightarrow{\lambda M^n} & HM^{n+1}. \end{array}$$

Proof. We prove the result by induction on n . The base case $n = 0$ is clear: both composites in the desired square are simply $\lambda : MH \rightarrow HM$. For the induction step we need to verify that the diagram below commutes:

$$\begin{array}{ccccc} M^{n+1}MH & \xrightarrow{M^{n+1}\lambda=MM^n\lambda} & M^{n+1}HM & & \\ MM\lambda^n \downarrow & & \downarrow M\lambda^nM & & \\ MMHM^n & \xrightarrow{M\lambda M^n} & MHM^nM & & \\ M\lambda M^n \downarrow & & \downarrow \lambda M^nM & & \\ MHM^{n+1} & \xrightarrow{\lambda M^{n+1}} & HM^{n+1}M & & \end{array}$$

The left-hand and right-hand parts both commute due to the definition of λ^{n+1} . The lower square obviously commutes, and for the commutativity of the upper one apply the functor M to the induction hypothesis. Thus the desired outside square commutes. \square

Next we need to establish a couple of properties connecting the three natural transformations χ , ε and λ^* .

Lemma A.6. *The following diagram of natural transformations commutes:*

$$\begin{array}{ccc} SMM & \xrightarrow{\chi M} & SM \\ \varepsilon M \downarrow & & \downarrow \varepsilon \\ MSM & \xrightarrow{M\chi} & MS. \end{array}$$

Proof. To verify that the square in the statement commutes we extend that square by the injections into the colimit SMM . This yields the following diagram:

$$\begin{array}{ccccc} M^nMM & \xlongequal{\quad} & & & M^{n+1}M \\ \text{inj}^nMM \searrow & & & & \swarrow \text{inj}^{n+1}M \\ & SMM & \xrightarrow{\chi M} & SM & \\ \varepsilon M \downarrow & & \downarrow \varepsilon & & \\ MSM & \xrightarrow{M\chi} & MS & & \\ \text{Minj}^nM \nearrow & & & & \swarrow \text{Minj}^{n+1} \\ MM^nM & \xlongequal{\quad} & & & MM^{n+1} \end{array}$$

The left-hand and right-hand inner squares commute by the definition of ε , and the upper and lower inner square commute by the definition of χ . Since the outside commutes obviously, so does the desired middle square when extended by any injection inj^nMM of the colimit SMM . Thus, the desired middle square commutes. \square

Lemma A.7. *The following square of natural transformations commutes:*

$$\begin{array}{ccc} SMH & \xrightarrow{S\lambda} & SHM \xrightarrow{\lambda^* M} HSM \\ \chi H \downarrow & & \downarrow H\chi \\ SH & \xrightarrow{\lambda^*} & HS. \end{array}$$

Proof. It suffices to verify that the desired square commutes when we extend it by an arbitrary colimit injection $\text{inj}^n MH$ of SMH . To this end we consider the diagram below:

$$\begin{array}{ccccc} M^n MH & \xrightarrow{M^n \lambda} & M^n HM & \xrightarrow{\lambda^n M} & HM^n M \\ \text{inj}^n MH \searrow & & \downarrow \text{inj}^n HM & & \swarrow H\text{inj}^n M \\ & SMH & \xrightarrow{S\lambda} & SHM & \xrightarrow{\lambda^* M} HSM \\ \chi H \downarrow & & & \downarrow H\chi & \\ & SH & \xrightarrow{\lambda^*} & HS & \\ \text{inj}^{n+1} H \nearrow & & & \swarrow H\text{inj}^{n+1} & \\ M^{n+1} H & & \xrightarrow{\lambda^{n+1}} & & HM^{n+1} \end{array}$$

The left-hand and right-hand parts commute by the definition of χ , and the lower and the upper right-hand parts commute by the definition of λ^* . The upper left-hand part commutes by the naturality of inj^n . Finally, the outside commutes by the definition of λ^{n+1} . Thus, the desired middle square commutes when extended by any colimit injection $\text{inj}^n MH$ of the colimit SMH . \square

Lemma A.8. *The following diagram of natural transformations commutes:*

$$\begin{array}{ccc} SMH & \xrightarrow{S\lambda} & SHM \xrightarrow{\lambda^* M} HSM \\ \varepsilon H \downarrow & & \downarrow H\varepsilon \\ MSH & \xrightarrow{M\lambda^*} & MHS \xrightarrow{\lambda S} HMS \end{array}$$

Proof. Once more it is sufficient to verify that the desired square commutes when extended by any injection of the colimit SMH . So consider the diagram below:

$$\begin{array}{ccccc} M^n MH & \xrightarrow{M^n \lambda} & M^n HM & \xrightarrow{\lambda^n M} & HM^n M \\ \text{inj}^n MH \searrow & & \downarrow \text{inj}^n HM & & \swarrow H\text{inj}^n M \\ & SMH & \xrightarrow{S\lambda} & SHM & \xrightarrow{\lambda^* M} HSM \\ \varepsilon H \downarrow & & & \downarrow H\varepsilon & \\ & MSH & \xrightarrow{M\lambda^*} & MHS & \xrightarrow{\lambda S} HMS \\ M\text{inj}^n H \nearrow & & \uparrow M\text{Hinj}^n & & \swarrow H\text{Minj}^n \\ MM^n H & \xrightarrow{M\lambda^n} & MHM^n & \xrightarrow{\lambda M^n} & HMM^n \end{array}$$

The left-hand and right-hand parts commute by the definition of ε , and the lower left-hand and upper right-hand parts commute by the definition of λ^* . The upper left-hand and the lower right-hand parts both commute due to the naturality of inj^n and λ , respectively. Finally, the outside commutes by

Lemma A.5. Thus, the desired inner square commutes when extended by any colimit injection $\text{inj}^n MH : M^n MH \rightarrow SMH$. \square

We are now prepared to prove the statement of Theorem A.4.

Proof of Theorem A.4. Let $e : X \rightarrow HMX$ be any λ -equation. We form the following H -coalgebra:

$$\bar{e} \equiv SX \xrightarrow{Se} SHMX \xrightarrow{\lambda_{MX}^*} HSMX \xrightarrow{H\chi_X} HSX. \quad (\text{A.5})$$

Since $c : C \rightarrow HC$ is a terminal H -coalgebra there exists a unique H -coalgebra homomorphism h from (SX, \bar{e}) to (C, c) . We shall prove that the morphism

$$e^\dagger \equiv X \xrightarrow{\text{inj}_X^0} SX \xrightarrow{h} C \quad (\text{A.6})$$

is the desired unique solution of the λ -equation e .

(1) e^\dagger is a solution of e . It is our task to establish that the outside of the diagram below commutes (cf. Diagram A.4):

$$\begin{array}{ccccc}
 & & e^\dagger & & \\
 & \swarrow & \curvearrowright & \searrow & \\
 X & \xrightarrow{\text{inj}_X^0} & SX & \xrightarrow{h} & C \\
 & \downarrow Se & \downarrow & & \downarrow c \\
 & & SHMX & & \\
 & \uparrow & \downarrow \lambda_{MX}^* & & \\
 & \uparrow \text{inj}_{HMX}^0 & \uparrow H\chi_X & & \\
 HMX & \xrightarrow{\text{Hinj}_{MX}^0} & HSMX & \xrightarrow{Hh} & HC \\
 & \uparrow \text{Hinj}_X^1 & \uparrow & \uparrow Hb & \\
 & & HSX & & \\
 & & \xrightarrow{HMe^\dagger} & &
 \end{array}$$

The upper part commutes by the definition of e^\dagger , and the upper right-hand square commutes since h is a coalgebra homomorphism. The upper left-hand part commutes due to the naturality of inj^0 , the triangle below that commutes by the definition of λ^* , and the lowest triangle commutes by the definition of χ . It remains to verify that the lowest part commutes. To this end we will now establish the following equation

$$b \cdot Me^\dagger = h \cdot \text{inj}_X^1. \quad (\text{A.7})$$

Consider the diagram below:

$$\begin{array}{ccccc}
 MX & \xrightarrow{Me^\dagger} & MC & & \\
 \downarrow & \searrow \text{Minj}_X^0 & \downarrow Mh & & \\
 & MSX & & & \\
 \downarrow \text{inj}_X^1 & \downarrow \text{inj}_{MX}^0 & \downarrow \varepsilon_X & & \downarrow b \\
 & SMX & & & \\
 \downarrow & \swarrow \chi_X & \downarrow h & & \\
 SX & \xrightarrow{h} & C & &
 \end{array}$$

The upper triangle commutes by the definition of e^\dagger , the left-hand triangle commutes by the definition of χ and the inner triangle commutes by the definition of ε . In order to establish that the right-hand part commutes we will use that C is a terminal H -coalgebra. Thus, we shall exhibit H -coalgebra structures on the five objects and then show that all edges of the right-hand part of the diagram are H -coalgebra homomorphisms. Then by the uniqueness of coalgebra homomorphisms into the terminal coalgebra (C, c) , we conclude that the desired part of the above diagram commutes.

For C , we use $c : MC \rightarrow C$, and for MC we use $\lambda_C \cdot Mc$. We already know that $b : MC \rightarrow C$ is a coalgebra homomorphism (see (A.3)). For SX , we use \bar{e} from (A.5) the composite from SX to HSX ; again, we already know that $h : SX \rightarrow C$ is a coalgebra homomorphism. For MSX we use $\lambda_{SX} \cdot M\bar{e}$. The verification that Mh is a coalgebra morphism comes from the diagram below:

$$\begin{array}{ccc}
 MSX & \xrightarrow{Mh} & MC \\
 \downarrow M\bar{e} & & \downarrow Mc \\
 MHSX & \xrightarrow{MHh} & MHC \\
 \downarrow \lambda_{SX} & & \downarrow \lambda_C \\
 HMSX & \xrightarrow{HMh} & HMC
 \end{array}$$

To see that the upper square commutes, remove M and recall that h is a coalgebra homomorphism from (SX, \bar{e}) to (C, c) . The lower square commutes by the naturality of λ .

Now we show that $\varepsilon_X : SMX \rightarrow MSX$ is a coalgebra homomorphism, where the structure on SMX is the composite on the left below:

$$\begin{array}{ccccc}
SMX & \xrightarrow{\varepsilon_X} & MSX & & \\
SMe \downarrow & & \downarrow MSE & & \\
SMHMX & \xrightarrow{\varepsilon_{HMX}} & MSHMX & & \\
S\lambda_{MX} \downarrow & & \downarrow M\lambda_{MX}^* & & M\bar{e} \\
SHMMX & & MHS MX & & \\
\lambda_{MMX}^* \downarrow & & \downarrow MH_{XX} & & \\
HSM MX & \xrightarrow{H\varepsilon_{MX}} & HMS MX & & \\
H\chi_{MX} \downarrow & & \downarrow \lambda_{SX} & & \\
HSMX & \xrightarrow{H\varepsilon_X} & HMSX & &
\end{array}$$

λ_{SMX} λ_{HSMX}

The upper square commutes by the naturality of ε , and the inner triangle commutes by the naturality of λ . To see that the right-hand part commutes, remove M and consider the definition of \bar{e} . The lowest part commutes due to Lemma A.6, and the middle part commutes by Lemma A.8.

Finally, we show that $\chi_X : SMX \rightarrow SX$ is a coalgebra homomorphism. To do this we consider the following diagram:

$$\begin{array}{ccc}
SMX & \xrightarrow{\chi_X} & SX \\
SMe \downarrow & & \downarrow Se \\
SMHMX & \xrightarrow{\chi_{HMX}} & SHMX \\
S\lambda_{MX} \downarrow & & \downarrow \lambda_{MX}^* \\
SHMMX & & \\
\lambda_{MMX}^* \downarrow & & \downarrow H\chi_X \\
HSM MX & \xrightarrow{H\chi_{MX}} & HSMX \\
H\chi_{MX} \downarrow & & \downarrow H\chi_X \\
HSMX & \xrightarrow{H\chi_X} & HSX
\end{array}$$

The upper square commutes by the naturality of χ , the middle square commutes by Lemma A.7, and the lower square commutes obviously. This concludes the proof that e^\dagger is a solution of e .

(2) e^\dagger in (A.6) is the unique solution of e . Suppose now that e^\dagger is any solution of the λ -equation e . Recall that the object SX is a colimit of the diagram D at object X with the colimit injections $\text{inj}_X^n : M^n X \rightarrow SX$, $n \in \mathbb{N}$. We will use the universal property of that colimit to define a morphism $h : SX \rightarrow C$. To this end we need to give a cocone $h_n : M^n X \rightarrow C$, $n \in \mathbb{N}$, for the appropriate diagram. We define this cocone inductively as follows:

$$\begin{aligned}
h_0 &= e^\dagger : M^0 X = X \rightarrow C; \\
h_{n+1} &= M^{n+1} X = MM^n X \xrightarrow{Mh_n} MC \xrightarrow{b} C, \quad n \in \mathbb{N}.
\end{aligned}$$

We now verify by induction on n that the morphisms $h_n, n \in \mathbb{N}$ do indeed form a cocone. For the base case consider the diagram below:

$$\begin{array}{ccc}
 M^0 X = X & \xrightarrow{\eta_X} & MX = M^1 X \\
 h_0 \downarrow & \searrow h_0 & \downarrow Mh_0 \\
 & C & \xrightarrow{\eta_C} MC \\
 & \searrow & \downarrow b \\
 & & C
 \end{array}$$

The upper part commutes by the naturality of η , the lower triangle commutes since $b : MC \rightarrow C$ is an algebra for the pointed endofunctor M , and the left-hand part is trivial. For the induction step consider for any natural number $n = i + j$ the following diagram:

$$\begin{array}{ccccc}
 M^{n+1} X = MM^{i+j} X & \xrightarrow{MM^i \eta_{M^j X}} & MM^i MM^j & = M^{n+2} X \\
 h_{n+1} \downarrow & \searrow Mh_n & & \swarrow Mh_{n+1} & \downarrow h_{n+2} \\
 & MC & & & \\
 & \downarrow b & & & \\
 & C & & &
 \end{array}$$

This diagram commutes: for the upper triangle remove M and use the induction hypothesis, and the remaining two inner parts commute by the definition of h_{n+1} and h_{n+2} , respectively.

Now we obtain a unique morphism $h : SX \rightarrow C$ such that for any natural number n the triangle below commutes:

$$\begin{array}{ccc}
 M^n X & & \\
 \downarrow \text{inj}^n X & \searrow h_n & \\
 SX & \xrightarrow{h} & C
 \end{array} \tag{A.8}$$

Next we show that $h : SX \rightarrow C$ is a coalgebra homomorphism from (SX, \bar{e}) to the terminal coalgebra (C, c) . To this end we will now verify that the lower part in the diagram below commutes:

$$\begin{array}{ccccccc}
 M^n X & \xrightarrow{M^n e} & M^n HMX & \xrightarrow{\lambda_{MX}^n} & HM^n MX & = & HM^{n+1} X \\
 \downarrow \text{inj}_X^n & & \downarrow \text{inj}_{HMX}^n & & \downarrow H\text{inj}_{MX}^n & & \downarrow H\text{inj}_X^{n+1} \\
 SX & \xrightarrow{Se} & SHMX & \xrightarrow{\lambda_{MX}^*} & HSMX & \xrightarrow{H\chi_X} & HSX \\
 h \downarrow & & & & & & \downarrow Hh \\
 C & & & & & & HC
 \end{array}$$

It suffices to show that the desired lower part commutes when extended by any colimit injection inj_X^n . Indeed, the left-hand part of the above diagram commutes by Diagram (A.8), and for the commutativity of the right-hand part, remove H and use Diagram (A.8) again. The upper left-hand square commutes by the naturality of inj^n , the upper middle square commutes by the definition of λ^* , and for the commutativity of the upper right-hand part remove H and use the definition of χ . It remains to verify that the outside of the diagram commutes. We will now prove this by induction

on n . For the base case $n = 0$ we obtain the following diagram

$$\begin{array}{ccccc}
 X & \xrightarrow{e} & HMX & & \\
 h_0 \downarrow & & \downarrow HMh_0 & & \\
 & & HMC & & \\
 & & \downarrow Hb & & \\
 C & \xrightarrow{c} & HC & &
 \end{array}
 \quad \text{Hh}_1 \curvearrowright$$

This diagram commutes: for the commutativity of the right-hand part remove H and use the definition of h_1 , and the left-hand part commutes since $h_0 = e^\dagger$ is a solution of the λ -equation e .

Finally, for the induction step we consider the diagram below:

$$\begin{array}{ccccccc}
 & & & & \lambda_{MX}^{n+1} & & \\
 & & M^{n+1}X & \xrightarrow{M^{n+1}e} & M^{n+1}HMX & \xrightarrow{M\lambda_{MX}^n} & MHM^nMX \xrightarrow{\lambda_{M^{n+1}X}} HM^{n+2}X \\
 h_{n+1} \left\{ \begin{array}{l} Mh_n \downarrow \\ MC \xrightarrow{Mc} MHC \xrightarrow{\lambda_C} HMC \\ b \downarrow \\ C \xrightarrow{c} HC \end{array} \right. & \xrightarrow{Mh_{n+1}} & & & \downarrow Mh_{n+1} & \xrightarrow{HMh_{n+1}} & \downarrow Hb \\
 & & & & & &
 \end{array}
 \quad \text{Hh}_{n+2} \curvearrowright$$

We see that this diagram commutes as follows: the lower part commutes by the definition of b (see (A.3)), the left-hand part commutes by the definition of h_{n+1} , for the commutativity of the right-hand part remove H and use the definition of h_{n+2} , the upper right-hand square commutes by the naturality of λ , and finally, to see the commutativity of the upper left-hand square remove M and use the induction hypothesis.

We have finished the proof that $h : SX \rightarrow C$ is a coalgebra homomorphism from (SX, \bar{e}) to the terminal coalgebra (C, c) . Since h is uniquely determined, it follows that the solution $e^\dagger = h \cdot \text{inj}_X^0$ is uniquely determined, too. This completes our proof. \square

Remark A.9. As explained by Bartels in [12], Theorem A.4 extends to the case where a distributive law λ of M over the cofree copointed functor $H \times \text{Id}$ is given. We briefly explain the ideas.

Let $D = H \times \text{Id}$ and $\varepsilon = \pi_1 : D \rightarrow \text{Id}$.

(1) A coalgebra for the copointed functor (D, ε) is a pair (X, x) where $x : X \rightarrow DX$ is such that $\varepsilon_X \cdot x = \text{id}_X$. Homomorphisms of coalgebras for (D, ε) are the usual D -coalgebra homomorphisms. It is trivial to prove that

$$C \xrightarrow{\langle c, \text{id}_C \rangle} HC \times C$$

is a terminal coalgebra for (D, ε) .

(2) One verifies that λ -equations $e : X \rightarrow HMX$ are in bijective correspondence with morphisms $f : X \rightarrow DMX$ such that

$$\begin{array}{ccc}
 X & \xrightarrow{f} & DMX \\
 & \searrow \eta_X & \downarrow \varepsilon_{MX} \\
 & & MX
 \end{array}$$

and that solutions of e correspond bijectively to solutions of f , i.e., morphisms $f^\dagger : X \rightarrow C$ such that Diagram (A.4) commutes with H replaced by D and c replaced by $\langle c, \text{id}_C \rangle$:

$$\begin{array}{ccc} X & \xrightarrow{f} & DMX \\ f^\dagger \downarrow & & \downarrow DMf^\dagger \\ C & \xrightarrow{\langle c, \text{id}_C \rangle} & DC \xleftarrow{Db} DMC \end{array}$$

See [12], Lemma 4.3.9.

(3) The same proof as the one for Theorem A.4 shows that for every $f : X \rightarrow DMX$ as in (2) above there exists a unique solution f^\dagger . One only replaces H by D , c by $\langle c, \text{id}_C \rangle$, and one has to verify that the coalgebra $\bar{e} : SX \rightarrow DSX$ from (A.5) is a coalgebra for the copointed endofunctor, see [12], Lemma 4.3.7.

To sum up, we obtain the following

Corollary A.10. *Let λ be a distributive law of the pointed functor M over the copointed one $H \times \text{Id}$. Then for every $e : X \rightarrow HMX$ there exists a unique solution, i.e., a unique $e^\dagger : X \rightarrow C$ such that (A.4) commutes.*

Theorem A.11. *Let λ be a distributive law of the pointed functor M over the copointed one $H \times \text{Id}$, and let $b : MC \rightarrow C$ be its λ -interpretation. Consider the algebra*

$$k = (HMC \xrightarrow{Hb} HC \xrightarrow{c^{-1}} C).$$

Then (C, k) is a cia for HM .

Indeed, to prove this result copy the proof of Theorem 3.4 replacing $\hat{b} : \hat{K}C \rightarrow C$ by $b : MC \rightarrow C$.

However, for our version of Theorem 3.5 in the current setting we need a different proof. We start with an auxiliary lemma.

Lemma A.12. *Let $\lambda : MH \rightarrow HM$ be a distributive law of the pointed functor M over the functor H , and let $b : MC \rightarrow C$ be its interpretation. Then the natural transformation $\lambda' = \lambda M \cdot M\lambda : MMH \rightarrow HMM$ is a distributive law of the pointed functor MM over H , and $Mb \cdot b$ is the λ' -interpretation in C .*

Proof. Clearly $(MM, \eta M \cdot \eta = M\eta \cdot \eta : \text{Id} \rightarrow MM)$ is a pointed endofunctor. The following commutative diagram

$$\begin{array}{ccccc} & & H & & \\ & \swarrow \eta H & & \searrow H\eta & \\ MH & & & & HM \\ \swarrow M\eta H & \searrow MH\eta & & \swarrow \eta HM & \searrow H\eta M \\ MMH & \xrightarrow{M\lambda} & MHM & \xrightarrow{\lambda M} & HMM \end{array}$$

shows that $\lambda' = \lambda M \cdot M\lambda$ is a distributive law of the pointed functor MM over H . In fact, the triangles commute by the assumption on λ , and the remaining upper square commutes by naturality of η ; thus the outside triangle commutes.

To see that $b \cdot Mb$ is the λ' -interpretation in C , consider the following diagram:

$$\begin{array}{ccccccc}
 MMC & \xrightarrow{MMc} & MMHC & \xrightarrow{M\lambda_C} & MHMC & \xrightarrow{\lambda_{MC}} & HMMC \\
 Mb \downarrow & & & & MHb \downarrow & & HMb \downarrow \\
 MC & \xrightarrow{Mc} & MHC & \xrightarrow{\lambda_C} & HMC & & \\
 b \downarrow & & & & & & Hb \downarrow \\
 C & \xrightarrow{c} & HC & & & &
 \end{array}$$

It commutes since b is the λ -interpretation in C and by the naturality of λ . In addition, $b \cdot Mb$ is easily seen to be an algebra for the pointed functor $(MM, \eta M \cdot \eta)$ since b is one for (M, η) and η is a natural transformation:

$$\begin{array}{ccccc}
 & C & & & \\
 & \eta_C \downarrow & & & \\
 & MC & \xrightarrow{b} & C & \\
 & \eta_{MC} \downarrow & & \eta_C \downarrow & \\
 MMC & \xrightarrow{Mb} & MC & \xrightarrow{b} & C
 \end{array}$$

This concludes the proof. □

Theorem A.13. *Let $\lambda : MH \rightarrow HM$ be a distributive law of the pointed functor M over the functor H , and let $b : MC \rightarrow C$ be its λ -interpretation. Consider the algebra*

$$k' = (MHMC \xrightarrow{Mk} MC \xrightarrow{b} C),$$

where $k = c^{-1} \cdot Hb$ as in Theorem A.11. Then (C, k') is a cia for MHM .

Proof. We have to prove that for every flat equation morphism $e : X \rightarrow MHMX + C$ for MHM there is a unique solution $e^\dagger : X \rightarrow C$ in $k' = b \cdot Mc^{-1} \cdot MHb : MHMC \rightarrow C$, i.e., a unique morphism e^\dagger such that the outside of the diagram

$$\begin{array}{ccccc}
 X & \xrightarrow{e^\dagger} & C & & \\
 e \downarrow & \searrow \bar{e} & & & \\
 & HMMX + C & & & \\
 & \xrightarrow{\lambda_{MX+C}} & & & \\
 & & HC + C & & \\
 & & \xrightarrow{\lambda_{C+C}} & & \\
 & & HMC + C & \xleftarrow{HMe^\dagger + C} & MC + C \\
 & & \xleftarrow{Hb + C} & & \xleftarrow{Mc + C} \\
 & & HMMC + C & \xleftarrow{\lambda_{MC+C}} & Mc^{-1} + C \\
 & & & & \xleftarrow{Hb + C} \\
 & & & & MHB + C
 \end{array}$$

commutes. To this end, we define the flat equation morphism

$$\bar{e} = (X \xrightarrow{e} MHMX + C \xrightarrow{\lambda_{MX+C}} HMMX + C)$$

for HMM (this is the left triangle). According to Lemma A.12 and Theorem A.11,

$$HMMC \xrightarrow{HMb} HMC \xrightarrow{Hb} HC \xrightarrow{c^{-1}} C$$

is a cia for HMM . So \bar{e} has a unique solution e^\dagger in this cia, i.e., the big inner part of the diagram commutes. In the upper right-hand part, b is the λ -interpretation in C . Since that part and the two remaining squares also commute (due to naturality of λ), the desired outside commutes. Thus, e^\dagger also is a solution of e in the algebra $k' : MHMC \rightarrow C$.

This solution is unique, since any other solution e^{\ddagger} of e in k' (i.e., the outside of the diagram with e^{\ddagger} in lieu of e^\dagger commutes) is a solution of \bar{e} in the cia $c^{-1} \cdot Hb \cdot HMb : HMMC \rightarrow C$ (i.e., the inner part commutes with e^{\ddagger} in lieu of e^\dagger), thus $e^{\ddagger} = e^\dagger$. \square